An Introduction to Algebraic Semiotics, with Application to User Interface Design

Joseph Goguen

Dept. Computer Science & Engineering, Univ. of California at San Diego

Abstract: This paper introduces a new approach to user interface design and other areas, called *algebraic semiotics*. The approach is based on a notion of *sign*, which allows complex hierarchical structure and incorporates the insight (emphasized by Saussure) that signs come in *systems*, and should be studied at that level, rather than individually. A user interface can be considered as a *representation* of the underlying functionality to which it provides access, and thus user interface design can be considered a craft of constructing such representations, where both the interface and the underlying functionality are considered as (structured) sign systems. In this setting, representations appear as mappings, or *morphisms*, between sign systems, which should preserve as much structure as possible. This motivates developing a calculus having systematic ways to combine signs, sign systems, and representations. One important mode of composition is *blending*, introduced by Fauconnier and Turner; we relate this to certain concepts from the very abstract area of mathematics called category theory. Applications for algebraic semiotics include not only user interface design, but also cognitive linguistics, especially metaphor theory and cognitive poetics. The main contribution of this paper is the precision it can bring to such areas. Building on an insight from computer science, that discrete structures can be described by algebraic theories, **sign systems** are defined to be algebraic theories with extra structure, and **semiotic morphisms** are defined to be mappings of algebraic theories that (to some extent) preserve the extra structure. As an aid for practical design, we show that the quality of representations is closely related to the preservation properties of semiotic morphisms; these measures of quality also provide the orderings needed by our category theoretic formulation of blending.

1 Introduction

Analogy, metaphor, representation and user interface have much in common: each involves signs, meaning, one or more people, and some context, including culture; moreover each can be looked at dually from either a design or a use perspective. Recent research in several disciplines is converging on a general area that includes the four topics in the first sentence above; these disciplines include (aspects of) sociology, cognitive linguistics, computer science, literary criticism, user interface design, psychology, semiotics, and philosophy. Of these, semiotics takes perhaps the most general view, although much of the research in this area has been rather vague. A goal of the research reported here is to develop a mathematically precise theory of semiotics, called **algebraic semiotics**, that avoids the error of **reification**, that is, of identifying its abstractions with real world phenomena, making only the more modest claim of developing potentially useful models. This paper focuses on applications to user interface design, but the mathematical formalism also applies to the other areas mentioned above, especially metaphor theory and cognitive poetics, within cognitive linguistics.

The job of user interface designers is to build good metaphors (representations, translations, etc.). In this area, the domains to be represented are often very clear, though prosaic, the designers are often engineers, the intended users are often mass market consumers, and quality can often be tested, e.g., by laboratory experiments and statistics. Therefore user interface design provides a good laboratory for studying the general area that we have identified. It is interesting to contrast user interface design with (say) poetry, where the objects of interest are unique brilliant creations, and analysis is difficult (but rewarding). Nevertheless, they have much in common, including the applicability of semiotic morphisms and blends.

User interface designers have long wanted the same capability as electrical and mechanical engineers to make models and reason about them, instead of having to build prototypes and test them, because proper experiments can be both time consuming and expensive. Clearly this requires an effective understanding of what user interfaces are and what makes some better than others. A major difference from the more established engineering disciplines is that social factors must be taken into account in setting up the models. Therefore purely mechanistic procedures are unlikely to be achieved in the near future. My claims are that user interfaces are representations, that their quality is determined by what they preserve, and that this can be an effective basis for design.

User interface issues are exceedingly common, despite a persistent tendency to ignore them, to downplay their importance, or to minimize their difficulty. A coffee cup is an interface between the coffee and the coffee drinker; questions like thickness, volume, and handle shape are interface design issues. A book can be considered a user interface to its content. Buildings can be seen as providing interfaces to users who want to navigate within them, e.g., a directory in the lobby, buttons outside and inside the elevators, "EXIT" signs, doorknobs, stairways, and even corridors (you make choices with your body, not your mouse). A technical research paper can be seen as a user interface, that to succeed must take account of its intended user community. Returning to the obvious, medical instruments have user interfaces (for doctors, nurses, and even patients) that can have extreme consequences if badly designed. By perhaps stretching a bit, almost anything can be seen as a user interface; doing so will highlight certain issues of design and representation that might otherwise remain obscure, though of course it will not include all possible relevant issues.

User interface issues are also important in mathematics, and have been given particular attention in relation to choice of notation and to education. As Leibniz put it,

> In signs, one sees an advantage for discovery that is greatest when they

> express the exact nature of a thing briefly and, as it were, picture it; then, indeed, the labor of thought is wonderfully diminished.

A good example is the difference in plane geometry between doing proofs with diagrams and doing proofs with axioms (see also Appendix D). The above quotation also draws attention to *signs* and their use, and indeed, our previous discussion about coffee cups, elevator buttons, etc. can be re-expressed very nicely in the language of *semiotics*, which is the study of signs. Signs are everywhere: not just icons on computer screens and corporate logos on T-shirts or racing cars, but more significantly, the organization of signs is the very nature of language, natural human language both spoken and written, artificial computer languages, and visual languages, as in architecture and art, both fine and popular, including cinema.

We will see that the following ideas are basic to our general theory:

- Signs appear as members of *sign systems*[1], not in isolation.
- Most signs are complex objects, constructed from other, lower level signs.
- Sign systems are better viewed as *theories* – that is, as declarations for symbols plus sentences, called "axioms," that restrict their use – than as (set-based) models.
- Representations in general, and user interfaces in particular, are "morphisms" (mappings) between sign systems.

Charles Sanders Peirce [49], a nineteenth century logician and philosopher working in Boston, coined the word "semiotics" and introduced many of its basic concepts. He emphasized that meanings are not directly attached to signs, but that instead, signs *mediate* meaning, through events (or processes) of **semiosis**, each involving a **signifier** (i.e., a sign), a **signified** (an "object" of some kind – e.g., an idea), and an **interpretant**[2] that links these two; these three things are often called the *semiotic triad*, and occur wherever there is meaning. Signs, meanings, and referents only exist for a particular semiosis, which must include its social context; therefore meaning is always *embedded* and *embodied*. In general, the signified is not given, but must be inferred by some person or persons involved. Designers work in the reverse direction, creating signs for a given signified. Peirce's approach may sound simple, but it is very different from more common and naive approaches, such as the use of denotational semantics for programming languages. Peirce's theory of signs is not a *representational theory of meaning*, in which a sign has a denotation; instead, the interpretant makes it a *relational theory of meaning*. Peirce's important notions of icon, index and symbol are discussed below in Section 4. In addition, we use the term **signal** for a physical configuration that may or may not be a sign.

[1] There is a difficulty with terminology here: the phrase "semiotic system" sounds too broad, while "sign system" may sound too narrow, since it is intended to include (descriptions of) conceptual spaces, as well as systems of physical signs.

[2] This is Peirce's original terminology; "interpretant" should not be confused with "interpreter," as it refers to the link itself.

Ferdinand de Saussure [54] was a late nineteenth century Swiss linguist, whose work inspired the recent French structuralist and poststructuralist movements; if he were around today, he might not wish to be called a semioticist. Nevertheless, he had an important insight that is perhaps most clearly expressed using the language of signs: it is that signs are always parts of **sign systems**. He also gave perhaps the first, and certainly one of the most influential examples of a sign system, with his theory that phonemes, the smallest recognized units of spoken language, are organized into systems of binary oppositions[3], which may be thought of as features. More complex linguistic signs are then constructed from lower level signs: words ("lexemes") are sequences of phonemes; sentences are sequences of words; and tense, gender, number etc. are indicated by various syntactic features. (Recent research qualifies and modifies this classical view in various ways, but it is a useful model, still widely used in linguistics.)

Composing signs from other signs is a fundamental strategy for managing the complexity of non-trivial communication, regarding complex signs at one level as individual signs at a higher level. This is illustrated by the linguistic levels discussed above. A simple computer graphics example might have as its levels pixels (individual "dots" on the screen), characters, words, simple geometrical figures, and windows, which are collections of signs at lower levels plus other windows; each sign at each level has attributes for location and size, and perhaps for color and intensity. This whole/part hierarchical structure puts each sign in a context of other signs with which it forms still higher level signs. Note the recursivity in the definition of windows.

More recent uses of sign systems, for example in the classic literary study *S/Z* by Roland Barthes [4], tend to be less rigid than the linguistics of Saussure or the anthropology of Levi-Strauss. Instead of binary oppositions, there are multi-valued, even continuous, scales; instead of constructing higher level signs by sequential composition, there are more complex relations of interpenetration and influence; and perhaps most importantly, there is a much greater sensitivity to context. Indeed, the "structuralist" tendency of classical semiotics has been severely criticized by the post-structuralist and deconstructionist schools for its limited ability to deal with context. Although Lyotard, Derrida, Baudrillard, and the rest are surely correct in such criticisms, there is a danger of throwing out the baby of structure with the dirty bathwater of decontextualization. Although meaning, as human experience, certainly does not confine itself to rigid systems of features, however complexly structured, it is equally undeniable that we see structure everywhere, and not least in language.

Structure is part of our experience, and though seemingly more abstract than

[3] A sign system that has just one element can't convey any information, because there are no differences. For example, imagine if Paul Revere, in describing how lamps in the church tower would indicate British invasion plans for Boston, had said "One if by land and one if by sea." instead of "One if by land and two if by sea." More technically, with just one sign, the Shannon information content of a message is zero. If there are two or more signs in a system, there must be some systematic way to distinguish among them. Or as Gregory Bateson said, information is a difference that makes a difference.

immediate sensations, emotions, evaluations, etc., there is strong evidence that it too plays a crucial role in the formation of such experiences (e.g., consider how movies are structured). Context, which for spoken language would include the speaker, can be at least as important for meaning as the signs involved. For an extreme example, "Yes" can mean almost anything given an appropriate context. Moreover, work in artificial intelligence has found contextual cues essential for disambiguation in speech understanding, machine vision, and elsewhere.

The vowel systems of various accents within the same language show that the same sign system can be realized in different ways; let us call these different **models** of the sign system. For computer scientists, it may be helpful to view sign systems as **abstract data types**, because this already includes the idea that the same information can be represented in different ways; for example, dates, times, and sports scores each have multiple familiar representations. The Greek, Roman and Cyrillic alphabets show that the sets underlying models can overlap; this example also shows that a signal that is meaningful in one sign system may not be in another, even though they share a medium. The same signal in a different alphabet is a different sign, because it is in a different sign system. The vowel system example also shows that different models of the same sign system can use exactly the same signals in different ways; therefore it is how elements are used that makes the models different, not the elements themselves. Here are some further useful concepts:

– A **medium** expresses dimensions within which signs can vary; for example, standard TV is a two dimensional pixel array with certain possible ranges of intensity and color, plus a monophonic audio channel with a certain possible range of frequency, etc.
– A **genre** is a collection of conventions for using a medium; these can be seen as further delimiting a sign system. For example, the daily newspaper is a genre within the medium of multisection collections of large size pages. Soap operas are a genre for TV. Obviously, genres have subgenres; e.g., soap operas about rich families.
– **Multimedia** are characterized by multiple simultaneous perceptual channels. So TV is multimedia, and so (in a weak sense) are cartoons, as well as books with pictures.
– **Interactive media** allow inputs as well as outputs. So PCs are (potentially) interactive multimedia. The web provides (at least one) genre within this medium; email is another.

We can even say that a book is interactive, because users can mark and turn pages, and can go to any page they wish; indices, glossaries, etc. are also used in an interactive manner. Many museums have interactive multimedia exhibits, and every museum is interactive in a more prosaic sense.

This paper proposes a precise framework for studying sign systems and their representations, as well as for studying what makes some representations better than others, and how to combine representations. The framework is intended for application to aspects of communication and cognition, such as designing and understanding interfaces, coordinating information in different media, and choosing effective representations in, e.g., natural language, video clips, interactive media, etc. One goal is to get a calculational approach to user interface design, like that found in other engineering disciplines. Although our official name for this approach is **algebraic semiotics**, it might also be called **structural semiotics** to emphasize that meaning is structural, or (in its philosophic guise) even **morphic semiotics**, to emphasize that meaning is dynamic, contextual, embodied and social. In a sense, this paper proposes a general theory of meaning, although it denies the possibility of traditional context-independent meaning. The social nature of information is discussed in [21], using ideas from ethnomethodology [53].

Familiarity with (a little bit of) OBJ3 and algebraic specification is needed for the examples in Appendix A, and familiarity with basic category theory is needed for Appendix B; references for these two topics are [28] and [33, 16, 17], respectively. Most philosophical discussion has been banished to Appendix C, while Appendix D is an essay on the social nature of proofs, which provides more concrete illustrations of some points in Appendix C.

1.1 Semiotic Morphisms

One of the great insights of twentieth century mathematics, with consequences that are still unfolding, is that structure preserving morphisms are often at least as important as the structures themselves. For example, linear algebra is more fundamentally concerned with linear maps (often represented by matrices) between vector spaces, than with vector spaces themselves (though the latter are of course not to be despised); without giving details, there are also computable functions in recursion theory, embeddings and tangent maps in geometry, analytic and meromorphic functions in complex analysis, continuous maps in topology, and much more – all of them structure preserving maps.

This conceptual revolution took a more definite and systematic form with the invention of category theory in the early 1940's by Eilenberg and Mac Lane; see [41]. Technical developments within category theory have in turn spurred further and deeper uses of morphisms within mathematics, and more recently in applied fields like computer science. This process has not ceased, and applications continue to inspire new theory, such as the $\frac{3}{2}$-categories and $\frac{3}{2}$-pushouts that are discussed in Appendix B of this paper.

Semiotics has escaped this particular revolution, probably in part due to its increasing alienation from formalization during the relevant period. But I claim there is much to be gained from this unlikely marriage of semiotics and category theory (with cognitive linguistics as bridesmaid), not the least of which is a theory of representation that can be applied to topics of current interest, like user interface design, metaphor theory, and natural language understanding. The essential idea is that interfaces, representations, metaphors, interpretations, etc. are morphisms from one sign system to another.

A user interface for a computer system can be seen as a semiotic morphism from (the theory of) the underlying abstract machine (what the system does) to a sign systems for windows, buttons, menus, etc. [31]. A web browser can be seen as a map from HTML (plus JavaScript, etc.) into the capabilities of a particular computer on which it is running[4]. Metaphors can be seen as semiotic morphisms from one system of concepts to another [10, 12, 58]. A given text (spoken utterance, etc.) can be seen as the image under a morphism from some (usually unknown) structure into the sign system of written English (or spoken English, or whatever). Conversely, we may be given some situation, and want to find the best way to describe it in natural language, or in some other medium or combination of media, such as text with photos, or cartoon sequences, or video, or online hypertext or hypermedia [27].

In these and many other cases, representations are signs in one system that relate systematically to signs in another system. Generally it is just as fruitless to study representations of single signs as to study single isolated signs. For representations also occur in systems, just as signs do: usually there are systematic regularities in how signs of one system are represented as signs of another. Let us use the notation $M\colon \mathbf{S_1} \to \mathbf{S_2}$ for a morphism from sign system $\mathbf{S_1}$ to sign system $\mathbf{S_2}$. Of course, in all but the most trivial cases, there is no unique morphism $\mathbf{S_1} \to \mathbf{S_2}$. Think, for example, of the difficulties of translating from one language to another. Moreover in general, morphisms are partial, that is, not defined for all the signs in the source system; some signs may be untranslatable, or at least, not translated by a given morphism.

Here are some very simple examples. Let $\mathbf{N_1}$ be the familiar decimal Arabic numerals and let $\mathbf{N_2}$ be the Roman numerals. Then there is a natural morphism $M\colon \mathbf{N_1} \to \mathbf{N_2}$ but it is undefined for Arabic 0, since the Romans did not have the concept of zero. We can also consider transliterations between the English and Greek alphabets: then certain letters just don't map. Similarly, Scandinavian alphabets make some distinctions that the English alphabet does not; Chinese and Sanskrit raise still other problems. Ciphers (i.e., "secret codes") are also representations, simple in their input and output alphabets, but deliberately complex in their algorithmic construction.

Further examples and details about the systematic organization of signs are discussed later, but it should now be clear that an ambitious enterprise is being proposed, taking a wide interpretation of the notion of sign, and treating sign systems and their morphisms with great rigor. However, because this enterprise is still at an early stage, our examples cannot be both complex and detailed. Hoping that readers will forgive the ambition and effrontery of combining such diverse elements, I acknowledge the deep indebtedness of this work to its precedents, and hope to have the help of readers of this paper in developing its potential.

1.2 Some Related Work

An adequate survey of related work in semiotics, cognitive science, linguistics, user interface design, literary criticism, etc. would consume many volumes. We

[4] These two examples highlight the important but subtle point that theory morphisms go in the opposite direction from the maps of models that they induce; this duality is explained at an abstract level by the theory of institutions [24], but is well outside the scope of this paper.

have already mentioned the work of Peirce [49] and Saussure [54], whose influence is pervasive; this brief subsection only sketches a few especially closely related items of more recent vintage. First is joint work with Linde begun more than 15 years ago [27], which contains the seeds for the main ideas of this paper. Analogies and file names were studied by Gentner [14] and Carroll [8] respectively, using set-based formalisms that can capture structure, but without axioms, levels or constructors; Sacks' ethnomethodological notion of "category system" [52] seems similar to our notion of sign system, but is very informal. We build on work of cognitive linguists Lakoff, Johnson and others [40] on metaphors, and Fauconnier and Turner's exciting proposal of blending (also called "conceptual integration") as a fundamental cognitive operation for metaphor, grammar, etc. [11, 12]; see also [58]. Shneiderman [55] is a good textbook on user interface design, and Norman [48] gives a good overview of broader design issues. Latour [43] gives a fascinating case study of design emphasizing the importance of social context, and [38] contains a number of case studies in the area of computer systems design. Andersen [2] has done some fascinating work applying semiotics and catastrophe theory to the design of interactive systems.

1.3 On Formalization

Sapir said *all systems leak*; he was referring to the fact that no grammatical system has ever successfully captured a real natural language, but it is natural to generalize his slogan to the formalization of any complex natural sign system. There are always "loose ends"; some deep reasons for this, having to do with the social nature of communication, are discussed in [21]. Thus we cannot expect our semiotic models to be perfect. However, a precise description that is somewhat wrong is better than a description so vague that no one can tell if it's wrong. We do not seek to formalize actual living meanings, but rather to express our partial understandings more exactly. Precision is also needed to build computer programs that use the theory. I do not believe that meaning in the human sense can be captured by formal sign systems; however, human analysts can note the extent to which the meanings that they see in some sign system are preserved by different representations. Thus we seek to formalize particular understandings of analysts, without claiming that such understandings are necessarily correct, or have some ideal kind of Platonic existence.

Acknowledgements

The proofs in Appendix B were provided by Grigore Roşu, and the basic definitions were worked out in collaboration with Grigore Roşu and Răzvan Diaconescu. Further results on $\frac{3}{2}$-colimits should eventually appear in a separate paper. I wish to thank the students in my Winter 1998 class CSE 271 on user interface design, for their patience, enthusiasm, and questions. I also thank Gilles Fauconnier, Masako Hiraga, and Mark Turner for their valuable comments on earlier drafts of this paper, and Michael Reddy for intensifying my interest in metaphor, as I supervised his PhD thesis at the University of Chicago.

2 Sign Systems

Sign systems usually have a classification of signs into certain **sorts**[5], and some rules for combining signs of appropriate sorts to get a new sign of another sort; we call these rules the **constructors** of the system. Constructors may have **parameters**. For example, a "cat" sign on a computer screen may have parameters for the size and location of its upper lefthand corner; changing these values does not change the identity of the cat.

Constructors may have what we call **priority**: a **primary constructor** has greatest priority; **secondary** constructors have less priority than the primary constructor but more than any non-primary or non-secondary constructor; **tertiary** constructors, etc. follow the same pattern. Priority is a *partial* ordering, not total. Experiments of Goguen and Linde [27] (testing subjects after multimedia instruction in various formats about a simple electronic device) support the hypothesis that the reasoning discourse type [32] has a primary constructor that conjoins reasons supporting a statement[6].

Semiotics should focus on the structure of sign systems rather than on *ad hoc* properties of individual signs and their settings, just as modern biology focuses on molecular structures like DNA rather than on descriptive classification schemes. For example, formalizing the handwritten letter "a" (or the spoken sound "ah") in isolation, is both far harder and less useful than formalizing relations between written letters and words (or phonemes and spoken words).

It is natural to think of a sign system as a set of signs, grouped into sorts and levels, not necessarily disjoint, with "constructor" functions at each level that build new signs from old ones. But such a set-based approach does not capture the openness of sign systems, that there might be other signs we don't yet know about, or haven't wanted to include, because we are always involved in constructing only *partial* understandings. It is therefore preferable to view sign systems as *theories* than as pre-given set theoretic objects. This motivates the following:

Definition 1: A **sign system** S consists of:

1. a set S of **sorts** for signs, not necessarily disjoint;
2. a partial ordering on S, called the **subsort** relation and denoted $\leq$;
3. a set V of **data sorts**, for information about signs, such as colors, locations, and truth values;
4. a partial ordering of sorts by **level**, such that data sorts are lower than sign sorts, and such that there is a unique sort of maximal level, called the **top sort**;

[5] We deliberately avoid the more familiar word "type" because it has had so many different uses in computer science. The so called parts of speech in syntax, such as noun and verb, are one example of sorts in the sense that we intend.

[6] The primary constructor of a given discourse type is its "default" constructor, i.e., the constructor assumed when there is no explicit marker in the text. In narrative, if one sentence follows another we assume they are connected by a SEQUENCE constructor; this is called the *narrative presupposition* [39].

5. a set C_n of level n **constructors** used to build level n signs from signs at levels n or less, and written $r\colon\ s_1...s_k d_1...d_l \to s$, indicating that its ith argument must have sort s_i, its jth parameter data sort d_j, and its result sort is s; constants $c\colon \to s$ are also allowed;
6. a **priority** (partial) **ordering** on each C_n;
7. some relations and functions on signs; and
8. a set A of sentences (in the sense of logic), called **axioms**, that constrain the possible signs.

$\square$

We can illustrate some parts of this definition with a very simple *time of day* sign system. It has just one sort, namely time, and just two constructors, one the constant time 0 (for midnight), and the other a successor operation s, where for a time t, $s(t)$ is the next minute. There are no subsorts, data sorts, levels, or priorities. But there is one important axiom,

$$s^{1440}(t) = t\ ,$$

where s^{1440} indicates 1440 applications of s, or more prosaically[7],

$$s^{1440}(0) = 0\ .$$

These axioms capture the cyclic nature of time over a day; any reasonable representation for time of day must satisfy this condition. Let's denote this sign system **TOD**.

An example illustrating some further parts of Definition 1 is a 24 line by 80 character display for a simple *line-oriented text editor*. The main sorts of interest here are: `char` (for character), `line`, and `window`. The sort `char` has two important subsorts: `alphanum` (alphanumeric) and `spec` (special); and `alphanum` has subsorts `alpha` and `num`. Among the special characters should be characters for space and punctuation, including comma, period, etc. The subsort relations involved here have the following graph,

```
            char
           /    \
   alphanum  spec
    /      \
alpha     num
```

where of course `alpha` and `num` are also subsorts of `char`. These sorts have levels in a natural way: `window` is the most important and therefore[8] has level 1, `line` has level 2, `char` has level 3, `alphanum` and `spec` have level 4, and `alpha` and `num` have level 5 (or we could give all subsorts of `char` level 4, or even 3; such choices can be a bit arbitrary until they are forced by some definite application). There are various choices for the constructors of this sign system. Since lines are strings of characters, one choice is an operation $\frown$ that concatenates a character with a line to get a longer line, and another operation, also denoted

[7] An additional assumption that is explained later is needed to show the equivalence of these two axioms.

[8] This assumes the ordering of sorts by level takes 1 as the maximum level.

$\frown$, that concatenates a line and a window to get another window; there must also be constant constructors for the empty line and the empty window. (The constraints on the lengths of lines and windows are given by axioms that are discussed below.) For each sort, the concatenation operations have priority over the constant operations.

This editor also has **data sorts** for fixed data types that are used in an auxiliary way in describing its signs: these include at least the natural numbers, and possibly colors, fonts, etc., depending on the capabilities we want to give our little editor. Functions include `windowidth` and `windowlength`, and there could also be predicates for the subsorts, such as a `numeric` predicate on characters. Then the constraints of length can be expressed by the following axioms:

$(\forall L : \texttt{line})\ \texttt{windowidth}(L) \leq 24$.
$(\forall W : \texttt{window})\ \texttt{windowlength}(W) \leq 80$.

Let us denote this sign system **W**.

If we want to study how *texts* can be displayed in this window, we should define a sign system for texts. One simple way to do this has sorts `char`, `word`, `sent` (sentence), and `text`, in addition to the data sorts and the subsorts of `char` as in **W** above; the sort `text` is level 1, `sent` level 2, `word` level 3, and `char` level 4. There are several choices for constructors, one of which defines any concatenation of alphanumeric characters to be a word, any concatenation of words to be a sentence, and any concatenation of sentences to be a text. Let us denote this sign system **TXT**. Clearly there are many different ways to display texts in a window, and each one is a different semiotic morphism; we will see some of these later.

A somewhat different sign system is given by simple *parsed sentences*, i.e., sentences with their "part of speech" (or syntactic category) explicitly given. The most familiar way to describe these is probably with a context free grammar like that below, where `S`, `NP`, `VP`, `N`, `Det`, `V`, `PP` and `P` stand for sentence, noun phrase, verb phrase, noun, determiner, verb, prepositional phrase, and preposition, respectively:

```
S  -> NP VP
NP -> N
NP -> Det N
VP -> V
VP -> V PP
PP -> P NP
.....
```

The "parts of speech" `S`, `NP`, `VP`, etc. are the *sorts* of this sign system, and the rules are its constructors. For example, the first rule says that a sentence can be constructed from a `NP` and a `VP`. There should also be some constants of the various sorts, such as

```
N -> time
N -> arrow
```

```
V -> flies
Det -> an
Det -> the
P -> like
......
```

There is a systematic way to view context free rules as operations that "construct" things from their parts (introduced in [15]), which in this case gives the following:

```
  sen : NP VP -> S
  nnp : N     -> NP
  np  : N Det -> N
  vvp : V     -> VP
  vp  : V PP  -> VP
  pp  : P NP  -> PP
  .....
 time :       -> N
flies :       -> V
  .....
```

It is a more elegant use of the machinery we have available to regard `N` as a subsort of `NP`, and `V` as a subsort of `VP`, than to have monadic operations `N -> NP` and `V -> VP`. Let's call the resulting sign system **PS**. It gives what computer scientists call *abstract syntax* for sentences, without saying how they are to be realized. We can of course still get "real" sentences, such as "time flies like an arrow", but this traditional linear form fails to show the syntactic structure, which is typically done using trees, as in

```
           S
          / \
        NP   VP
       /    /  \
      N    V    PP
     /     |    / \
  time flies  P   NP
             /   /  \
          like an  arrow
```

Another approach is to view a sentence as a "term" involving the operations above (terms are compositions of constructors); here's how our little sentence looks in that notation:

`sen(time, vp(flies, pp(like, np(an, arrow))))` .

So called bracket (or bracket-with-subscript) notation, as used in linguistics, also shows syntactic structure; it is surely a bit harder to read, and looks like this:

$$[[time]_{\mathrm{N}}[[flies]_{\mathrm{V}}[[like]_{\mathrm{P}}[[an]_{\mathrm{Det}}[arrow]_{\mathrm{N}}]_{\mathrm{NP}}]_{\mathrm{PP}}]_{\mathrm{VP}}]_{\mathrm{S}} \ .$$

(Another example of bracket notation appears in Section 4). In this setting, we can also use equations to express constraints on sentences, for example, that the number of the subject and of the verb agree (i.e., both are singular or both are plural).

Each of these concrete ways to realize abstract syntax (trees, terms, bracket notation, and lists) can be considered to give a *model* of the sign system, providing a *set* of signs for each sort, and operations on those sets which build new signs from old ones. You might have expected this to be the definition of sign system, instead of what we gave, which is a *language* for talking about such models. Our sign systems are *theories* rather than models. The distinction is that a model provides concrete *interpretations* for the things in the theory: sorts are interpreted as sets; constant symbols are interpreted as elements; constructors are interpreted as functions, etc. This allows much flexibility for what can be a model.

We often wish to exclude models where two different terms denote the same thing; otherwise, for example, two different times of day might be represented the same way[9]. This is called the **no confusion** condition; more precisely, it says that if two terms cannot be proved equal using axioms in the theory, then they must denote different elements in the model. Also it is often desirable to restrict to models where all signs are denoted by terms in their theory; these are called the **reachable** models[10]. An important point brought out in the next section is that semiotic morphisms do the same conceptual work as models, but in a way that is more convenient for many purposes.

It has been shown that any computable model can be defined using only equations as axioms[11]. Therefore we lose no generality by using equational logic for examples, as has been advocated in the situated abstract data type approach described in [20]. More precisely, our examples (in the appendices) use order sorted equational logic over a fixed data algebra [18, 29], although the reader does not need to be familiar with the technicalities of this logic.

What Fauconnier and Turner [10, 12] call **conceptual spaces** are also sign systems, of a rather simple kind, where there are (usually) no constructors except constants, and where in addition, there are some relations defined among these constants. Typical conceptual spaces are little theories of some everyday concept, including only as much detail as is needed to analyze some particular text. For example, a theory of houses might have constants `house`, `owner` and `resident`, with relations `own` and `live-in` making the obvious assertions. Similarly, a boat theory might have constants `boat`, `owner` and `passenger`, with relations `own` and `ride`. These two spaces are illustrated in Figure 1. No sorts are shown, but for

[9] But see the 12 hour analog clock representation denoted A below, where this *seems* to happen.

[10] This is also called the **no junk** condition. It is now an interesting exercise to prove that the two different equations previously given for expressing the cyclic nature of days are equivalent for reachable models.

[11] Although an additional technical condition called initiality is needed; see [47] for a survey of this and related results.

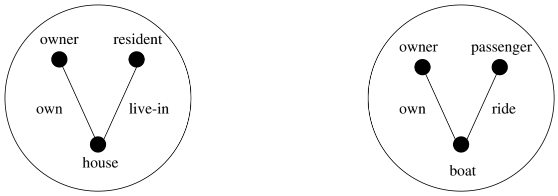


Fig. 1. Two Simple Conceptual Spaces

this simple example, one is enough, say `Thing`. That a relation such as `own`, holds of two things is given by a line in the figure, and in the corresponding logical theory is given by an axiom, e.g., `own(owner,house)`. It is usually assumed that relation instances that are not shown (such as `ride(boat,owner)`) do not hold, i.e., are false (one way to formalize this, which is related to the so called *frame problem* in artificial intelligence, is given in Chapter 8 of [23]). Let us call this the **default negation** assumption. But sometimes whether or not a relation holds may be unknown. Humans generally do a good job of figuring all this out, using what is called "common sense". However, the deductions involved can actually be extremely complex; some hints of this complexity may be found in the discussion of the blending examples in Section 5.

Formalism and representation feature in much recent work in sociology of science, with many fascinating examples. For example, Latour [42] shows how representation by cartographic maps was essential for European colonization, and Bowers [6] discusses the politics of formalism, including CSCW systems. Latour leaves representation undefined, while Bowers has a slightly formal notion of formalism. I believe that such discussions could be given greater precision by using the framework proposed in this paper.

3 Semiotic Morphisms

The purpose of semiotic morphisms[12] is to provide a way to describe the movement (mapping, translation, interpretation, representation) of signs in one system to signs in another system. This is intended to include metaphors as well as representations in the more familiar user interface design sense. Generating a good icon, file name, explanation or metaphor, or arranging text and graphics together in an appropriate way, each involves moving signs from one system to another. Just as we defined sign systems to be theories rather than models, so their morphisms are between theories, translating from the *language* of one sign system to the language of another, instead of just translating the concrete signs in the models. This may sound a bit indirect, but it has important advantages

[12] Although the root "morph" of the noun "morphism" means "form," this word has recently also become a verb meaning "to change form."

over a model based approach; moreover, theories and their morphisms determine models and their mappings.

A good semiotic morphism should preserve as much of the structure in its source sign system as possible. Certainly it should map sorts to sorts, subsorts to subsorts, data sorts to data sorts, constants to constants, constructors to constructors, etc. But it turns out that in many real world examples, not everything is preserved. So these must all be *partial* maps. Axioms should also be preserved — but again in practice, sometimes not all axioms are preserved.

Definition 2: Given sign systems S_1, S_2, a **semiotic morphism** $M\colon S_1 \to S_2$, from S_1 to S_2, consists of the following partial functions (all denoted M):

1. sorts of $S_1 \to$ sorts of S_2,
2. constructors of $S_1 \to$ constructors of S_2, and
3. predicates and functions of $S_1 \to$ predicates and functions of S_2,

such that

1. if $s \leq s'$ then $M(s) \leq M(s')$,
2. if $c\colon s_1 ... s_k \to s$ is a constructor (or function) of S_1, then (if defined) $M(c)\colon M(s_1) ... M(s_k) \to M(s)$ is a constructor (or function) of S_2,
3. if $p\colon s_1 ... s_k$ is a predicate of S_1, then (if defined) $M(p)\colon M(s_1) ... M(s_k)$ is a predicate of S_2, and
4. M is the identity on all sorts and operations for data in S_1.

More generally, a semiotic morphism can map source system constructors and predicates to compound terms defined in the target system[13]. □

A semiotic morphism $S_1 \to S_2$ gives representations in S_2 for signs in S_1. If we know how a semiotic morphism maps constructors, then we can compute how it maps complex signs. For example, if $M(a) = a', M(b) = b', M(c)(x, y) = c'(x, y + 1) + 1$, and $M(f)(x, y) = x + y + 1$, then

$$M(c(a, f(3, b))) = c'(a', b' + 5) + 1 \ .$$

We now consider some examples. First, suppose we want to represent time of day, **TOD**, in the little window, **W**. Clearly there are many ways to do this; each of them must map the sort time to the sort window, map the constructor 0 to some string of (less than 25) strings of (less than 81) characters, and map the constructor s to a function sending each such string of strings to some other string of strings. There isn't anything else to preserve in this very simple example except the axiom, which however is very important here.

Recall that the items of abstract syntax in **TOD** are strings of up to 1439 s's followed by a single 0. One simple representation just maps these strings directly to strings of strings of s's plus a final 0, such that the total number of s's is the same; this is a kind of unary notation. Let $N(t)$ be the number of s's in some t from **TOD**. Let $Q(t)$ and $R(t)$ be the quotient and remainder after dividing $N(t)$ by 80. Then there will be $Q(t)$ lines of 80 s's followed by one line of $R(t)$ s's and a final 0. This is guaranteed to fit in our window because $Q(1439) = 17$

[13] This is illustrated by $M(c)$ is the example just below this definition.

is less than 24, and $R(t)+1 \leq 80$. For humans, this representation is so detailed that it is more or less analog: I think after getting familiar with it, a user would have a "feel" for the approximate number of (these strange 80 minute) hours in a window and of minutes in the last line, just from its appearance. Let us call this representation U. Figure 2 shows the time that we would call "1:15 pm" in it.

```
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss0
```

Fig. 2. A Strange "Unary" Clock

Another obvious but naive representation just displays $N(t)$ in decimal notation, giving a string of 1 to 4 decimal digits. This is very different from our usual representations; but we could imagine a culture that divides its days into 14 "hours" each having 100 minutes, except the last hour, which only has 40 (this is less strange than what we do with our months, with their varying numbers of days!). Here $N(0)$ is 0, and `s` just adds 1, except that $\mathtt{s}(1439) = 0$. Figure 3 shows quarter after one in the afternoon in this representation; the last two digits give the number of minutes, and those to the left of that give the number of "hours". Let us call this representation N.

$\boxed{795}$

Fig. 3. A Naive Digital Clock

A more familiar representation is constructed as follows: Let $N1$ and $N2$ be the quotient and remainder of N divided by 60, both in base 10, with 0's added in front if necessary so that each has exactly 2 digits. Now form the string of characters "`N1 : N2`". This is the so-called "military" representation of time; let's denote it by M. Then $M(0) =$ `00:00`, and of course you know how `s` goes. Figure 4 shows our usual afternoon time in a slight variant of this representation.

$\boxed{13}\boxed{15}$

Fig. 4. A Military Time Clock

Notice that this representation has been defined as a composition of N with a re-representation of **TOD** to itself. The spoken variant of military time has the form "$N1$ hundred $N2$ hours" (unless $N2 = \mathtt{00}$, in which case N_2 is omitted). The use of "hundred" and "hours" may seem odd here, because it isn't hundreds and it isn't hours! — but at least it's clear — and that's the point. Part of this clarity stems from the phonology: the aspirated "h" sound at the beginning of "hundred" and "hour" does not occur in any of the numerals, and hence makes a good separator, especially over radio, where it gets exaggerated.

Readers should now be able to construct other representations of time as semiotic morphisms, including even the "analog" representation of a clock face. Here 0 has both hands up, satisfaction of the axiom follows because something even stronger is true, namely $\mathtt{s}(719) = 0$, which is built into the circular nature of this geometrical representation. This is an example where no confusion seems to fail – but does it really?[14] Let's call this representation A.

Now let's consider displaying parsed sentences, from the sign system **PS**, in the window **W**; this means constructing a semiotic morphism E from **PS** to **W**. One issue is placing spaces so that words are always separated. The designer will also have to do something about words that want to go past the end of a line; choices include wrapping around, hyphenating, and filling in with spaces; the limit of 24 lines will also pose problems. This can clearly get complex. The designer may also want to consider more sophisticated issues, such as automatic insertion of commas to show some of the syntactic structure.

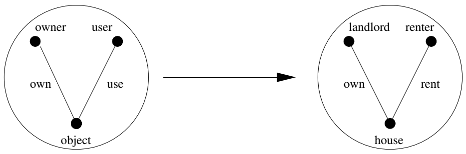


Fig. 5. A Conceptual Space Morphism

Morphisms of conceptual spaces are relatively simple; we map sorts to sorts, constants to constants and relations to relations, so as to preserve the sorts of constants and relations. Because assertions that relations hold are given by axioms, preservation of axioms implies preservation of the corresponding relation instances, positive or negative. If no axiom is given for some relation instance in the source theory, then either value is possible, and since there is nothing to preserve, the same holds for the target theory. The pairs of constants, one from

[14] Think about why a clock modulo 720 might work, but (say) modulo 120 or modulo 300 would not work; what about modulo 360? Why? Hint: consider the context in which the representation is typically used. Answers to these exercises appear in Section 4.

the source and one from the target, that are mapped by a semiotic morphism are called **connectors** by Fauconnier and Turner [10, 12]. For example, there is a connector from `object` to `house` (though we have not drawn it, as Fauconnier and Turner would).

A natural task in the experiments of [27] is to describe the condition of four lights on the face of an experimental device. This involves constructing a sequence of clauses arranged in narrative order, and can be seen as a semiotic morphism from the sign system of lights to that of English; a typical sentence would (in part) have the form "the first light is on, the second is off, ...".

For another example, a source sign system might contain instructions for repairing some piece of equipment, with target sign system a non-color graphics screen plus a speech chip. We can then ask how to generate instructional material using these particular capabilities in the most effective way.

Star [56] introduced the notion of **boundary object**, which (for our purposes) can be seen as a sign system that is interpreted in different ways by different social groups. These interpretations can be seen as semiotic morphisms from the sign system of the boundary object into the more specific sign systems of each group. For example, bird sightings are taken in different ways by amateur and professional ornithologists; the former make lists, emphasizing rare birds in friendly competition with other amateurs, whereas the latter integrate sightings with much larger datasets to construct migration patterns, population densities, long term trends, etc. A sighting may first appear as a field notebook entry, and then move into the two very different contexts. (A similar example is given in [56].)

3.1 First Steps Towards a Calculus of Representation

Semiotic morphisms can be composed; this is important because it is very common to compose representations, for example, in processes of iterative design, where details are added in progressive stages. The composition of semiotic morphisms satisfies some simple equations, which show that sign systems together with semiotic morphisms form what is called a *category* in mathematics; surprisingly, this is enough to define some important concepts and derive their main properties (much more information on the category of sign systems is given in Appendix B).

Given semiotic morphisms $M\colon S_1 \to S_2$ and $M'\colon S_2 \to S_3$, then their **composition**, which is denoted[15] $M;M'\colon S_1 \to S_3$, is formed just by composing the component functions of each morphism, not forgetting that these are partial functions, so that the result is also partial. For example, the sort set function for $M;M'$ is the composite of the sort set functions of M and M'. Then it is easy to see that $M;M'$ is also a semiotic morphism, and that if in addition we are given $M''\colon S_3 \to S_4$, then

[15] We use the symbol ";" for composition, to indicate the opposite order from that associated with the usual symbol "$\circ$" – that is, "$f;g$" means first do f, then do g, as with commands in a programming language.

$$(M;M');M'' = M;(M';M'') \ ,$$

i.e., composition of semiotic morphisms is **associative**. Moreover, for every sign system S, there is an **identity** semiotic morphism $1_S\colon S \to S$, which is defined to have as its component functions the identities on each set from S. This trivial morphism is rarely useful in practical design, but it plays an important role in the theory, just as do identity elements in algebra. For any $M\colon S_1 \to S_2$, the following equations are satisfied,

$$1_{S_1};M = M \ ,$$
$$M;1_{S_2} = M \ .$$

Composition of semiotic morphisms can be used to factor design problems, i.e., to separate concerns into different stages. For example, designing a text editor using the window **W** involves constructing a semiotic morphism E from **TXT** to **W**. In addition to the issues already mentioned for displaying sentences, we need to mark the boundaries of sentences, e.g., with a period followed by two spaces at the end. Issues that concern separating units, e.g., adding spaces and periods, can be separated from issues that concern the size of the window, e.g., hyphenation, by "factoring" the morphism E into two morphisms, $E1\colon$ **TXT** $\to$ **STG** and $E2\colon$ **STG** $\to$ **W**, where **STG** is a sign system of strings of characters, and E is the composition $E1;E2$.

Composition and identity of semiotic morphisms allow us to define the notion of **isomorphism** for sign systems, as a morphism $M\colon S_1 \to S_2$ such that there exists an **inverse** morphism $M'\colon S_2 \to S_1$ such that $M;M' = 1_{S_1}$ and $M';M = 1_{S_2}$. Isomorphic sign systems define exactly the same structure, and have the same models. Remarkably, it now follows from just the associative and identity laws that the inverse is unique if it exists, and that the relation of **isomorphy** (i.e., of being isomorphic) on sign systems is reflexive, symmetric and transitive. If we write $\cong$ for the isomorphy relation, then these facts may be written

$$S \cong S \ ,$$
$$S_1 \cong S_2 \ \text{ implies } \ S_2 \cong S_1 \ ,$$
$$S_1 \cong S_2 \text{ and } S_2 \cong S_3 \ \text{ implies } \ S_1 \cong S_3 \ .$$

Moreover, if we denote the inverse of M by M^{-1}, then the following equations are also easily proved,

$$(1_S)^{-1} = 1_S \ ,$$
$$(M^{-1})^{-1} = M \ , \ \text{ and}$$
$$(M;M')^{-1} = M'^{-1};M^{-1} \ .$$

The last equation particularly has some not entirely trivial content, and can be useful in thinking about the composition of isomorphism representations.

4 The Quality of Semiotic Morphisms

The goal of user interface design is to produce high quality representations; unfortunately, it has not been very clear how to determine quality. Also, as in other areas of engineering, design is subject to constraints, and typically involves tradeoffs, i.e., compromises between competing measures of success, such as cost, size, complexity and response time. Limits on human capability for dealing with complex displays implies that some information may have to be compressed, deleted, or moved elsewhere. This in turn implies that we need to understand priorities on what should be preserved.

In determining what makes one representation better than another, the entire structure of the sign systems involved should be considered. The structure that is preserved by semiotic morphisms provides an important way to compare their **quality**. First, notice that because a semiotic morphism $M: S_1 \to S_2$ need not be total, some signs in S_1 may have no representation in S_2; moreover, some of the complex internal structure of signs in S_1 could be lost. This might seem undesirable, but if representations in S_2 get too complex, they will not be useful in practice. For example, if S_1 is English sentences and S_2 is bracket notation, then the representation (from the data of [27])

$$[[[[the]_{Det}[light]_N]_{NP}[[on]_{Prep}[[the]_{Det}[left]_N]_{NP}]_{PP}]_{NP}[[comes]_V[on]_{Part}]_{VP}]_{Sent}$$

is not as useful for human communication as the linear representation would be. In fact, we very often want what Latour [42] calls a *re-representation*, which concentrates or abstracts information. For example, the representation "$[NP\ PP]\,VP$" is more useful than that above for some purposes, precisely because it *omits* some information. Statistics, such as the mean and median of a population, are also re-representations in this sense, as are cartographic maps.

Peirce [49] introduced a well-known three-fold classification of signs into icon, index, and symbol. These terms have precise technical meanings that differ from their everyday use. Peirce defined an **icon** as a "sign which refers to the Object that it denotes merely by virtue of characters of its own ... such as a lead-pencil streak representing a geometrical line." In contrast, a sign x is an **index** for an object y if x and y are regularly connected, in the sense "that always or usually when there is an x, there is also a y in some more or less exactly specifiable spatio-temporal relation to the x in question" [1]. "Such, for instance, is a piece of mould with a bullet-hole in it as sign of a shot" [49]. In this example, the spatio-temporal relation is a causal one, which applies with great generality. However many indices only work in very particular spatio-temporal contexts, e.g., the use of first names for persons. Finally, Peirce defines a **symbol** as a "sign which is constituted a sign merely or mainly by the fact that it is used and understood as such." In addition, we use the term **signal** for a physical configuration that may or may not be a sign.

Thus, an iconic representation preserves some important *properties* of signified signs; for a semiotic morphism, these might appear as axioms and/or data valued functions (for which the word "attribute" is commonly used in the object oriented community). An indexical representation participates in some larger situation (i.e., theory) within which we can deduce the connection between the signified and signifying signs. For a symbol, there is no such more basic relationship between source and target signs.

For purposes of design, other things being equal, there is a natural ordering to these three kinds of sign: icons are better than indices, and indices are better than symbols. However, things are *not* always equal. For example, base 1 notation for natural numbers is iconic, e.g., 4 is represented as `||||`, 3 as `|||`, and we get their sum just by copying and appending,

`|||| + ||| = |||||||` ,

which is iconic. But base one notation is very inefficient for representing large numbers. With Arabic numerals, the use of 1 for "one" is iconic (one stroke), but the others are symbolic[16]. Using the blank character for "zero" would be iconic, but of course this would undermine the positional aspect of decimal notation and introduce ambiguities. Chinese notation for several of the small numerals is iconic.

Peirce's three classes of sign overlap, so some signs will be hard to classify. Also, complex situations may involve all three kinds of sign, interacting in complex ways; indeed, different aspects of the same sign can be iconic, indexical, and symbolic. It is often necessary to consider the context of a sign, e.g., how is it used in practice, and of course its relation to other signs in the same system. See [19, 35] for further examples and discussion, the former mainly from computer science, and the latter mainly from language.

The following definition gives some precise ways to compare the quality of representations:

Definition 3: Given a semiotic morphism $M: S_1 \to S_2$, then:

(1) M is **level preserving** iff the partial ordering on levels is preserved by M, in the sense that if sort s is lower level than sort s' in S_1, then $M(s)$ has lower (or equal) level than $M(s')$ in S_2.
(2) M is **priority preserving** iff $c < c'$ in S_1 implies $M(c) < M(c')$ in S_2.
(3) M is **axiom preserving** iff for each axiom a of S_1, its translation $M(a)$ to S_2 is a logical consequence of the axioms in S_2.
(4) Given also $M'\colon S_1 \to S_2$, then M' is (at least) **as defined as** M, written $M \subseteq M'$, iff for each constructor c of S_1, $M'(c)$ is defined whenever $M(c)$ is.
(5) Given also $M'\colon S_1 \to S_2$, then M' **preserves all axioms that** M does, written $M \preceq M'$, iff whenever M preserves an axiom, then so does M'.
(6) Given also $M'\colon S_1 \to S_2$, then M' is (at least) **as inclusive as** M iff $M(x) = x$ implies $M'(x) = x$ for each sign x of S_1.
(7) Given also $M'\colon S_1 \to S_2$, then M' **preserves** (at least) **as much content as** M, written $M \ll M'$, iff M' is as defined as M and M' preserves every selector that M does, where a morphism $M: S_1 \to S_2$ **preserves** a **selector** f_1 of S_1 iff there is a selector f_2 for S_2 such that for every sign x of S_1 where M is defined, then $f_2(M(x)) = f_1(x)$, where
(8) a **selector** for a sign system S is a function $f\colon s \to d$, where s is a sign sort and d a data sort of S, such that there are axioms A' such that adding f and A' to S is consistent and defines a unique value $f(x)$ for each sign x of sort s. For example, each parameter of a constructor has a corresponding selector to extract its value.

[16] Though there is a trick for regarding several of the small Arabic numerals as symbolic.

□

The intuition for (7) is that content is preserved if there is some way to retrieve each data value of the source sign from its image in the target sign system; the definition of selector in (8) is unfortunately rather technical.

It may be that neither M nor M' preserves strictly more than the other; for example, M might preserve more constructors while M' preserves more content. Also, each of these orderings is itself partial, not total. Still other orderings on morphisms than those defined above may be useful for some applications; for example, special measures may be important at certain levels of some signs systems, such as phonological complexity (which is the effort of pronunciation) for spoken language. In general, specific "designer orderings" which combine various preservation properties in a specific way, may be needed to reflect the design tradeoffs of specific applications (e.g., see the end of Appendix B). As a result of this, given sign systems S_1, S_2, we can assume a partial ordering on the collection of semiotic morphisms from S_1 to S_2, as is needed for the $\frac{3}{2}$-categories of Appendix B.

We can see some of the complexities involved in comparing the quality of representations by considering simple examples where there is not very much structure to preserve. For example, in the time of day representations, simplicity, uniformity, and precision of the display are important: the naive decimal representation N lacks uniformity in the size of its "hours"; the strange unary representation U lacks precision (at least for humans who refuse to count very carefully) as well as simplicity and (to an extent) uniformity. The representations that are most straightforward mathematically may not be very close to the ones we use every day; for example, military time and analog clock time require mathematically more complex operations for their definition than do the decimal and strange unary representations.

Now let's consider the no confusion condition in regard to the cyclicity of clocks. The military clock M satisfies the condition. But for the standard 12 hour analog clock A, we could say it is only "half satisfied," because one extra bit, as found on most watches, to indicate "am" or "pm," is necessary and sufficient to avoid confusion. Of course this extra bit is often available just by looking out the window to see if it's day or night; but if you lived underground for a few weeks with just a 12 hour clock and no other information (such as radio), you might well lose track of that one bit. A 6 hour analog clock would only satisfy the no confusion condition only "one quarter," because two extra bits are needed. A 5 or 7 or 17 hour analog clock would be much worse, because these numbers are relatively prime to 24. An alternative way to talk about this is to say that a selector for the number of elapsed minutes from midnight can be defined that is not preserved. So although the general rule is that the more preservation the better, sometimes we can recover lost information some other way, and then less preservation may be better, because it allows for a more compact representation.

For another example, let's consider representing (abstract) texts as strings, i.e., let's consider semiotic morphisms $M\colon$ **TXT** $\to$ **STG**. The sign system **TXT** has sorts for sentences, words, and characters, while the sign system **STG** only has sorts for strings and characters. Because characters are a data sort, any morphism $M:\ \mathbf{TXT} \to \mathbf{STG}$ must preserve the sort `char`, and there is also no choice about how to map the other sorts of **TXT**: they must all go to the sort `string`. The top level constructor of **TXT** forms texts by concatenating sentences, while its second level constructor concatenates words to form sentences, and its third level constructor concatenates characters to form words. Since the only constructor for **STG** concatenates characters to form strings, the obvious thing to do is map each concatenation of **TXT** to the concatenation of **STG**. However, the sign resulting from a text would now be just one huge ugly string which "mushes" everything together. As we know, it is usual to insert spaces between words, and a period and two spaces after each sentence. It is easy to define a morphism that does this, though it is more complex than the "mushing" representation.

Both these morphisms preserve the structure of **TXT**. But what would it mean for a morphism $M:\ \mathbf{TXT} \to \mathbf{STG}$ *not* to preserve this structure? There are many possibilities, including dropping some characters, words, and/or sentences, and permuting them in a random order. Phenomena like these will clearly produce a low quality display.

Experiments reported in [27] show that preserving high levels is more important than preserving priorities, which in turn is more important than preserving content. They also show a strong tendency to preserve higher levels at the expense of lower levels when some structure must be dropped. This may be surprising, because of emphasis by cognitive psychologists on the "basic level" of lexical concepts (e.g., Rosch [50, 51]). For natural language, the sentential level was long considered to be basic, but research like that of [27] shows that the discourse level is higher in our technical sense, and thus more important. This is consistent with the important general principle that structure has priority over content, i.e., *form is more important than content* (if something must be sacrificed to limit the complexity of the display).

Much more detailed empirical work is needed to determine more precisely the tradeoffs among various preservation and other optimality criteria for semiotic morphisms. At start is being made by assembling a collection of examples of bad design arising from failures of semiotic morphisms to fully preserve structure in the "world-famous"[17] UC San Diego Semiotic Zoo. Although not all the explanations are available yet, the animals can be visited at any hour of the day or night, at

`http://www.cs.ucsd.edu/users/goguen/zoo/`

where much additional information (and some bad jokes about zoos) can also be found. Most of the exhibits there involve color and/or interactive graphics, and so cannot easily be discussed in this traditional medium of print.

The tatami project at UCSD is applying semiotic morphisms and their orderings to design the user interface of a system to supports cooperative distributed proofs over the world wide web [31, 25]. We found that certain ways we had used

[17] For some reason, the real San Diego Zoo, which really is world famous, almost always precedes its name with "world-famous," with the hyphen.

to represent proofs were not semiotic morphisms, which then led us to construct better representations; we also used semiotic morphisms to determine aspects of window layout, button location, etc. Details can be found especially in [22, 25], and of course on the project website

`http://www.cs.ucsd.edu/groups/tatami/`

which should always have the very latest information.

5 Blending, Ambiguity and Pushouts

Fauconnier and Turner [10, 12] study the "blending" of conceptual spaces, to obtain new spaces that combine the parts of the input spaces. Blends are common in natural language, for example, in words like "houseboat" and "roadkill," and in phrases like "artificial life" and "computer virus," as well as in metaphors that have more than one strand (as is usually the case).

The most basic kind of blend may be visualized using the diagram below, where I_1 and I_2 are called the **inputs**, G the **generic**, and B the **blend**[18]. More precisely, we define a **blend** of sign systems I_1 and I_2 over G (using given semiotic morphisms $G \to I_1$ and $G \to I_2$) to be a sign system B with morphisms $I_1 \to B$, $I_2 \to B$, and $G \to B$, which are all called **injections**, such that the diagram **weakly commutes**, in the sense that both the compositions $G \to I_1 \to B$ and $G \to I_2 \to B$ are **weakly equal** to the morphism $G \to B$, in the sense that each sign in G gets mapped to the same sign in B under them, provided that both morphisms are defined on it[19]. It follows that the compositions $G \to I_1 \to B$ and $G \to I_2 \to B$ are also weakly equal when $G \to B$ is totally defined, but not necessarily otherwise. The special case where all sign systems are conceptual spaces is called a **conceptual blend**. In general, we should expect the morphisms to the blend to preserve as much as possible from the inputs and generic.

$$
\begin{tikzcd}
 & B & \\
I_1 \arrow[ur] & & I_2 \arrow[ul] \\
 & G \arrow[ul] \arrow[ur] \arrow[uu] &
\end{tikzcd}
$$

[18] The form of this diagram is "upside down" from that used by Fauconnier and Turner, in that our arrows go up, with the generic G on the bottom, and the blend B on the top; this is consistent with the metaphor (or "image scheme" [40]) that "up is more" as well as with conventions for drawing such diagrams in mathematics. Also, Fauconnier and Turner do not include the map $G \to B$.

[19] Strict commutativity, which is usually called just **commutativity**, means that the compositions are strictly equal.

Mathematically, it is more perspicuous to think of blending the two morphisms $a_i : G \to I_i$ than the two spaces I_1, I_2, and for this reason we will sometimes use the notation $a_1 \diamond a_2$ to stand for an arbitrary blend of a_1 and a_2; this will be especially helpful in writing formulae for our calculus of blending.

Blends have applications in computer interface design, some of which are described in [31]. For a simple example, suppose we want to display both temperature and time of day on the same device. This is an example of the product of sign systems: if `TMP` is a sign system for temperature; then the sign system for our device is `TOD` $\times$ `TMP`. Before giving the technical definition, let $\mathbb{1}$ denote the "trivial" sign system that has only one sort (its top sort) and no operations (except those for data). Now given sign systems S_1 and S_2, their **product**, denoted $S_1 \times S_2$, is the blend of S_1 and S_2 over $\mathbb{1}$ with the obvious (and only) morphisms $\mathbb{1} \to S_i$, formed by taking the disjoint union[20] of S_1 and S_2, and then identifying their top sorts to get a new sort called the product sort. Both injections are injective and both triangles strictly commute.

It is not hard to prove some simple properties of product, including the following, where S, S_1, S_2, S_3 are arbitrary sign systems,

$$
\begin{aligned}
&S \times \mathbb{1} \cong S \ , \\
&\mathbb{1} \times S \cong S \ , \\
&S_1 \times S_2 \cong S_2 \times S_1 \ , \\
&S_1 \times (S_2 \times S_3) \cong (S_1 \times S_2) \times S_3 \ .
\end{aligned}
$$

These are only a modest addition to our calculus of representation, but the notion of product becomes more interesting later on, when extended from sign systems to representations. Forms of the commutative and identity laws also hold for blends, and may be written as

$$
\begin{aligned}
&a_1 \diamond a_2 = a_2 \diamond a_1 \ , \\
&a \diamond 1_G = a \ , \\
&1_G \diamond a = a \ ,
\end{aligned}
$$

where the first should be read as saying that any blend of a_1, a_2 is also a blend of a_2, a_1, and the next two as saying that one blend of any space with its generic space is the space itself.

Before doing a slightly more complex example in some detail, we generalize the concept of blend to a labeled graph, with sign systems on its nodes and morphisms on its edges, such that if e is an edge from n_0 to n_1, then the morphism on e has as its source the sign system on n_0 and as its target the one on n_1. We will call this labeled graph the **base graph**. Some morphisms in the base graph may be designated as **auxiliary**[21], indicating that the relationships

[20] This involves renaming sorts and operations, if necessary, so that there are no overlaps except for the data sorts and operations. Thus this blend is a sort of "amalgamated sum" of its two inputs (this phrase is used in algebraic topology, among other places). Due to the duality between theories and models (as formalized in the theory of institutions [24]), this corresponds to taking products of models.

[21] More technically, it is the *edges* that are designated as auxiliary, because it is possible that the same morphism appears on more than one edge, where not all instances of

that they embody do not need to be preserved. Then a **blend** for a given base graph is some sign system, together with a morphism called an **injection** to it from each sign system in the graph, such that any triangle of morphisms involving two injections and one non-auxiliary morphism in the base graph weakly commutes. The exclusion of auxiliary morphisms is important, because commutativity should not be expected for auxiliary information; this is illustrated in the example below. The base graph for the basic kind of blend considered at the beginning of this section has a "V" shape; let us use the term **V-blends** for this case. Also, let us call a node in the base graph **auxiliary** if all morphisms to and from it in the base graph are auxiliary[22].

Appendix B develops the above ideas more precisely, and puts blending in the rich mathematical framework of category theory, relating V-blends to what are called "pushouts", and the more general blend of a base graph to what are called colimits. In addition, Appendix B develops a special kind of category, called a $\frac{3}{2}$-category, and shows that (what we there call) $\frac{3}{2}$-pushouts and $\frac{3}{2}$-colimits give blends that are "best possible" in a certain precise sense that involves ordering semiotic morphisms by quality, e.g., that they should be as defined as possible, should preserve as many axioms as possible, and should be as inclusive as possible (see Definition 3).

We now show several ways to blend spaces for the words "house" and "boat"; see Figure 6, in which the generic space is auxiliary. We do not aspire to great accuracy in linguistic modeling here; certainly much more detail could be added to the various spaces, and some details could be challenged[23]. Our interest is rather to illustrate the mathematical machinery introduced in this section with a simple, intuitive example. The generic space has three constants, `object`, `medium`, and `person`, plus two relations, `on` and `use`. The house input has constants for `house`, `land`, and `resident`; these are mapped onto by `object`, `medium`, and `person` from the generic space, respectively; the relations are `live-in`, and `on`, where the first is mapped onto by `use`, and where the `house` is `on land`. Similarly, the boat input space has constants for `boat`, `water`, and `passenger`, which are mapped onto by `object`, `medium`, and `person`, respectively; and it has relations `ride` and `on`, where the first is mapped onto by `use`, and where the `boat` is `on water`. In forming a blend, there is a conflict between being on water and being on land, and for "houseboat", water wins. Here all triangles commute. The blend for boathouse chooses land instead of water. But the most interesting things to notice about the boathouse blend are that the boat becomes the resident, and that this leads to a non-commutative triangle of morphisms on the right side.

There are also some other, more surprising, blends for these two conceptual spaces: one gives a boat for transporting houses, and another gives an amphibious

---

it are auxiliary.

[22] I thank Grigore Roşu for the suggestion to generalize from auxiliary nodes to auxiliary edges.

[23] This is consistent with our belief that unique best possible theories do not exist for most real world concepts [21].

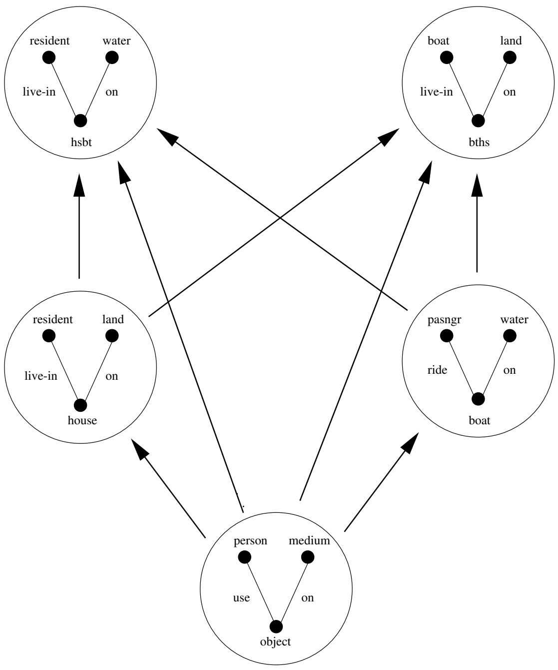


Fig. 6. Two Different Blends of Two Input Spaces

house! See Figure 7. The first blend (to the left in Figure 7) is dual to houseboat: instead of the boat ending up in the house, the house ends up on the boat; there's nothing strange about this except that we don't have any established word for it, and it doesn't correspond to anything in (most people's) experience[24]. The second blend (to the right in Figure 7) is more exotic, since the resulting object can be either on land or on water, and the user both rides and lives in it. Although no such thing exists in our world now, we can easily imagine some mad engineer trying to build one. Now it is interesting to see which triangles commute for each of these, and then to compare the naturalness of each blend with its degree of commutativity. The left triangle of the first blend fails to commute (again just dual to "boathouse"). For the second, although both its triangles commute, the situation here is actually worse than if they didn't, because the injections fail to preserve some of the relevant structure, namely the (implicit) *negations* of relation instances, such as that the boat is *not* on land.

The above is a good illustration of the very important fact that blends are not unique. Ambiguity and its resolution are pervasive in natural language understanding. A word, phrase or sentence with an "obvious" meaning in one context,

[24] But we can easily imagine a construction project on an island where prefabricated houses are transported by boat.

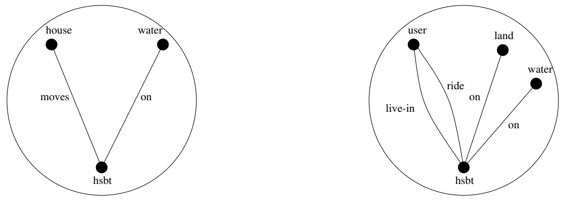


Fig. 7. Two Less Familiar Blends

or in isolation, can have a very different meaning in another context. What is amazing is that we resolve ambiguities so effortlessly that we aren't even aware that they existed, so that it takes some effort to discover the other possibilities that were passed over so easily! For another example, Appendix A constructs a context in which the old aphorism "Time flies like an arrow" undergoes a drastic change of meaning, and also gives a formal specification of the conceptual spaces involved, using the OBJ system [28] to compute the blend, parse the sentence, and then evaluate it to reveal the "meaning". A different way to illustrate the ambiguity of blends can be seen in the beautiful analyses done by Hiraga [37, 36] of haiku by the great Japanese poet Basho; she shows that several different blends coexist for these haiku, and argues that this is a deliberate exploitation of ambiguity as a poetic device.

Ambiguity also plays an interesting role in so called "oxymorons" (like "military intelligence"): these involve two different blends of two given words, one of which has a standard meaning, and the other of which has some kind of conflict in it. The second meaning only arises because the word "oxymoron" has been introduced, and this deliberate creation of a surprising ambiguity is what makes these a form of humor. For "military intelligence" the standard meaning is an agency that gathers intelligence (i.e., information, especially secret information) for military purposes, while the second, conflictual meaning is something like "stupid smartness", playing off the common (but incorrect) prejudice that the military are stupid, plus the more usual meaning of intelligence. A lot of humor seems to have a similar character: an informal survey of cartoons in the local newspaper found that more than half of the intendedly humorous cartoons achieved their effect by recontextualization, through blending a given conceptual space with some new conceptual space, to give some parts of the old one surprising new meanings.

Semiotic morphisms can also arise when signs should have some additional structure in order to be considered "good". For example, typical recent Hollywood movies have a three act structure with two specific "plot points" that move the action from one act into the next; let's call this the "Syd Field" structure after an author who advocates it [13]. Blending this with the "film medium" structure (consisting of shots, scenes, angles, etc.) gives a precise sign system that can help with understanding certain aspects of films. (This is a rather different approach to applying semiotics to cinema than that of the large literature found in the semiotics of film, e.g., [9], but is still compatible with it.)

Now let's consider products of representations. For example, we might have reprepresentations M_1 and M_2 for time and of temperature that we want to use to realize the sign system $\mathtt{TOD} \times \mathtt{TMP}$, where say $M_1 : \mathtt{TOD} \to S_1$ and $M_2 : \mathtt{TMP} \to S_2$. Then what we want is a semiotic morphisms $M_1 \times M_2 : \mathtt{TOD} \times \mathtt{TMP} \to S_1 \times S_2$, defined to be M_1 on TOD and M_2 on TMP, except that the product sort of each source maps to that of its target. We can now prove the following laws, analogous to those for products of sign systems, where M, M_1, M_2, M_3, M_4 are semiotic morphisms, and $\mathbb{1}$ now denotes the identity semiotic morphism on the trivial sign system $\mathbb{1}$, and where $\cong$ now refers to a consistent family of isomorphisms, one for each choice of the morphisms involved[25]:

$$
\begin{aligned}
&M \times \mathbb{1} \cong M \\
&\mathbb{1} \times M \cong M \\
&M_1 \times M_2 \cong M_2 \times M_1 \\
&M_1 \times (M_2 \times M_3) \cong (M_1 \times M_2) \times M_3 \\
&(M_1 \times M_2); (M_3 \times M_4) \cong (M_1; M_3) \times (M_2; M_4)\ .
\end{aligned}
$$

It is clear that a great deal more could be done along these lines, for example, giving laws for the more general forms of blending. This introductory paper does not seem the right place for such a development, but a few more laws are found in Appendix B.

A traditional view is that a metaphor is a mapping from one cognitive space to another, as in the formalization of Gentner [14]. However, the work of Fauconnier and Turner [12] suggests a different view, in which the existence of such a "cross space mapping" between two input spaces is a special asymmetric condition that may occur if one input space dominates the other in the blend. In general, there may be more than two input spaces, and the information about links between the contents of these spaces is distributed among the injections in a complex way that cannot be summarized in any single map.

6 Discussion

This paper has introduced *algebraic semiotics*, a new approach to user interface design, cognitive linguistics, and other areas, based on a notion of sign allowing complex hierarchical structure, thus elaborating Saussure's insight that signs come in systems. Representations are mappings, or *morphisms*, between sign systems, and a user interface is considered a representation of the underlying functionality to which it provides access. This motivates a calculus for combining signs, sign systems, and representations. One important mode of composition is *blending*, introduced by Fauconnier and Turner, which is related to certain concepts from category theory. The main contribution of this paper is the precision that its approach can bring to applications. Building on an insight from computer science, that discrete structures can be described by algebraic theories, **sign systems** are defined as algebraic theories with some extra structure, and

[25] In the technical language of category theory, they are natural isomorphisms.

semiotic morphisms are defined as mappings of algebraic theories that preserve the extra structure to some extent; the quality of representations was found to correlate with the degree to which structure is preserved.

When one sees concrete examples of sign systems like graphical user interfaces, it is easy to believe that these sign systems "really exist". It is amazing how quickly and easily we see signs as actually existing with all their structure "out there" in the "real world". Nevertheless, what "really exists" (in the sense of physics) are the photons coming off the screen; the structure that we see is our own construction. This paper provides a way to describe and study perceived regularities, as modeled by sign systems, without claiming that these regularities correspond to real objects, let alone that best possible descriptions exist for any given phenomenon. This is consistent with ordinary engineering practice, which constructs models for bridges, aircraft wings, audio amplifiers, etc. that are good enough for the practical purpose at hand, without claiming that the models *are* the reality, and indeed, with a deep awareness, based on practical experience, that the models are definitely *not* adequate in certain respects, some known and some unknown[26]. Another advantage of our approach is that it enables us to avoid a lot of distracting philosophical problems, e.g., having to do with the doctrine of realism.

The use of morphisms of theories for representations instead of morphisms of models relates to the above point, in that we tend to think of models as finally grounding the representation process in something "real", whereas morphisms never claim more than to be re-representations, which may add more detail, but do not exhaust all of the possibilities for description.

William Burroughs said *language is a virus* [7], meaning (for example) that peculiarities of accent, vocabulary, attitude, disposition, confusion, neurosis, etc. are contagious, and tend to spread within communities. Mikhael Bakhtin [3] emphasized that language is never a single homogeneous system, using the word "heteroglossia". Paraphrasing Burroughs in the light of Bakhtin, we might say that *language is an ecology of interacting viruses*. So despite our use of formal mathematical description techniques, we should not expect such a realm to be characterized by formal modernist order, but rather to exhibit multiple species of interacting chaotic evolution: signs and interpretations are co-evolving co-emergent social phenomena that are too complex to be fully described; order appears in our multiple, partial descriptions, and such descriptions are what we can formalize. But these descriptions should never be confused with "reality". In contrast with situation theory [5], we do not consider that signs and representations are pre-existing residents of some Platonic heaven, but instead claim that they arise in a context of social interaction. (Further philosophical discussion related to this appears in Appendix C, where in brief, we find that realism is difficult to reconcile with the practice of engineering design, and that phenomenology is more congenial.)

The dynamic aspect of sign systems that emerges from the above discussion

[26] For example, Hook's law for the length of a spring as a function of the weight it is holding, fails if the weight is too heavy, because the spring will be damaged.

brings out an important limitation of the formal apparatus introduced in this paper: it does not address the history-sensitive aspects that are needed for many applications to user interface design. For example, most text editors have an `undo` command that takes one back to the state before the last command was executed. By a fortunate coincidence, a recent advance in algebraic semantics called *hidden algebra* [29, 30, 18] provides exactly the technical apparatus that is needed for this extension, by using hidden sorts for internal states. The extension is actually very straightforward mathematically, but to develop the methodology for its application will require some further work.

There are many other dynamic aspects of sign systems. Real world sign systems evolve; for example, in natural languages, words change their meanings, new words are added, old words disappear, syntax changes, and of course the huge contextual background changes, as social experience changes. In yet another important kind of dynamics, a listener or reader (or "user") constructs meanings dynamically and incrementally, in real time. How this happens is a very difficult problem, about which little information is directly available. It is however clear that no simple algorithm based on just the structure of the sign systems involved can be used compute meanings, because even for the simple blend of two conceptual spaces, selection among the manifold possibilities is governed by external contextual factors in a very complex way, crucially including the *values* of the person doing the blend. Moreover, the perception that the blend is in some way *coherent* seems to be at least as important as any of the more mechanical measures of optimality. This paper does not attempt to solve these difficult problems, but only the simpler problem of providing a precise language for describing structural aspects of particular understandings. In a dynamic context, these static descriptions will be snapshots of evolving structures.

Another area that needs further work is "higher order" signification, which concerns explicit references to meaning; one approach to this problem is to provide some form of "meta-spaces." Meaning is one of the deepest and most difficult of all subjects, and it should not be thought that the explorations in the present paper are more than early steps down one particular path into a great jungle.

References

1. William P. Alston. Sign and symbol. In Paul Edwards, editor, *Encyclopaedia of Philosophy, Volume 7*, pages 437–441. Macmillan, Free Press, 1967. In 8 volumes; republished 1972 in 4 books.
2. Peter B. Andersen. Dynamic logic. *Kodikas*, 18(4):249–275, 1995.
3. Mikhail Bakhtin. *The Dialogic Imagination: Four Essays.* University of Texas at Austin, 1981.
4. Roland Barthes. *S/Z: An Essay and Attitudes.* Hill and Wang, 1974. Trans. Richard Miller.
5. Jon Barwise and John Perry. *Situations and Attitudes.* MIT (Bradford), 1983.
6. John Bowers. The politics of formalism. In Martin Lea, editor, *Contexts of Computer-Mediated Communication.* Harvester Wheatsheaf, 1992.
7. William S. Burroughs. *The Adding Machine: Selected Essays.* Arcade, 1986.

8. John Carroll. Learning, using, and designing filenames and command paradigms. *Behavior and Information Technology*, 1(4):327–246, 1982.
9. Alain Cohen. *Blade Runner*: Aesthetics of agonistics and the law of response. *Il Cannocchiale*, 3:43–58, 1996.
10. Gilles Fauconnier and Mark Turner. Conceptual projection and middle spaces. Technical Report 9401, University of California at San Diego, 1994. Dept. of Cognitive Science.
11. Gilles Fauconnier and Mark Turner. Blending as a central process of grammar. In Adele E. Goldberg, editor, *Conceptual Structure, Discourse and Language*, pages 113–129. CSLI, 1996.
12. Gilles Fauconnier and Mark Turner. Conceptual integration networks. *Cognitive Science*, 22(2):133–187, 1998.
13. Syd Field. *Screenplay: The Foundations of Screenwriting*. Dell, 1982. Third edition.
14. Deidre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.
15. Joseph Goguen. Semantics of computation. In Ernest Manes, editor, *Proceedings, First International Symposium on Category Theory Applied to Computation and Control*, pages 151–163. Springer, 1975. (San Fransisco, February 1974.) Lecture Notes in Computer Science, Volume 25.
16. Joseph Goguen. What is unification? A categorical view of substitution, equation and solution. In Maurice Nivat and Hassan Aït-Kaci, editors, *Resolution of Equations in Algebraic Structures, Volume 1: Algebraic Techniques*, pages 217–261. Academic, 1989.
17. Joseph Goguen. A categorical manifesto. *Mathematical Structures in Computer Science*, 1(1):49–67, March 1991.
18. Joseph Goguen. Types as theories. In George Michael Reed, Andrew William Roscoe, and Ralph F. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 357–390. Oxford, 1991. Proceedings of a Conference held at Oxford, June 1989.
19. Joseph Goguen. On notation (a sketch of the paper). In Boris Magnusson, Bertrand Meyer, and Jean-Francois Perrot, editors, *TOOLS 10: Technology of Object-Oriented Languages and Systems*, pages 5–10. Prentice-Hall, 1993. The extended version of this paper may be obtained from `http://www.cs.ucsd.edu/users/goguen/ps/notn.ps.gz`.
20. Joseph Goguen. Requirements engineering as the reconciliation of social and technical issues. In Marina Jirotka and Joseph Goguen, editors, *Requirements Engineering: Social and Technical Issues*, pages 165–200. Academic, 1994.
21. Joseph Goguen. Towards a social, ethical theory of information. In Geoffrey Bowker, Leigh Star, William Turner, and Les Gasser, editors, *Social Science, Technical Systems and Cooperative Work: Beyond the Great Divide*, pages 27–56. Erlbaum, 1997.
22. Joseph Goguen. Social and semiotic analyses for theorem prover user interface design, submitted for publication 1998.
23. Joseph Goguen. *Theorem Proving and Algebra*. MIT, to appear.
24. Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, January 1992.
25. Joseph Goguen, Kai Lin, Akira Mori, Grigore Roşu, and Akiyoshi Sato. Distributed cooperative formal methods tools. In Michael Lowry, editor, *Proceedings,*

Automated Software Engineering, pages 55–62. IEEE, 1997.
26. Joseph Goguen, Kai Lin, Akira Mori, Grigore Roşu, and Akiyoshi Sato. Tools for distributed cooperative design and validation. In *Proceedings, CafeOBJ Symposium*. Japan Advanced Institute for Science and Technology, 1998. Nomuzu, Japan, April 1998.
27. Joseph Goguen and Charlotte Linde. Optimal structures for multi-media instruction. Technical report, SRI International, 1984. To Office of Naval Research, Psychological Sciences Division.
28. Joseph Goguen and Grant Malcolm. *Algebraic Semantics of Imperative Programs*. MIT, 1996.
29. Joseph Goguen and Grant Malcolm. A hidden agenda. Technical Report CS97–538, UCSD, Dept. Computer Science & Eng., May 1997. To appear in special issue of *Theoretical Computer Science* on Algebraic Engineering, edited by Chrystopher Nehaniv and Masamo Ito. Early abstract in *Proc., Conf. Intelligent Systems: A Semiotic Perspective, Vol. I*, ed. J. Albus, A. Meystel and R. Quintero, Nat. Inst. Science & Technology (Gaithersberg MD, 20–23 October 1996), pages 159–167.
30. Joseph Goguen and Grant Malcolm. Hidden coinduction: Behavioral correctness proofs for objects. *Mathematical Structures in Computer Science*, to appear 1999.
31. Joseph Goguen, Akira Mori, and Kai Lin. Algebraic semiotics, ProofWebs and distributed cooperative proving. In Yves Bartot, editor, *Proceedings, User Interfaces for Theorem Provers*, pages 25–34. INRIA, 1997. (Sophia Antipolis, 1–2 September 1997).
32. Joseph Goguen, James Weiner, and Charlotte Linde. Reasoning and natural explanation. *International Journal of Man-Machine Studies*, 19:521–559, 1983.
33. Robert Goldblatt. *Topoi, the Categorial Analysis of Logic*. North-Holland, 1979.
34. Martin Heidegger. *Being and Time*. Blackwell, 1962. Translated by John Macquarrie and Edward Robinson from *Sein und Zeit*, Niemeyer, 1927.
35. Masako K. Hiraga. Diagrams and metaphors: Iconic aspects in language. *Journal of Pragmatics*, 22:5–21, 1994.
36. Masako K. Hiraga. Rough seas and the milky way: 'Blending' in a *haiku* text. In *Plenary Working Papers in Computation for Metaphors, Analogy and Agents*, pages 17–23. University of Aizu, 1998. Technical Report 98-1-005, Graduate School of Computer Science and Engineering.
37. Masako K. Hiraga. 'Blending' and an interpretation of *haiku*: A cognitive approach. *Poetics Today*, to appear 1998.
38. Marina Jirotka and Joseph Goguen. *Requirements Engineering: Social and Technical Issues*. Academic, 1994.
39. William Labov. The transformation of experience in narrative syntax. In *Language in the Inner City*, pages 354–396. University of Pennsylvania, 1972.
40. George Lakoff and Mark Johnson. *Metaphors We Live By*. Chicago, 1980.
41. Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.
42. Bruno Latour. *Science in Action*. Open, 1987.
43. Bruno Latour. *Aramis, or the Love of Technology*. Harvard, 1996.
44. John Lechte. *Fifty Key Contemporary Thinkers*. Routledge, 1994.
45. Eric Livingston. *The Ethnomethodology of Mathematics*. Routledge & Kegan Paul, 1987.
46. Grant Malcolm and Joseph Goguen. Signs and representations: Semiotics for user interface design. In Ray Paton and Irene Nielson, editors, *Visual Representations and Interpretations*. Springer Workshops in Computing, 1998. Proceedings of an international workshop held in Liverpool.

47. José Meseguer and Joseph Goguen. Initiality, induction and computability. In Maurice Nivat and John Reynolds, editors, *Algebraic Methods in Semantics*, pages 459–541. Cambridge, 1985.
48. Donald A. Norman. *The Design of Everyday Things*. Doubleday, 1988.
49. Charles Saunders Peirce. *Collected Papers*. Harvard, 1965. In 6 volumes; see especially Volume 2: Elements of Logic.
50. Eleanor Rosch. On the internal structure of perceptual and semantic categories. In T.M. Moore, editor, *Cognitive Development and the Acquisition of Language*. Academic, 1973.
51. Eleanor Rosch. Cognitive reference points. *Cognitive Psychology*, 7, 1975.
52. Harvey Sacks. On the analyzability of stories by children. In John Gumpertz and Del Hymes, editors, *Directions in Sociolinguistics*, pages 325–345. Holt, Rinehart and Winston, 1972.
53. Harvey Sacks. *Lectures on Conversation*. Blackwell, 1992. Edited by Gail Jefferson.
54. Ferdinand de Saussure. *Course in General Linguistics*. Duckworth, 1976. Translated by Roy Harris.
55. Ben Shneiderman. *Designing the User Interface*. Addison Wesley, 1997.
56. Susan Leigh Star. The structure of ill-structured solutions: Boundary objects and heterogeneous problem-solving. In Les Gasser and Michael Huhns, editors, *Distributed Artificial Intelligence*, volume 2, pages 37–54. Pitman, 1989.
57. Lucy Suchman. *Plans and Situated Actions: The Problem of Human-machine Communication*. Cambridge, 1987.
58. Mark Turner. *The Literary Mind*. Oxford, 1997.

A Two Examples in OBJ3

Consider the following "science fiction" fragment, which constructs a context in which the old aphorism "Time flies like an arrow" undergoes a drastic change of meaning:

> A gravity kink forced the ship enough off course that realtime was needed to calculate corrections. Taking realtime in a wormhole creates a local space-time vector, and time flies were already buzzing about, making the corrections even harder. "They hang onto any vector they can find out here," Randi said. "Time flies like an arrow. We may never get out."

Here the original verb "flies" becomes the subject; the original subject "time" now modifies "flies"; the preposition "like" becomes the verb; and "arrow" becomes the object of "like". The only word that doesn't change its syntactic role is the lowly article "an"! How does this happen? The "local space-time vector" (whatever that is) prepares the reader for "an arrow", and then "time flies" are introduced explicitly. These two conceptual spaces blend into another, where our sentence gets its new interpretation; they share a subspace where a ship takes realtime in a wormhole.

We describe these three conceptual spaces, form a blend, and then parse and evaluate our sentence using the OBJ language (for more on OBJ and its underlying theory, see [28]), which is especially suitable because of its rich facilities for combining theories. The keyword pair `th...endth` delimits OBJ modules that introduce "theories" which allow any model that satisfies the axioms. The two "`pr SHIP`" lines indicate importation of the theory `SHIP` in such a way that it is shared; `+` tells OBJ to form a blend (which is actually their colimit in the sense of Appendix B below), which is then named `POUT` as part of the `make...endm` construct, which just builds and names a module. Predicates appear as `Bool`(ean) valued functions. Finally, `red` tells OBJ to parse what follows, apply equations as left to right rewrite rules, and then print the final result (if there is one):

```
th SHIP is sort Thing .
  ops (the ship) wormhole vector : -> Thing .
  op _in_ : Thing Thing -> Bool .
  op _makes_ : Thing Thing -> Bool .
  eq the ship in wormhole = true .
  var X : Thing .
  cq X makes vector = true if X in wormhole .
endth

th FLIES is pr SHIP .
  op time flies : -> Thing .
  ops (_like_)(_buzz around_) :
 Thing Thing -> Bool .
  eq time flies buzz around the ship = true .
  var X : Thing .
  cq time flies like X = true if X == vector .
endth

th ARROW is pr SHIP .
  op an arrow : -> Thing .
  eq an arrow = vector .
endth

make POUT is FLIES + ARROW . endm

red the ship makes an arrow .
red time flies like an arrow .
```

Of course, as an understanding of the text, this formal system is grossly oversimplified; however, it is *precise*. These sign systems have only two levels (for "words" and "sentences"), no priority, and few signs; the morphisms are just inclusions. Probably the hardest part, that `an arrow` is a `vector`, has been just posited, because OBJ cannot do this kind of "selection" process, although it is well suited for defining and blending sign systems, and for parsing and evaluating expressions. Here is OBJ's output from the above:

```
                   \||||||||||||||||||/
                 --- Welcome to OBJ3 ---
                   /||||||||||||||||||\
OBJ3 version 2.04 built 1994 Feb 28 Mon 15:07:40
```

```
   Copyright 1988,1989,1991 SRI International
   1997 Jan 18 Sat 22:25:11
OBJ>
==========================================
obj SHIP
==========================================
obj FLIES
==========================================
obj ARROW
==========================================
make POUT
==========================================
reduce in POUT : the ship makes an arrow
rewrites: 3
result Bool: true
==========================================
reduce in POUT : time flies like an arrow
rewrites: 3
result Bool: true
OBJ> Bye.
```

This shows OBJ parsing both sentences and then "understanding" that they are "`true`"; note that neither sentence parses outside the blend. I hope the reader is as pleased as the author at how easy[27] all this is. Of course, we could get the usual understanding of the sentence by evaluating it in a different context.

We now consider a somewhat more complex example, a proof that one metaphor is better than another, under certain assumptions. The assumptions are given in the five theories, the metaphors in the two views, and the proof in the four reductions. The first metaphor, "The internet is an information tornado," comes from a press release from the Federal Communications Commission, while the second, "The internet is an information volcano," comes from a poster that the author of this paper prepared for a course on material in this paper at UCSD. The keyword "`us`" (from "using") indicates importation by copying rather than sharing, and `*(op A to B)` indicates a renaming of the operation `A` to become `B`.

```
th COMMON is
  sorts Agent Effect .
  op  effect : Agent Agent  -> Effect .
  ops hurt nil helped :     -> Effect .
endth

th PROCESS is us COMMON .
  sort Volume .
  ops subject process :      -> Agent .
```

[27] It took about 15 minutes to write the code, and less than a second for OBJ to process it, most of which is spent on input-output, rather than on processing the various declarations and doing the 6 applications of rewrite rules.

```
  op  flow : Agent Agent     -> Volume .
  ops low medium high huge : -> Volume .
endth

th INTERNET is us (PROCESS *(op subject to user)
                  *(op process to internet)).
  eq flow(internet,user)   = huge .
  eq flow(user,internet)   = low .
  eq effect(internet,user) = hurt .
endth

th VOLCANO is us (PROCESS *(op subject to victim)
                 *(op process to volcano)).
  eq flow(volcano,victim)   = huge .
  eq flow(victim,volcano)   = low .
  eq effect(volcano,victim) = hurt .
endth

th TORNADO is us (PROCESS *(op subject to victim)
                 *(op process to tornado)).
  eq flow(tornado,victim)   = low .
  eq flow(victim,tornado)   = huge .
  eq effect(tornado,victim) = hurt .
endth

*** The internet is an information tornado.
view TORNADO from TORNADO to INTERNET is
  op victim to user .
  op tornado to internet .
endv

th TESTT is us (TORNADO + INTERNET). endth
red flow(victim,tornado) == flow(user,internet).
red flow(tornado,victim) == flow(internet,user).

*** The internet is an information volcano.
view VOLCANO from VOLCANO to INTERNET is
  op victim to user .
  op volcano to internet .
endv

th TESTV is us (VOLCANO + INTERNET). endth
red flow(victim,volcano) == flow(user,internet).
red flow(volcano,victim) == flow(internet,user).
```

The OBJ3 output from this shows that the first two reductions give `false` and the second two give `true`. This means that the first semiotic morphism does not preserve the axioms (which concern the flow of material between the user and the object, either tornado or volcano), while the second morphism does, which implies that the second metaphor is better than the first with respect to preserving these axioms. (On the other hand, the tornado metaphor resonates with many common phrases such as "winds of change," which are part of our culture, whereas we have less collective experience and associated language for volcanos.)

B Categories, Blends, Pushouts, $\frac{3}{2}$-Categories and $\frac{3}{2}$-Pushouts

Although this appendix is written under the assumption that readers already know some basic category theory[28], it is nonetheless essentially self-contained, though terse, in order to fix notation for the new material. The essential intuition behind categories is that they capture mathematical structures; for example, sets, groups, vector spaces, and automata, along with their structure preserving morphisms, each form a category, and their morphisms are an essential part of the picture.

Definition 4: A **category C** consists of: a collection, denoted $|\mathbf{C}|$, of **objects**; for each pair A, B of objects, a set $\mathbf{C}(A, B)$ of **morphisms** (also called **arrows** or **maps**) from A to B; for each object A, a morphism 1_A from A to A called the **identity** at A; and for each three objects A, B, C, an operation called **composition**, $\mathbf{C}(A, B) \times \mathbf{C}(B, C) \to \mathbf{C}(A, C)$ denoted ";" such that $f;(g;h) = (f;g);h$ and $f;1_A = f$ and $1_A;g = g$ whenever these compositions are defined. We write $f\colon A \to B$ when $f \in \mathbf{C}(A, B)$, and call A the **source** and B the **target** of f. □

Results in the body of this paper show that sign systems with semiotic morphisms form a category. We will review the notions of pushout, cone and colimit for ordinary categories, relate this to blending, and then consider the more general setting of $\frac{3}{2}$-categories, which captures more of the phenomenology of blending.

The intuition for colimits is that they put some components together, identifying as little as possible, with nothing left over, and with nothing essentially new added [17]. This suggests that colimits should give some kind of optimal blend. We will see that there are problems with this, so that the traditional categorical notions are not quite appropriate for blending. Nevertheless, they provide a good place to begin our journey of formalization.

Definition 5: Given a category **C**, a **V** in **C** is a pair $a_i\colon G \to I_i$ $(i = 1, 2)$ of morphisms, and a **cone** with **apex** B over a V a_1, a_2 is a pair $b_i\colon I_i \to B$ $(i = 1, 2)$ of morphisms; then a_1, a_2 and b_1, b_2 together are said to form a **diamond** (or a **square**). The cone (or its diamond) **commutes** iff $a_1;b_1 = a_2;b_2$, and is a **pushout** iff given any other commutative cone $c_i\colon I_i \to C$ over a_1, a_2, there is a unique arrow $u\colon B \to C$ such that $b_i;u = c_i$ for $i = 1, 2$.

A **diagram** D in a category **C** is a directed graph with its nodes labeled by objects from **C** and its edges labeled by arrows from **C**, such that if an arrow

[28] See [33, 16, 17] for relatively gentle introductions to some basic ideas of category theory; there are also many many other papers and many other books.

$f\colon D_i \to D_j$ labels an edge $e\colon i \to j$, then the source node i of e is labeled by D_i and the target node j of e is labeled by D_j. A **cone over** D is an object B, called its **apex**, together with an arrow $b_i\colon D_i \to B$, called an **injection**, from each object of D to B, and is **commutative** iff for each $f\colon D_i \to D_j$ in D, we have[29] $b_i = f; b_j$. A **colimit** of D is a commutative cone $b_i\colon D_i \to B$ over D such that if $c_i\colon D_i \to C$ is any other commutative cone over D, then there is a unique $u\colon B \to C$ such that[30] $b_i; u = c_i$ for all nodes i of D. $\square$

Pushouts are the special case of colimits where the diagram is a V. However there seems to be a discrepancy in the definitions, because pushouts are not required to have an arrow $G \to B$. But when the diagram is a V, this missing arrow is automatically provided by the morphism $a_1; b_1 = a_2; b_2$.

There is a short proof that any two colimits of a diagram D are isomorphic. Let the cones be $b_i\colon D_i \to B$ and $b'_i\colon D_i \to B'$. Then there are unique arrows $u\colon B \to B'$ and $v\colon B' \to B$ satisfying the appropriate triangles, and there are also unique arrows $B \to B$ and $B' \to B'$ satisfying their appropriate triangles, namely the respective identities 1_B and $1_{B'}$; but $u; v$ and $v; u$ also satisfy the same triangles; so by uniqueness, $u; v = 1_B$ and $v; u = 1_{B'}$.

Following the suggestion of Section 5 that blends are commutative cones, it follows that colimits should be some kind of optimal blend. For example, the "houseboat" blend of "house" and "boat" is a colimit. But the fact that colimits are only determined up to isomorphism seems inconsistent with this, because the names attached to the elements in a blend are important; that is, isomorphic cones do *not* represent the same blend. This differs from the situation in group theory or topology, where it is enough to characterize an object up to isomorphism. However the requirement (also motivated by the examples in Section 5) that the injections should be inclusions to as great an extent as possible, causes the actual names of elements to be captured by blends, and thus eliminates the apparent inconsistency.

Another problem with defining blends to be commutative cones is that, as shown in Section 5, not all blends actually have fully commutative cones; for "house" and "boat", only the "houseboat" blend has all its triangles commutative. But as suggested there, the notion of auxiliary morphism solves this problem. The **auxiliary morphisms** in D are those whose triangles are not required to commute; these morphisms can be removed from D, to yield another diagram D' having the same nodes as D. Commutative cones over D' are then cones over D that commute except possibly over the auxiliary morphisms. Now we can also form a colimit of D', to get a "best possible" such cone over D. It therefore makes sense to define a blend to be a commutative cone over a diagram with the auxiliary morphisms removed.

One advantage of formalization is that it makes it possible to prove general laws, in this case, laws about blends based on general results from category theory, such as that "the pushout of a pushout is a pushout." This result suggests

[29] These equations are called **triangles** below, after the corresponding three node commutative diagrams.

[30] These equations may also be called "triangles" below.

proving that "the blend of a blend is a blend," so that compositionality of the kind of optimal blends given by pushouts follows from the above quoted result about pushouts. The meaning of these assertions will be clearer if we refer to the following diagram:

$$\begin{tikzcd}
 & \bullet & & \\
\bullet \arrow[ur, "c_2"] & & \bullet \arrow[ul, "c_3"'] & \\
 & \bullet \arrow[ul, "a_1"] \arrow[ur, "b_2"'] & & \bullet \arrow[ul, "b_3"'] \\
 & & \bullet \arrow[ul, "a_2"] \arrow[ur, "a_3"'] &
\end{tikzcd}$$

Here we assume that b_2, b_3 is a blend of a_2, a_3, and c_2, c_3 is a blend of a_1, b_2, i.e., that $a_2; b_2 = a_3; b_3$ and $a_1; c_2 = b_2; c_3$; then the claim is that $c_2,\ b_3; c_3$ is a blend of $a_2; a_1,\ a_3$, which follows because $a_2; a_1; c_2 = a_3; b_3; c_3$. Using the notation $a_2 \diamond a_3$ for an arbitrary blend of a_2, a_3, we can write this result rather nicely in the form

$$a_1 \diamond (a_2 \diamond a_3) = (a_2; a_1) \diamond a_3 \ ,$$

taking advantage of a convention that $a_1 \diamond (a_2 \diamond a_3)$ indicates blending a_1 with the left injection of $(a_2 \diamond a_3)$ (the top left edge of its diamond).

The pushout composition result (proved e.g. in [33, 41]) states that if b_2, b_3 is a pushout of a_2, a_3, and c_2, c_3 is a pushout of a_1, b_2, then $c_2,\ b_3; c_3$ is a pushout of $a_2; a_1,\ a_3$. If we write $a_2 \bowtie a_3$ for the pushout of a_2, a_3, then this result can also be written neatly, as

$$a_1 \bowtie (a_2 \bowtie a_3) = (a_2; a_1) \bowtie a_3 \ .$$

We can also place a second blend (or pushout) on top of b_3 instead of b_2; corresponding results then follow by symmetry, and after some renaming of arrows can be written as follows:

$$(a_1 \diamond a_2) \diamond a_3 = a_1 \diamond (a_2; a_3) \ .$$
$$(a_1 \bowtie a_2) \bowtie a_3 = a_1 \bowtie (a_2; a_3) \ .$$

We can further generalize to any pattern of diamonds: if they all commute, then so does the outside figure; and if they are all pushouts, then so is the outside figure. Another very general result from category theory says that the colimit of any connected diagram can be built from pushouts of its parts. Taken all together, these results give a good deal of calculational power for blending.

Now it's time to broaden our framework. The category of sign systems with semiotic morphisms has some additional structure over that of a category: it is an *ordered category*, because of the orderings by quality of representation that can be put on its morphisms. This extra structure gives a richer framework for considering blends; I believe this approach captures what Fauconnier and Turner have called "emergent" structure, without needing any other machinery. Moreover, all the usual categorical compositionality results about pushouts and colimits extend to $\frac{3}{2}$-categories.

Definition 6: A $\mathbf{\frac{3}{2}}$**-category**[31] is a category $\mathbf{C}$ such that each set $\mathbf{C}(A, B)$ is partially ordered, composition preserves the orderings, and identities are maximal. $\Box$

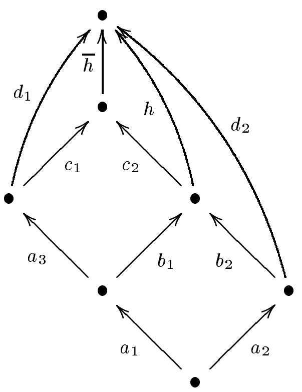
Because we are concerned here with ordered categories, a somewhat different notion of pushout is appropriate, and for this notion, the uniqueness property is (fortunately!) lost:

Definition 7: Given a V, $a_i\colon G \to I_i$ $(i = 1, 2)$ in a $\frac{3}{2}$-category $\mathbf{C}$, a cone b_1, b_2 over a_1, a_2 is **consistent** iff there exists some $d\colon G \to B$ such that $a_1; b_1 \leq d$ and $a_2; b_2 \leq d$, and is a $\mathbf{\frac{3}{2}}$**-pushout** iff given any consistent cone $c_i\colon I_i \to C$ over a_1, a_2, the set

$$\{h\colon B \to C \mid b_1; h \leq c_1 \ \text{ and } \ b_2; h \leq c_2\}$$

has a maximum element. $\Box$

Proposition 8: The composition of two $\frac{3}{2}$-pushouts is also a $\frac{3}{2}$-pushout.
Proof: Let b_1, b_2 be a $\frac{3}{2}$-pushout of a_1, a_2, and let c_1, c_2 be a $\frac{3}{2}$-pushout of a_3, b_1; we will show that c_1 $b_2; c_2$ is a $\frac{3}{2}$-pushout of $a_1; a_3,\ a_2$.



Suppose d_1, d_2 together with $a_1; a_3$ and a_2 form a consistent diamond. Then $a_3; d_1$ and d_2 with a_1, a_3 also form a consistent diamond, and because b_1, b_2 is a $\frac{3}{2}$-pushout for a_1, a_2, the set $\{g \mid b_1; g \leq a_3; d_1,\ b_2; g \leq d_2\}$ has a maximum element, which we denote h. Note that d_1, h with a_3, b_1 form a consistent diamond. Then because c_1, c_2 is a $\frac{3}{2}$-pushout of a_3, b_1, the set $\{g \mid c_1; g \leq d_1,\ c_2; g \leq h\}$ has a maximum element, which we denote $\overline{h}$. We now claim that the following two sets are equal:

[31] In the literature, similar structures have been called "one and a half" categories, because they are half way between ordinary ("one dimensional") categories and the more general "two (dimensional)" categories.

$\mathcal{M}_1 = \{g \mid c_1; g \leq d_1 \text{ and } c_2; g \leq h\}$; and
$\mathcal{M}_2 = \{g \mid c_1; g \leq d_1 \text{ and } b_2; c_2; g \leq d_2\}$.
First let $g \in \mathcal{M}_1$. Then $b_2; (c_2; g) \leq b_2; h \leq d_2$. Therefore $g \in \mathcal{M}_2$. Conversely, suppose $g \in \mathcal{M}_2$; then all we have to prove is that $c_2; g \leq h$. Because $b_2; (c_2; g) \leq d_2$ and $b_1; (c_2; g) = (a_3; c_1); g \leq a_3; d_1$, and because h is the maximum element satisfying the inequalities above, we get $c_2; g \leq h$. Therefore $\mathcal{M}_1 = \mathcal{M}_2$, which implies they have the same maximum, namely $\overline{h}$. □

However, unlike the situation for ordinary pushouts, the composition of consistent diamonds need not be consistent, and two different $\frac{3}{2}$-pushouts need not be isomorphic; this means that ambiguity is natural in this setting. The following is another compositionality result for $\frac{3}{2}$-pushouts:

Proposition 9: In the diagram below, if the four small squares are $\frac{3}{2}$-pushouts, then so is the large outside square.

$$
\begin{tikzcd}
 & & \bullet & & \\
 & \bullet \arrow[ur, "d_1"] & & \bullet \arrow[ul, "d_2"'] & \\
\bullet \arrow[ur, "c_1"] & & \bullet \arrow[ul, "c_2"'] \arrow[ur, "c_3"] & & \bullet \arrow[ul, "c_4"'] \\
 & \bullet \arrow[ul, "b_1"] \arrow[ur, "b_2"'] & & \bullet \arrow[ul, "b_3"] \arrow[ur, "b_4"'] & \\
 & & \bullet \arrow[ul, "a_1"] \arrow[ur, "a_2"'] & &
\end{tikzcd}
$$

Proof: Applying Proposition 8 twice gives two $\frac{3}{2}$-pushouts shown below,

$$
\begin{tikzcd}
 & & \bullet & \\
 & \bullet \arrow[ur, "d_1"] & & \bullet \arrow[ul, "c_4;d_2"'] \\
\bullet \arrow[ur, "c_1"] & & \bullet \arrow[ul, "b_3;c_2"'] \arrow[ur, "b_4"'] & \\
 & \bullet \arrow[ul, "a_1;b_1"] \arrow[ur, "a_2"'] & &
\end{tikzcd}
$$

and applying Proposition 8 once more gives us that the big square is a $\frac{3}{2}$-pushout. □

Passing from V's to arbitrary diagrams of morphisms generalizes $\frac{3}{2}$-pushouts to $\mathbf{\frac{3}{2}}$-colimits, and provides what seems a natural way to blend complex interconnections of meanings. The notion of consistent diamond extends naturally to arbitrary diagrams, as follows:

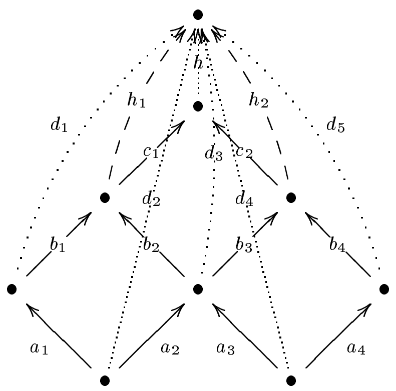
Definition 10: Let D be a diagram. Then a family $\{\alpha_i\}_{i \in |D|}$ of morphisms is D**-consistent** iff $a; \alpha_j \leq \alpha_i$ whenever there is a morphism $a : i \to j$ in D. Similarly, given $J \subseteq |D|$, we say a family of morphisms $\{\alpha_i\}_{i \in J}$ is D-consistent iff $\{\alpha_i\}_{i \in J}$ extends to a D-consistent family $\{\alpha_i\}_{i \in |D|}$. $\square$

Fact 11: A diamond a_1, a_2, b_1, b_2 is consistent if and only if $\{b_1, b_2\}$ is $\{a_1, a_2\}$-consistent.
Proof: If the diamond is consistent then there is some d such that $a_1; b_1 \leq d$ and $a_2; b_2 \leq d$. But then $\{b_1, b_2, d\}$ is $\{a_1, a_2\}$-consistent, i.e., $\{b_1, b_2\}$ is $\{a_1, a_2\}$-consistent. Conversely, if $\{b_1, b_2\}$ is $\{a_1, a_2\}$-consistent, then some d exists such that $\{b_1, b_2, d\}$ is $\{a_1, a_2\}$-consistent, which says exactly that $a_1; b_1 \leq d$ and $a_2; b_2 \leq d$, i.e., that the diamond is consistent. $\square$

Definition 12: Let D be a diagram. Then a family $\{\alpha_i\}_{i \in |D|}$ is a $\frac{3}{2}$**-colimit** of D iff it is a cone and for any D-consistent family $\{\beta_i\}_{i \in |D|}$, the set $\{h \mid \alpha_i; h \leq \beta_i,$ for each $i \in |D|\}$ has a *maximum* element. $\square$

The following is another typical result that extends from ordinary colimits to $\frac{3}{2}$-colimits:

Theorem 13: Let a **W diagram** consist of two V's connected at the middle top. If D is a W diagram, then a $\frac{3}{2}$-colimit of D is obtained by taking a $\frac{3}{2}$-pushout of each V, and then taking a pushout those two pushouts, as shown below.
Proof: Let D contain the morphisms a_1, a_2, a_3, a_4, let b_1, b_2 be a $\frac{3}{2}$-pushout of a_1, a_2, let b_3, b_4 be a $\frac{3}{2}$-pushout of a_3, a_4, and let c_1, c_2 be a $\frac{3}{2}$-pushout of $b_2 b_3$. Then we must show that the family of morphisms $\{b_1; c_1,\ a_2; b_2; c_1,\ b_2; c_1,\ a_3; b_3; c_2,\ b_4; c_2\}$ is a $\frac{3}{2}$-colimit of D.



Let $\{d_1, d_2, d_3, d_4, d_5\}$ be a D-consistent family. Then d_1 and d_3 with a_1, a_2 form a consistent diamond (because $a_1; d_1 \leq d_2$ and $a_2; d_3 \leq d_2$), and because b_1, b_2 is a $\frac{3}{2}$-pushout, we deduce that there exists h_1 (as the maximum of a set of morphisms) such that $b_1; h_1 \leq d_1$ and $b_2; h_1 \leq d_3$. Similarly there exists h_2 such that $b_3; h_2 \leq d_3$ and $b_4; h_2 \leq d_5$. Now note that h_1, h_2 with b_2, b_3 give a consistent diamond (because there is d_3 such that $b_2; h_1 \leq d_3$ and $b_3; h_2 \leq d_3$). We next claim that the following two sets are equal:

$\mathcal{M}_1 = \{h \mid c_1; h \leq h_1 \text{ and } c_2; h \leq h_2\}$; and
$\mathcal{M}_2 = \{h \mid (b_1; c_1); h \leq d_1 \text{ and } (b_2; c_1); h \leq d_3 \text{ and } (b_4; c_2); h \leq d_5\}$.
(The corresponding inequalities for d_2 and d_4 are omitted from $\mathcal{M}_2$ because they are implicit). First we show $\mathcal{M}_1 \subseteq \mathcal{M}_2$. If $h \in \mathcal{M}_1$ then

- $(b_1; c_1); h = b_1; (c_1; h) \leq b_1; h_1 \leq d_1$.
- $(b_2; c_1); h = b_2; (c_1; h) \leq b_2; h_1 \leq d_3$.
- $(b_4; c_2); h = b_4; (c_2; h) \leq b_4; h_2 \leq d_5$.

This implies $h \in \mathcal{M}_2$. Conversely, if $h \in \mathcal{M}_2$ then $b_1; (c_1; h) \leq b_1; h_1 \leq d_1$ and $b_2; (c_1; h) \leq b_2; h_1 \leq d_3$. Then by maximality of h_1, we get $c_1; h \leq h_1$. In a similar way, we can show $c_2; h \leq h_2$, and thus $h \in \mathcal{M}_1$. Therefore $\mathcal{M}_1 = \mathcal{M}_2$, which implies that these sets have the same maximum element. □

Extending our pushout notation $\bowtie$ to $\frac{3}{2}$-categories, the above result can be rather neatly written in the form

$$(a_1 \bowtie a_2) \bowtie (a_3 \bowtie a_4) = \mathit{Colim}(W) ,$$

where W is the bottom part of the diagram, with edges labeled a_1, a_2, a_3, a_4. A generalization of the above result implies that $\frac{3}{2}$-pushouts can be used to compute the $\frac{3}{2}$-colimit of any connected diagram. Observe that the notion of auxiliary morphism carries over to the framework of $\frac{3}{2}$-categories without any change.

It is natural to use the conditions that morphisms should be as defined as possible, should preserve as many axioms as possible, and should be as inclusive as possible, to define a quality ordering (see Definition 3). More precisely, given morphisms $f, g \colon A \to B$ between conceptual spaces A, B, let us define $f \leq g$ iff g preserves as much content as f, preserves all axioms that f does, and is as inclusive as f. Although more work should be done to determine whether this particular "designer ordering" really works the best for this particular application, the situation with respect to our house and boat example from Section 5 really is quite satisfying, in that the most natural blend is an ordinary pushout, all the other good blends are $\frac{3}{2}$-pushouts, and various blends that fail to preserve as much structure as they could are not any kind of pushout.

C Some Philosophical Issues

The research program of which this paper is part, is primarily concerned with practical applications, and the goal of this paper is to provide some of the theory that is needed to support such applications. By contrast, most work in semiotics has had a much more philosophical focus. As a result, a great deal of philosophical discussion could be generated concerning the heretical approach of this paper. This appendix confines itself to just a few points that seem to have some practical significance.

Today humanists of nearly all schools reject the notion that some kind of "Cartesian coordinates" can be imposed on experience, despite partial evidence to the contrary from fields like linguistics and music. This rejection is understandable as a reaction to the scientistic reductionism that nearly always accompanies projects to impose structure on experience. Such tendencies are deeply ingrained in Western civilization, going back at least to Pythagoras and Plato. But evidence from a wide range of fields now makes it clear that traditional reductionism has serious limitations. The following are brief descriptions of some better known examples:

1. Work on mechanical speech recognition has shown that contextual information is essential for determining what phoneme some raw acoustic waveform represents (if anything); this contextual information may include not just prior but also subsequent speech, a profile for the individual speaker (accent, eccentricities, etc.), the topic of discourse, and much more, up to arbitrary shared cultural knowledge.
2. In music, the same acoustic event in a different context can have a radically different impact, ranging from ugly and incongruous, to great beauty and elegance. Moreover, the background of the listener is crucial; for example, naive listeners have little chance of appreciating the subtleties and beauties of Cecil Taylor or Ornette Coleman, however familiar with theories of psychoacoustics and harmony they might be.
3. Similar things happen in cinema and poetry, and indeed any art or craft, from architecture and interior design, to basket weaving, pottery, and flower arranging. Often a great deal of cultural context is needed to appreciate (in any deep sense) a single artifact; buildings, rooms, baskets and pots are used by ordinary people in their ordinary lives, as part of the complex social fabric. The "Gucci" label on a purse is not lovely in itself, but nonetheless it has a meaning to those who go out of their way to acquire it. A brightly colored postmodern bank building in Lisbon has a complex cultural meaning that does not transfer to Paris, London, or New York.
4. Despite the stunning success of applying simple atomic theory to basic molecular chemistry, physics has found it necessary to postulate nonlocalized quantum fields to explain many important phenomena, some of which appear even in applied chemistry, to say nothing of more rarefied areas.
5. Metamathematics has had great success in formalizing mathematics, and in studying what is provable. But its greatest successes have been results, like Gödel's incompleteness theorem, that demonstrate the *limitations* of formalization. Moreover, formal proofs lack the comprehensibility, and the human interest, of well done informal proofs. See Appendix D for more discussion along these lines, demonstrating the importance of context for making proofs "come alive."

Returning now to our main point, there is a justifiable opposition to totalizing reductionist structuralist systems, while at the same time, there is the utterly pervasive presence of structured signs. What are we to do about this seemingly contradictory situation?

Two alternatives have been most explored, each with some valuable results. The first is to pursue the quest for structure, digging deeper wherever it seems to work, and avoiding the (very many) areas where things just seem too slippery to admit much precision. This inevitably results in a partial view, which is open to criticism in various ways (as post-structuralism has criticized the structuralism of Saussure, Levi-Strauss, Barthes, etc.). The second alternative is to abandon structure and work with intuitive experiences and descriptions (some currently fashionable words are "rich," "nuanced," "textured," and "postmodern"). This too inevitably results in a partial view, which in the extreme avoids criticism by refusing to be pinned down, even to the extent of using inconsistent, incoherent language. Through both are extreme positions, it seems difficult to find a clear, consistent, defensible middle ground. (A general reference for continental philosophy is [44].)

It seems to me that ethnomethodology provides some valuable hints on a way out of this impasse. Often presented as a principled criticism of traditional sociology, especially its normative category schemes (gender, race, status, etc.), ethnomethodology can perhaps better be seen positively as an approach to understanding social phenomena (such as signs!) by seeing how members of some group come to see those sign as present. Thus, ethnomethodology wants to know what categories the members of a social group use, and what methods they use to determine instances of those categories. This requires careful attention to real social interaction, and avoids the Platonist assumption that the categories have a pre-given existence "in nature." Rather, we see how members of a group achieve categorization in actual practice, without having to give these either the categories or their instances any status other than what has been achieved in a particular way at a particular time. The branch of ethnomethodology called *conversation analysis* has taken a rather radical approach to the social context of language, showing that even simple features such as whose turn it is to speak are always negotiated in real time by actual social groups [52, 53], and should not be considered as given. Words like "reification" and "transcendentalizing" are used to describe approaches that take the opposite view. (Of course, any one paragraph description of ethnomethodology is necessarily a gross oversimplification; more information may be found in [57] and [21] among many other places, some of which may be very difficult to read.)

Although this paper is not the place to discuss it, phenomenology has also been an important influence on our formulation of a philosophical foundation for semiotics, particularly in its insistence that the only possible starting point is the ground of our own actual experience, with all metaphysical principles firmly bracketed.

The sign, object, interpretant triad of classical semiotics (Peirce, Morris, Eco, etc.) presupposes an objective world, whereas our morphic semiotics is consistent with the view that mind (usually unconsciously) constructs models by selecting and blending (abstractions from) immediate and past experience, using (e.g.) templates derived from embodied motion [40], so that what we see as "objects" are actually parts of these models. This does not deny that a "world" exists, but it does deny that we experience it directly. As Heidegger observed, we come closest to experiencing "reality" when our models break down [34]. Similarly, we may reinterpret the syntax, semantics, pragmatics triad of classical semiotics, by claiming that its instances can probably be better understood through the use of semiotic morphisms.

The above ideas suggest various ways to avoid the extremes of mindless reductionism and mindless holism. The most straightforward approach is to admit that while each individual analysis no doubt has biases and limitations, it nonetheless embodies certain structures, values, insights, etc. A given analysis, if it is clear, coherent and consistent, can be formalized, and may have some value as such; for example, its limitations will be easier to spot. Such an analysis should not pretend to be objective, factual, complete, universal, or even self-contained; it is a momentary snapshot of a partial understanding of one (or more) interested party, and of course, can only be understood by other interested parties who have a more or less comparable background. It has frozen out the fluid processes of interpretation that actually produced the understanding.

The previous paragraph may claim too little, because sometimes analyses can have great impact, with broad acceptance, important applications, etc., e.g., Newtonian mechanics[32]. However this paper is not the place to try to understand why some analyses may work better than others in some given social context. It is enough for our purposes that analyses exist, exhibit structure, and can be formalized, without requiring a totalizing, reductionist, or realist stance.

D What is a Proof?

Mathematicians talk of "proofs" as real things. But all we can ever actually find in the real world of actual experience are *proof events*, or "provings", each of which is a social interaction occurring at a particular time and place, involving particular people, who have particular skills as members of an appropriate mathematical social community.

A proof event minimally involves a "proof observer" with the relevant background and interest, and some mediating physical objects, such as spoken words, gestures, hand written formulae, 3D models, or printed words, diagrams or formulae. But none of these can be a "proof" by itself, because each must be interpreted in order to come alive as a proof event.

The efficacy of some proof events depends on the marks that constitute a diagram being seen to be drawn in a certain order; e.g., Euclidean geometric proofs, and commutative diagrams in algebra; in some cases, the order may not be easily inferred from just the diagram. Therefore we must generalize from proof objects to proof processes, such as diagrams being drawn, movies being shown, and Java applets being executed.

Mathematicians habitually and professionally reify, and it seems that what they call proofs are idealized Platonic "mathematical objects," like numbers, that cannot be found anywhere on this earth. So let us agree to go along with

[32] We should not forget that, according to today's science, Newtonian mechanics, despite its tremendous utility, is not a correct physical theory, but only a practical approximation that holds within certain (not entirely well specified) limits.

this confusion (I almost wrote "joke") and call any object or process a "proof" if it effectively mediates a proof event, not forgetting that an appropriate context is also needed. Then perhaps surprisingly, almost anything can be a proof! For example, 3 geese joining a group of 7 geese flying north is a proof that $7 + 3 = 10$, to an appropriate observer. Peirce's notion of semiosis takes a cognitive view of examples like this, placing emphasis on a sign having a relation to an interpretation.

Notice that a proof event can have many different outcomes. For a mathematician engaged in proving, the most satisfactory outcome is that all participants agree that "a proof has been given." Other outcomes may be that most are more or less convinced, but want to see some further details; or they may agree that the result is probably true, but believe there are significant gaps; or they may think that the proof is bad and the result is false. And of course, some observers may be lost or confused. In real provings, outcomes are not always just 'true' or 'false'. Moreover, a group of proof observers need not agree among themselves, in which case there may not be any definite socially negotiated "outcome" at all!

Going a little further, the distinction between a proof giver and a proof observer is often artificial or problematic; for example, a group of mathematicians working collaboratively on a proof may argue among themselves about whether or not some given person has contributed substantively to "the proof". Hence we should speak of "proof participants", however they happen to be distributed in space and time, and be aware that the nature of their participation is subject to social negotiation, like everything else.

The above deconstruction of "proofs" as objectively existing real things is only the first part of a more complex story. In addition to a proof object (or process), certain *practices* (also called *methods*) are needed to establish an interpretation of a proof object as a proof event. For example, to interpret the flying geese as a proof about addition requires a practice of counting. This runs counter to the tendency, in mathematics as well as in literature and linguistics, to insist on the "primacy of the text" ignoring the practices required to bring texts to life, as well as the communities that embody those practices.

In fact, practices and their communities are at least as important as proof objects; in particular, it is clear that they are indispensable for interpreting some experience as a proof; if you can't count, then you can't see goose patterns as proofs, and if you haven't been taught about the numerals '7', '3', '10', then you can't explain your proof to the decimal digit speaking community. Of course, this line of thought takes us further from the objective certainties that mathematics likes to claim, but if we look at the history of mathematics, it is clear that there have been many different communities of proving practice; for example, what we call "mathematical rigor" is a relatively very new viewpoint, and even within it, there are various competing schools, including formalists, intuitionists and constructivists, each of which itself has many variants. Moreover, the availability of calculators and computers is even now once more changing mathematical practice.

Mathematical logic restricts attention to small sets of simple mechanical methods, called *rules of inference*, and claims that all proofs can be constructed as finite sequences of applications of such rules. While this approach is appropriate for foundational studies, and has been interesting and valuable in many ways, it is far from capturing the great diversity and vital living quality of natural proofs.

Unfortunately, we lack the detailed studies that would reveal the full richness of mathematical practice, but it is already clear that proof participants bring a tremendous variety of resources to bear on proof objects (see [45] for an excellent discussion). For example, a discussion among a group of mathematicians at a blackboard will typically involve the integration of writing, drawing, talking and gesturing in real time multimedia interaction. In at least some cases, this interaction has a high level "narrative" structure, in which sequentially organized proof parts are interleaved with evaluation and motivation in complex ways.

Aristotle said "Drama is conflict", meaning that the dramatic interest, or excitement, of a play comes from conflict, that is, from obstacles and difficulties. Anyone who has done mathematics knows that many difficulties arise. But the way proofs are typically presented *hides* those difficulties, showing only the specialized bulldozers, grenades, torpedos, etc. that were built to eradicate them. Thus reading a conventional proof can be a highly alienating experience, since it is difficult or impossible to understand why these particular weapons have been deployed. No wonder the public's typical response to mathematics is something like "I don't understand it. I can't do it. I don't like it". I believe that mathematicians' systematic elision of conflict must take a significant part of the blame for this. (Note the military metaphor used above; it is suggestive, and also very common in mathematical discourse.)

So called "natural deduction" (due to Gentzen) is a proof structure with some advantages, but it is very far from "natural" in the sense of being what provers do in natural settings; natural deduction presents proofs in a purely top down manner, so that, for example, lemmas cannot be proved before they are used. We need to move beyond the extreme poverty of the proof structures that are traditional in mathematical logic, by developing more flexible and inclusive structures. A first step towards accommodating conflict in proofs might be to allow alternative proofs that are incomplete, or even incorrect. For example, to show why a lemma is needed, it is helpful to first show how the proof fails without it; or to show why transfinite induction is needed, it may help to show how ordinary induction fails. A history of attempts to build a proof records conflicts, and hence reintroduces drama, which can make proofs more interesting and less alienating. Of course, we should not go too far with this; no proof reader will want to see all the small errors a proof author makes, e.g., bad syntax, failure to check hypotheses of a theorem before applying it, etc. As in a good movie, conflict introduction should be carefully structured and carefully timed, so that the clarity of the narrative line is not lost, but actually enhanced. The tatami system, which embodies many of these ideas, is described in [25, 26], and more detail on the application of ideas in this paper to that system can be found in

[31, 22]; for a less formal introduction to some of the ideas of algebraic semiotics, see also [46].

The narrative structures of natural proofs seem to have much in common with cinema: there is a hierarchical structuring (of acts, scenes, shots in cinema, and of proof parts in mathematics); there are flashbacks and flashforwards; there is a rich use of multimedia; etc. The traditional formal languages for proofs are also very impoverished in the mechanisms they provide for structuring proofs into parts, and for explaining these structures and parts. Probably we could learn much about how to better structure proofs by studying movies, because a movie must present a complex world, involving the parallel lives of many people, as a linear sequence of scenes, in a way that holds audience interest, e.g., see [9]. No doubt there are many other exciting areas for further exploration in our quest to improve the understandability of proofs. Success in this quest could have a significant impact on mathematics education, given the impending pervasiveness of computers in schools, and the mounting frustration with current mathematical education practices.

(The essay in this appendix was in part inspired by remarks of Eric Livingston, whom I wish to thank, though I may still have got it wrong. The remarks on narrative draw on detailed studies by the sociolinguist William Labov [39]. See [21] for some related discussion and background.)

This article was processed using the LaTeX macro package with LLNCS style