The Conversion of Diagrams to Knowledge Bases

Robert P. Futrelle

Biological Knowledge Laboratory
College of Computer Science
Northeastern University
Boston, MA 02115
Internet: futrelle@corwin.ccs.northeastern.edu

Abstract

If future electronic documents are to be truly useful, we must devise ways to automatically turn them into knowledge bases. In particular, we must be able to do this for *diagrams*. This paper discusses biological diagrams.

We describe the three major aspects of diagrams: *visual salience*, *domain conventions* and *pragmatics*. We next describe the organization of diagrams into *informational* and *substrate* components. The latter are typically collections of objects related by *Generalized Equivalence Relations*.

To analyze diagrams, we define *Graphics Constraint Grammars* (GCGs) that can be used for both syntactic and semantic analysis. Each grammar rule describes a *rule object* and consists of a *Production*, describing the constituents of the object, *Constraints* that must hold between the constituents and *Propagators* that build properties of the rule object from the constituents. We discuss how a mix of parsing and constraint satisfaction techniques are used to parse diagrams with GCGs.

1. Introduction

Procedures are being developed in our laboratory to automatically convert biological research articles to a knowledge base that is then accessed by scientists using an interactive *Scientist's Assistant*. We work with a corpus of 1500 papers from the biological research literature, covering all of the field of bacterial chemotaxis from its beginning in 1965.

Our basic assumption is that diagrams and text in scientific papers are both forms of expository language and both need similar analysis. The emphasis in this paper will be on parsing diagrams (line drawings) using our *Graphics Constraint Grammars*. This is a major first step in converting them to knowledge bases. The approach is similar to [4, 6 and 8].

2. A biological diagram example

The example chosen for this discussion is the gene diagram in Figure 1, based on a diagram in [7]. Though x,y data graphs are the most common diagram in all scientific publications, we have already discussed them elsewhere [1-3]. Figure 1 shows three related gene segments, in schematic form. The figure includes an expansion of a portion of a segment to the level of the individual DNA bases, A, T, G and C.

3. Diagram organization

The effectiveness of diagrams is dependent on three factors: visual salience, domain conventions and pragmatics. These are best understood from the examples below, keyed to Figure 1. Understanding these three organizational principles is necessary if we are to write grammars and parsers to analyze diagrams.

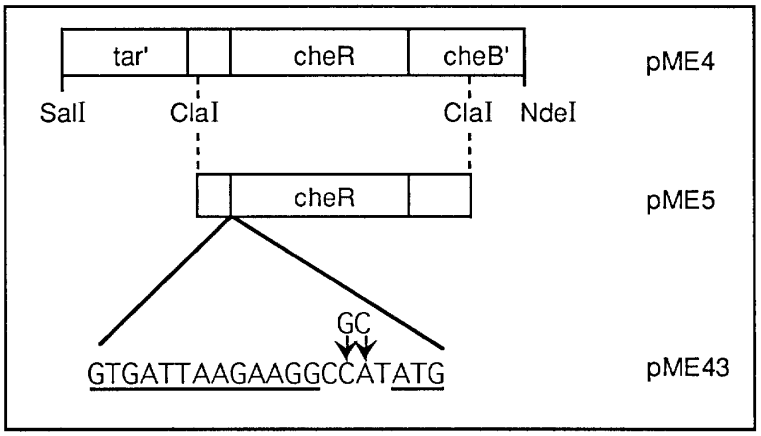

Figure 1 . A diagram showing three gene segments, after ref. [7]. Only a 20-base sequence region of pME43 that differs from pME5 is shown (the letters GTGA....). The diagram is strongly organized in a horizontal, vertical design.

1049-2615/92 $3.00 ©1992 IEEE

Visually salient organization describes how well the diagram exploits the nature of visual perception:

- Containment — the labels cheR in the boxes
- Alignment — horizontal and vertical alignment of the rectangles, alignment of labels such as SalI with the short labeling line above them and pME4 with the rectangle to its left
- Nearness — of the SalI label to the line it labels
- Connectivity — the diagonal lines connect the segment division line at the top to the ends of the detailed sequence below
- Horizontal/vertical organization — the entire diagram is organized in this way

Conventions of the domain are crucial for interpretation and disambiguation:

- Genes — indicated by segmented rectangles
- Cut sites — indicated by vertical lines labeled with restriction enzyme names
- Morphology of abbreviations — gene names such as cheR are three-letter lower case with additional letters as needed, etc., conventions that are universal in the biological community.

Pragmatics most often deals with "spatial real estate" — designs that fit the required elements into the space, preserving their relations:

- The rectangles are just large enough to contain their labels.
- The vertical lines that label the cut sites extend below the plasmid rectangle by about one character height
- Parsimony operates so that the same choices are made wherever possible, e.g., the heights of all the rectangles are the same.

4. Generalized Equivalence Relations

One of the major conventions in diagrams is the separation of the *informational* components from the *substrate* on which the information is presented. In Figure 1, the primary informational parameters are the order, length and labels of the rectangles and the labeling of their termini. Many other parameters in the diagram are not informational, e.g., the height of the rectangles (short vertical lines), the exact positions and font size of the labels, the vertical separation of the upper and middle gene segments, etc. These parameters are part of the substrate.

Many sets of substrate items can be described (and recognized) by *equivalence relations* — some of their parameters are equal. For example, the short vertical lines that border each rectangle are of equal length and, for a single gene segment, they all have the same vertical position on the page. Classes of items related by equivalence contain less information (in the Shannon, Kolmogorov sense) than the those with distinct parameters. In order to deal with phenomena of this type, we have extended the notion of equivalence relation to the *Generalized Equivalence Relation (GER)*. A GER may only demand approximate equality of parameters, e.g., *Coincident* is a true equivalence relation while the analogous *Near* is a GER. Also, a GER may include relations that are not normally thought of as equivalence relations, such as *Equal_spaced*. GERs are related closely to issues of visual salience.

5. Graphics Constraint Grammars

We describe diagrams using grammars that specify objects, and the object attributes and relations. In the *Graphics Constraint Grammars* we have developed [3], low-level elements are objects such as lines and polygons. High-level objects are more complex structures such as a *Connected_box_row*. These diagram grammars differ from others [4,6,8] because of our use of Generalized Equivalence Relations [3], hierarchical spatially indexed data structures [3], and constraint satisfaction techniques [5].

Graphics Constraint Grammars are collections of *rules* and each rule has three components:

- The *production* names the *rule object* on its left-hand side , and the *constituents* of the object on its right-hand side
- The *constraints* consist of spatial relations such as *Near, Horizontal, Aligned,* etc. as well as *type* constraints, e.g., that an object be a line or text.
- The *propagators* describe the relations between the attributes of the rule object and the attributes of the constituents.

The following simple rule could describe a labeled rectangle in a gene segment:

Production: Labeled_box => Label, Box
Constraints:
(Type-of Label 'text)
(Type-of Box 'rectangle)
(Orientation Box 'horizontal)
(Contains Box Label)
Propagators: Length <= (Length Box)

Note that the first three constraints are unary and the fourth is a binary relation that must hold between *Box* and *Label*. When the rule is satisfied, one or more *Labeled_box* objects are created, each with two constituents and a *Length* attribute.

Additional information about these grammars can be found in [3]. Instead of repeating that discussion, we'll discuss three additional issues.

6. Solution strategies for parsing

Solving (parsing) a Graphics Constraint Grammar is a potentially expensive computation. For each rule, a number of assignments of objects to constituents may have to be tried. For example, in the *Labeled_box* rule, any two pairs of vertical lines can define a rectangle, so for the top gene segment, there are 10 possible rectangles. Similarly, when the parser attempts to generate a set of horizontally aligned vertical line segments in the top gene segment (lines related by a GER), any set in the power set of the five lines would be a legitimate answer. We use *extremal principles* to deal with these problems. For GER constraints the parser always returns the *maximal* number of items that satisfy the constraint. For rectangles, it returns the *smallest* ones. Often, pragmatic "real estate" problems in diagram design lead to diagrams that demand more complex solution strategies.

8. Semantics — Massive low-level ambiguity

Graphic objects are very different from words. A simple item such as a rectangle is massively ambiguous. There is no "graphics dictionary" in which you can look up the meaning of "rectangle". We cannot depend on compositional semantics as we do in natural language, where the meaning of the whole is built up from the meaning of the parts.

To show how semantic analysis can be done in spite of these ambiguities, consider how the system could identify a *Labeled_box* as a gene segment portion. This can be done by domain conventions, e.g., in molecular biology, a *Labeled_box* has a high probability of being a gene segment portion. It can also be done by coupling to text semantic analysis, in this case by using a lexicon that tells us that cheR is a gene name and SalI is a restriction enzyme name. A rectangle labeled in this way could then be interpreted properly. Still another way would be to parse using high-level grammar productions such as: *Gene_segment => Connected_box_row*. The constituent on the right-hand-side is derived from syntactic analysis that considers only geometry. The rule object on the left has semantic meaning in the biological domain.

9. Conversion to knowledge bases

A grammatical analysis, even using semantic rules as described in the previous section, can at best produce a tree-structured description of a diagram. To allow the system to answer queries usefully, a richer knowledge base has to be built, e.g., a frame-based representation. An example of the type of knowledge that would be captured in this would be a frame for pME5, the second gene segment in Figure 1. The portion of the frame for pME5 that represents the part-whole relation would be:

pME5	
Part-of:	pME4
Has-all-of:	cheR
Has-parts-of:	left-of-cheR, cheB'
....	

where the system has generated the label, left-of-cheR.

Building knowledge frames is a complex enterprise. It consists first of building ontologies for object types and relations. Then tools are designed to map from parse trees to knowledge frames that take advantage of the ontologies. For example, gene segments would be in the class of objects that can participate in part-whole structures.

Acknowledgments

Thanks to the ERC for providing a productive research environment. This research was supported in part by NSF grant DIR-8814522.

References

1. Futrelle, R. P. (1985). A Framework For Understanding Graphics In Technical Documents. In *Expert Systems in Government Symposium* (pp. 386-390): IEEE Computer Society.

2. Futrelle, R. P. (1990). Strategies for Diagram Understanding: Object/Spatial Data Structures, Animate Vision, and Generalized Equivalence. In *Proceedings of the 10th ICPR* (pp. 403-408): IEEE Press.

3. Futrelle, R. P., Kakadiaris, I. A., Alexander, J., Carriero, C. M., Nikolakis, N., & Futrelle, J. M. (1992). Understanding Diagrams in Technical Documents. *Computer*(July, 1992), 75-78.

4. Helm, R., Marriott, K., & Odersky, M. (1991). Building Visual Language Parsers. In S. P. Robertson, G. M. Olson & J. S. Olson (Ed.), *CHI '91*,(pp. 105-112). New Orleans, Louisiana: Addison-Wesley Pub. Co.

5. Kumar, V. (1992). Algorithms for Constraint-Satisfaction Problems: A Survey. *AI Magazine*, **13**(1), 32-44.

6. Searls, D. B., & Liebowitz, S. A. (1990). Logic Grammars as a Vehicle for Syntactic Pattern Recognition. In *Pre-Proc. of SSPR-90*, (pp. 402-422). Murray Hill, NJ: Intl. Assoc. for Pattern Recognition.

7. Simms, S. A., Stock, A. M., & Stock, J. B. (1987). Purification and characterization of the *S*-adenosylmethionine:glutamyl methyltransferase that modifies membrane chemoreceptor proteins in bacteria. *J. Biological Chemistry*, **262**(18), 8537-8543.

8. Wittenburg, K., & Weitzman, L. (1990). Visual Grammars and Incremental Parsing for Interface Languages. In *Proc. of IEEE Workshop on Visual Languages*, (pp. 111-118). Skokie, IL.