A Logical Analysis of Graphical Consistency Proofs

Atsushi Shimojima
School of Knowledge Science
Japan Advanced Institute of Science and Technology

Suppose Mr. and Mrs. Murata have a small living room with a large sectional, two side tables, a center table, and a large TV cabinet. Mrs. Murata wants to rearrange the furniture to create a large pathway across the living room to the kitchen. Mr. Murata is rather reluctant about rearrangement, fearing that it may result in a less convenient setting of the TV cabinet and the sectional. Thus, Mrs. Murata needs to show that it is possible to create a desired pathway without sacrificing a good viewing angle of the TV from the sectional. For this purpose, she draws a diagram depicting their living room after a would-be rearrangement (Figure 1). "Look," she says. "It's possible. Let's do it."

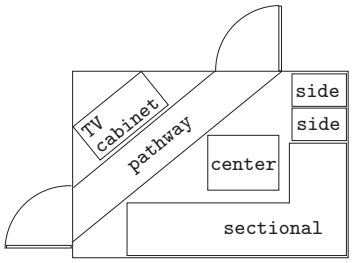

Figure 1: Room Map Drawn by Mrs. Murata

The type of "proof" that she has just conducted is the main subject of this paper. She has shown the possibility, or consistency, of an arrangement of furniture that satisfies specific requirements. Interestingly, she has done so *by constructing a graphical expression of the arrangement in question.* For some reason, the constructibility of such an expression is taken to guarantee the consistency of the represented conditions. Let us call this type of proof *graphical consistency proof.*

Although this procedure is quite natural and ubiquitous, its validity should not be taken for granted. A diagram is, after all, just a *representation* of something, not the thing itself. How can a construction of a representation be a proof of the consistency of what is expressed? Let Γ be the set of specifications of the

desired furniture arrangement, concerning the desired viewing angle of the TV display from the sectional, the desired width and route of the pathway, and the horizontal dimensions of Mr. and Mrs. Murata's living room and furniture, the desired viewing angle of the TV display from the sectional. Let s be Mrs. Murata's map, which expresses Γ. The question is how the construction of s can be a proof of the consistency of Γ.

Note that *not every* representation that expresses Γ is taken to be a proof of Γ's consistency. Suppose you specify all the requirements in the form of a list of English sentences. This would be a representation expressing all the requirements Γ, just as the room map s is. Yet, nobody would count it as a proof of the consistency of Γ. So, there must be something in representations such as s that is missing from lists of sentences. What is it?

A quick answer to this question is "an auto-consistency property of a representation system." Roughly, an auto-consistency property of a representation system is its incapability of expressing a certain range of inconsistent conditions. For example, the system of room maps of the kind Mrs. Murata has produced cannot express spatially impossible arrangements of furniture, and in this respect, the system is auto-consistent. This means that if some arrangement of furniture can be expressed in a room map, it guarantees that the expressed arrangement is spatially possible. The system of English sentences, on the other hand, is not auto-consistent in this respect, and for this reason, expressibility of a furniture arrangement in English does not guarantee the spatial possibility of the arrangement. Auto-consistency, defined as the incapability of expressing a certain range of inconsistent conditions, is clearly responsible for a system's capacity of graphical consistency proofs.

Researchers in philosophy, logic, AI, and cognitive psychology have paid some attention to auto-consistency properties of representation systems. Although the formal notion of auto-consistency was not introduced until later, Gelernter's Geometry Machine (1959) exploited the auto-consistency of geometry diagrams to short-cut the search for provable theorems. Sloman (1971) explicitly suggested non-expressibility of inconsistent information as a characteristic of "analogical" representations. Lindsay (1988) proposed a general framework of knowledge representation that exploits the auto-consistency property. Barwise and Etchemendy (1994) formally introduced the notion of auto-consistency and showed how their system of Hyperproof diagrams exploits this property to enable graphical consistency proofs. They also proved that a sub-system of Hyperproof diagrams is in fact auto-consistent (1995). Stenning and Inder (1995) discussed the trade-off of the expressive power of a representation system and its auto-consistency property.

Despite these exceptions, studies of the phenomenon of auto-consistency have been rather scattered and cursory so far. In particular, the questions still remains on the semantic mechanism *behind* an auto-consistency property of a representation system. Exactly what makes a representation system auto-consistent? What prevents a member of a representation system from expressing a certain range of inconsistencies? Is there any common semantic property shared by various systems that allow graphical consistency proofs?

This paper has two main goals. The first goal is to develop a semantic analysis adequate to answer the questions just posed. We will start with examining more examples of auto-consistent representation systems and their potentials for graphical consistency proofs (section 1). We will then propose our model of auto-consistency properties of representation systems in a simplified semantic framework of channel theory (Barwise and Seligman, 1997), and characterize, in its terms, the conditions for a graphical consistency proof to be valid in a representation system (section 2). As it turns out, a special types of matching of constraints in the source and the target of the system is responsible for auto-consistency. This indicates an important connection of the phenomenon of "free ride" discussed in the literature of diagrammatic reasoning and the phenomenon of graphical consistency proof, and our analysis motivates the general concept of *physical on-site inference* that covers both types of inferences.

The second goal of this paper is then to formulate this concept as clearly as possible. After giving an analysis of the exact procedures and requirements for free rides, we will highlight three characters shared by free rides and graphical consistency proofs: both procedures utilize *perceptually accessible* objects, such as graphics on a sheet of paper, as *inferential surrogates* by applying *physical operations* on them (section 3). Thus, seen in connection with the on-going research on model-based reasoning, this paper defines one exact sense in which perceptual objects are used as *models* for inference in manipulative processes (Magnani, 2001). In connection with the framework of distributed cognition, this paper is a case study of an important class of inferences that use visual representations as parts of distributed cognitive systems (Giere, 2001).

1 Examples

To get a surer grasp of our target, let us examine various examples of representation systems with auto-consistency properties and the kinds of graphical consistency proofs allowed in the systems.

Example 1 Mrs. Murata's room map is a member of an auto-consistent system of representation, since no representation of this type can express a spatially inconsistent arrangement of furniture. Imagine that the sectional in the living room were twice as large. Then it would be impossible even to lay out, without stacking, all the furniture in their small living room. *Correspondingly*, it is impossible to express this condition in a room map, for it is impossible to lay out, without overlapping, furniture icons of appropriate shapes and sizes in a small rectangular that stands for the living room.

This auto-consistency property of the system of room maps in turn allows us to conduct graphical consistency proofs, such as the one conducted by Mrs. Murata. You arrange furniture icons in a bounded area on a room map, and the expressed arrangement is thereby guaranteed to be spatially consistent.

It is important to note that the auto-consistency of a representation system is always relative to some particular range of inconsistencies. A representa-

tion system that is auto-consistent for some range of inconsistencies can fail to be some other range of inconsistencies. For example, although the furniture arrangement expressed by a room map is always spatially possible, it may well be psychologically impossible for Mr. Murata, or culturally impossible in a Japanese community. Accordingly, an graphical consistency proof with a room map can establish only spatial consistency, and not psychological or cultural consistency.

Example 2 Consider ordinary drawings of people such as Figure 2. Clearly, one cannot be taller than oneself—one's being taller than oneself is an inconsistent condition. And in fact, we cannot draw a line drawing of a person with such a height. That would amount to drawing a personal figure that is longer than itself. Similarly, it is impossible for a person A to be taller than a person B who is taller than A, nor for A to be taller than B who is taller than C who is taller than A. Correspondingly, we cannot draw line drawings of people of such relative heights. In terms of Figure 2, expressing the first condition amounts to drawing a figure that is both longer and shorter than the "B"-figure, and expressing the second condition amounts to drawing a personal figure that is longer than the "B"-figure but shorter than the "C"-figure. Clearly, both are impossible endeavors.

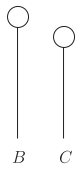

Figure 2: A drawing of people. Try to add a personal figure that is both longer and shorter than the left figure, or try to add a personal figure that is longer than the left figure but shorter than the right figure.

Generally, a representation system that expresses relative magnitude of one sort (relative height or percentage) by means of relative magnitude of another sort (relative length, size, or height) is auto-consistent against violations of the quasi-linearity of the expressed relation. Thus, the systems of bar charts, line graphs, pie charts, and other graphics used for quantitative analysis have analogous auto-consistency properties.

Such an auto-consistency property of course gives the system potential for simple graphical consistency proofs. For example, consider if it is possible for the following conditions to hold together:

(1) A is not taller than B.

(2) B is not taller than C.

(3) C is not taller than A.

Using the auto-consistency of the system of line drawings, one can draw a line drawing expressing all these conditions and make it a proof of their consistency. Figure 3 shows one of the many line drawings that would serve this purpose.

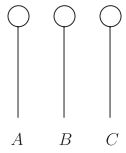

Figure 3: A line drawing that serves as a consistency proof of the conditions (1), (2), and (3).

In contrast, we can easily express the inconsistent conditions of people's height with different types of representations. You can write them up in English sentences such as: "A is taller than himself," "A is taller than B but shorter than B," and "A is taller than B, B is taller than C, and C is taller than A." You could also use first-order sentences to express these conditions. Or you can use a directed graph, such as Figure 4, where the edge denotes the *taller-than* relation. Thus, the systems of English sentences, first-order sentences, and directed graphs are not generally auto-consistent for people's relative height.

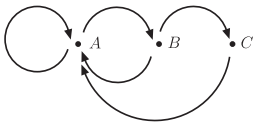

Figure 4: A directed graph expressing inconsistent conditions of people's height.

Example 3 Returning to an example of auto-consistency, consider a route map of a subway system of the kind shown in Figure 5. In common subway maps, a line with a particular pattern stands for a particular subway line (Jubilee line, Midosuji line, etc.). A connection of two stations via a particular subway line is then expressed by an corresponding type of line segments between two station icons, while a non-connection is expressed by the absence of such line segments. For example, the black line directly connecting the circles labeled "J" and "K" in Figure 5 indicates that Line U connects the stations J and K directly, with no station in between. On the contrary, the absence of a line segment directly connecting the circles labeled "J" and "E" indicates the absence of a direct connection between the stations J and E. Yet, the "J" circle and the "E" circle is indirectly connected by two black line segments, and this indicates that

Line U connects the stations J and E indirectly, with one station in between. There is not even such an indirect connection between the circles labeled "J" and "D," and this indicates that no single subway line connects the stations J and D.

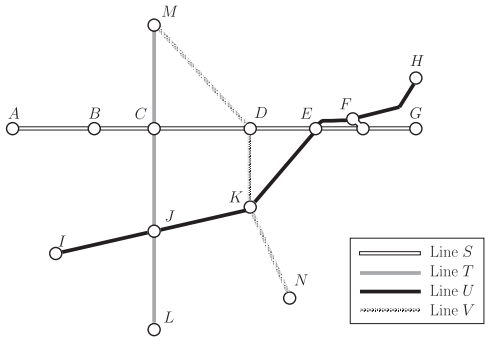

Figure 5: A route map of a subway system.

Now, it is not possible for two stations to be both connected and disconnected by the same subway line, nor for a station not to be connected to another station when the first station is connected to a station that is connected to the second station, nor for five stations to be connected to each other with less than four direct connections. *Correspondingly*, it is not possible to draw a route map that expresses any of these conditions: line segments cannot both connect and disconnect two circles, line segments of a particular color or pattern must connect two circles if line segments of that kind connect both circles to a third circle, and it is not possible to connect five circles to each other with less than four line segments. The system of route maps is auto-consistent against a certain type of graph-theoretic inconsistencies.

Example 4 The system of Euler diagrams is auto-consistent for certain set-theoretic inconsistencies. For instance, it is not possible for a set B to intersect with a proper subset A of a set C without intersecting with the superset C. Correspondingly, we cannot draw an Euler diagram that depicts sets in such relationship: try to draw a circle that intersects with the circle labeled "A" in Figure 6, and your circle will necessarily intersect with the circle labeled "B." Thus, however hard you may try, you cannot express a set disjoint from B that intersects with its proper subset A.

Interestingly, we can draw a Venn diagram of such a set. In Figure 7, the x-sequence indicates that B intersects with A, the y-sequence and the shading in the "A"-circle in combination indicate that A is a proper subset of C, and the shading in the intersection of the "B"- and the "C"-circle indicates that B does not intersect with C. Thus, it expresses the set-theoretically inconsistent condition of the three sets. The system of Venn diagrams is *not* auto-consistent

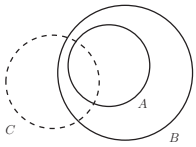

Figure 6: An Euler diagram. Try to add a circle that overlaps with the "A"-circle but not with the "B"-circle.

for this set-theoretic inconsistency and therefore cannot be used for a graphical consistency proof in the standard set-theoretic domain.

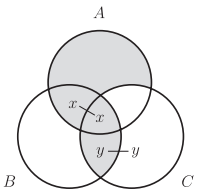

Figure 7: Venn diagram expressing an inconsistent condition.

2 Analysis

These examples should have given an adequate evidence that a wide variety of representations systems have auto-consistency properties, allowing us graphical consistency proofs. What is then the semantic mechanism behind auto-consistency properties?

We will explore this issue with mathematical tools of channel theory (Barwise and Seligman, 1997), although we will keep a part of our discussions informal. Mathematically oriented readers should consult chapter 20 and related chapters of Barwise and Seligman (1997) for formal details of the concepts used in this paper, such as "constraint," "indication," and "representation system."

Intuitively, auto-consistency involves some correspondence between inconsistent conditions in the domain of representations and inconsistent conditions in the domain of things represented. Take the example of the system of Euler diagrams (Example 4). Due to some spatial constraints holding in the domain of Euler diagrams, a certain arrangement of Euler circles is just impossible, and this impossibility corresponds to the impossibility of a certain inclusion or jointness relation in the domain of sets.

Let us make this intuition more precise. What exactly is impossible in the domain of representations in the case of Euler diagrams? Due to spatial constraints, it is impossible that the following three conditions hold together in

a single Euler diagram[1]:

(4*) A circle labeled "A" is inside a circle labeled "B."

(5*) A circle labeled "C" is outside a circle labeled "B."

(6*) A circle labeled "C" overlaps with a circle labeled "A."

Due to the semantic conventions associated with the system of Euler diagrams, the conditions (4*), (5*), and (6*) respectively indicate the following conditions on the represented sets:

(4) The set A is a proper subset of the set B.

(5) The set C is disjoint from the set B.

(6) The set C has an intersection with the set A.

Corresponding to the mutual inconsistency of (4*), (5*), and (6*), these indicated conditions cannot hold together in a single situation. That is, the conditions (4), (5), and (6) are also mutually inconsistent. Figure 8 depicts this situation schematically.

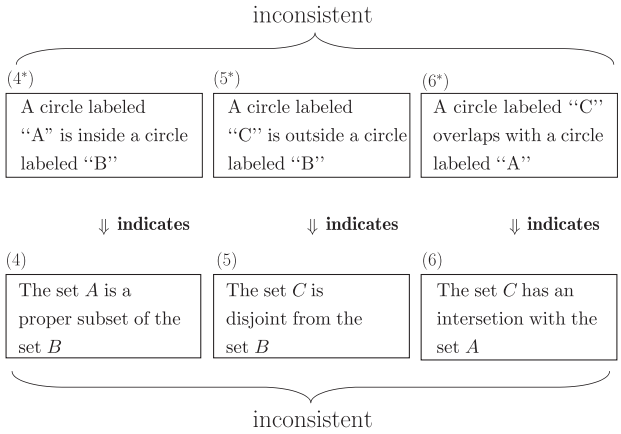

Figure 8: Correspondence of inconsistencies in the system of Euler diagrams.

Now, (4*), (5*), and (6*) must hold in any old Euler diagram expressing (4), (5), and (6), but (4*), (5*), and (6*) cannot hold together in a single Euler diagram. It follows that no Euler diagram can express the conditions (4), (5),

[1]We are assuming that no distinct circles can have the same label in a well-formed Euler diagram, making distinct sets denoted by distinct circles. "One-one denotation" rules of this sort apply to objects in many different kinds of graphical representations, including circles in Venn diagrams and station icons in subway route maps.

and (6), which are mutually inconsistent. Here we see the correspondence of inconsistent sets of conditions through the indication relation, and it accounts for the inability of Euler diagrams to express a particular inconsistent set of conditions. The system of Euler diagrams obviously involves many other cases of semantic correspondence of inconsistent sets, and they combine to define a significant range of inconsistent sets of conditions that cannot be expressed in Euler diagrams.

Figures 9 shows another instance of corresponding inconsistent sets in the system of route maps (Example 3). The conditions listed in the upper part of the figure must hold in any route map if it is to express the conditions in the lower part. Yet the set of conditions in the upper part is inconsistent. Hence the incapability of the system to express the inconsistent set of conditions in the lower part.

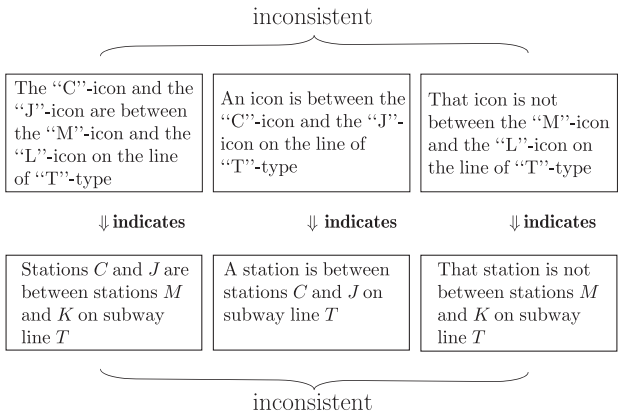

Figure 9: Correspondence of inconsistencies in the system of route maps.

How can we characterize these correspondences of inconsistent sets in more general terms? Let us introduce certain terminology to ease our analysis. By *source types*, we mean conditions that (potentially) hold in a representation, such as the three conditions of Euler diagrams listed in the upper part of Figure 8. In contrast, *target types* are conditions that (potentially) hold in a represented situation, such as the three conditions of sets A, B, and C listed in the lower part. If Ω is a set of source or target types, we call Ω *inconsistent* if there is no possible situation in which all members of Ω hold. Let us introduce the notion of "projections of sets of source types" in the following sense:

Definition 1 (Projection of Sets of Source Types) A set Γ of source types is projected to a set Δ of target types in a representation system $\mathcal{R}$ if:

- Each member of Γ indicates at least one member of Δ in $\mathcal{R}$,
- Each member of Δ is indicated by at least one member of Γ in $\mathcal{R}$.

For example, the set $\{(4^*), (5^*), (6^*)\}$ of source types is projected to the set $\{(4), (5), (6)\}$ of target types in the system of Euler diagrams. The projection is a one-one correspondence between the sets in this particular case, although the above definition does not require a one-one correspondence. Also, the set of source types listed in the upper part of Figure 9 is projected to the set of target types listed in the lower part in the system of line drawings.

With this preparation, we can now give a general characterization of "inconsistency inducement":

Definition 2 (Inducement of Inconsistency) A set Δ of target types is an *induced inconsistency in* a representation system $\mathcal{R}$ if:

- There is a set Γ of source types projected to Δ in $\mathcal{R}$,
- Every set Γ of source types projected to Δ is inconsistent.

For example, the set $\{(4), (5), (6)\}$ is an induced inconsistency in the system of Euler diagrams. For there is a set, $\{(4^*), (5^*), (6^*)\}$, projected to it, and while every set projected to it entails this set, it is inconsistent. Hence every set projected to $\{(4), (5), (6)\}$ is inconsistent. Likewise, the set of target types listed in the lower part of Figure 9 is an induced inconsistency in the system of line drawings. Each of these sets of target types cannot be expressed in the respective representation system that induces it.

Note that an induced inconsistency in a given system is not necessarily inconsistent. Definition 2 requires the inconsistency of the sets of source types projected to the target type, but not of the set of target types itself. Thus, the semantic mechanism of a system may "deem" a given set of target types as inconsistent, while it is in fact consistent.

On the other hand, if a system $\mathcal{R}$ never makes this type of "errors," we will say that *a constraint matching of type 1* holds in $\mathcal{R}$. That is, a constraint matching of type 1 is the following condition:

Constraint matching, type 1 For every set Δ of target types, if some set Γ of source types is projected to Δ in $\mathcal{R}$ and every set of source types projected to Δ in $\mathcal{R}$ is inconsistent, then Δ is inconsistent.

Typically, if a system ever induces an inconsistency, it induces more than one inconsistencies. Every representation system cited in section 1 induces more than one inconsistencies, as the discussions in that section show. So, it makes sense to talk about the *set* of inconsistencies induced in a given system $\mathcal{R}$. If, in addition, a constraint matching of type 1 holds in $\mathcal{R}$, we can think of that set as a special range of inconsistencies that are correctly "tracked" by the semantic mechanism of $\mathcal{R}$. We will call this set "$K_{\mathcal{R}}$," and call the members of this set "*$K_{\mathcal{R}}$-inconsistent*" to distinguish them from inconsistent sets of target types not tracked by $\mathcal{R}$.

We will also call any set of target types outside this set "*$K_{\mathcal{R}}$-consistent*." Thus, even when a set of target types is $K_{\mathcal{R}}$-consistent, it may be inconsistent.

Being $K_{\mathcal{R}}$-consistent only guarantees that the set is consistent so far as the range $K_{\mathcal{R}}$ of inconsistencies is concerned. The set may be inconsistent with respect to some other range of inconsistencies not tracked by the system $\mathcal{R}$.

Earlier, we roughly characterized the auto-consistency property of a representation system as the inability of the system to express a certain range of inconsistent conditions. We can now refine this characterization. That is, the auto-consistency of a representation system $\mathcal{R}$ is nothing but a constraint matching of type 1, where the set $K_{\mathcal{R}}$ of induced inconsistencies correspond to the "range of inconsistencies" that $\mathcal{R}$ cannot express. For, by the definition of $K_{\mathcal{R}}$, every member of $K_{\mathcal{R}}$ is a set of target types inexpressible in the system $\mathcal{R}$, and by the definition of type 1 matching, every member of $K_{\mathcal{R}}$ is in fact inconsistent.

More importantly, a constraint matching of type 1 holding in a system $\mathcal{R}$ guarantees the following form of inferences to be valid:

Graphical Consistency Proof Express the set Δ of information in the representation system $\mathcal{R}$. Conclude, from the success of that operation, that Δ is $K_{\mathcal{R}}$-consistent.

This procedure is valid since, if some consistent set Γ is projected to Δ in $\mathcal{R}$, then Δ is not an induced inconsistency in $\mathcal{R}$ and not a member of $K_{\mathcal{R}}$. Given the constraint matching of type 1, this just means Δ is $K_{\mathcal{R}}$-consistent. Now, one of the most efficient ways of verifying that some consistent set Γ of source types is projected to Δ in $\mathcal{R}$ is to actually construct a representation in $\mathcal{R}$ that expresses Δ, and this is exactly what is done through actual drawing of diagrams, charts, and maps. Thus, a graphical consistency proof is valid under a representation system with a constraint matching of type 1.

Note, however, the consistency result obtained through this procedure is just the $K_{\mathcal{R}}$-consistency of the set Δ, not the unconditional consistency of Δ. As we noted above, the $K_{\mathcal{R}}$-consistency of a set Δ of target sets only means that Δ is consistent with respect to the particular range of inconsistencies tracked by the system $\mathcal{R}$, and Δ may be actually inconsistent with respect to some other range of inconsistencies. And this is just natural as a model of graphical consistency proofs performed under a particular representation system. For example, the consistency of a particular furniture arrangement established in Example 1 is just the *spatial* consistency of the arrangement, and not necessarily the social or psychological or cultural consistency of the arrangement. Likewise, the consistency of a subway route established in Example 3 is just the *graph-theoretical* consistency of the expressed route, and not necessarily the commercial or economical or urbanologistic consistency. Generally, consistency established through a graphical consistency proof in a particular representation system is *limited* in nature. Our model reflects this limitation in terms of the limitation implied by the notion of $K_{\mathcal{R}}$-consistency.

3 Physical On-Site Inferences

Shimojima (1995a, 1995b) has introduced the notion of *free ride* to capture another common form of inferences typically done with graphical representations. Free rides are similar to graphical consistency proofs in that they exploit a certain type of constraint matching between representations and represented situations and that they essentially involve physical operations on representations for this exploitation. In this section, we will first review a few examples of free rides and extend the analysis given in Shimojima (1995a). We will then specify the common elements of graphical consistency proofs and free rides in order to define the general notion of "physical on-site inference."

3.1 Free Rides

Let us start with looking at two simple examples of free rides.

Example 5 Harry is asked to describe the geographical features of the village in which he grew up, as accurately as possible, by memory. After some failed trials of recollecting the geographical features of his home village with no tools, he decides to draw an approximate map of his home town. On the basis of fragments of his memory, he draws lines and curves on a sheet of paper to represent the streets, pathways, rivers, and such. He then uses wood blocks to represent the buildings that he remembers to have existed, and places them on his map, to represent the approximate locations of those buildings. (He keeps revising and supplementing the map, and eventually obtains a map that represents his home town to the best of his memory.)

At the beginning of this procedure, Harry remembers the locations of a river, two roads, and several houses, and constructs a tentative map (Figure 10, left).

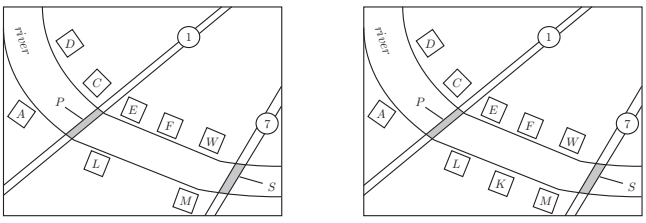

Figure 10: A manipulation of memory maps producing free rides.

Then he recollects one more piece of information about his home village, that is:

(7) The house K was halfway between the houses L and M.

To present this new fragment of memory in his map, Harry puts a wood block standing for the house K between the wood blocks standing for the houses L and M (Figure 10, right).

As the result of this simple operation, Harry's map now presents many pieces of new information, *other than* (7), that were absent from the initial map. Among them are:

(8) The house K was across the house F over the river.

(9) The house K was closer to road 1 than the house M was.

(10) The house M was closer to the bridge S than the house K was.

(11) The house K and the house A had the road 1 in between.

To get the sense of utility of the system of Harry's memory map, imagine how many deduction steps would be needed if he tried to obtain the same results with pure thought on the basis of the principles of geometry. By operating on his map in the way described above, Harry has skipped all these complications of computation, and obtain the information (8), (9), (10), and (11) almost "for free." The operation is extremely efficient, for the purpose of updating the information content of his map toward the solution of the problem.

Example 6 We use Venn diagrams to check the validity of the following syllogism:

(12) All Cs are Bs.

(13) No Bs are As.

(14) (Therefore) no Cs are As.

We start with drawing three circles, labeled "As," "Bs," and "Cs" respectively. On the basis of the premises (12) and (13) of the syllogism, we shade the complement of the "C"-circle with respect to the "B"-circle (Figure 11, left) and shade the intersection of the "B"-circle and the "A"-circle (Figure 11, right). Observing that the intersection of the "C"-circle and the "A"-circle is shaded as a result, we read off the conclusion (14), and decide that the syllogism is valid.

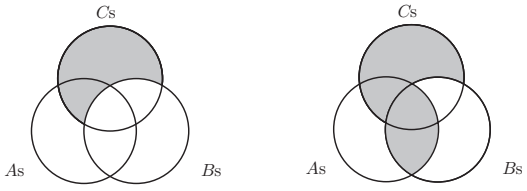

Figure 11: A manipulation of Venn diagrams producing a free ride.

Again, we obtain a piece of information "for free" just by operating on diagrams: updating a diagram on the basis of the information (12) and (13) lets the diagram generate the information (14), which in turn lets us decide that the syllogism is valid.

In a nut shell, a free ride is an inferential procedure where we express the set Δ of information in a representation system $\mathcal{R}$, observe that this operation results in the condition σ that indicates the information θ, and conclude that Δ entails θ.

Let us analyze the free ride in example 6 in more detail. Let us assume that we start with a blank sheet of paper, say s. We apply a certain operation on s to create a Venn diagram that express the target types (12) and (13). Due to the semantic rules associated with Venn diagrams, this requires the resulting sheet of paper, say s', to support the following source types:

(12*) The complement of a "C"-circle to a "B"-circle is shaded.

(13*) The intersection of a "B"-circle and an "A"-circle is shaded.

These types respectively indicate (12) and (13), and make s' express these pieces of information. Interestingly, (12*) and (13*) not only indicate (12) and (13), but also *entail* another source type, namely:

(14*) The intersection of a "C"-circle and an "A"-circle is shaded.

That is, if you shade the complement of a "C"-circle to a "B"-circle and shade the intersection of the "B"-circle and an "A"-circle, you end up shading the intersection of the "C"-circle and the "A"-circle! Now, according to the semantic rules associated with Venn diagrams again, this source type indicates the target type (14). Thus, the Venn diagram s' ends up with expressing the information (14) too. Thus, one can simply read off the information (14) from s', and since (14) is entailed by the original information (12) and (13), one is making a *valid* inference in this way. The major part of the inference, however, is not done by the user's thinking, but taken over by the entailment relation from (12*) and (13*) to (14*).

Figure 12 shows this analysis schematically, where a is the operation of drawing applied to the blank sheet of paper s. Here we see a constraint on representations that makes (14*) a consequence of the set $\{(12^*), (13^*)\}$ of source types, as well as a constraint on represented situations that makes (14) a consequence of the set $\{(12), (13)\}$ of target types. Moreover, the antecedent $\{(12^*), (13^*)\}$ is projected to the antecedent $\{(12), (13)\}$ via the indication relation $\Rightarrow$ associated with the representation system, while the consequent (14*) indicates the consequent (14). Thus, our analysis implies that a free ride is also a form of inference that utilizes the semantic matching of a constraint governing representations with a constraint governing the represented situations. This much is the analysis of Example 6 given in Shimojima (1995a), which can be easily extended to other cases of free rides such as Example 5.

What exactly is the type of constraint matching required for free rides then? It is slightly different from type 1 of constraint matching, required for physical consistency proofs. We therefore call it "type 2":

Constraint matching, type 2 For every set Δ of target types and every target type θ, if some set Γ is projected to Δ in the system $\mathcal{R}$ and every set

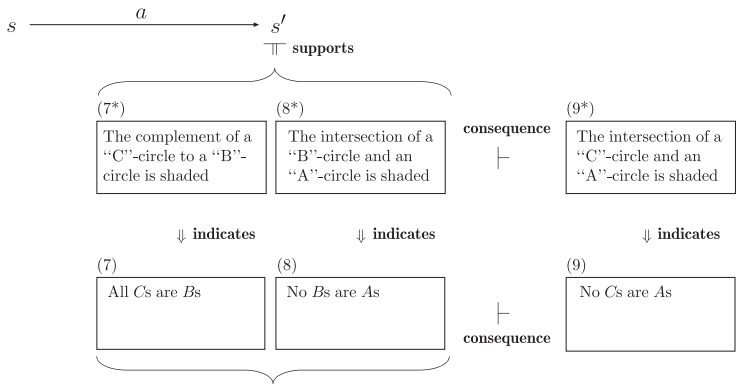

Figure 12: Analysis of a free ride in the system of Venn diagrams.

of source types projected to Δ in $\mathcal{R}$ entails a source type σ that indicates θ in $\mathcal{R}$, then Δ entails θ.

Note that this condition by itself does not allow the simple inferential procedure we called "free ride." For, if we are to simply use a constraint matching of this type to conclude that Δ entails θ, we need verify that every set Γ of source types projected to Δ entails a source type σ that indicates θ in $\mathcal{R}$. This amounts to exploring *every* way of expressing Δ in the system and finding what will happen, while intuitively, a free ride consists in just expressing Δ in a *particular* way and observing what happens.

In actual cases of free rides, this gap is filled by the *homogeneity* of the relevant representation systems. Intuitively, a system is homogeneous if, for each set Δ of target types, there is at most one "line" of expressing Δ in the system. For example, when you want to express the information (15) and (16) in a line drawing, you must have three figures labeled "A," "B," and "C" satisfying the conditions (15*) and (16*), and there is no way of expressing (15) and (16) without having such figures.

(15) A is not taller than B.

(16) B is not taller than C.

(15*) A figure labeled "A" is not longer than a figure labeled "B."

(16*) A figure labeled "B" is not longer than a figure labeled "C."

Likewise, there is no way of expressing (12) and (13) in a Venn diagram without having three circles satisfying (12*) and (13*). This is because the systems of line drawings and Venn diagrams are both homogeneous systems. In fact, all representation systems that have been considered in this paper are homogeneous. A typical example of heterogeneous system is the full representation

system used in *Hyperproof* (Barwise and Etchemendy, 1994), which contains, as its subsystems, a system of diagrams as well as a first-order language.

Let us define homogeneity and heterogeneity of a system more explicitly:

Definition 3 (Primitive Indicator and Homogeneous System)

- A set Δ of target types is *expressible* in a representation system $\mathcal{R}$ if there is a set Γ of source types projected to Δ in $\mathcal{R}$.
- A set Γ of source types is the *primitive indicator* of a set Δ of target types in a representation system $\mathcal{R}$ if Γ is projected to Δ in $\mathcal{R}$ and every set of source types projected to Δ in $\mathcal{R}$ entails Γ.
- A representation system $\mathcal{R}$ is *homogeneous* if every expressible set of target types has its primitive indicator; it is *heterogeneous* otherwise.

For example, the set $\{(15^*), (16^*)\}$ of source types is the primitive indicator of the set $\{(15), (16)\}$ of target types in the system of line drawings, and $\{(12^*), (13^*)\}$ is the primitive indicator of $\{(12), (13)\}$ in the system of Venn diagrams.

When a system is homogeneous, we can express any expressible set Δ of target types with its primitive indicator, and observing the result of expressing Δ in this way amounts to finding the *common* result of expressing Δ in every other way. Thus, we can exploit a constraint matching of type 2 without directly exploring every possible way of expressing Δ. If you express Δ with a primitive indicator Γ, and if you find new information θ expressed as the result, it means that Γ entails some source type σ indicating θ; but since Γ is entailed by every set of source types projected to Δ, it follows that σ is entailed by every set of source types projected to Δ; hence by a constraint matching of type 2, θ is guaranteed to be an entailment of Δ.

Thus, in its exact form, a free ride is the following inferential procedure:

Free ride Express the set Δ of information with its primitive indicator Γ. By observing that the operation results in the condition σ that indicates the information θ, conclude that θ is a consequence of Δ.

3.2 Three Characters of Physical On-Site Inference

So far, we have seen two different forms of inferences exploiting constraint matching between representations and their targets. These forms of inferences share several interesting properties: both use (1) perceptually accessible external representations (2) as inferential surrogates (3) through applications of physical operations on them. In view of these common properties, we call inferences in either of these forms "*physical on-site inferences*."

To make this notion more precise, let us clarify what each of these common properties are. We start with the property (2).

Inferential Surrogate Suppose we are thinking about a particular object t. Let us say we are using another object s as an *inferential surrogate* when we exploit the matching of constraints on s and constraints on t to make an inference about t.

In this sense, external representations used in free rides, such as a Venn diagram (Example 6), a memory map (Example 5), and other graphical representations, are inferential surrogates. The target object t is a particular situation represented by the representation at hand, such as a particular situation with several sets in a certain inclusion relation (Examples 6) or a particular region with a certain geographical configuration (Example 5). The relevant constraint matching is specified as type 2, namely, the projection of constraints of the form $\Omega \vdash \beta$ to constraints of the same form. According to our analysis, a free ride is an inference that exploits this particular type of constraint matching between a representation s and the represented situation t, and hence it is an instance of inference using an inferential surrogate.

Likewise, a graphical consistency proof uses an inferential surrogate. The surrogate s is again a representation at hand, such as a room map (Example 1) or a route map (Example 3), and the target object t is a particular room with several pieces of furniture (Example 1) or a subway route with several subway lines (Example 3). The constraint matching exploited is specified as type 1, where a consistent set on s is projected to another consistent set on t through the indication relation.

In this regard, our analysis is a clarification of the semantic mechanism behind an important species of distributed cognition, namely, the case emphasized by Giere (2001) when he says, "The visual representation is not merely an aid to human cognition; it is part of the system engaged in cognition" (p. 8). In physical on-site inferences, a cognitive burden of inference is partly transferred from human brains to physical constraints on external representations: we put the information to be processed into diagrams, charts, and others; the physical constraints on these representations "calculate" a consequence of the expressed information (free rides) or its consistency in a limited sense (graphical consistency proofs); we can then observe the results of calculation by attending to the new information expressed in representations or simply by checking whether the information to be expressed has been actually expressed. Later, we will see two more forms of physical on-site inferences that calculate non-consequence or inconsistency. Conceptually, inferential surrogates in our sense are special cases of "mediating structures" for distributed cognition (Hutchins, 1995).

Perceptual Presence Inferential surrogates used in physical on-site inferences are objects such as Venn diagrams, Euler diagrams, route maps, memory maps, and room maps. They are all representations on a piece of paper, a computer display, or some other physical media, and they are accessible to our vision, and in certain case, to tactile perception too.

Compare these cases with analogical reasoning in general. As far as they use a particular object as a source for reasoning about another object, physical on-

site inferences are a species of analogical reasoning. Yet the source object used in analogical reasoning does not have to be perceptually present, and indeed, cases typically cited as analogical reasoning involve source objects that are not perceptually present at the time of inference (a fictional event of a troop attacking a castle, planetary revolutions around the sun, and so on). In contrast, the notion of physical on-site inference emphasizes the fact that perceptually present representations, such as graphics on a piece of paper, often serve as source objects for inference. Although it is not in the scope of this book to study the exact internal processes involved in analogical inferences, it is clear that the internal process of using a perceptually present objects as the source is significantly different from that of using a perceptually inaccessible object.

Physicality It should be clear by now that free rides and graphical consistency proofs have essential physical components. Expressing a certain information set Δ is a physical operation on a representation, and each form of inference draws a different type of conclusion from the result of that operation: a consequence conclusion when a piece of information is automatically expressed (free ride) and a consistency conclusion when Δ is successfully expressed (physical consistency proof). In each case, the physical operation and the accompanying result plays the role of verifying the existence or non-existence of a constraint on the representation, and under certain types of constraint matching, the existence or non-existence of a constraint on the representation guarantees the existence of the constraint or non-constraint to which it is projected. Thus, a physical operation plays a significant, or even dominant, role in this inferential process, and it saves a significant amount of inferential task on the part of the user.

In this regard, our analysis should capture at least *some* instances of what Magnani (2001) calls "manipulative abductions." Magnani's focus is on production of explanatory hypotheses in scientific practices through physical manipulations of experimental devises. Yet the notion also covers manipulations of concrete models and diagrams (pp. 62–63), and our account could be considered specifications of the exact inferential procedures and semantic requirements involved in (some of) these cases. Although we have not considered any particular cases of manipulative abductions in scientific practices, how much of them are physical on-site inferences in our sense is an intriguing question.

3.3 Other Forms of Physical On-Site Inferences

Once the notion of physical on-site inference is thus clarified, it is obvious that there are at least two other forms of inferences that fit the definition. They are:

Physical non-consequence proof Express the set Δ of information in the representation system $\mathcal{R}$. By observing that the operation does not result in some condition σ that indicates the information θ, conclude that θ is not a $K_{\mathcal{R}}$-consequence of Δ.

Physical inconsistency proof Try to express the set Δ of information with its primitive indicator Γ. Conclude, from the impossibility of that operation, that Δ is inconsistent.

Remember that a free ride is the form of inference that utilizes a constraint matching of type 2 to conclude the existence of a constraint on represented situations from the existence of a constraint on representations. The first inferential procedure listed above is a flip side of this procedure, and it utilizes a constraint matching of type 2 to conclude the non-existence of a constraint on represented situations from the non-existence of a constraint on representations.

Here, the notion of $K_{\mathcal{R}}$-consequence is defined analogously as the notion of $K_{\mathcal{R}}$-inconsistency. A target type θ is an *induced consequence* of a set Δ of target types in a representation system $\mathcal{R}$ if some set Γ is projected to Δ in $\mathcal{R}$ and every set of source types projected to Δ in $\mathcal{R}$ entails at least one source type σ that indicates θ in $\mathcal{R}$; if every induced consequence is in fact a consequence of the relevant set of target types, an induced consequence in $\mathcal{R}$ is called $K_{\mathcal{R}}$-*consequence*. A constraint matching of type 2 is exactly this condition, and it is straightforward to prove that every physical non-consequence proof is valid in a system satisfying this condition.

As the name suggests, the second inferential procedure listed above is a flip side of graphical consistency proofs: while physical consistency proofs utilize a constraint matching of type 1 to conclude the non-existence of a constraint on represented situations from the non-existence of a constraint on representations, physical inconsistency proofs utilize a constraint matching of the same type to conclude the existence of a constraint on represented situations from the existence of a constraint on representations.

Although we do not have space to give actual examples of physical non-consequence proofs and physical inconsistency proofs, these forms of inferences are as common as free rides and graphical consistency proofs, and they are conductible on the basis of ordinary graphical representations such as Euler diagrams, route maps, line drawings, and geometry diagrams.

4 Summary

In this paper, we investigated the semantic mechanism of graphical consistency proofs, where one constructs a chart, a diagram, or some other external representation that expresses certain conditions, and uses its existence as a proof of the consistency of the expressed conditions. We found that an auto-consistency property responsible for a system's capacity of such a proof can be characterized as a matching of constraints (specified as type 1) between representations and represented situations. We then extended our analysis to another types of graphics-based inferences called "free rides," and showed that they also rely on a matching of constraints (specified as type 2) between representations and represented situations. Comparisons of these procedures let us see three commonalities between them, and define the general notion of physical on-site inference as procedures using perceptually present objects as inferential surrogates

through physical operations. Our analysis is therefore clarifications of the exact processes and semantic requirements under which a visual representation participate in distributed cognition (Giere, 2001) or manipulative inferences (Magnani 2001).

References

Barwise, J. and Etchemendy, J., 1994, *Hyperproof*, CSLI Publications, Stanford.

Barwise, J. and Etchemendy, J., 1995, Heterogeneous logic, in: *Diagrammatic Reasoning: Cognitive and Computational Perspectives*, J. I. Glasgow, N. H. Narayanan and B. Chandrasekaran eds., MIT Press and AAAI Press, Cambridge and Menlo Park, pp. 211-234.

Barwise, J. and Seligman, J., 1997, *Information Flow: the Logic of Distributed Systems*. Cambridge University Press, Cambridge.

Gelernter, H., 1959, Realization of a geometry-theorem proving machine, in: *Computers and Thought*, E. A. Feigenbaum and J. Feldman eds., McGraw Hill, New York.

Giere, R. N., 2001, Scientific cognition as distributed cognition, Manuscript to be published in: *Cognitive Bases of Science*. P. Carruthers, S. Stitch and M. Siegel, eds., Cambridge University Press, Cambridge.

Hutchins, E., 1995, *Cognition in the Wild*, MIT Press, Cambridge.

Lindsay, R. K., 1988, Images and inference, in: *Diagrammatic Reasoning: Cognitive and Computational Perspectives*, J. I. Glasgow, N. H. Narayanan and B. Chandrasekaran eds., MIT Press and AAAI Press, Cambridge and Menlo Park, pp. 111-135.

Magnani, L., 2001, *Abduction, Reason, and Science: Processes of Discovery and Explanation*, Kluwer Academic/Plenum Publishers, London.

Shimojima, A., 1995a, Reasoning with diagrams and geometrical constraints, in: *Language, Logic and Computation*, D. Westeståhl and J. Seligman eds., CSLI Publications, Stanford.

Shimojima, A., 1995b, Operational constraints in diagrammatic reasoning, in: *Logical Reasoning with Diagrams*, J. Barwise and G. Allwein eds., Oxford University Press, Oxford.

Sloman, A., 1971, Interactions between philosophy and AI: the role of intuition and non-logical reasoning in intelligence. *Artificial Intelligence* 2: 209–225.

Stenning, K. and Inder, R., 1995, Applying semantic concepts to analyzing media and modalities, in: *Diagrammatic Reasoning: Cognitive and Computational Perspectives*, J. I. Glasgow, N. H. Narayanan and B. Chandrasekaran eds., MIT Press and AAAI Press, Cambridge and Menlo Park, pp. 303-338.