Sketching Formal Semantics of Graphical Meaning Derivation

Atsushi Shimojima

1. The Phenomenon

Let us start with an example. Suppose Jon, Ken, Gil, Bob, and Ron run races. They have a way to resolve ties in arrival, so that runners have exclusive finishing places, from the first to the fifth, in each race. Suppose we use the system $\mathcal{P}$ of “position diagrams” to express the finishing places of the five runners in particular races. Every well-formed diagram in this system is required to have the names “Jon”, “Ken”, “Gil”, “Bob”, and “Ron” in a horizontal row, with each name appearing exactly once. Figure 1 shows an example of a diagram in this system:

Gil Jon Bob Ken Ron

Fig.1: An example of a well-formed diagram of $\mathcal{P}$.

The semantic convention of this system is that if the name X' appears in the n-th position from left, it means that the bearer of X' arrived in n-th place (in the represented race). As the name “Gil” is in the first position from left, we learn that Gil arrived in first place; and as the name “Jon” is in the second position from left, we learn that Jon arrived in second place, and so on. This way of reading the diagram is, of course, a valid one, directly ligitimized by the semantic convention of the system.

There are, however, *other* legitimate ways of reading the diagram. For example, you see the name “Ken” appearing to the right of the name “Jon”, and you learn that Ken arrived later than Jon. If you *count* the number of names between the names “Jon” and “Ron”, you see how many people arrived between Jon and Ron in the race; if you notice that there are more names to the left of “Ken” than to its right, you learn that more people arrived before Ken than after him.

Note that the meaning relations underlying these reading practices are different from the ones directly legitimized by the basic semantic convention. To make this point clearer, (1) shows an example of the meaning relations directly legitimized by the semantic convention, and (2)–(4) show the meaning relations that we have just cited:

(1) If the name “Jon” appears in the second position from left, it means that Jon arrived in second place

(2) If the name “Ken” appears to the right of the name “Jon”, it means that Ken arrived later than Jon

(3) If there are two names between the names "Jon" and "Ron", it means that two people arrived between Jon and Ron

(4) If there are more names to the left of "Ken" than to its right, it means that more people arrived before Ken than after him

The meaning relations (2)–(4) are clearly different from (1) in kind. Yet each of them appears to be a valid evaluation of what is meant by the relevant aspect of the diagram. Although deviant from the system's basic semantic convention, they are surely not arbitratry creations by the interpreter.

In fact, the validity of (2)–(4) is partly based on the particular semantic convention that we have chosen for the system $\mathcal{P}$. Imagine that the system $\mathcal{P}$ has a totally different semantic convention, say, interpreting the appearance of a name in the *n*-th position *from right* to mean the person's finishing place in the race. Then all the meaning relations (2)–(4) would no longer hold and some alternatives would hold instead. In this respect, (2)–(4) are *derivatives* of the system's basic semantic convention.(1)

This phenomenon is quite prevalent in graphical representation systems. Specifically, *statistical charts* are sources of relatively clear cases of derivative meanings, and practioners and researchers discussed them under various conceptual frameworks. For example, bar charts enable the viewer to see "higher-level abstractions" constructed from the basic numerical information they carry (Guthrie, Weber & Kimmerly, 1993); visual patterns made by nearby lines in Cartesian line graphs carry "conceptual messages" about the data trend (Pinker, 1990); the shape of "clouds" made by dots in scatter plots signal the relationship between the represented variables (Kosslyn, 1994). Both practioners and researchers often distinguish "levels of questions" to be asked for statistical charts (Bertin, 1981; Wainer, 1992; Ratwani, Trafton & Boehm-Davis, 2003; Lohse, 1993) where higher-level questions are apparently directed to derivative meanings carried by the relevant charts.

Although less frequently, maps were cited as sources of derivative meanings in our sense. Lowe (1994) discussed "secondary structure", where adjacent isobars on a meteorological map together indicate a global trend of the area's barometric situation, including the presense of a trough. Gilhooly et al. found the use of "specialist schemata" in geographers' reading of contour maps, where visual patterns formed by several contour lines indicate some global structures in the area, such as valleys and interlocking spurs (Gilhooly, Wood, Kinnear & Green, 1988).

Node-edge graphs and even tables support derivative meanings. Olivier (2001) discussed the case of tree diagrams, where an extended path formed by consecutive edges indicates the presence of a descent or chain in the represented relational structure. In London's tube map, the concentration of edges touching a node indicates the presense of a "hub" station (Shimojima, 1999). Many tables are designed to allow the viewer to do "column-wise" or "row-wise" readings, in addition to basic "cell-wise" readings (Shimojima, 1999).

Graphical meaning derivation is a *functionally important* phenomenon as well. Some researchers hypothesize that the utility of a graphical system depends on what repeirtore of derivative meanings it allows the viewer to extract, and how easily the viewer can do so (Pinker, 1990; Lohse, 1993). A related hypothesis is that the proficiency or expertise of reading graphics depends on the ability to appreciate derivative meanings in the graphics (Lowe, 1989, 1994; Guthrie et al., 1993; Pinker, 1990; Gilhooly et al., 1988). It has been also hypothesized that evaluations of derivative meanings in a graphic forms a class of mental operations relatively independent from evaluations of basic meanings, whose occurences depend on the given purposes of reading the graphic (Guthrie et al., 1993; Kinnear & Wood, 1987; Ratwani et al., 2003).

Of course, all this is still hypothetical, and more research is required to determine the exact functional implications of the derivative meaning phenomenon. The issue of the scope of graphical meaning derivation, namely, to what range of instances the conept is coherently extendable, also requires more careful treatment based on detailed case studies. Yet, we largely set aside these issues in this paper, in order to concentrate on a more basic question concerning the logical origin of a derivative meaning relation in a graphical system. Specifically: under what informational conditions derivative meaning relations hold? What confers them their apparent legitimacy?

Despite its apparent prevalence and functional importance, the phenomenon of graphical meaning derivation has received little explicit attention in the literature of graphics semantics. No semantic theories of graphics, either grammatical, model-theoretic, or algbraic approaches, have ever attempted to track its logical origin. Also, the phenomenon is apparently independent of any informational relations, such as "secondary notations" (Petre & Green, 1992) and "graphical implicatures" (Marks & Reiter, 1990), that have been studied in pragmatic accounts of graphics. Pinker's theory (1990) offers a systematic account of the conditions for a coginitive system to comprehend derivative meanings, yet as a psychological theory, it is silent about the logical relationship between derivative meanings and basic meanings.

In the following, we try to build a framework of graphics semantics in which meaning derivation properties of graphical systems are explicitly modeled and accounted for. In our account, derivative meaning relations in a graphical system are results of the logical interactions involving (i) constraints installed by the semantic conventions, (ii) constraints originated in the domain of representations, and (iii) constraints originated in the domain of represented objects. We borrowed the concept of *constraint*, as well as its formal characterizations, from channel theory (Barwise & Seligman, 1997). Thus, our exploration has two broad parts: an exposition of the relevant part of channel theory, and a development of our own account. The point of this paper is to closely trace the logical origin of the meaning derivation phenomenon, so we will confine our analysis to the relatively simple instances (2)–(4) in the system of position diagrams. We refer the reader to Shimojima (1999) for broader (but less formal) treatment of the phenomenon.

2. Channel Theory

Definition 1 (Classification) A *classification* $\mathbf{A} = \langle \mathrm{tok}(\mathbf{A}), \mathrm{typ}(\mathbf{A}), \models_{\mathbf{A}} \rangle$ consists of a set $\mathrm{tok}(\mathbf{A})$ of *tokens*, a set $\mathrm{typ}(\mathbf{A})$ of *types*, and a binary relation $\models_{\mathbf{A}}$ on $\mathrm{tok}(\mathbf{A}) \times \mathrm{typ}(\mathbf{A})$. We say a token a *supports* a type α *in* the classification $\mathbf{A}$ if $a \models_{\mathbf{A}} \alpha$.

As the definition indicates, "classification" is a quite general concept, applicable wherever there is a collection of objects to which a specific set of properties are attributable. The following examples in particular show how we use this concept to model "the domain of representations" and "the domain of represented objects".

Example 1 Recall the system $\mathcal{P}$ of position diagrams representing running races, where the class of well-formed diagrams was defined by the following syntactic stipulations: (i) the names " Jon ", " Ken ", " Gil ", " Bob ", and " Ron " appear in a horizontal row, and (ii) each name appears exactly once. We can take the class of well-formed diagrams of $\mathcal{P}$ as the set of tokens of a classification, say, $\mathbf{P}_S$. Thus, $\mathrm{tok}(\mathbf{P}_S)$ consists of all the inidvidual position diagrams, produced in past or future, on paper or sand, in red or black.
As the set of types $\mathrm{typ}(\mathbf{P}_S)$, one can take whatever set of properties that classify objects in $\mathrm{tok}(\mathbf{P}_S)$. Let us assume that $\mathrm{typ}(\mathbf{P}_S)$ is the set of types taking the following forms:

- The name X' is in the n-th position from left (where $n \in \{1, 2, 3, 4, 5\}$)
- The name X' is to the right of the name Y'
- There are n names between the names X' and Y' (where $n \in \{0, 1, 2, 3\}$)
- There are more names to the left of the name X' than to its right

For brevity, we symbolize these types respectively as:

$$\textsc{In}(n, X') \qquad \textsc{Right}(X', Y') \qquad \textsc{Between}(n, X', Y') \qquad \textsc{MoreLeft}(X')$$

We use J', B', R', K', and G' to denote the names "Jon", "Bob", "Ron", "Ken", and "Gil" respectively. Thus, for example, $\textsc{Between}(2, B', R')$is the state of affairs that there are two names between the names "Bob" and "Ron". Let us use "d" to refer to the particular position diagram in Figure 1. Using the above symbols, we can describe some of d's properties in the following way:

$$d \models_{\mathbf{P}_S} \textsc{In}(5, R') \qquad d \models_{\mathbf{P}_S} \textsc{Right}(K', G')$$
$$d \models_{\mathbf{P}_S} \textsc{Between}(3, G', R') \qquad d \models_{\mathbf{P}_S} \textsc{MoreLeft}(K')$$

Example 2 Remember that the system $\mathcal{P}$ of position diagrams was designed to represent various running races run by Jon, Ken, Gil, Bob, and Ron and to express their finishing places in individual races. Thus, the target of this repersentation system can be considered another classification, say $\mathbf{P}_T$, where the set of tokens $\mathrm{tok}(\mathbf{P}_T)$ consists of all running races run by these people in future or past. So, if these five men run 30 races together in their life times, $\mathrm{tok}(\mathbf{P}_T)$ contains 30 tokens. The set of types of this classification could then consist of types of the following forms:

- The runner X arrives in the n-th place (where $n \in \{1, 2, 3, 4, 5\}$)
- The runner X arrives later than the runner Y
- There are n runners who arrive between the runners X and Y (where $n \in \{0, 1, 2, 3\}$)
- More runners arrive before X than after him

For brevity, we symbolize these types as:

$$\textsc{Arrive}(n, X) \qquad \textsc{Later}(X, Y) \qquad \textsc{Between}(n, X, Y) \qquad \textsc{MoreBefore}(X)$$

We use J, B, R, K, and G to denote the people Jon, Bob, Ron, Ken, and Gil, respectively. So, if these men arrived in the order of Gil, Jon, Bob, Ken, and Ron in a particular race r, the following is a partial description of the properties of r:

$$r \models_{\mathbf{P}_T} \textsc{Arrive}(3, B) \qquad r \models_{\mathbf{P}_T} \textsc{Later}(R, B)$$
$$r \models_{\mathbf{P}_T} \textsc{Between}(1, J, K) \qquad r \models_{\mathbf{P}_T} \textsc{MoreBefore}(R)$$

Generally speaking, a constraint is a regularity over a class of objects. It is a regularity in the sense that it is a recurrent pattern of properties shared by a class of objects. We use the notions of *sequent* and *satisfaction* to express such patterns of properties.

Definition 2 (Sequent) Let Σ be a set. A *sequent of* Σ is a pair $\langle \Gamma, \Delta \rangle$ of subsets of Σ.

Definition 3 (Satisfaction) Let $\mathbf{A}$ be a classification, and $\langle \Gamma, \Delta \rangle$ be a sequent of the set typ(A) of types in $\mathbf{A}$. Given a token a, we say *a satisfies* $\langle \Gamma, \Delta \rangle$ if:

> If $a \models_{\mathbf{A}} \alpha$ for every member α of Γ, then $a \models_{\mathbf{A}} \beta$ for some member β of Δ.

Thus, intuitively, when we talk about satisfaction, a sequent $\langle \Gamma, \Delta \rangle$ is taken as a material conditional where the antecedent is the conjunction of the types in Γ and the subsequent is the disjunction of the types in Δ.

Using the three concepts introduced so far, we can paraphrase our intuitive expression "recurrent patterns of properties" into a more precise "sequents of types satisfied by a class of tokens in a classification". Most generally, then, we can capture a precise notion of constraint in terms of a triple, consiting of a classification $\mathbf{A}$, a set of sequents of $\mathbf{A}$'s types, and a set of $\mathbf{A}$'s tokens satisfying those sequents. The notion of *local logic* in channel theory is an attempt to capture a system of constraints along this line.

Definition 4 (Local logic) Let $\mathbf{A}$ be a classification. A *local logic* $\mathcal{L}$ on $\mathbf{A}$ is a triple $\langle \mathbf{A}, \vdash_{\mathcal{L}}, N_{\mathcal{L}} \rangle$ consisting of:

1. A classification $\mathbf{A}$,
2. A set $\vdash_{\mathcal{L}}$ of sequents of typ($\mathbf{A}$) such that $\langle \text{typ}(\mathbf{A}), \vdash_{\mathcal{L}} \rangle$ makes a regular theory on typ($\mathbf{A}$),
3. A subset $N_{\mathcal{L}}$ of tok($\mathbf{A}$), called the *normal tokens* of $\mathcal{L}$, which satisfy all the sequents of $\vdash_{\mathcal{L}}$.

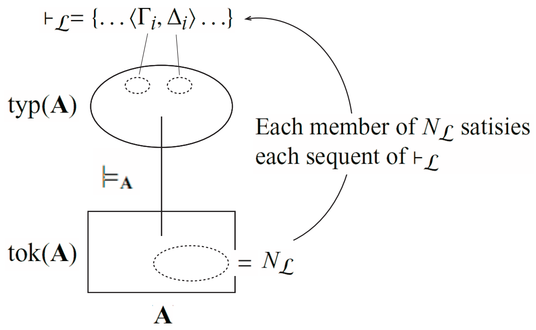

Fig.2: A local logic $\langle \mathbf{A}, \vdash_{\mathcal{L}}, N_{\mathcal{L}} \rangle$ on the classification **A**.

Figure 2 shows how the local logic $\mathcal{L}$ is related to a classification **A**. As we defined in Definition 2, a sequent $\langle \Gamma_i, \Delta_i \rangle$ is a pair of subsets of typ(**A**), and the component $\vdash_{\mathcal{L}}$ of a local logic is a collection of such sequents. Another component $N_{\mathcal{L}}$ of the local logic is a subset of typ(**A**), and each member of $N_{\mathcal{L}}$ is required to satisfy each sequent in $\vdash_{\mathcal{L}}$. According to this definition, the component $\vdash_{\mathcal{L}}$ of a local logic is not just any set of sequents, but a special set of sequents called a *regular theory*. This requirement is necessary as the definition is trying to capture a *system* of constraints. The following definitions specify what "regular theory" means.

Definition 5 (Theory) Let Σ be a set. A *theory on* Σ is a pair $T = \langle \Sigma, \vdash \rangle$ where $\vdash$ is a set of sequents of Σ.

Definition 6 (Regular theory) A theory $T = \langle \Sigma, \vdash \rangle$ is called *regular* if it satisfies the following closure conditions:

- (Identity) $\{\alpha\} \vdash \{\alpha\}$,
- (Weakening) If $\Gamma \vdash \Delta$, then $\Gamma, \Gamma' \vdash \Delta, \Delta'$.
- (Global Cut) If $\Gamma, \Sigma_0 \vdash \Delta, \Sigma_1$ for each partition $\langle \Sigma_0, \Sigma_1 \rangle$ of Σ', then $\Gamma \vdash \Delta$.(2)

Thus, a theory $T = \langle \Sigma, \vdash \rangle$ is called "regular" if its set of sequents $\vdash$ is "systematic" enough to satisfy the closure conditions of Identity, Weakening, and Global Cut. (3)

Remember that we have modeled the domain of representing objects in a representation system as a classification. Accordingly, the system of constraints governing this domain should be modeled as a local logic on the classification. Example 3 specifies how this could be done. It also shows the same for the domain of represented objects.

Example 3 Recall the clasification $\mathbf{P}_S$ of position diagrams from Example 1. Consider the class of all constraints govering the tokens of $\mathbf{P}_S$, and they will make up a special local logic on $\mathbf{P}_S$. Let $\mathcal{S}_{\mathcal{P}} = \langle \mathbf{P}_S, \vdash_{\mathcal{S}_{\mathcal{P}}}, \mathrm{tok}(\mathbf{P}_S)\rangle$ be that local logic. The following are obvious examples of constraints in $\mathcal{S}_{\mathcal{P}}$:

$$\textsc{In}(2, J'), \textsc{In}(2, B') \vdash_{\mathcal{S}_{\mathcal{P}}} \emptyset \qquad \textsc{Right}(J', R'), \textsc{Right}(R', K') \vdash_{\mathcal{S}_{\mathcal{P}}} \textsc{Right}(J', K')$$

$$\textsc{Between}(3, G', B') \vdash_{\mathcal{S}_{\mathcal{P}}} \textsc{Between}(3, B', G') \qquad \textsc{In}(5, R') \vdash_{\mathcal{S}_{\mathcal{P}}} \textsc{MoreLeft}(R')$$

The same position cannot be occupied by more than one name, so (1) holds. The constraint (2) holds because of the syntactic stipulation that names appear in a horizontal line, plus the transitivity of the Right relation in the horizontal ordering. It is important to note that this constraint is based on *both* stipulative constraints (an enforcement of linear horizontal arrangement) *and* natural constraints on the Between relation of names (transitivity). The constraint (3) comes from the nature of the tertiary relation Between plus the stipulation that each name appears exactly once. By the synatctic stipulation, every position diagram in this system have exactly fives names in it. Thus, if a name is placed in the fifth position from left, then certainly there are more names to its left than to its right. Hence the constraint (4).

Turn to the clasification $\mathbf{P}_T$ of running races from Example 2, and let $\mathcal{T}_{\mathcal{P}} = \langle \mathbf{P}_T, \vdash_{\mathcal{T}_{\mathcal{P}}}, \mathrm{tok}(\mathbf{P}_T)\rangle$ be the local logic listing all constraints on the races in tok($\mathbf{P}_T$). The following are obvious examples of constraints in $\mathcal{T}_{\mathcal{P}}$:

$$\textsc{Arrive}(2, J), \textsc{Arrive}(2, B) \vdash_{\mathcal{T}_{\mathcal{P}}} \emptyset \qquad \textsc{Later}(J, R), \textsc{Later}(R, K) \vdash_{\mathcal{T}_{\mathcal{P}}} \textsc{Later}(J, K)$$

$$\textsc{Between}(3, G, B) \vdash_{\mathcal{T}_{\mathcal{P}}} \textsc{Between}(3, B, G) \qquad \textsc{In}(5, R) \vdash_{\mathcal{T}_{\mathcal{P}}} \textsc{MoreLeft}(R)$$

These constraints simply follow from the irreflexive, asymmetric and transitive nature of the defeating relation. The logic $\mathcal{T}_{\mathcal{P}}$ also houses the following constraints because of the regularity condition on it:

(5) $\textsc{Defeat}(J, R) \vdash_{\mathcal{T}_{\mathcal{P}}} \textsc{Defeat}(J, R)$

(6) $\textsc{Defeat}(J, J), \textsc{Defeat}(K, B) \vdash_{\mathcal{T}_{\mathcal{P}}} \textsc{Defeat}(G, R)$

(7) $\textsc{Defeat}(J, R), \textsc{Defeat}(R, K), \textsc{Defeat}(K, B) \vdash_{\mathcal{T}_{\mathcal{P}}} \textsc{Defeat}(J, B)$

Recall that these constraints (2)–(7) had exact analogues in the local logic $\mathcal{T}_{\mathcal{P}}$ in Example 3. As the reader might have anticipated, this matching of constraints plays an important role in explaining the functional properties of the representation system $\mathcal{P}$.

Thus, we can use local logics to model systems of constraints both for the domain of representations and for the domain of represented objects. However, the tool kit obtained so far is not quite sufficient for our purpose. As we will see shortly, our account of graphical meaning derivation crucially assumes that semantic conventions in a representation system produce constraints holding *between* the source domain and the target domain. Although we have shown that local logics can model systems of constraints *in individual domains* (such as constraints in the source domain and the target domain of a representation system), we still want a machinery to extend this model to cover constraints holding *across multiple domains*.

Before we start developing the wanted machinery, let us see exactly how we can view the semantic conventions in a representation system as a producer of constraints.

Example 4 Representations are tools, and they are used for certain purposes—usually to convey information about certain objects. We then can individuate different *uses* of representations—particular events or situations in which representations are used to represent certain objects. For example, when I produced a sketch of my garden to show it my colleagues yesterday, the event makes up one particular use of a representation, the skecth. When my colleage said, "I will be available on September the sixth," that is another use of a representation—this time, a linguistic representation produced as a sound sequence.

Likewise, if somebody produces a particular position diagram d_k in the system $\mathcal{P}$ to convey information about a particular race r_k, this act makes up a particular use u_k of a position diagram. You may produce another position diagram d_j on a cocktail napkin to report the result of another race r_j to your friend. This act of yours makes up another use of u_j of a position diagram.

Let us put together these individual uses of position diagrams in the system $\mathcal{P}$, and call the class tok($\mathbf{P}_U$). Now suppose that Ken regularly reports the results of the races in his newsletters to his friends. Supppose Ken never lies about the results of races (although he sometimes skips reporting races in which he came in last). Given that Ken, as a runner, is always in the positon of directly witnessing the races, Ken's reports in his newsletter are reliably accurate. Thus, when his newsletter describes that Jon arrived first in a race, then Jon arrived first in the race. In the same vein, when his newsletter shows a position diagram where the name "Jon" appears in the first position from left, then Jon actually arrived first in the race represented by the diagram.

This relationship is, in fact, a lawful constraint, enpowered by Ken's determination to issue accurate reports on the races he covers. It is a constraint from a type $\text{In}(1, J')$ of the classification $\mathbf{P}_S$ to a type $\text{Arrive}(1, J)$ of the classification $\mathbf{P}_T$, guaranteed to hold in the uses of position diagrams in Ken's newsletter. Generally, Ken's uses of position diagrams support any constraints, from the type $\text{In}(n, X')$ to the type $\text{Arrive}(n, X)$, where that the name X' denotes the person X.

Let N_K be the subclass of tok($\mathbf{P}_U$) consisting of those uses of position diagrams in Ken's private newsletters to his friends. There may, of course, be other uses, outside N_K, that genuinely respect these constraints. For example, Jon too may have strong determination toward accurate race reports in his personal newsletters. Bob may have similar determination for his uses of position diagrams in cocktail-napkin reports to his wife. And there are other reasons for accuracy constraints. Whatever the reason might be, there are various circumstanes in which position diagrams are constrained to be accurate. Let $N_{\mathcal{L}_P}$ be the class of all such uses of position diagrams. Then, the members of $N_{\mathcal{L}_P}$ all respect the constraints from the type $\text{In}(n, X')$ to the type $\text{Arrive}(n, X)$.

Note that the mapping from the types $\text{In}(n, X')$ to the types $\text{Arrive}(n, X)$ is precisely the basic semantic indication relation of the representation system $\mathcal{P}$. This is no coincidence, of course, as the lawful constraints in question are the results of Ken's and other people's efforts to make produce accurate reports according to the very semantic convention that they employ.

Any reasonable semantic convention thus produces lawful constraints on the uses of a certain class of representations, or more precisely, installing such lawful constraints is the the purpose of instituting a semantic convention. When you and I agree that I will wink when Tom comes in, we are installing a new little constraint in our immediate environment—a constraint from my winking and Tom's presence. Such an additional constraint in our environment makes our life significantly easy, and that is why we agree on the little semantic convention in question.

Semantic conventions in a representation system are thus a producer of constraints from the domain of representing objects to the domain of represented objects. How do we go about modeling such inter-domain constraints? Here again, the basic tools of channel theory prove to be useful.(4) Specifically, two of its central concepts, *infomorphism* and *channel*, are suitable for our purpose.

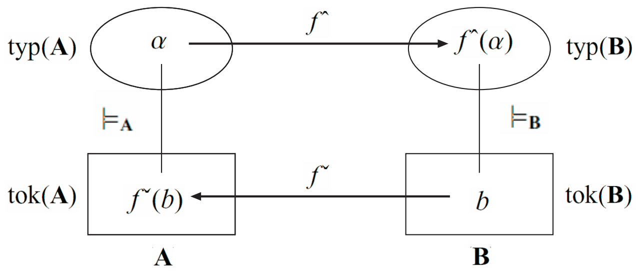

Fig.3: An infomorphism $f = \langle f^\wedge, f^\vee \rangle$ consisting of the function $f^\wedge$ from typ(**A**) to typ(**B**) and the function $f^\vee$ from tok(**B**) to tok(**A**), where $f^\vee(b) \models_{\mathbf{A}} \alpha$ is required to be equivalent to $b \models_{\mathbf{B}} f^\wedge(\alpha)$.

Definition 7 (Infomorphism) Let **A** and **B** be classfications. An *infomorphism* $f : \mathbf{A} \rightleftarrows \mathbf{B}$ from **A** to **B** is a contravariant pair of functions $f = \langle f^\wedge, f^\vee \rangle$ such that:

1. $f^\wedge : \text{typ}(\mathbf{A}) \to \text{typ}(\mathbf{B})$,
2. $f^\vee : \text{tok}(\mathbf{B}) \to \text{tok}(\mathbf{A})$,
3. $f^\vee(b) \models_{\mathbf{A}} \alpha$ iff $b \models_{\mathbf{B}} f^\wedge(\alpha)$ for each token $b \in \text{tok}(\mathbf{A})$ and each type $\alpha \in \text{typ}(\mathbf{B})$. (We will call this condition "the fundamental property of infomorphisms".)

Figure 3 shows the general form of infomorphism $f = \langle f^\wedge, f^\vee \rangle$ described in this definition. The fundamental property requires the equivalence of the condition $f^\vee(b) \models_{\mathbf{A}} \alpha$ with the condition $b \models_{\mathbf{B}} f^\wedge(\alpha)$.

Generally speaking, the notion of infomorphism models a particular classificatory correspondence between two classifications (**B** and **A** above). Intuitively, every token b of the classification **B** has a unique token $f^\vee(b)$ of the classification **A** as one of its components playing a specific role in it. Thus, every type α describing the token $f^\vee(b)$ of **A** is a description of a component of the token b of **B** and hence can be considered a description of b. As such, α has its "translation" $f^\wedge(\alpha)$ in the collection of types of **B**, and $f^\vee(b) \models_{\mathbf{A}} \alpha$ is equivalent to $b \models_{\mathbf{B}} f^\wedge(\alpha)$. The folowing are examples of infomorphism directly relevent to our analysis.

Example 5 Consider the class tok($\mathbf{P}_U$) of individual uses of position diagrams that we discussed in Example 4. We can think of various classifications of this class of "objects". In this example, we will describe a particular classification $\mathbf{P}_U$ among them, and show there is a natural infomorphism $p_S : \mathbf{P}_S \rightleftarrows \mathbf{P}_U$ from the classification $\mathbf{P}_S$ of position diagrams (Example 1) to this classification $\mathbf{P}_U$ classifying particular uses of position diagrams.

Remember that the set tok($\mathbf{P}_S$) of tokens of $\mathbf{P}_S$ collects all the position diagrams in the system $\mathcal{P}$ produced in past and future. Then, there is a natural function from tok($\mathbf{P}_U$) to tok($\mathbf{P}_S$), mapping each individual use u_i of a position diagram to the position diagram d_i used in that use. Call this function ${p_S}^\vee$.

Thus, in the above examples concerning Ken and Jon, $p_S\check{}(u_k) = d_k$ and $p_S\check{}(u_j) = d_j$. Given a particular use u_i in tok($\mathbf{P}_U$), we call $s_i\check{}(u_i)$ *the source diagram of u*.

Now, this mapping $p_S\check{}$ lets us define a particular collection of properties classifying tok($\mathbf{P}_U$) with reference to the properties classifying tok($\mathbf{P}_S$). For example, take the type $\textsc{In}(1, J')$ classifying tok($\mathbf{P}_S$). From this, we can define a type classifying tok($\mathbf{P}_U$), which might be expressed as $\textsc{In}(1, J')(\textsc{Source})$. Its intuitive meaning is "having the source diagram in which the name 'Jon' appearing to the right of the name 'Ken' ". (Contrast this with the type $\textsc{In}(1, J')$, whose intuitive meaning would be "having the name 'Jon' appearing to the right of the name 'Ken' ".) A member u_i of tok($\mathbf{P}_U$) supports$\textsc{In}(1, J')$ if and only if the name "Jon" appearing to the right of the name "Ken" in the source diagram of u_i.

Suppose typ($\mathbf{P}_U$) contains all types that can be defined in this way from typ($\mathbf{P}_S$). This implies that typ($\mathbf{P}_U$) contains all the types of the following forms:

$\textsc{In}(n, X')(\textsc{Source})$, $\textsc{Right}(X', Y')(\textsc{Source})$, $\textsc{Between}(n, X', Y')(\textsc{Source})$, $\textsc{MoreLeft}(X')(\textsc{Source})$

Under this assumption, there is a natural function from typ($\mathbf{P}_S$) to typ($\mathbf{P}_U$), mapping each type α in typ($\mathbf{P}_S$) to the type $\alpha(\textsc{Source})$. Calling this function $p_S\hat{}$, it should be clear that the pair $s_i = \langle p_S\hat{}, p_S\check{}\rangle$ is an infomorphism from the classification $\mathbf{P}_S$ to the classification $\mathbf{P}_U$. In particular, it satisfies the following fundamental condition:

(8) For every token $u_i \in$ tok($\mathbf{P}_U$) and every type $\alpha \in$ typ($\mathbf{P}_S$),

$$p_S\check{}(u_i) \models_{\mathbf{P}_S} \alpha \text{ iff } u_i \models_{\mathbf{P}_U} p_S\hat{}(\alpha).$$

Example 6 Continuing on Example 5, we find another natural mapping that departs from the class tok($\mathbf{P}_U$). For in many cases, position diagrams in the system $\mathcal{P}$ are used to convey information about particular races run by Jon, Ken, Ron, Gil, and Bob. Thus, for each particular use u_i in this category, there corresponds a particular race r_i. Recalling the set tok($\mathbf{P}_T$) of the classification $\mathbf{P}_T$ consists of all the races run by these five men (Example 2), this means that there is a natural mapping, say $p_T\check{}$, from tok($\mathbf{P}_U$) to tok($\mathbf{P}_T$). Given a particular use u_i in tok($\mathbf{P}_U$), we call $p_T\check{}(u_i)$ *the target object of* u_i.

Thus, again, we find a class of properties defined on the basis of this mapping $p_T\check{}$ and the properties typ($\mathbf{P}_T$) classifying tok($\mathbf{P}_T$). That is the class $\{\beta(\textsc{Target}) : \beta \in \text{typ}(\mathbf{P}_T)\}$. Since typ($\mathbf{P}_T$) contains all the types of the form $\textsc{Arrive}(n, X)$, this class contains all the types of the form $\textsc{Arrive}(n, X)(\textsc{Target})$.

Now, the types in this class classify the members of tok($\mathbf{P}_U$), so let us assume that they belong to typ($\mathbf{P}_U$). Then, there is a natural mapping from typ($\mathbf{P}_T$) to typ($\mathbf{P}_U$), which maps each type β to the type $\beta(\textsc{Target})$. Calling this function $p_T\hat{}$, we see that the pair $p_T = \langle p_T\hat{}, p_T\check{}\rangle$ makes up another example of infomorphism, this time from the classification $\mathbf{P}_T$ to the classification $\mathbf{P}_U$.

Recall that the infomorphism $p_S = \langle p_S\hat{}, p_S\check{}\rangle$ in Example 5 is from the classification $\mathbf{P}_S$ to the classification $\mathbf{P}_U$. Thus, we have two infomorphisms going to the common core $\mathbf{P}_U$. The combination of these infomorphisms is an example of "channel", and we will see shortly that a channel of this kind plays a crucial role in modeling inter-domain constraints bridging the source domain and the target domain of a representation system.

Generally, a channel is a group of infomorphisms going toward a common core. The following is the general definition:

Definition 8 (Channel) A *channel* C is an indexed family $\{f_i : \mathbf{A}_i \rightleftarrows \mathbf{C}\}_{i \in I}$ of infomorphisms with a common codomain $\mathbf{C}$, called the *core* of C. The classifications $\mathbf{A}_i$ are called *component classifications of* C.

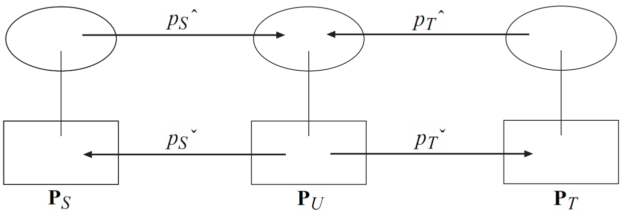

Fig.4: The binary channel $C_{\mathcal{P}} = \{p_i : \mathbf{P}_i \rightleftarrows \mathbf{P}_U\}_{i\in\{S,T\}}$, consisting of two infomorphisms, $p_S : \mathbf{P}_S \rightleftarrows \mathbf{P}_U$ and $p_T : \mathbf{P}_T \rightleftarrows \mathbf{P}_U$.

Thus, a channel is a collection of informorphisms going to the common core. In particular, the particualr channel discussed above can be expressed as the indexed family $\{p_i : \mathbf{P}_i \rightleftarrows \mathbf{P}_U\}_{i\in\{S,T\}}$. This is a *binary channel* in that it has two coponent classification, $\mathbf{P}_S$ and $\mathbf{P}_T$. We call this channel $C_{\mathcal{P}}$.

Now, a binary channel of this kind lets us model inter-domain constraints bridging the domain of representing objects and the domain of represented objects. The key idea is to define a local logic on the core of such a channel. Then some constraints in that local logic will be a model of such inter-domain constraints. The following example illutrates this method for the case of the channel $C_{\mathcal{P}}$.

Example 7 Recall tok($\mathbf{P}_U$) collects all individual uses of position diagrams in the system $\mathcal{P}$, and one of its subsets $N_{\mathcal{L}_P}$ is a collection of those uses under the accuracy pressure. We have seen that this group of uses supports an inter-domain constraint from the type $\textsc{In}(n, X')$ to the type $\textsc{Arrive}(n, X)$. Given the channel $C_{\mathcal{P}} = \{p_i : \mathbf{P}_i \rightleftarrows \mathbf{P}_U\}_{i\in\{S,T\}}$, we may express such inter-domain constraints as *sequents in a local logic on the core of* $\mathbf{P}_U$. This local logic, call it $\mathcal{L}_P$, has $N_{\mathcal{L}_P}$ as the set of its normal tokens. The inter-domain constraints in question are then described in the following way:

(9) $\textsc{In}(n, X')(\textsc{Source}) \vdash_{\mathcal{L}_P} \textsc{Arrive}(n, X)(\textsc{Target})$

According to the definitions in Examples 5 and 6, this is synonymous to:

(10) $p_S\hat{}(\textsc{In}(n, X')) \vdash_{\mathcal{L}_P} p_T\hat{}(\textsc{Arrive}(n, X))$

Given the construction of the channel $C_{\mathcal{P}}$, these constraints assert lawful relationship between the two classifications $\mathbf{P}_S$ and $\mathbf{P}_T$. That is, it asserts:

(11) For every use u in $N_{\mathcal{L}_P}$ if $p_S\check{}(u)$ supports the type $\textsc{In}(n, X')$ in the classification $\mathbf{P}_S$, then $p_T\check{}(u)$ supports the type $\textsc{Arrive}(n, X)$ in the classification $\mathbf{P}_T$.

To wit, assume (11), and let u be an arbitrary use of a position diagram belonging to the class $N_{\mathcal{L}_P}$. Then $p_S\check{}(u)$ is the source diagram of u, and $p_T\check{}(u)$ is the target object of u. Suppose $p_S\check{}(u) \models_{\mathbf{P}_S} \textsc{In}(n, X')$. Then by the fundamental property of the infomorphism p_S, it follows $u \models_{\mathbf{P}_U} p_S\hat{}(\textsc{In}(n, X'))$. Now $u \in N_{\mathcal{L}_P}$ by asuumption. So, by the constraint (11), it follows $u \models_{\mathbf{P}_U} p_T\hat{}(\textsc{Arrive}(n, X))$. Then by the fundamental property of the infomorphism p_T, we obtain $p_T\check{}(u) \models_{\mathbf{P}_T} \textsc{Arrive}(n, X)$.

Let us return to a simple infomorphism $f : \mathbf{A} \rightleftarrows \mathbf{B}$ from the classification $\mathbf{A}$ to $\mathbf{B}$. We have seen that an infomorphism models a classificatory correspondence between two classifications.

Now, let $\mathcal{L}_A$ be the local logic capturing the system of constraints holding on the classification **A** and $\mathcal{L}_B$ the local logic doing the same for the classification **B**. Since **A** and **B** have a classificatory correspondence (due to the infomorphism $f : \mathbf{A} \rightleftarrows \mathbf{B}$), there should be some correspondence between their respective logics $\mathcal{L}_A$ and $\mathcal{L}_B$. What exactly is it? The following examples illustrate this point.

Example 8 Consider the infomorphism $p_S : \mathbf{P}_S \rightleftarrows \mathbf{P}_U$ discussed in Example 5. Let $\mathcal{S}_\mathcal{P}$ and $\mathcal{L}_\mathcal{P}$ be the local logics capturing the system of constraints governing the classifications $\mathbf{P}_S$ and $\mathbf{P}_U$ respectively. We saw in Example 3 that the following is a constraint in the local logic $\mathcal{S}_\mathcal{P}$ on the classification $\mathbf{P}_S$:

(12) $\textsc{Right}(J', R'), \textsc{Right}(R', K') \vdash_{\mathcal{S}_\mathcal{P}} \textsc{Right}(J', K')$

Intuitively, (12) asserts the following:

(13) Having "Jon" to the right of "Ron" and having "Ron" to the right of "Ken" entails having "Jon" to the right of "Ken".

Since this is a constraint governing position diagrams, it should be intuitively clear that the following hold as a constraint governing the situations in which position diagrams are used:

(14) Having the source diagram in which "Jon" appearing to the right of "Ron" and having the source diagram in which "Ron" appearing to the right of "Ken" entails having "Jon" appearing to the right of "Ken"

Thus, a constraint in the local logic $\mathcal{S}_\mathcal{P}$ on the classification $\mathbf{P}_S$ of position diagrams is transferred to a constraint in the local logic $\mathcal{L}_\mathcal{P}$ on the classification $\mathbf{P}_U$ of situations in which position diagrams are used. This point may become more perspicuous if we rewrite (14) more formally in terms of the notations involving the infomorphism $p_S : \mathbf{P}_S \rightleftarrows \mathbf{P}_U$:

(15) $p_S\hat{}(\textsc{Right}(J', R')), p_S\hat{}(\textsc{Right}(R', K')) \vdash_{\mathcal{L}_\mathcal{P}} p_S\hat{}(\textsc{Right}(J', K'))$

Generally, every constraint $\Gamma \vdash_{\mathcal{S}_\mathcal{P}} \Delta$ in the logic $\mathcal{S}_\mathcal{P}$ on $\mathbf{P}_S$ is reflected as the constraint $p_S\hat{}(\Gamma) \vdash_{\mathcal{L}_\mathcal{P}} p_S\hat{}(\Delta)$ in the logic $\mathcal{L}_\mathcal{P}$ on $\mathbf{P}_U$. This is a natural assumption, given the classificatory correspondence between $\mathbf{P}_S$ to $\mathbf{P}_U$ modeled by the infomorphism $p_S : \mathbf{P}_S \rightleftarrows \mathbf{P}_U$.

Example 9 The logics $\mathcal{S}_\mathcal{P}$ and $\mathcal{L}_\mathcal{P}$ have a correspondece in their normal tokens, too. Consider a token u of the classification $\mathbf{P}_U$. As such, u is a particular situation in which a position diagram is used. Suppose u is *normal* in the sense that it respects the system of constraints $\mathcal{L}_\mathcal{P}$ on the classification $\mathbf{P}_U$. Well, then, it is a natural expectation that every component of u be also normal, respecting the local logic on on its own classification. In particular, the source diagram $p_S\check{}(u)$ used in this particular use u is expected to be normal, respecting the local logic $\mathcal{S}_\mathcal{P}$ on its own classification $\mathbf{P}_S$. The idea is that the two logics $\mathcal{L}_\mathcal{P}$ and $\mathcal{S}_\mathcal{P}$ must be associated so that no use of a position diagram respecting $\mathcal{L}_\mathcal{P}$ may involve a source diagram violating $\mathcal{S}_\mathcal{P}$. This means that for every token u of $\mathbf{P}_U$, if $u \in N_{\mathcal{L}_\mathcal{P}}$, then $p_S\check{}(u) \in N_{\mathcal{S}_\mathcal{P}}$. Put another way, $p_S\check{}(N_{\mathcal{L}_\mathcal{P}}) \subseteq N_{\mathcal{S}_\mathcal{P}}$.

The two kinds of correspondences illustrated by Examples 8 and 9 are quite natural assumptions when we deal with logics of infomorphic classifications. When an infomorphism $f : \mathbf{A} \rightleftarrows \mathbf{B}$ relates the logics $\mathcal{L}_A$ and $\mathcal{L}_B$ in this way, we call f a *logic infomorphism* from $\mathcal{L}_A$ to $\mathcal{L}_B$. Here is the formal definition:

Definition 9 (Logic infomorphism) Let f be an infomorphism from the classification **A** to the classification **B**, and $\mathcal{L}_A, \mathcal{L}_B$ be local logics of **A**, **B** respectively. We call f a *logic infomorphism* from **A** to **B**, writing $f : \mathcal{L}_A \rightleftarrows \mathcal{L}_B$ if:

1. For all sets Γ, Δ of types of **A**, if $\Gamma \vdash_{\mathcal{L}_A} \Delta$, then $f\hat{}(\Gamma) \vdash_{\mathcal{L}_B} f\hat{}(\Delta)$,
2. $f\check{}(N_{\mathcal{L}_B}) \subseteq N_{\mathcal{L}_A}$.

3. The Account

Now that we have presented the relevant part of channel theory, we can start developing our own concepts to be used for our account of graphical meaning derivation. We start with the notion of "abstraction."

Definition 10 (Abstraction relation) Let $\mathcal{L} = \langle \mathbf{A}, \vdash_{\mathcal{L}}, N_{\mathcal{L}} \rangle$ be a local logic, α be a member of typ(**A**), and $\mathcal{G}$ be a set of subsets of typ(**A**). We say α is an *abstraction over $\mathcal{G}$ in $\mathcal{L}$*, writing $\alpha \bowtie_{\mathcal{L}} \mathcal{G}$, if:

1. $\alpha \vdash_{\mathcal{L}} \Gamma$ for every choice set Γ of $\mathcal{G}$,
2. $\Gamma \vdash_{\mathcal{L}} \alpha$ for every member Γ of $\mathcal{G}$.

Intuitively, the type α abstracting over a collection $\mathcal{G}$ is in fact an "abstract" type that can be realized in various concrete ways, and $\mathcal{G}$ is the exhaustive collection of these concrete ways for α to be realized. As the definition indicates, the abstraction relation can be modeled as a special type of bi-directional constraints in a local logic.

More specifically, (i) each member Γ of $\mathcal{G}$ constitutes a particular way in which α can be realized, and (ii) the members of $\mathcal{G}$ exhausts all the ways in which α can be realized. Because of (i), if a token satisfies any member Γ of $\mathcal{G}$, then it supports α, and because of (ii), if a token supports α, then it satisfies some member of Γ. The following proposition makes this point more precise:

Proposition 1 *Let $\mathcal{L} = \langle \mathbf{A}, \vdash_{\mathcal{L}}, N_{\mathcal{L}} \rangle$ be a local logic, and suppose $\alpha \bowtie_{\mathcal{L}} \mathcal{G}$. For every token a in $N_{\mathcal{L}}$, $a \models_{\mathbf{A}} \alpha$ iff a satisfies some member of $\mathcal{G}$.*

Proof. For left to right, suppose $a \in N_{\mathcal{L}}$ and $a \models_{\mathbf{A}} \alpha$. Suppose for reductio that α satisfies no member of $\mathcal{G}$. Then, every member of $\mathcal{G}$ has a member β such that $a \not\models_{\mathbf{A}} \beta$. So, if we define Σ as the set of all members of $\bigcup \mathcal{G}$ that are not supported by a in **A**, then Σ is a choice set over $\mathcal{G}$. But since $\alpha \bowtie_{\mathcal{L}} \mathcal{G}$, it must hold that $\alpha \vdash_{\mathcal{L}} \Sigma$. Then since $a \in N_{\mathcal{L}}$ and $a \models_{\mathbf{A}} \alpha$ by supposition, a must support some member of Σ in **A**. But this is not the case by the definition of Σ. This is a contradiction. For right to left, suppose $a \in N_{\mathcal{L}}$ and a satisfies some member Δ of $\mathcal{G}$. Since

$$\mathcal{G}_1 = \left\{\begin{array}{ll} \{\textsc{In}(1,J'), & \textsc{In}(2,K')\} \\ \{\textsc{In}(1,J'), & \textsc{In}(3,K')\} \\ \{\textsc{In}(1,J'), & \textsc{In}(4,K')\} \\ \{\textsc{In}(1,J'), & \textsc{In}(5,K')\} \\ \{\textsc{In}(2,J'), & \textsc{In}(3,K')\} \\ \{\textsc{In}(2,J'), & \textsc{In}(4,K')\} \\ \{\textsc{In}(2,J'), & \textsc{In}(5,K')\} \\ \{\textsc{In}(3,J'), & \textsc{In}(4,K')\} \\ \{\textsc{In}(3,J'), & \textsc{In}(5,K')\} \\ \{\textsc{In}(4,J'), & \textsc{In}(5,K')\} \end{array}\right\} \quad \mathcal{G}_2 = \left\{\begin{array}{ll} \{\textsc{In}(1,J'), & \textsc{In}(4,R')\} \\ \{\textsc{In}(2,J'), & \textsc{In}(5,R')\} \\ \{\textsc{In}(4,J'), & \textsc{In}(1,R')\} \\ \{\textsc{In}(5,J'), & \textsc{In}(2,R')\} \end{array}\right\} \quad \mathcal{G}_3 = \left\{\begin{array}{l} \{\textsc{In}(4,K')\} \\ \{\textsc{In}(5,K')\} \end{array}\right\}$$

Fig.5: The collections $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ of sets of source types of the system $\mathcal{P}$

$$\mathcal{D}_1 = \left\{\begin{array}{ll} \{\textsc{Arrive}(1,J), & \textsc{Arrive}(2,K)\} \\ \{\textsc{Arrive}(1,J), & \textsc{Arrive}(3,K)\} \\ \{\textsc{Arrive}(1,J), & \textsc{Arrive}(4,K)\} \\ \{\textsc{Arrive}(1,J), & \textsc{Arrive}(5,K)\} \\ \{\textsc{Arrive}(2,J), & \textsc{Arrive}(3,K)\} \\ \{\textsc{Arrive}(2,J), & \textsc{Arrive}(4,K)\} \\ \{\textsc{Arrive}(2,J), & \textsc{Arrive}(5,K)\} \\ \{\textsc{Arrive}(3,J), & \textsc{Arrive}(4,K)\} \\ \{\textsc{Arrive}(3,J), & \textsc{Arrive}(5,K)\} \\ \{\textsc{Arrive}(4,J), & \textsc{Arrive}(5,K)\} \end{array}\right\} \quad \mathcal{D}_2 = \left\{\begin{array}{ll} \{\textsc{Arrive}(1,J), & \textsc{Arrive}(4,R)\} \\ \{\textsc{Arrive}(2,J), & \textsc{Arrive}(5,R)\} \\ \{\textsc{Arrive}(4,J), & \textsc{Arrive}(1,R)\} \\ \{\textsc{Arrive}(5,J), & \textsc{Arrive}(2,R)\} \end{array}\right\} \quad \mathcal{D}_3 = \left\{\begin{array}{l} \{\textsc{Arrive}(4,K)\} \\ \{\textsc{Arrive}(5,K)\} \end{array}\right\}$$

Fig.6: The collections $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ of sets of target types of the system $\mathcal{P}$

$\alpha \bowtie_{\mathcal{L}} \mathcal{G}$, we know $\Gamma \vdash_{\mathcal{L}} \alpha$ for every member Γ of $\mathcal{G}$. In particular, $\Delta \vdash_{\mathcal{L}} \alpha$. Since $a \in N_{\mathcal{L}}$ and a satisfies Δ, it follows that $a \models_{\mathbf{A}} \alpha$.

The following examples give instances of the abstraction relation that are directly relevant to our account of derivative meaning.

Example 10 Consider the local logic $\mathcal{S}_{\mathcal{P}}$ on the classification $\mathbf{P}_S$ of position diagrams. In this logic, the type $\textsc{Right}(K', J')$ is an abstraction over the collection $\mathcal{G}_1$ of sets of types, where $\mathcal{G}_1$ is as shown in Fig. 5. Note that each member of $\mathcal{G}_1$ is a particualr way in which $\textsc{Right}(K', J')$ is realized. For example, $\{\textsc{In}(1, J'), \textsc{In}(2, K')\}$ represents the state in which the name "Jon" appears in the leftmost position and the name "Ken" appears in the second position from left. It is certainly a particular way in which the name "Ken" appears to the right of the name "Jon". Note also that this collection exhausts all particular ways in which "Ken" appears to the right of "Jon".

We cite two more instances of the abstraction relation in the logic $\mathcal{S}_{\mathcal{P}}$: $\textsc{Between}(2, J', R')$ is an abstraction over $\mathcal{G}_2$ and $\textsc{MoreLeft}(K')$ is an abstraction over $\mathcal{G}_3$, where $\mathcal{G}_2$ and $\mathcal{G}_3$ are as shown in Fig. 5.

Example 11 Consider the local logic $\mathcal{T}_{\mathcal{P}}$ on the classification $\mathbf{P}_T$ of position diagrams. We find instances of the abstraction relation that closely match the instances in Example 10. That is, $\textsc{Later}(K, J)$ is an abstraction over $\mathcal{D}_1$, $\textsc{Between}(2, J, R)$ is an abstraction over $\mathcal{D}_2$, and $\textsc{MoreBefore}(K)$ is an abstraction over $\mathcal{D}_3$, where the collections $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ are as shown in Fig. 6.

Barwise & Seligman (1997) developed the notion of *information system* to model the constraints over whatever complex objects ("distributed systems") with any number of inter-related components. A *binary information system* defined below is useful when we want to model relatively simple objects with two immediate components.

Definition 11 (Binary information system) A *binary information system* is an indexed family of logic infomorphisms $\{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}\}_{i \in I}$, where I has exactly two members.

Specifically, the component logics $\mathcal{L}_i$ specify intra-domain regularities in the two component classifications Cla($\mathcal{L}_i$). The core logic $\mathcal{L}$ specifies inter-domain regularities across the two classifications. Since the informorphisms f_i are also logic infomorphisms, every intra-domain regularity in $\mathcal{L}_i$ is transferred to a regularity in $\mathcal{L}$. Thus, the logic $\mathcal{L}$ is a central platform in which inter-domain constraints logically interact with the intra-domain constraints originated in component logics. This way, $\mathcal{L}$ predicts what collection of constraints are available in the entire binary system. This capability of (the concept of) a binary information system plays a crucial role in our account for meaning derivation in a graphical representation system.

We model representation systems as a special kind of binary information systems, developed from initial institution of basic semantic conventions but equipped with a richer set of semantic rules:

Definition 12 (Representation system) A *representation system* $\mathcal{R}$ consists of a binary information system $\{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}_\mathcal{R}\}_{i \in \{S,T\}}$ and a pair of binary relations $\Vdash_\mathcal{R}$ and $\vdash_\mathcal{R}$ on dom($f_S\hat{}$)$\times$dom($f_T\hat{}$) such that:

1. $\Vdash_\mathcal{R}$ is a sub-relation of $\vdash_\mathcal{R}$,
2. $\vdash_\mathcal{R}$ is a sub-relation of $\vdash_{\mathcal{L}_\mathcal{R}}$.

We call $\vdash_\mathcal{R}$ the *basic semantic conventions* of $\mathcal{R}$, and call $\vdash_\mathcal{R}$ the *semantic rules* of $\mathcal{R}$.

Earlier, we individuated particular *uses* of diagrams as individual events or situations in which a diagram is used to represent a certain object. As such, a particular use of a diagram is itself a complex object with two main components—namely, the source diagram and the target object. The binary information system $\{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}_\mathcal{R}\}_{i \in \{S,T\}}$ in Definition 12 is intended to capture regularities over a certain class of diagram uses and their components. In particular, $\mathcal{L}_\mathcal{R}$, $\mathcal{L}_S$, and $\mathcal{L}_T$ are local logics capturing the systems of constraints on diagram uses, source diagrams, and target objects, respectively.

As illustrated in Examples 4 and 7, we can consider establishment of semantic conventions as installment of new constraints into our immediate environment. They are constraints governing the relevant class of diagram uses, and as such constraints in the core logic $\mathcal{L}_{\mathcal{R}}$. They take the form of inter-domain constraints $f_S\hat{}(\alpha') \vdash f_T\hat{}(\alpha)$, connecting an individual type $f_S\hat{}(\alpha')$ to another individual type $f_T\hat{}(\alpha)$. Thus, the class of these constraints is a binary relation $\Vdash_{\mathcal{R}}$ defined on $\mathrm{dom}(f_S\hat{}) \times \mathrm{dom}(f_T\hat{})$. One may think of the members of $\Vdash_{\mathcal{R}}$ as "axioms" in the logical system $\mathcal{L}_{\mathcal{R}}$, since as initial conventions, they are not derived from any more basic constraints.

Now, a diagram use involves two main components, namely, a source diagram and a target object. As such, individual diagram uses are also subject to local logics $\mathcal{L}_S$ and $\mathcal{L}_T$ governing these component objects, so that all constraints in these logics are reflected in the core logic $\mathcal{L}_{\mathcal{R}}$. In our definition, this correlation is expressed by the requirement that f_S and f_T are logic infomorphisms, making a binary information system.

Thus, three kinds of constraints coexist in the core logic $\mathcal{L}_{\mathcal{R}}$: (i) the set of basic semantic conventions $\Vdash_{\mathcal{R}}$, (ii) the set of constraints originated in the local logic $\mathcal{L}_S$ on source diagrams, and (iii) the set of constraints originated in the local logic $\mathcal{L}_T$ on target objects. The semantic conventions interact with the transferred constraints, and they spin out new constraints ("theorems") in the core logic $\mathcal{L}_S$. (The interactions are closure operations of Identity, Weakening, and Global Cut under $\mathcal{L}_S$, to be exact.)

Some of these theorems take the form $f_S\hat{}(\alpha') \vdash_{\mathcal{L}_{\mathcal{R}}} f_T\hat{}(\alpha)$, allowing a valid inference from the type α' of a source diagram to the type α of the represented object. Depending on their utility, ease, and other factors, selected items of these inferences are automized and stabilized as "semantic rules". Combined with initial semantic conventions, these new rules make up the collection of semantic rules available in the system at the moment. The binary relation $\vdash_{\mathcal{R}}$ in our definition stands for this collection. (Hence the conditions 1 and 2 in Definition 12.) Our definition of representation system is designed to capture a snapshot of the representation system in this development.

In our view, derivative meaning relations in graphical representation systems are special instances of the new semantic rules thus adopted. Before we can specify their exact logical origin, however, we need introduce some auxiliary concepts.

Definition 13 (Source types and target types) Let $\mathcal{R} = \langle \{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}_{\mathcal{R}}\}_{i \in \{S,T\}}, \Vdash_{\mathcal{R}}, \vdash_{\mathcal{R}} \rangle$ be a representation system, where $\mathcal{L}_S$ is the local logic $\langle \mathbf{A}_S, \vdash_{\mathcal{L}_S}, N_{\mathcal{L}_S} \rangle$ and $\mathcal{L}_S$ is the local logic $\langle \mathbf{A}_T, \vdash_{\mathcal{L}_T}, N_{\mathcal{L}_T} \rangle$. We call types of the classification $\mathbf{A}_S$ *source types* of $\mathcal{R}$, and types of the classification $\mathbf{A}_T$ *target types* of $\mathcal{R}$.

Definition 14 (Conventional meaning and projection) Let $\mathcal{R} = \langle \{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}_{\mathcal{R}}\}_{i \in \{S,T\}}, \Vdash_{\mathcal{R}}, \vdash_{\mathcal{R}} \rangle$ be a representation system.

1. Given a source type α and a target type β of $\mathcal{R}$, we say α *conventionally means* β *in* $\mathcal{R}$, writing $\alpha \rightharpoonup_{\mathcal{R}} \beta$, if $f_S\hat{}(\alpha) \Vdash_{\mathcal{R}} f_T\hat{}(\beta)$.
2. Let Γ be a set of source types and Δ be a set of target types. We say Γ *is projected to* Δ *through* $\rightharpoonup_{\mathcal{R}}$, writing $\Gamma \rightharpoonup_{\mathcal{R}} \Delta$, if the relation $\rightharpoonup_{\mathcal{R}}$ restricted to the domain Γ is a one-one correspondence from Γ to Δ.
3. Let $\mathcal{G}$ be a collection of sets of source types and $\mathcal{D}$ be a collection of sets of target types. We say $\mathcal{G}$ *is projected to* $\mathcal{D}$ *through* $\rightharpoonup_{\mathcal{R}}$ if the projection relation $\rightharpoonup_{\mathcal{R}}$ restricted to the domain $\mathcal{G}$ is a one-one correspondence from $\mathcal{G}$ to $\mathcal{D}$.

Intuitively, $\rightharpoonup_{\mathcal{R}}$ is the semantic relation from source types to target types directly legitimized by the semantic conventions $\Vdash_{\mathcal{R}}$ of the system. The projection relation holds when a set Γ of source types is “just enough” to conventionally mean all target types in Δ: if you subtract any member from Γ, it no longer suffices to cover all members in Δ, and if you subtract any member from Δ, some member of Γ is no longer necessary to cover all members in Δ. The same idea applies to *collections of sets of types*, and the projection relation holds from $\mathcal{G}$ to $\mathcal{D}$ when $\mathcal{G}$ is just enough to conventionally mean all collections of sets of target types in $\mathcal{D}$.

It is our main proposal that every derivative meaning relation is based on the following relationship between a source type α and a target type β:

Definition 15 (Parallel abstraction pair) Let $\mathcal{R} = \langle \{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}_{\mathcal{R}}\}_{i \in \{S,T\}}, \Vdash_{\mathcal{R}}, \vdash_{\mathcal{R}} \rangle$ be a representation system. Let α' be a source type and α be a target type. We call $\langle \alpha, \beta \rangle$ a *parallel abstraction pair* in $\mathcal{R}$ if there exist a collection $\mathcal{G}$ of sets of source types and a collection $\mathcal{D}$ of sets of target types such that:

1. $\alpha' \bowtie_{\mathcal{L}_S} \mathcal{G}$,
2. $\mathcal{G}$ is projected to $\mathcal{D}$ through $\rightharpoonup_{\mathcal{R}}$,
3. $\alpha \bowtie_{\mathcal{L}_T} \mathcal{D}$,

Figure 7 shows what it amounts to for types α' and α to be an parallel abstraction pair. Note that the abstraction relations $\bowtie_{\mathcal{L}_S}$ and $\bowtie_{\mathcal{L}_T}$ are determined by the local logics $\mathcal{L}_S$ and $\mathcal{L}_T$, respectively, whereas the projection relation $\rightharpoonup_{\mathcal{R}}$ is largely the matter of the local logic $\mathcal{L}_U$.

Example 12 It is not difficult to see that the condition of parallel abstraction applies to the source type and the target type in each derivative meaning relation (2)–(4) discussed in the beginning of this paper. In our current terminology, these meaning relations can be expressed in the following way:

(16) RIGHT(K', J') means LATER(K, J)

(17) BETWEEN($2, J', R'$) means BETWEEN($2, J, R$)

(18) MORELEFT(K') means MOREBEFORE(K)

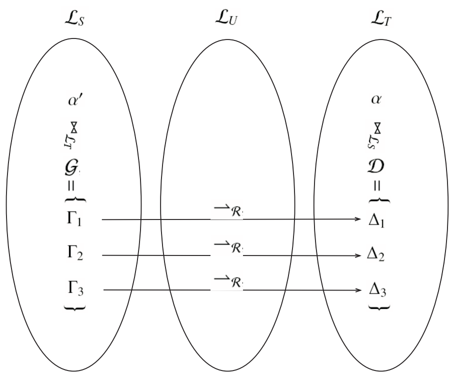

Fig.7: The source type α' and the target type α being a parallel abstraction pair, where $\mathcal{G}$ and $\mathcal{D}$ are depicted to have only three members for simplicity.

For (16), we have already seen that Right(K', J') is an abstraction of $\mathcal{G}_1$ in the local logic $\mathcal{S}_{\mathcal{P}}$, and also that Later$(K, J)$ is an abstraction of $\mathcal{D}_1$ in the local logic $\mathcal{T}_{\mathcal{P}}$ (see Examples 10 and 11). Now the collections $\mathcal{G}_1$ and $\mathcal{D}_1$ are clearly in a one-one correspondence under the conventional meaning relation $\rightharpoonup_{\mathcal{R}}$. Thus, on Definition 15, Right(K', J') and Later(K, J) are a parallel abstraction pair in the representation system $\mathcal{P}$.

As for (17), Between$(2, J', R')$ is an abstraction of $\mathcal{G}_2$ in the local logic $\mathcal{S}_{\mathcal{P}}$, and also that Between$(2, J, R)$ is an abstraction of $\mathcal{D}_2$ in the local logic $\mathcal{T}_{\mathcal{P}}$. The collections $\mathcal{G}_2$ and $\mathcal{D}_2$ are again in a one-one correspondence under $\rightharpoonup_{\mathcal{R}}$. Thus, Between$(2, J', R')$ and Between$(2, J, R)$ are a parallel abstraction pair in the system $\mathcal{P}$. We leave the reader to check that the definition also applies to (18).

In the beginning of this paper, we claimed that derivative meaning relations are not arbitrary creations of the interpreter, but valid evaluations of the information carried by the relevant aspects of the diagrams. The following proposition demonstrates that a source type and a target type in fact stand in a valid information relation whenever they make a parallel abstraction pair.

Proposition 2 *Let* $\mathcal{R} = \langle\{f_i : \mathcal{L}_i \rightleftarrows \mathcal{L}_{\mathcal{R}}\}_{i\in\{S,T\}}, \Vdash_{\mathcal{R}}, \vdash_{\mathcal{R}}\rangle$ *be a representation system. If* $\langle\alpha', \alpha\rangle$ *is a parallel abstraction pair in* $\mathcal{R}$*, then* $f_S\hat{}(\alpha') \vdash_{\mathcal{L}_{\mathcal{R}}} f_T\hat{}(\alpha)$.

Proof. The antecedent implies that there are collections $\mathcal{G}$ and $\mathcal{D}$ such that:

(1) $\alpha' \bowtie_{\mathcal{L}_S} \mathcal{G}$,

(2) $\mathcal{G}$ is projected to $\varDelta$ through $\rightharpoonup_{\mathcal{R}}$,

(3) $\alpha \bowtie_{\mathcal{L}_T} \mathcal{D}$.

We will show (a) and then (b):

(a) $f_S\hat{}(\varGamma) \vdash_{\mathcal{L}_{\mathcal{R}}} f_T\hat{}(\alpha)$ for every member $\varGamma$ of $\mathcal{G}$.

(b) $f_S\hat{}(\alpha') \vdash_{\mathcal{L}_{\mathcal{R}}} f_T\hat{}(\alpha)$.

For (a), let $\varGamma$ be an arbitrary member of $\mathcal{G}$. By (2), there exists a member $\varDelta$ of $\mathcal{D}$ to which $\varGamma$ is projected through $\rightharpoonup_{\mathcal{R}}$. This implies:

(4) For every member δ of $\varDelta$, there exists a member γ of $\varGamma$ such that $f_S\hat{}(\gamma) \Vdash_{\mathcal{R}} f_T\hat{}(\delta)$.

Consider the set $f_T\hat{}(\varDelta)$. If $\varSigma_1, f_S\hat{}(\varGamma) \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha), \varSigma_2$ is shown for every partition $\langle \varSigma_1, \varSigma_2 \rangle$ of $f_T\hat{}(\varDelta)$, then we obtain $f_S\hat{}(\varGamma) \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha)$ by Global Cut. So, let $\langle \varSigma_1, \varSigma_2 \rangle$ be an arbitrary partition of $f_T\hat{}(\varDelta)$. We divide the cases into (i) when $\varSigma_2 \neq \emptyset$ and (ii) when $\varSigma_2 = \emptyset$. The proof for the first case is trivial. In the second case, $\varSigma_2 = \emptyset$ and so $\varSigma_1 = f_T\hat{}(\varDelta)$. Since $\alpha \bowtie_{\mathcal{L}_T} \mathcal{D}$ by (3), the following holds:

(5) $\varDelta \vdash_{\mathcal{L}_T} \alpha$ for every member $\varDelta$ of $\mathcal{D}$.

Since $\varDelta \in \mathcal{D}$, it follows $\varDelta \vdash_{\mathcal{L}_T} \alpha$. But $f_T\hat{}$ is a logic infomorphism by the definition of representation systems. Hence, $f_T\hat{}(\varDelta) \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha)$. So by Weakening, $\varSigma_1, f_S\hat{}(\varGamma) \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha), \varSigma_2$. We thus obtain (a).

For (b), consider the set $f_S\hat{}(\bigcup \mathcal{G})$. Due to Global Cut, it suffices to show $\varSigma_1, f_S\hat{}(\alpha') \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha), \varSigma_2$ for every partition $\langle \varSigma_1, \varSigma_2 \rangle$ of $f_S\hat{}(\bigcup \mathcal{G})$. So let $\langle \varSigma_1, \varSigma_2 \rangle$ be an arbitrary partition of $f_S\hat{}(\bigcup \mathcal{G})$. Note that $\langle f_S\hat{}^{-1}(\varSigma_1), f_S\hat{}^{-1}(\varSigma_2) \rangle$ is a partition of $\bigcup \mathcal{G}$. We divide the cases into (i) when $f_S\hat{}^{-1}(\varSigma_2) \cap \varGamma \neq \emptyset$ for every member $\varGamma$ of $\mathcal{G}$ and (ii) $f_S\hat{}^{-1}(\varSigma_2) \cap \varGamma = \emptyset$ for some member $\varGamma$ of $\mathcal{G}$. In the first case, there is a choice set Θ of $\mathcal{G}$ such that $\Theta \subseteq f_S\hat{}^{-1}(\varSigma_2)$. Since $\alpha' \bowtie_{\mathcal{L}_S} \mathcal{G}$ by (1), the following holds:

(6) $\alpha' \vdash_{\mathcal{L}_S} \varGamma$ for every choice set $\varGamma$ of $\mathcal{G}$

In particular, $\alpha' \vdash_{\mathcal{L}_S} \Theta$. Since $\Theta \subseteq f_S\hat{}^{-1}(\varSigma_2)$, we obtain $f_S\hat{}^{-1}(\varSigma_1), \alpha' \vdash_{\mathcal{L}_S} f_S\hat{}^{-1}(\varSigma_2)$ by Weakening. But $f_S\hat{}$ is a logic infomorphism. It follows that $\varSigma_1, f_S\hat{}(\alpha') \vdash_{\mathcal{L}_\mathcal{R}} \varSigma_2$. By Weakening, $\varSigma_1, f_S\hat{}(\alpha') \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha), \varSigma_2$. The proof for the second case is trivial.

Earlier, we indicated that a derivative meaning relation arises out of the logical interaction of three kinds of constraints in the core logic: (i) a set of basic semantic conventions, (ii) a set of constraints originated in source diagrams, and (iii) a set of constraints originated in target objects. The proof of proposition 2 traces the logical interaction in question. In particular, (4), (5), and (6) in the proof are respectively the places where the constraints in the first, the first, and the second kind are appealed to. As the proof demonstrates, the logical interaction of these constraints in fact spins out a new constraint, $f_S\hat{}(\alpha') \vdash_{\mathcal{L}_\mathcal{R}} f_T\hat{}(\alpha)$, in the core logic $\mathcal{L}_\mathcal{R}$. This is an inter-domain constraint, making β a logical consequence of α, as far as the diagrams uses in $N_{\mathcal{L}_\mathcal{R}}$ are concerned. This is why we can validly interpret Right(K', J') to mean Later(K, J), Between($2, J', R'$) to mean Between($2, J, R$), and MoreLeft(K') to mean MoreBefore(K) in the system $\mathcal{P}$ of position diagrams. When a source type and a target make a parallel abstraction pair, they are logically qualified to stand in a derivative meaning relation.

4. Conclusion

Derivative meaning is quite prevalent in graphical representation systems. It is also a functionally important property of such systems. However, our account shows that doing justice to this phenomenon requires a drastic change in the way we view meaning of graphical representations. Derived meanings are completely different animals than basic meanings. Their treatment would require an explicit attention to three different kinds of constraints on the domain of diagram uses, the domain of diagram themselves, and the domain of target objects. Meaning derivation arises when the last two kinds of constraints are "aligned" by the first kind of constraints, so we need a mathematical tool to keep track of this alignment. As far as I could see, no existing framework of graphics semantics was equipped with such a tool, nor even prepared for a separate treatment of these kinds of constraints.

Channel theory, with its explicit attentions to local logics of separate domains and infomorphisms connecting them, seems like an ideal framework, so we sketched a framework of graphics semantics on top of it. I do not expect that the present paper is an strong enough argument to change every graphics semanticist's mind to adopt the particular framework proposed in it. However, I do believe that it has done enough to show that *something like this* is necessary in order for formal semantics to really extend its coverage into the domain of graphical representations

Notes

(1) The classifications of convenional and derivative meanings are not an absolute matter determined solely by syntactic structures of a given diagram. We can easily imagine an alternative system of position diagrams where (2) is a basic semantic convention while (1) is a derivative meaning. The point is that whatever the initially chosen semantic convention may be, additional meaning relations are often derived from them in graphical systems.

(2) Note on the theory notation $\vdash$: for readability, we write $\Gamma \vdash \beta$ instead of $\Gamma \vdash \{\beta\}$, and write $\alpha \vdash \Delta$ instead of $\{\alpha\} \vdash \Delta$. We also write $\Sigma, \Gamma \vdash \Delta$ instead of $\Sigma \cup \Gamma \vdash \Delta$, and write $\Gamma \vdash \Delta, \Sigma$ instead of $\Gamma \vdash \Delta \cup \Sigma$.

(3) In fact, Barwise and Seligman Barwise & Seligman (1997) have shown that these conditions must be satisfied by any set of sequents if it is to be an exhaustive list of constraints on some class of objects. They showed the converse too, namely, that any set of sequents satisfying these closure conditions can be considered an exhaustive list of constraints on some class of objects. We refer interested readers to Corollary 9.34 in Barwise & Seligman (1997).

(4) And this is certainly no accident, given the fact that chnnel theory is developed from a situation semantics, a quintessential project of analyzing meaning as constraints.

References

Barwise, J. & Seligman, J. (1997). *Information Flow: the Logic of Distributed Systems*, Cambridge, UK: Cambridge University Press.

Bertin, J. (1981). *Graphics and Graphic Information*, Berlin: Walter de Gruyter. Originally published in France in 1977.

Gilhooly, K. J., Wood, M., Kinnear, P. R., & Green, C. (1988). 'Skill in map reading and memory for maps,' *Quarterly Journal of Experimental Psychology*, *40A*, 87–107.

Guthrie, J. T., Weber, S., & Kimmerly, N. (1993). 'Searching Documents: Cognitive Processes and Deficits in Understanding Graphs, Tables, and Illustrations,' *Contemporary Educational Psychology*, *18*, 186–221.

Kinnear, P. R. & Wood, M. (1987). 'Memory for Topographic Vontour Maps,' *British Journal of Psychology*, *78*, 395–402.

Kosslyn, S. M. (1994). *Elements of Graph Design*, New York: W. H. Freeman and Company.

Lohse, G. L. (1993). 'A Cognitive Model for Understanding Graphical Perception,' *Human-Computer Interaction*, *8*, 353–388.

Lowe, R. K. (1989). 'Search Strategies and Inference in the Exploration of Scientific Diagrams,' *Educational Psychology*, *9*, *1*, 27–44.

——— (1994). 'Diagram Prediction and Higher Order Structures in Mental Representation,' *Research in Science Education*, *24*, 208–216.

Marks, J. & Reiter, E. (1990). 'Avoiding Unwanted Conversational Implicatures in Text and Graphics,' in *AAAI-90: Proceedings of Eighth National Conference on Artificial Intelligence*, *1*, 450–455, Menlo Park, CA/Cambridge, MA: AAAI Press/MIT Press.

Olivier, P. (2001). 'Diagrammatic Reasoning: an Artificial Intelligence Perspective,' in Blackwell, A. F. (Ed.), *Thinking with Diagrams*, Dordrecht: Kluwer Academic Publishers, 63–78.

Petre, M. & Green, T. R. (1992). 'Requirements of Graphical Notations for Professional Users: Electronics CAD Systems As a Case Study,' *Le Travail humain*, *55*, *1*, 47–70.

Pinker, S. (1990). 'A Theory of Graph Comprehension,' in *Aritificial Intelligence and the Future of Testing*, Hilsdale, NJ: L. Erlbaum Associates, 73–126.

Ratwani, R. M., Trafton, J. G., & Boehm-Davis, D. A. (2003). 'Thinking Graphically: Extracting Local and Global Information,' in *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*, 958–963.

Shimojima, A. (1999). 'Derivative Meaning in Graphical Representations,' in *Proceedings of the 1999 IEEE Symposium on Visual Languages*, Washington, D. C.: IEEE Computer Society, 212–219.

Wainer, H. (1992). 'Understanding Graphs and Tables,' *Educaional Researcher*, *21*, 14–23.

〔同志社大学准教授〕