Letter Spirit:
An Emergent Model of the Perception and Creation of Alphabetic Style

Douglas Hofstadter & Gary McGraw

Indiana University
Center for Research on Concepts and Cognition
Department of Computer Science
510 North Fess Street
Bloomington, Indiana 47405
dughof@cogsci.indiana.edu gem@cogsci.indiana.edu

CRCC Technical Report number 68

February 15, 1993

Abstract

The Letter Spirit project is an attempt to model central aspects of human high-level perception and creativity on a computer, focusing on the creative act of artistic letter-design. The aim is to model the process of rendering the 26 lowercase letters of the roman alphabet in many different, internally coherent styles. Two important and orthogonal aspects of letterforms are basic to the project: the *categorical sameness* possessed by instances of a single letter in various styles (*e.g.,* the letter 'a' in Baskerville, Palatino, and Helvetica) and the *stylistic sameness* possessed by instances of various letters in a single style (*e.g.,* the letters 'a', 'b', and 'c' in Baskerville). Starting with one or more seed letters representing the beginnings of a style, the program will attempt to create the rest of the alphabet in such a way that all 26 letters share the same style, or *spirit.* Letters in the domain are formed exclusively from straight segments on a grid in order to make decisions smaller in number and more discrete. This restriction allows much of low-level vision to be bypassed and forces concentration on higher-level cognitive processing, particularly the abstract and context-dependent character of concepts.

Letter Spirit is motivated by the belief that creativity is an automatic outcome of the existence of flexible and context-sensitive concepts. Its architecture is based on the principles of emergent computation, wherein complex high-level behavior emerges as a statistical consequence of the bottom-up cooperation of many small computational actions influenced by many dynamically changing top-down conceptual pressures. Micro-agents known as "codelets" build and destroy perceptual structures in a nondeterministic parallel manner, guided throughout by letter-concepts.

Viewed from a distance, the behavior of the program can be thought of as resulting from the interaction of just four large-scale emergent agents working together to form a coherent style and to design a complete alphabet. The four agents are: the *Imaginer* (which plays with the abstract concepts behind letterforms), the *Drafter* (which converts ideas for letterforms into graphical realizations), the *Examiner* (which combines bottom-up and top-down processing to perceive and categorize letterforms), and the *Adjudicator* (which perceives stylistic aspects of letterforms and dynamically builds a representation of the evolving style).

Creating a gridfont is an iterative process of guesswork and evaluation carried out by the four agents. This process is the "central feedback loop of creativity". The full realization of the Letter Spirit program will, we believe, shed significant light on the mechanisms of human creativity.

1 Creativity and Artificial Intelligence

The Letter Spirit project is an attempt to model central aspects of human creativity on a computer. It is based on the belief that creativity is an automatic outcome of the existence of sufficiently flexible and context-sensitive concepts — what are referred to herein as *fluid concepts*. Accordingly, our goal is to implement a model of fluid concepts in a challenging domain. Not surprisingly, this is a very complex undertaking and requires several types of dynamic memory structures, as well as a sophisticated control structure involving an intimate mixture of bottom-up and top-down processing. The full realization of such a model will, we believe, shed light on the mechanisms of human creativity.

The specific focus of Letter Spirit is the creative act of artistic letter-design. The aim is to model how the 26 lowercase letters of the roman alphabet can be rendered in many different but internally coherent styles. Starting with one or more seed letters representing the beginnings of a style, the program will attempt to create the rest of the alphabet in such a way that all 26 letters share that same style, or *spirit*. Letter Spirit involves a blend of high-level perception[1] and conceptual play that will allow it to *create* in a cognitively plausible fashion.

To put the goals of Letter Spirit in perspective, it is enlightening to analyze what is lacking in many AI programs that are touted as "models of creativity". The problem is that they *don't know anything about what they are doing*. This phrase actually has two quite distinct meanings, both of which are worth spelling out, since both tend to apply to the programs under discussion. The first meaning is "the program has no knowledge about the *domain* in which it is working". The other is "the program has no internal representation of the *actions* that it is itself taking, and no awareness of the *products* that it is creating". These gaps are both serious defects in anything that purports to be a model of creativity.

In this connection, a quintessential (though little-known) example is the DAFFODIL project [Nanard et al., 1986]. Many people, if they knew about DAFFODIL, would undoubtedly suggest it as a program that has already achieved our stated goals. To show why this is a serious misconception, we must briefly describe the project. DAFFODIL is a program that combines two types of input: (1) a (static) description of each letter of the alphabet in terms of a few stroke-types; and (2) a mapping that relates each possible stroke-type (of which there may be ten or twenty) to a corresponding "curlicue" (any shape whatsoever, but usually an ornate or florid design, and of course one concocted by a human). Given these two ingredients, the program carries out a systematic *stroke* $\Rightarrow$ *curlicue* substitution operation, which yields a set of 26 novel shapes made exclusively of curlicues. A new typeface has apparently been created by the program.

To be sure, a new typeface has been *generated*; however, to think of what DAFFODIL does as in any way resembling creation by humans is a serious error, for the following reasons:

- A tacit assumption behind the DAFFODIL project is that style does not involve manipulation of abstract concepts behind the scenes, but merely involves playing with how surface-level aspects are to be rendered;
- The curlicues to be "plugged in" by DAFFODIL are all designed by a human and supplied to the program, and for that reason any creativity involved is entirely external, not internal;
- No decisions whatsoever are made by DAFFODIL — it simply mechanically plugs in all the curlicues wherever they are required, and quits;
- DAFFODIL's "understanding" of each letter is very impoverished (just a list of names of a few stroke-types), and it has no knowledge whatsoever of how letters are interrelated;
- DAFFODIL's knowledge of letters (*i.e.,* the breakdown of each letter into stroke-types) is fed to it from the outside, and once it has been fed in, it remains fixed and static;
- DAFFODIL has no ability to perceive or judge what it has created — that is, it cannot tell whether a shape it has made actually looks like an 'A', whether it is attractive or ugly, or whether it goes well or poorly with other already-created letters.

Taken alone, any of these points speaks tellingly against DAFFODIL as a model of creativity; taken together, they pretty much destroy any claims that might be made for it as such a model. However, we did not go through this exercise simply to shoot down one rival (and an obscure one, to boot); rather, the objections we have raised point the way to a deeper understanding of creativity. To be specific, we insist that for a design program to be called "creative", it must meet the following requirements:

[1] High-level perception is the perceptual process whereby sensory data activate concepts at various levels of abstraction. For a discussion of high-level perception see [Chalmers, French, and Hofstadter, 1992].

- The program itself must arguably make its own decisions rather than simply carrying out a set of design decisions all of which have already been made by a human;
- The program's knowledge must be rich — that is, each concept must on its own be a nontrivial representation of some category, and among diverse concepts there must be explicit connections;
- The program's concepts and their interrelations must not be static, but rather must be flexible and context-dependent;
- The program must play at a deep conceptual level rather than at a shallow geometric level;
- The program must be able to judge its own tentative output and be able to accept it, reject it totally, or come up with plausible ideas for improving it;
- The program must gradually converge on a satisfactory solution through a continual process in which suggestions and judgments are interleaved.

We would in fact argue that the deep (and philosophically controversial) question raised implicitly by the first point — "What would it mean for a program to make its own decisions?" — is answered by the last five points taken together.

The Letter Spirit architecture is an attempt to model all these crucial aspects of creativity, albeit in a rudimentary way. As will be described below, several of its features — nondeterminism, parallelism, and especially their consequence, statistical emergence — are key elements in allowing it to achieve these goals.

2 The Motivation of Letter Spirit

2.1 Letters as concepts

Letter Spirit focuses on a fundamental question of cognitive science and artificial intelligence: *What are the mechanisms underlying the fluidity of human concepts?* One specific form this question takes in the Letter Spirit research is: *What is the conceptual essence of the letter* 'a'*?*

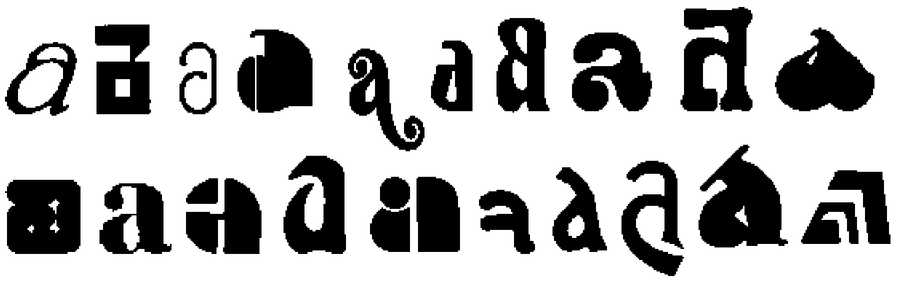

Figure 1: Twenty 'a's taken from typeface catalogues. What do all they all have in common? What is the *idea* behind lowercase 'a'?

Letterforms are subtler than people generally realize. Most literate people take letters for granted, not considering them deeply. For example, most people, if asked, would say that the letter 'a' is a *shape.* A closer look reveals an interconnected web of abstractions that make up the idea of 'a' itself (see Figure 1, which shows a number of different lowercase 'a's behind all of which lies a single shared *idea.*). Consider, for instance, how 'a' can be thought of as a marriage of two parts: a small 'c'-like loop at the bottom, and to its right an umbrella-handle-like curve that hangs over it. These two conceptual components, which we shall henceforth refer to as *roles*, are not explicit shapes *per se* but are *ideas* for what the shapes eventually drawn should be like, what their acceptable bounds are, how they should fit together, and how far they can be stretched before they no longer work.

In order to better distinguish the *concept* of a letter from various geometric shapes that may instantiate it, we introduce some new terminology. We shall distinguish three conceptual levels, which range from highly abstract to more concrete as they move towards the actual geometric letterform. We will use the term *letter-concept* to refer to the most abstract idea for drawing a letter without reference to its style. This level is comprised of a set of *letter-conceptualizations.* A typical letter-conceptualization would be the notion that a 'b' consists of two roles — a *post* on the lefthand side attached in two places to an open *bowl* on the righthand side, sitting on the baseline. A rival conceptualization for the same letter, 'b', also consists of two roles — a *post* on the lefthand side attached in one place to a closed *loop* on the righthand side, sitting on the baseline. These two conceptualizations, possibly augmented by others, constitute the *letter-concept* of 'b'. Which of these alternative conceptualizations will be used is a deep design decision. Once a specific letter-conceptualization has been chosen, notions of style give rise to a more specific and detailed letter-conceptualization that partially specifies *how each role should be realized* (of course this conceptualization still could be realized in infinitely many ways). This is called a *letter-plan*. A letter-plan is present in a designer's mind before any marks are put on paper. The actual shape drawn on paper will be called a *letterform*. Letter Spirit is concerned with all of these levels: play with letter-conceptualizations, creation of letter-plans, and instantiation of the latter as letterforms.

A vivid example of the shape/concept distinction involves lowercase 'x'. For most US-educated adults, the letter-concept for 'x' consists of a forward slash and a backward slash of the same size that cross somewhere near the middle. It is critical to stress that what is in people's minds is not a *picture* of two lines, but a set of *ideas*. English schoolchildren, in contrast to Americans, are taught to draw a lowercase cursive 'x' as a pair of small crescents facing away from each other but "kissing" in the middle. If we look at a printed 'x' in this way, we are suddenly struck by this new conceptualization. The shape drawn on paper is the same, but the conceptualization of it in our mind is very different.

The conceptual pieces into which a letter is broken in the mind's eye are its *roles*. (See [Blesser, 1973].) For example, the two crossing slashes in an imagined 'x' are roles. So also are their four tips, and the crossing-point in the middle. Each role has a different degree of *importance* to the letter — the degree to which its presence or absence matters. Of course, different shapes instantiate a given role more strongly or more weakly than others. In other words, roles are also concepts with somewhat nebulous boundaries, just as *wholes* (complete letters) are. The difference is, membership in a role is easier to characterize than in a whole, so that reducing wholes to collections of interacting roles is a step forward in simplification.

A central notion of the Letter Spirit project is the idea that the internal structure of a category is represented as a collection of interacting roles. Category membership at the whole-letter level is partially determined by category membership at the lower level of roles. In addition, *stylistic* appropriateness of a shape is judged in terms of *how roles are filled* — which is another way of saying, *how norms are violated*. For instance, consider the role *crossbar*, which belongs to conceptualizations of 'e', 'f', and 't' (at least). Norm violations include the following: "crossbar too high", "crossbar tilted upwards", "unusually short crossbar", "crossbar missing", etc. Any such violation is a stylistic hallmark that must be respected and propagated (via analogy) to other letters, even ones that lack the crossbar role.

How can a letter whose basic structure does not involve the notion of "crossbar" be influenced by a style in which crossbars are too high (say)? To carry such a stylistic trend across, one might make some *other* role be realized in an unusually high way, as long as that role can be seen as analogically close to the concept of "crossbar", and as long as the distortion does not make the resultant letterform unrecognizable as a member of its desired letter-category.

In alphabet design there is much give-and-take between local letter-category pressures and global stylistic pressures. We characterize this give-and-take as a fight between two forces: an *incentric* force, which tends to pull the shape being created *close to the center* of the intended letter-category (so that it is a strong instance of the *letter*), and an *eccentric* force, which tends to push the shape *away from the center* of the letter category (so that it better fits the desired *style*). The modeling of this conflict between incentric and eccentric forces lies at the heart of the Letter Spirit project.

2.2 The study of design and style as cognitive science

Letter perception and creation provide an elegant window onto the workings of the mind. If through working in the domain of letter design we can gain general insight into the workings of concepts and their fluid interrelations, we feel we will have made important headway on the problem of intelligence.

Like music composition or novel writing, letter design is an art form that requires years of practice. We could not possibly aspire to make a model that operates at the level of skill of an experienced human letter-designer. Instead we have drastically simplified the domain. Nonetheless, we hope to have preserved the *deep* aspects of the art of letter design while removing many of the shallower artistic qualities that, for the most part, rely on well-practiced, refined motor skills, such as: the use of elegant curves and tapering lines, knowledge of typographic conventions, and the need for a well-trained hand. Our domain, by contrast, involves very stripped-down letterforms, which are much closer to the concepts behind them than are fully fleshed-out letterforms. The specifics of this will be shown in Section 3.

Although much work has been done on character recognition and the reading of handwriting, AI has had little to say about letter-*concepts* [Hofstadter, 1985a]. So far, attempts at letter recognition and creation in AI have mostly been based on the notion that letters are basically syntactic variants of a single underlying *shape*. Such an approach ignores the cognitively and philosophically central question: *What is* 'a'*-ness all about conceptually?* Although this is already a very challenging question, an even more challenging question regarding letterforms is: *How are the letters in a given style related to one another?* This can be cast in terms of an analogy problem. Given an 'a' in a certain style, how would you make an 'e' in the same style? Of course the problem here is figuring out what "the same style" means. Transferring stylistic aspects from one letter to another involves the sort of "slippability" discussed in [Hofstadter, 1979], [Hofstadter, 1987a], [Hofstadter and Mitchell, 1991], and [Mitchell, 1993]. Stylistic qualities of one letter cannot usually be *directly* transferred to another; rather, they must be "slipped" into reasonable variants so that they fit within the conceptual framework of the new letter without bending it beyond recognition.

Letter Spirit is thus meant to simultaneously address two important and complementary aspects of letterforms: the *categorical sameness* possessed by instances of a single letter in various styles (*e.g.,* the letter 'a' in Baskerville, Palatino, and Helvetica) and the *stylistic sameness* possessed by instances of various letters in a single style (*e.g.,* the letters 'a', 'b', and 'c' in Baskerville). A simple picture illustrates these two ideas and shows their relationship to one another (see Figure 2).

Given a few letters in a *gridfont* (our name for the highly constrained, gridbound alphabets used in the Letter Spirit project), the challenge to Letter Spirit is to figure out what category a given seed letter belongs to and what the stylistic tendencies suggested by the seed letter(s) are, and then design the remaining letters in what it considers to be the same style. Of course, there is never just one good answer, since seed letters do not uniquely specify a style, nor does a style uniquely specify its constituent letterforms.

Letter Spirit is primarily intended to be a cognitive model. However, Letter Spirit is also an AI program. Even if its cognitive mechanisms turned out to be very different from human mechanisms, Letter Spirit would still be interesting as an approach to machine creativity. Traditional AI has often been criticized for its brittleness [Holland, 1986], which could be considered the antithesis of creativity. New types of mechanisms are thus needed if AI is to successfully model creativity (along with a host of other human activities that require intelligence). We believe that the Letter Spirit architecture, which is based on subcognitive, emergent computation, will extend the ability of computers to act intelligently.

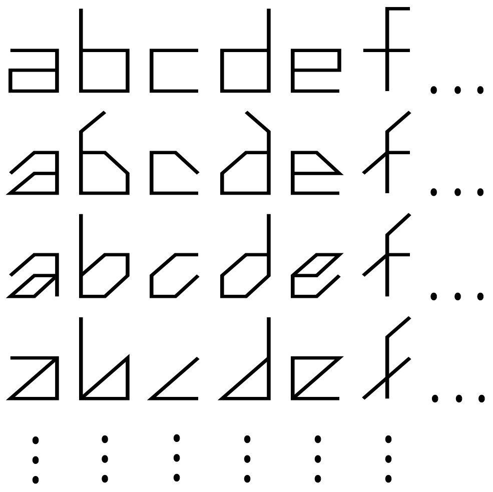

Figure 2: Items in any column have *letter* in common. Items in any row have *spirit* in common.

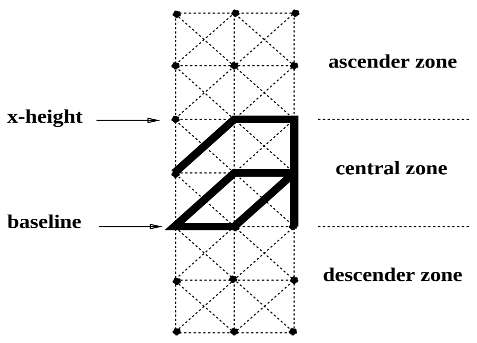

Figure 3: The grid, with one of the many possible sets of quanta instantiating an 'a' turned on.

3 The Domain

3.1 Real-world and microworld typeface design

The initial ambition of the Letter Spirit project was to write a computer program to design full-fledged typefaces having the same order of complexity as conventional typefaces such as Helvetica, Times, Baskerville, and so on. This quickly revealed itself to be too hard and of little interest to cognitive science. The problem is that real-world typeface design involves far too many nuances to allow practical simulation in a computer model. Moreover, full-scale simulation of this human ability would involve the modeling of so many domain-specific details that the cognitive issues meant to be the core of the project — the nature of fluid concepts and the nature of style — would be lost among a welter of irrelevant noncognitive concerns.

For this reason we deliberately decided to eschew all the complexities involved in curves, stroke-taperings, and other aspects of curvilinearity that seem to involve low-level or intermediate-level vision rather than the high conceptual level that was our main focus. We felt that forbidding the manipulation of surface-level aspects of letterforms would simultaneously reduce the complexity of the programming task and force concentration on deeper, more *conceptual* levels of the design process (what we call *deep style*). At the deepest levels, style involves playing with the conceptual foundations of letters (*e.g.,* conceptualizing an 'x' as *a 'v' on top of an inverted 'v'* instead of as *two intersecting slashes*). This distinction between deep and shallow style is of course not a black-and-white one. Instead there is a continuum, with shallow style at one end and deep style at the other. Letter Spirit is focused on the deep-style end of the continuum.

3.2 The grid

To avoid the need for modeling low-level vision and to focus attention on the deeper aspects of letter design, we eliminated all continuous variables, leaving only a small number of discrete decisions affecting each letterform. Specifically, letterforms are restricted to short line segments on a fixed grid having 21 points arranged in a 3×7 array [Hofstadter, 1985b]. Legal line segments, called *quanta*, are those that connect any point to any of its nearest neighbors horizontally, vertically, or diagonally. There are 56 possible quanta on the grid, as shown in Figure 3.

Letters created on the grid might be thought of as "skeletal" geometric representations of letter-concepts. To some people, this would suggest "fleshing them out" into "true" letterforms by adding curvature, thickness, and so on. (This could even be carried out by a DAFFODIL-like program!) Although this is possible, we choose to think of the gridbound letterforms as full-fledged letters. Seen as stand-alone typefaces, some gridfonts are visually fascinating — even beautiful (see Figures 2 and 6 as well as Figure 9) — and might make acceptable typefaces for certain purposes, but most of their beauty is more intellectual and flows from the ways in which they illustrate and explore the conceptual essences of letters, as opposed to the more sensual or "retinal" beauty of fully fleshed-out, curvilinear typefaces.

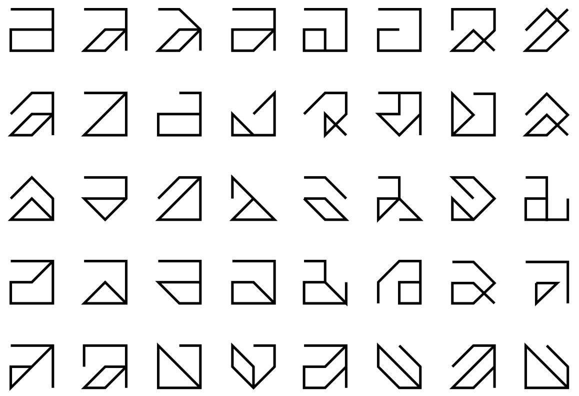
Figure 4: Several renditions of the letter 'a' on the grid. They are by no means all equally strong members of the category 'a' — in fact, some are rather weak.

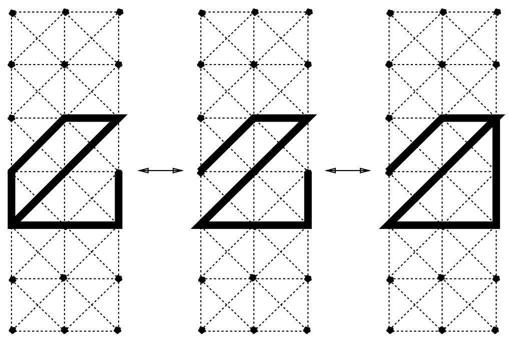
Figure 5: Rapid metamorphosis of an 'e' into an 'a' via 'z', with merely a one-quantum change in each step.

Since any quantum is either on or off, decisions on the grid are coarse. Surprisingly, the variety among letters of a given category is still huge — hundreds of versions of each letter have been designed by humans. So far, about 600 complete gridfonts have been designed.[2] Surveying them all, one is struck by the diversity present in each letter-category. Almost paradoxically, the Letter Spirit domain's severe limitations seem to engender this diversity. As an example of the richness the grid allows (even within one letter-category), see Figure 4, which shows 40 of the approximately 1500 different 'a's, ranging from very weak to very strong, that have been created by people.

[2]A number of representative gridfonts are included in Figure 9 of this proposal. For more see [Hofstadter, 1987b].

Since it is impossible to make tiny changes on the grid, any two instantiations of a given letter are significantly different from each other. Decisions to add or subtract even one quantum often fundamentally affect category membership. Figure 5 illustrates the large distance in letter-space one can travel with minimal changes on the grid. A "small" change in the letterform — the erasure of just one quantum — changes the first shape from a strong 'e' to a strong 'z'. A similar one-quantum addition transforms the 'z' into an 'a'. So category membership in the Letter Spirit domain is a tricky matter. In fact, modeling the process of deciding which of the 26 categories a given shape on the grid belongs to (if any) is one of the hardest aspects of the project and has represented the major focus of our initial implementation work.

The human-designed gridfonts *Benzene*, *Intersect*, *Poise*, and *Square Curl* (Figure 6) show how deeply style can affect the letterforms of a gridfont, and illustrate a variety of devices for assuring stylistic consistency, such as motifs and abstract rules (discussed further in Section 5). Note, for instance, the "benzene-ring" motif found in some form in every letter of *Benzene*. The main constraint that inspired *Intersect* was that two convex curves must intersect to form each letter. This is an abstract rule rather than a geometric motif. This constraint was not hard-and-fast — in fact, it had to be "slipped" occasionally in order to create acceptable versions of a few letters (*e.g.,* 'c', 'i', and 'j'). It is not surprising that style gets somewhat diluted in the name of creating letters that are easily recognizable; however, the converse is true, too. Occasionally, instances of letter-categories can be weakened to a surprising degree in the name of maintaining stylistic coherence. See, for instance, 'k', 'r', 'w', and 'z' in *Poise*. *Square Curl* involves both an abstract rule (total avoidance of diagonal quanta) and a motif (essentially the shape of the 'n').

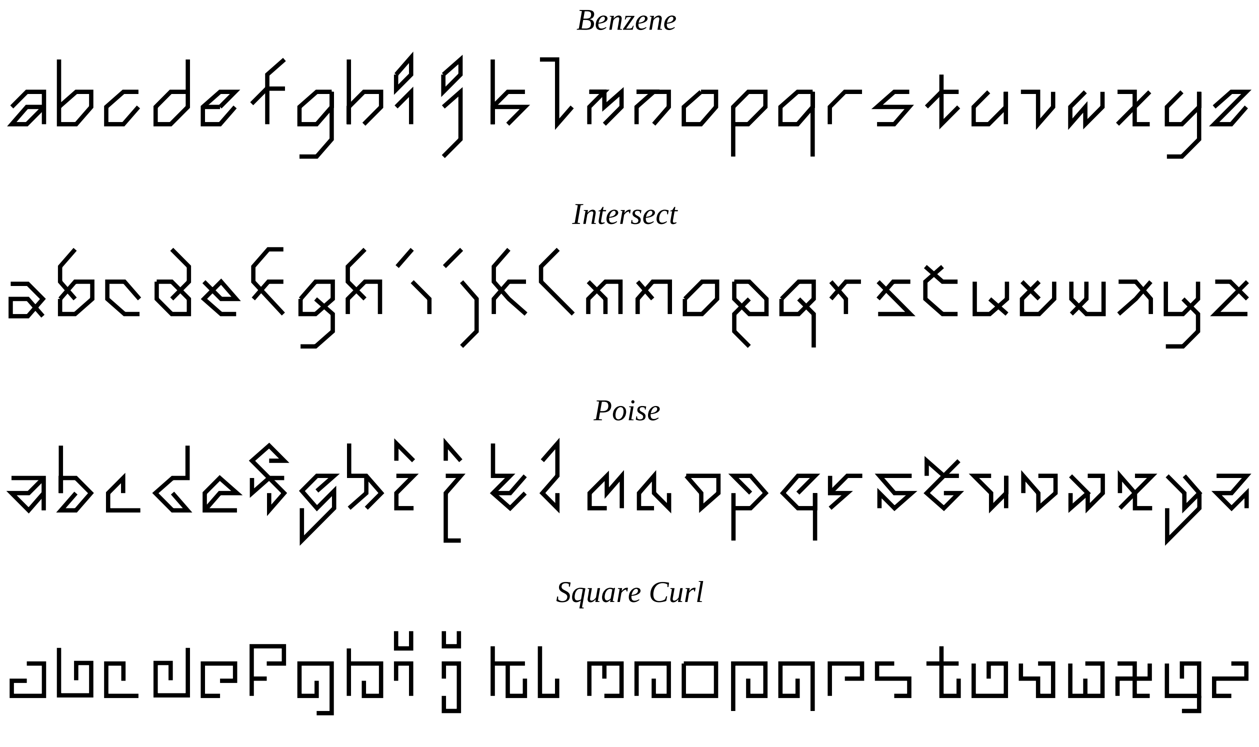

Figure 6: The human-designed gridfonts above illustrate how deeply style can affect letterforms in the gridfont domain.

As was mentioned above, the constraints defined by the grid actually encourage tampering with letter-concepts at a deep level. Because there are no fine-grained features to manipulate, one ends up playing at the boundaries of the 26 categories. Consequently, many gridfonts are wild, sometimes having angular, blocky, spiky, sparse, or otherwise bizarre letters. (See Figure 6 and Figure 9.) We must stress that we are not striving for typefaces with high readability or letterforms beautiful at the surface level. Rather, we are attempting to understand the conceptual nature of letterforms and thereby gain insight into what imbues concepts *in general* with their fluidity. Pushing the 26 categories to their edges in a coordinated way often results in an intellectual beauty not localized in any single letterform of a gridfont, but spread out over the gridfont as a whole — the beauty of *spirit* rather than of *letter*.

While at first glance, the Letter Spirit domain might be shrugged off as a "toy domain", this would be a gross underestimate of its subtlety. In spite of — or rather, *because* of — the reduction to the grid, the Letter Spirit challenge is, in terms of cognitive-science issues, extremely rich. The cognitive issues are magnified, not reduced, by the act of simplifying the domain. All that has been thrown out is the need for expertise. One does not need to be a professional typeface designer or lifelong student of letterforms to appreciate the consistency of a well-designed gridfont. Even a novice can design a passable gridfont, although doing a sophisticated one is very difficult.

4 Creating a Gridfont — A Broad View

For a person, designing a gridfont usually takes between ten minutes and three hours (after which the font still remains potentially subject to scrutiny and minor revision). The process involves myriad small operations, ranging from the highly mechanical to the highly inventive. A typical action (*e.g.,* creating a 'k' or even changing a single quantum in an already-designed letter) sets off repercussions that echo throughout the continuing design process, all over the gridfont, and at all levels of abstraction. This activity is largely guided by *a priori* notions of the interrelatedness of letter categories (*e.g.,* 'd' and 'b' are often considered to be near-reflections of each other, 'n' and 'u' to be near-rotations of each other, etc.). Many such actions occur, and eventually a stable gridfont emerges.

For a human designer, dissatisfaction and contradiction either within a given letterform or between letters are the prime movers of the design process. If Letter Spirit is to be faithful to how humans create, it must be capable of *recognizing* and *resolving* such conflicts, in order to create a coherent style. Sometimes the conflicts are subtle, and therefore require refined artistic judgment-calls. Modeling the ability to find conflicts, diagnose them, and convert diagnoses of problems into reasonable suggestions for solutions is a key aspect of the project.

Any design decision will affect not only the letter currently under consideration but also conceptually close letters. For example, a specific decision as to how to instantiate the role of *post* in 'b' is likely to have an enormous effect on the post of the 'd', as well as on many other letters with posts, and may also noticeably influence the *stems* of the 'p' and the 'q'. Of course, the extent and type of this influence are highly dependent on the particular style and are in no way mechanical or formulaic. Most aspects of letterforms are likely to propagate their influence to varying extents through the entire gridfont. The propagating wave will probably cause many retroactive adjustments (both major and minor) to some "already finished" letters, and give rise to ideas for letters not yet designed. One design decision will typically spark others, which in turn will engender others, and so on.

Eventually, when enough "decision waves" have washed over the entire gridfont, all the letterforms begin to have a high degree of internal consistency, and a clear style begins to emerge. Once the tension of inconsistency eases up, no more large-scale changes are required. Minor adjustments may continue, but for the most part, the large-scale creative act will be finished. This temporally-extended, serial process of integration and gradual tightening of internal consistency is an indispensable part of true creativity, and is a well-known property of such creative acts as musical composition, the writing of poetry and prose, the activities of painting and sculpture, the evolution of scientific theories, and even the design of AI programs.

The architectural structures required to model any dynamic creative process are necessarily complex. We now move on to a discussion of the implementation of such structures in Letter Spirit.

5 Four Global Memory Structures

The Letter Spirit program will contain four dynamic memories, each concerned with different levels of concreteness and abstraction of shapes (and concepts pertaining to shapes). These memories are:

- the **Scratchpad**, which can be thought of as *a virtual piece of paper* on which all the different letters of a given font are drawn and modified; as such it is more a type of external memory than an aspect of mental activity;
- the **Visual Focus**, which can be thought of as *the site where perception of a given letterform occurs*; in it, perceptual structures are built up and converge to stable categorical and stylistic interpretations;
- the **Thematic Focus**, which can be thought of as the program's *dynamically changing set of ideas about the stylistic essence of the gridfont under way*; in it are recorded stylistic observations of all sorts concerning letters already designed, and if and when some of these passive observations start to be perceived as falling into patterns, those patterns can be taken as determinant of the style, meaning

that they can be elevated to the status of explicit *themes* — ideas that play an active role in guiding further design decisions, in the sense of serving as "pressures" on the construction of further letters;

- the **Conceptual Memory**, which can be thought of as the program's *locus of permanent knowledge and understanding of its domain*, and which, for each full-fledged concept, has three facets: (1) a set of *category-membership criteria*, whose purpose is, roughly, to specify the recognition requirements for instances of the concept in terms of more primitive concepts; (2) a set of *explicit norms*, which encode certain aspects of the concept's "core"; and (3) an *associative halo*, consisting of links having time-varying lengths connecting the concept with various other related concepts, essentially giving a sense of where the concept is located in "conceptual space" by saying what it most resembles.

A useful perspective on these four structures is afforded by the following set of rough equivalences with various familiar types of memory. The Scratchpad can be thought of as an *external memory device*; the Visual Focus as a *subcognitive workspace* (that is, a very-short-term cache-like working memory in which parallel perceptual processes, mostly occurring below the system's threshold of awareness, collectively give rise to rapid visual classification of a shape, whose final category assignment is made cognitively accessible); the Thematic Focus as a *cognitive workspace* (that is, a much slower, and therefore more conscious, level of working memory in which abstractions derived from more concrete and primary perceptions are stored, compared, and modified); and finally, the Conceptual Memory can be thought of as a *permanent semantic memory* containing both procedural and declarative knowledge of the system's repertoire of concepts. We now describe each of these memory structures in more detail.

The **Scratchpad** is the place where experimental letterforms are created and critically examined. At the beginning of a run it is empty; by the end of a run it contains at least 26 completed letterforms. We say "at least" since it is certainly possible for the program to have created and stored a number of alternate forms for a given letter. This is a frequent occurrence in design by humans, and we see no reason to force the program to completely discard ideas that it creates, even if it winds up preferring other ones in the end. Such near-miss letterforms are in fact very useful windows onto the creative process, and often can serve as seed letterforms for another gridfont. In this way, at least theoretically, the program could generate its own seed letters rather than needing human input for each new gridfont. Hypothetically, one can envision the program finishing one gridfont and then being "eager" to use rejected ideas as the starting point for new gridfonts, thus becoming a self-driving creator!

Little needs to be said about how the Scratchpad is structured; it simply contains an arbitrary number of grids, each of which is a 56-bit data structure telling which of the 56 quanta in the grid are turned on and which are off. All higher levels of perception of the grids on the Scratchpad are the responsibility of the Visual Focus, to which we now turn.

The **Visual Focus** is where recognition of a single letterform takes place. It can be thought of as a busy construction site where quanta belonging to a letterform are fused together into small structures of various sizes and shapes. These structures are then interpreted as roles, and any particular combination of roles present suggests membership in one or more letter-categories.

At the beginning of a run, processing in the Visual Focus is purely bottom-up; gradually, however, as structures are built up, top-down influences enter the picture, with top-down processing guiding the "docking" of syntactic parts into semantic slots. This is explained further in Section 7.

Structure in the Visual Focus is built up in parallel by many small computational micro-agents called *codelets*. A useful image is that of a large structure (like a bridge) being built by a colony of hundreds of ants or termites. The ants work semi-independently, but cooperatively. Codelets (further described in Section 7) correspond to the ants in this metaphor, and perceptual structures to the bridge. So perceptual structures develop nondeterministically but not haphazardly. From the hundreds of tiny probabilistic decisions, a coherent view emerges.

The **Thematic Focus** is the site where stylistic attributes come to be recognized, especially if they crop up in one letter after another. The more a stylistic attribute is seen as a systematic pattern, the more chance it has of making an upward shift in status — rising from being merely a casual observation, essentially a passive entity, to becoming an active entity: an officially sanctioned guiding principle, or *theme*.

This type of "elevation to themehood" is a little-appreciated but pervasive aspect of creative acts. In working on a gridfont, a human designer starts by being inspired by (let us suppose) a single seed letter. Aspects of this letter, borrowed analogically, suggest further letters. But each of these new letters, when looked at by a style-conscious eye, will be seen to have stylistic attributes that are entirely its own and that were not implicit in or implied by the seed letter alone. These unexpected attributes are a result of the interaction of the constraints defining a perceived style with the completely unrelated constraints defining the given letter-category. In other words, these stylistic attributes are unpredictable emergent by-products of the creation of new letters. Once such an attribute is recognized and explicitly elevated to themehood, it becomes an active force in shaping yet further letters. Of course this means the process is in some sense recursive — new letterforms give rise to new emergent attributes, which in turn give rise to new letterforms, and so on. The upshot is that new stylistic attributes are continually emerging. All of this adds up to a highly unpredictable meandering in "style space", reflecting the extreme subtlety of the creative act.

Numerous types of stylistic pattern characterize a gridfont as a whole, including the following:

- A *role trait* characterizes how a specific role tends to be instantiated, independently of the specific letters it belongs to. In other words, a role trait is a "portable norm-violation" — a norm violation attached to a specific role and thus capable of affecting a number of different letters. For instance, in a particular style, the roles of *post* and *stem* might be realized several times over in a zigzaggy manner; in some other style, the role of *bowl* might be realized in various letters in a tall and narrow manner; in yet another style, the role of *crossbar* might be realized in one or more letters by passing through a "hole" or "gap" in the associated vertical stroke; and so on. Each of these stylistic features is a role trait: a description of how a specific role has been (or might become) instantiated in various letters.
- A *motif* is a geometric shape used over and over again in many letters. If it is very simple (*e.g.,* a two-quantum backslash crossing the central zone), it may be required to appear in complete form in every single letter. If the motif is a more complicated shape (*e.g.,* a perfect two-quantum-by-two-quantum square, or a hexagon that looks like a tilted benzene ring), then parts of it may be allowed to be absent from various letters, so long as a reasonably substantial portion of it remains. Some styles allow a motif to appear in reflected, rotated, and/or translated form; others allow translation but no reflection or rotation; yet others allow reflection and/or rotation but no translation; and so on.
- An *abstract rule* is a systematic self-imposed constraint such as: allowing no diagonal quanta; allowing only diagonal quanta; requiring each letter to consist of precisely two disjoint parts; forbidding any straight section of more than one quantum in length; and so on. It is not that there is some particular shape that is repeated, but rather that a rule of some sort is globally enforced.
- *Levels of enforcement.* The three preceding types of stylistic attribute pertain directly to shapes on the grid. A much more abstract determiner of style — in fact, a kind of "meta-level" aspect of style — is the degree to which any such constraint is considered "unslippable" (*i.e.,* absolute or inviolable), as opposed to being "slippable" (*i.e.,* allowed to be disrespected under sufficiently great pressure). The level of enforcement of a stylistic constraint — strict, lax, or somewhere in between — sets an abstract, almost intangible tone for the entire gridfont. Whenever a new stylistic attribute is elevated to themehood during the creation of a gridfont, the default assumption is made that it is unslippable. By definition, default assumptions are not noticed at the moment they are made, and so the new theme will start out being obeyed stringently without question. However, if down the road some letter/spirit conflict gets sufficiently severe, pressures may arise that call that automatic assumption into question, at which point the level of enforcement will emerge from hiding and become an explicit variable that can be adjusted. Although it takes considerable pressure to "unbury" any of them, all levels of enforcement are ultimately under the control of the program, and can thus, in principle, be tampered with during the design process.

All of these stylistic attributes are explicitly represented in the Thematic Focus and are thus globally accessible, not just in the sense of being *observable* by agents in the program but also in the sense of being *alterable* by agents in the program. Thus thanks to the Thematic Focus, all aspects of style are under complete control of the program itself.

The **Conceptual Memory** provides each concept in the domain with an internal definitional structure and a local conceptual neighborhood. Roughly speaking, a concept's "internal definitional structure" consists of its specification in terms of simpler concepts, and a concept's "local conceptual neighborhood" consists of its links to peer concepts in conceptual space. The internal definitional structure itself breaks down into two facets: *category-membership criteria* and *explicit norms.* Finally, a concept's local conceptual neighborhood is known as its *associative halo*, and one of its main purposes is to serve as the *source of conceptual slippability.* What this means is that, in times of severe pressure, the possibility arises that the concept itself might "slip to" some concept in its associative halo, with closer concepts of course being more likely slippages. This means that the nearby concept is tried out, and possibly accepted, as a substitute for the concept itself. We now say a bit more about each of these three aspects of a concept.

Category-membership criteria specify perceptual criteria that contribute toward membership in the category, with different weights attached to the various criteria, reflecting their level of importance. The purpose of this weighted set of criteria is, roughly, to "reduce" the concept to a collection of more primitive, more syntactic notions (*e.g.,* a letter is "reduced" to a set of interacting roles, or a role is "reduced" to a weighted set of desired properties of an adjusted part).

By contrast, a set of *explicit norms* exists (at least for roles and wholes, which are sufficiently "semantic" concepts), the purpose of which is to make the "core" of the concept explicitly accessible to agents seeking ways to make a weak instance of the concept stronger, or conversely, to make a strong instance somewhat weaker without casting its category membership into complete limbo. Norms represent the program's knowledge about the internal structure of each category in its domain.

A concept's *associative halo* consists of time-varying links of varying lengths connecting the concept with various other concepts. The links encode such information as standard resemblances between letters, analogical relationships between different types of roles, conceptual proximity of various descriptors used in specifying norm violations, and so on. Knowledge of letter–letter resemblances serves a dual function. It not only helps in designing new letters (*e.g.,* a good heuristic for a first stab at 'u' is to rotate 'n'), but also serves as a warning that one letter has a tendency to be confused with another (*e.g.,* 'b' and 'h' can easily be confused). The set of links from any concept effectively gives that concept a halo of conceptually close, potential-substitute concepts — concepts to which it might slip under sufficiently great contextual pressures.

6 Four Interacting Emergent Agents

Emerging from the many small actions of codelets are four conceptually separable types of large-scale activities:

1. the high-level conceptual activity of *devising a new letter-plan* (*i.e.,* either an idea for an as-yet undesigned letter or a possibility for improving an already-designed letter);

2. the intermediary activity of *translating a new letter-plan into a concrete shape* on the Scratchpad;

3. the relatively concrete perceptual activity of *examining a newly-drawn shape and categorizing it* (*i.e.,* deciding which letter of the alphabet it is, and how unambiguously so);

4. the more abstract perceptual activity of *recognizing the stylistic attributes of a newly-drawn letter, and judging them* (*i.e.,* finding "exportable" ways of describing how a given letterform violates norms, and deciding how well the letter's attributes fit with those of other letters in the developing gridfont).

It is often convenient to speak as if these emergent activities were carried out by four explicit and cleanly separable modules, together comprising the totality of the program. (These agents could be likened to the agents referred to in [Minsky, 1985].) The names we shall apply to these hypothetical modules or agents are, respectively, the *Imaginer*, the *Drafter*, the *Examiner*, and the *Adjudicator*, and we shall briefly describe each of them in turn. However, it must be borne in mind that these modules are in some sense convenient descriptive fictions, in that each one is simply an emergent by-product of the actions of many codelets, and their activities are so intertwined that they cannot be disentangled from each other in a clean way.

The **Imaginer** does not deal with, or even know anything about, the constraints defined by the Letter Spirit grid (*i.e.,* the fact that letterforms are made up of discrete quanta); rather, it functions exclusively at the abstract level of *roles*. Its job is to make suggestions regarding roles, which it then hands over to the Drafter (which will attempt to implement them concretely in conformity with the constraints of the grid — that is, as parts composed of quanta). The Imaginer can make suggestions of two distinct types — *norm-violation* suggestions and *role-regrouping* suggestions. Although both types can lead to highly novel instantiations of letters, suggestions of the first type tend to be tamer than ones of the latter type.

A *norm-violation* suggestion assumes that a set of interacting roles (*i.e.,* a particular conceptualization for the letter) has already been specified; then, for one or more roles in that set, it suggests one or more ways of violating associated norms. For instance, suggestions such as "Bend the tip of the ascender to the right", "Use a short crossbar", "Don't let the bowl touch the post at the x-height", "Make the bowl narrow", and so on (suitably expressed in an internal formalism) would be typical norm-violation suggestions. Though fairly specific, such suggestions still require more fleshing-out to be realized on the grid itself.

A *role-regrouping* suggestion is more radical and deep, in that it involves tampering with the very essence of the letter — in other words, coming up with a new conceptualization for the letter. This means taking apart one or more roles and making new roles that combine aspects of the old roles. An easy-to-grasp example is the above-described conceptual move from imagining 'x' as two intersecting slashes to imagining it as two "kissing" angle-brackets. Role-regrouping is very subtle because it takes place completely at an *abstract* level. That is, no *shapes* are involved at any time; rather, the Imaginer deals exclusively with abstractions — abstractions that, to be sure, have the general "feel" of shapes, in that they are associated with spatial locations and have spatial functionalities — but such abstractions are not shapes.

For the sake of concreteness, let us consider the conversion of one conceptualization of 'x' into another. The idea of a "linear unit" (shorthand for the concept "long thin shape that does not bend much") passing through some point suggests in a fairly obvious manner the idea of breaking the linear unit into two shorter linear sub-units, using the point as the divider. Applying this "breakup" idea to each of the two *slash* roles in an 'x', we obtain four shorter diagonal roles that converge like spokes to a hub. Now comes the regrouping operation. Whereas the "northwest" and "southeast" sub-units had formed one conceptual linear unit, and the "northeast" and "southwest" sub-units another, we now pull each of these units apart and regroup their pieces into two new conceptual units — one made up of the "northwest" and "southwest" sub-units, and the other of the "northeast" and the "southeast" sub-units. (Keep in mind that the pulling-apart and rebonding are being done *conceptually*, not as a splicing-operation on any genuinely pictorial entities.) We now have two new roles (and of course the unperturbed *central-point* role) that can serve as an alternate way of thinking about what 'x' is — that is, a novel starting-point for creating new 'x's. Norms for the roles in this new conceptualization have to be derived from the norms associated with the roles in the old conceptualization.

Once a new conceptualization has been produced, it can be handed over directly as a suggestion to the Drafter, or norm-violation suggestions can be made in addition, and then the whole thing handed over as a package to the Drafter.

The **Drafter**, unlike the Imaginer, does know about the grid; indeed, its main function is to take the Imaginer's grid-independent suggestions and adapt them to the down-to-earth and severe constraints of the grid. In other words, the Drafter must convert the Imaginer's abstract suggestions into concrete instructions for drawing shapes on a grid on the Scratchpad.

Here is an example that could easily come up in designing a 't' or an 'f'. A norm-violation suggestion like "Make the crossbar short", which in real-life circumstances would offer a letter-designer a full continuum of possibilities, offers much less freedom to a designer restricted to the grid. For a gridbound 't' or 'f', a conventional (*i.e.,* norm-respecting) crossbar would be a horizontal line segment composed of two quanta. Obeying the suggestion to make it short would thus seem to offer just three alternatives: dropping the left quantum, dropping the right one, or dropping both. Of course, dropping both quanta would seem very drastic, if not outrageous (although possibly the right solution for some very far-out styles); thus in all likelihood, the Drafter should opt for drawing just a single quantum, probably a horizontal one. Then the question is whether it should draw it to the left or to the right of the ascender. This choice will certainly depend in part on precedents in other letters in the developing gridfont (*e.g.,* if the decision "Draw a quantum on the *left* side of the ascender" had already been made in the case of the 'f', then the 't' might want to follow suit), but it will also depend on how strong the potential letter's category membership will be.

The Drafter will generally be uncritical of the suggestions it receives from "on high", but it is conceivable that it will encounter so much difficulty converting them into reasonable instructions for a grid-shape that it will send back a "complaint" to the Imaginer, asking it to revise its suggestion in some way. Generally, however, feedback to the Imaginer comes from the perceptual agents — the Examiner and the Adjudicator. This all-important feedback loop will be described below.

The **Examiner**'s responsibility is to take the specification of a grid-letter in terms of its quanta and to determine which (if any) of the 26 letter-categories it belongs to, and how strongly and unambiguously so. It is useful to cast the Examiner's work in terms of *syntactic* and *semantic* operations.

Syntactic operations are purely bottom-up chunking operations. They serve to put quanta together in a way that would be reasonable no matter what type of shape was being recognized. In other words, they are *context-free chunkings* that would presumably arise as the result of any naturally-evolved visual system. Semantic operations, on the other hand, depend on the set of categories into which the shapes are being channeled — a writing-system-specific repertoire that in a person has been acquired through experience. Semantic operations take the output of syntactic actions (which occur earlier in perceptual processing) and adjust it so that it conforms sufficiently well to expected abstract structures. The upshot is a "marriage" of bottom-up structures coming from sensory stimuli with top-down expectations defined by letter-concepts.

All processing in the Examiner takes place in the Visual Focus. Processing begins at the level of quanta and starts out being completely bottom-up. Quanta get syntactically chunked into *parts*, which are then assigned any number of syntactic labels (*e.g.,* "straight", "tall", "central-zone", "zigzag", "left-side", "slanting", "open on right", etc.). Top-down semantic influence enters the picture as the labeled parts are matched up with conceptual *roles.* As the interpretation rises towards the level of letters (which we call *wholes*), even more top-down influence is brought to bear. Much of the Examiner has already been implemented; for a detailed description of how it works, see Appendix A.

The **Adjudicator** is concerned with a more abstract type of category membership — namely, *stylistic consistency.* It is not enough for a candidate letterform to be perceived by the Examiner as a strong member of its intended letter-category; that letter must also be judged by the Adjudicator as embodying the same stylistic qualities as the seed letter(s) and any already-generated letters. This requires a set of high-level descriptions of stylistic qualities to be manufactured as the alphabet develops. No single letter contains all the information about style, so stylistic attributes from various letters, as they come into existence, must be assembled in a global list belonging to the gridfont as a whole. This is of course the Thematic Focus. Thus whereas the 26 letter-categories exist in the Conceptual Memory *prior* to any run, a single stylistic category gradually comes into existence in the Thematic Focus *during* a run.

The types of stylistic attributes that the Adjudicator looks at in order to judge a candidate letterform include *role traits*, *motifs*, *abstract rules*, and *levels of enforcement.* A given letterform is inspected for the presence of established themes, and is given a "stylistic-coherency rating" according to how many themes are echoed in the letter.

In addition to looking for attributes that have already been established as part of a style, the Adjudicator also tries to extract from any new letter *new* stylistic attributes, in order to extend the set of themes defining the emerging style. An attribute discovered in a single new letter may not be considered strong enough to be elevated to themehood, but if that observation is reinforced by finding it echoed in other new letters, then it stands a good chance of becoming a new theme and thus driving the design of further letters and even the retroactive modification of older letters.

Note that stylistic attributes can emerge in a completely unpredictable fashion. If the Adjudicator happens to notice that neither the seed letter nor the first new letter generated contains any vertical strokes, then it may generalize from these two examples and thereafter forbid vertical strokes. Such a decision will of course have global ramifications, and on a different run where the same two letters existed, a totally different course could be taken if that observation were not made, or if the interdiction were taken loosely.

As this brief survey shows, recognizing stylistic characteristics and formulating them in an appropriate manner (neither too specific nor too general; neither too rigid nor too loose; etc.) is most complicated, and the Adjudicator's job is therefore a very subtle one.

6.1 The central feedback loop of the creative process

The entire Letter Spirit challenge can be thought of as the problem of *attempting to do in one framework something that has already been done in a significantly different framework.* Here, the two frameworks are different letter-categories, such as 'd' and 'b'. A designer is handed a somewhat off-center member of the first category and the challenge is to transport its eccentricity — its stylistic essence — into the other category, or in other words, to reproduce the *spirit* of it in the second framework, despite the fact that the second framework is by no means isomorphic to the first. Such transport of spirit cannot be done in a reliable manner except by trial and error. Guesses must be made and their results evaluated, then refined and evaluated again, and so on, until something satisfactory emerges in the end. We refer to this necessarily iterative process of guesswork and evaluation as "the central feedback loop of creativity".

We now run through one example, and a rather simple one at that, in order to give a sense of how all four agents are involved in this loop. We suppose that Letter Spirit is given as its lone seed letter an 'f' with a conventional ascender (*i.e.,* a tall vertical stroke on the left side of the grid that curves over to the right at the top), but with *no crossbar at all* (see Figure 7). What kind of grid-letters might this seed inspire? What kind of overall style?

To begin with, the seed's letter-category must be identified (in itself a nontrivial task, given the eccentricity of the letterform). To this end, the Examiner is invoked. The quanta in the seed letter are quickly glued together, and because there is a fairly weak juncture near the top — namely, where the ascending line bends to the right to form an "overhang" — two distinct syntactic parts are made. After being suitably labeled, these parts wake up two semantic roles: *post* and *hook*, and since there is nothing else to see, no other roles are strongly activated. This pair of activated roles now activates two *wholes* — the letter-categories 'f' and 'l'. Counting against 'f' is the lack of crossbar, but counting against 'l' is the hook at the top. The power of two strongly-filled roles overwhelms the pressures for seeing 'l', and the shape winds up being seen as an 'f' whose primary stylistic attribute is the lack of anything to fill the *crossbar* role. Thus "crossbar suppressed" is the main stylistic note (*i.e.,* norm violation) attached to the letter.

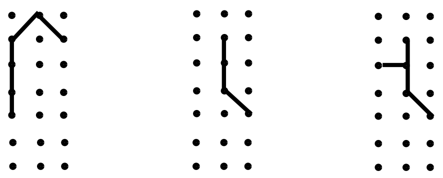

Figure 7: The 'f' with no crossbar (left) gives rise in the Imaginer to a 't' with no crossbar (middle). This is rejected by the Examiner since it is too 'l'-like. This leads the Imaginer to slip "no crossbar" to "short crossbar" and a better 't' is created (right).

We now move from the perceptual to the generative phase. Given that 'f' and 't' are linked as similar letters in the Conceptual Memory, there is a high probability that 't' would be the next letter tackled. An obvious idea for the 't' would be to suppress *its* crossbar. Like any good copycat, the Imaginer would have little trouble coming up with that analogy, since the role *crossbar* exists in both 'f' and 't'; all it would need to do is take the norm-violation that describes 'f' ("crossbar suppressed") and copy it literally into a suggestion for the 't'. Upon receiving this norm-violation suggestion, the Drafter would have no problem converting it into a grid-oriented instruction saying, in effect, "Draw no horizontal quanta at the x-height." (Let us assume that the 't's ascender would be a conventional one.)

The Drafter renders this attempt at 't' on the Scratchpad, leaving it up to the Examiner to look at it and make of it what it can. The quanta are quickly put together into a single perceptual part — namely, a vertical line rising from the baseline to somewhere above the x-height. This wakes up the role *ascender*, and since there is nothing else to see, no other roles are strongly activated. This rather sparse "combination" of activated roles now sharply activates one and only one whole — namely, the category 'l'. At this point, the Examiner, knowing that 't' was intended, pronounces the attempt at 't' a failure, and provides what is hopefully an accurate diagnosis: the fact that the role *crossbar* never got awakened at all.

This information goes back to the Imaginer, which was, after all, the source of the idea of suppressing the crossbar entirely. So the Imaginer is now caught in the crossfire of Letter and Spirit pressures: on the one hand, it knows that suppressing the crossbar leads to disaster (this is Letter pressure), but on the other hand, it wants to follow the stylistic lead of the 'f' (Spirit pressure). Something has to give!

Luckily, there is a way out, provided by *creative slippage*, which involves consulting the Conceptual Memory for potential substitutes provided by conceptual halos. In the halo of "suppress", the Imaginer finds such close neighbor-concepts as "austerity", "minimality", "sparsity", as well as the concept "underdo" (or a more formal structure representing that idea). Thus, under the pressure created by the failure of using the concept "suppress", it is quite likely that the Imaginer will make a *slippage* — namely, it will take the nearby idea "underdo" and try it on for size. In other words, the Imaginer supposes that "underdoing" the 't's crossbar is the next-best thing to all-out suppression of it. This slippage is of course the key creative breakthrough. It now just needs some fleshing-out, still to be done by the Imaginer.

In order to translate the vaguish "underdo" into a more specific operation, the Imaginer must have information about the *meaning* of "underdo". This is available through its internal definition, which (in a suitable formalism) is given as "reduce the key dimension of". Now the Imaginer must consult the norms attached to "crossbar" in order to find out if a crossbar has a key dimension, and if so, what it is. It finds that for "crossbar", there is only one norm involving size, and that is horizontal length. This allows the vague "underdo" suggestion to be straightforwardly translated into a norm-violation suggestion that says, in effect, "Make a short crossbar". This is what the Imaginer hands to the Drafter. From our discussion above, we know that this can lead to a 't' with a one-quantum crossbar — in other words, a perfectly acceptable and style-loaded 't'. It is, of course, debatable how faithfully this 't' preserves the truly austere spirit of the seed letter 'f', but certainly it is a reasonable attempt.

Note that this example shows how the program can, in some sense, understand and imitate the *spirit* of the seed letter, rather than copying it *literally.* A key role was played here by the *conceptual halo* of the concept "suppress", which yielded the conceptually close, potential-substitute concept "underdo".

Not all feedback in the central loop of creativity originates in the Examiner; it can originate in the Adjudicator too. In contrast to the example just described, it is possible that the Examiner will be satisfied with a given letterform but the Adjudicator will be dissatisfied. In such a case, the letterform is rejected and a message is sent to the Imaginer explaining how the letterform fell short of one or more stylistic criteria (as opposed to letter-category criteria). The Imaginer must then modify its suggestions accordingly.

The interaction of these four agents, wherein ideas are suggested, critiqued, revised, possibly abandoned and regenerated, and so on, meshes exactly with our intuitive sense of what human creativity really is. It seems to us fair to say that this kind of emergent, unpredictable processing constitutes a program's *making its own decisions.* This concurs with the views expressed in [Johnson-Laird, 1988] that free will and creativity are closely related.

7 The Implementation of Emergent Processing

We now descend one level further into the implementation details of Letter Spirit. Each of the four emergent agents described above is actually implemented as a large number of small codelets that are run in simulated parallel. Different types of codelets affect each of the four global memories according to their particular activities. The history of the codelet-based architecture is relevant here.

Early inspiration for the Letter Spirit architecture came from a model of low-level and high-level auditory perception — the Hearsay II speech-understanding project [Erman et al., 1980]. Hearsay II modeled the way in which low-level auditory input signals are converted to meaningful utterances, and pioneered in developing a parallel architecture in which bottom-up and top-down processes coexist and cooperate gracefully. It made use of a Blackboard system that integrated information from a variety of processing levels.

The three main predecessors of Letter Spirit are Jumbo [Hofstadter, 1983], Copycat[3] [Mitchell, 1990], and Tabletop [French, 1992]. The Letter Spirit architecture is closely related to their architectures. Copycat and Tabletop solve analogy problems in microdomains using psychologically plausible methods. Both are models of high-level perception making use of a novel type of architecture that falls somewhere between the symbolic and connectionist paradigms. Top-level behavior emerges from many low-level stochastic computational actions that occur in parallel. Copycat and Tabletop model the way in which a small number of relevant concepts can be awakened from dormancy in order to understand and reason about a situation. Letter Spirit will have a similar architecture, but will focus on a larger-scale creative process that unfolds over an extended period of time. Jumbo, though smaller and earlier than Copycat and Tabletop, pioneered the parallel-terraced-scan architecture (explained below). Its aim was to model the hierarchical perceptual clustering and fluid regrouping-operations that occur in the activity of looking for anagrams. This work is especially relevant to the creative process and bears much kinship to operations in Letter Spirit.

Implementation of emergence and parallelism in all these programs is made possible by the Coderack — a "stochastic waiting room" where computational micro-agents wait before being allowed to do their work. In contrast to a standard operating-systems queue, where processes wait before being deterministically given their slice of CPU time, the Coderack features stochastic selection of actions. To each codelet is attached an *urgency* value — a number that determines its probability of being chosen next. Urgency values are based on how well a codelet's possible effect coheres with structures already built. Over a long period of time, processes (a process consists of many codelets acting in concert) are interleaved in a manner reminiscent of time-sharing. The main difference is that the biased nondeterministic selection method allows processes to run at different speeds. The speeds themselves can be regulated over time to favor more-promising directions over less-promising ones. This emergent behavior is called the *parallel terraced scan.*

The Coderack is the stochastic control center of Letter Spirit. In the memories, acts of gluing quanta together, labeling parts, scanning, matching, adjusting, destroying, and so on are carried out by codelets. The effect of each codelet considered by itself is very slight; however, as many codelets run, their independent effects build upon one another into a coherent collective behavior.

As old codelets run and "die", new codelets are created and placed on the Coderack. Each new codelet is assigned an urgency representing an estimate of the importance of its possible action. Codelets that will most likely enhance the structures in the memories and dovetail with the emerging view will be assigned high urgencies and thus will have a good chance of getting to run. Codelets that seem less promising will

[3]Part of the Copycat work was funded by the NSF in 1984 under grant number DCR 8410409.

be assigned low urgencies and will probably have to wait a long time to run. The biased-random nature of codelet-picking ensures that low-urgency codelets have at least some chance of running, while ensuring that aimless processing is avoided.

Since codelets have very small effects, it is never critical that any particular codelet get selected. What does matter is that certain broad-stroked courses of action as a whole run faster than others. Probabilistic selection based on urgencies allows this to happen. The parallel terraced scan can be thought of as allocating processing power as a function of the degree of promise of a pathway. This idea has much in common with genetic algorithm search and the k-armed bandit problem [Holland et al., 1986].

New codelets are created in two ways: (1) they are posted as follow-ups to codelets that complete their tasks; (2) they are placed on the Coderack automatically as certain stages of processing are reached. The population of the Coderack thus dynamically adjusts to the system's needs.

Though Letter Spirit will be implemented on a serial computer, so that only one codelet will run at a time, the system is in principle a parallel system in which many different activities take place in parallel at different speeds. Viewing Letter Spirit as a parallel-processing system is justified since codelets work in local and independent ways. Hence the "parallel" in "parallel terraced scan".

7.1 Current status of the project

Work on implementation of the Examiner and the Adjudicator sections of Letter Spirit has progressed to the stage of part/role mating. Code allowing the Coderack to select and run codelets has been completed. Of the four global memories, the most work has gone into development of the Visual Focus and parts of the Conceptual Memory. The program has a graphical interface, allowing activity in the memories to be followed on the screen as a run progresses. (Letter Spirit is being developed as an X program so that it will be portable to multiple platforms.)

Other tools being developed at CRCC for the Letter Spirit project include the Gridfont program and the 'a'tabase. The Gridfont program is an X application that allows a human to design, store, and manipulate gridfonts. It greatly facilitates our own observations of the creative process in humans. The 'a'tabase is a collection of gridbound 'a's, all of which can be given ratings by people, representing their strength as 'a's. Its purpose is to facilitate exploration of the vast space of potential gridfont 'a's in order to deepen our understanding of abstract letter-concepts — particularly, what makes for strong and weak category membership.

8 Related Work

Besides DAFFODIL (see Section 1), we know of only one other program that focuses on letter design — the connectionist approach to the Letter Spirit problem by Grebert et al. There are also several AI programs focusing on creativity in general.

8.1 Other models of creativity

As a field, AI has a history of attempting to come up with creative programs. Many such approaches have obvious flaws. As the field has progressed, ideas about creativity have developed accordingly. Below are listed some of the main attempts to model the creative process:

- Meehan's TALE-SPIN program generated stories using a knowledge-based approach [Meehan, 1976]. It made up sensible stories with the aid of a planner, but in the process sacrificed flexibility and interest. Since TALE-SPIN had no idea what made a story entertaining, stories ended up being rather dull and brittle. It seems that too much top-down control can quash creativity.
- AM and Eurisko, two programs developed by Douglas Lenat, used the field of mathematical concepts as their domain [Lenat, 1982, Lenat, 1983a]. Starting with concepts like "set", "list", and "ordered pair", and a few functions such as union, intersect, and compose, both made conjectures (in some sense discoveries) based on examples they generated. One major flaw with both programs was their reliance on human perception to separate the wheat of the generated conjectures and examples from the chaff. Effectively, the user guided the search, making it hard to evaluate the power of the program. Two highly pertinent critiques are [Rowe and Partridge, 1991b] and [Ritchie and Hanna, 1990].
- The BACON family of programs uses a data-driven approach to discovery [Langley et al., 1987]. The idea is to start with a body of data and find a hypothesis that explains it. BACON actually does somewhat *ad hoc* curve fitting by playing "syntactic number games" until it finds an answer. The

problem with these programs is that the idea of what to look for is directly built into the heuristics. Behavior is extremely rigid without any notion of flexibility or ambiguity.

- Harold Cohen's Aaron is a program that draws pictures using a large amount of knowledge about the world [McCorduck, 1991]. Although the results are impressive, it is hard to figure out exactly what is going on in Aaron's processing from the published materials. In general, though, it seems that the program is missing the ability to perceive its work. Using its knowledge about the world and a set of heuristics for drawing, the program creates a picture plan, but that plan is never perceived or judged (either before or after it has been realized).
- Perhaps the model closest in heart to Letter Spirit is the GENESIS system developed by Jon Rowe and Derek Partridge [Rowe and Partridge, 1991a]. The GENESIS program, which plays the card game micro-Eleusis, recognizes the need for flexible representation and control. Their work, like Letter Spirit, stresses the importance of emergence and the interaction of autonomous agents in modeling creativity.

Few AI projects are concerned with the building-up of a large-scale esthetically consistent structure, such as a full alphabet. Most are concerned much more with *discovery* than with creation. Programs like Aaron and TALE-SPIN, which attempt to construct large coherent pieces of work, are exceptions.

8.1 Typeface creation by the GridFont network

Grebert and colleagues, inspired by the Letter Spirit domain and the creation task of Letter Spirit, decided to try a purely connectionist approach and see how well it could do [Grebert et al., 1991a].[4] Their model is based on what they call "generalizing production from examples" using a three-layer feedforward network that is supposed to *learn* how to create gridfonts by being trained on a few examples.

The network was trained on five human-designed gridfonts until it was able to reproduce them all with only a small amount of error. Then it was trained on fourteen letters from another human-designed gridfont, *Hunt Four*. The network's task was to construct the twelve letters of *Hunt Four* that it had never seen.

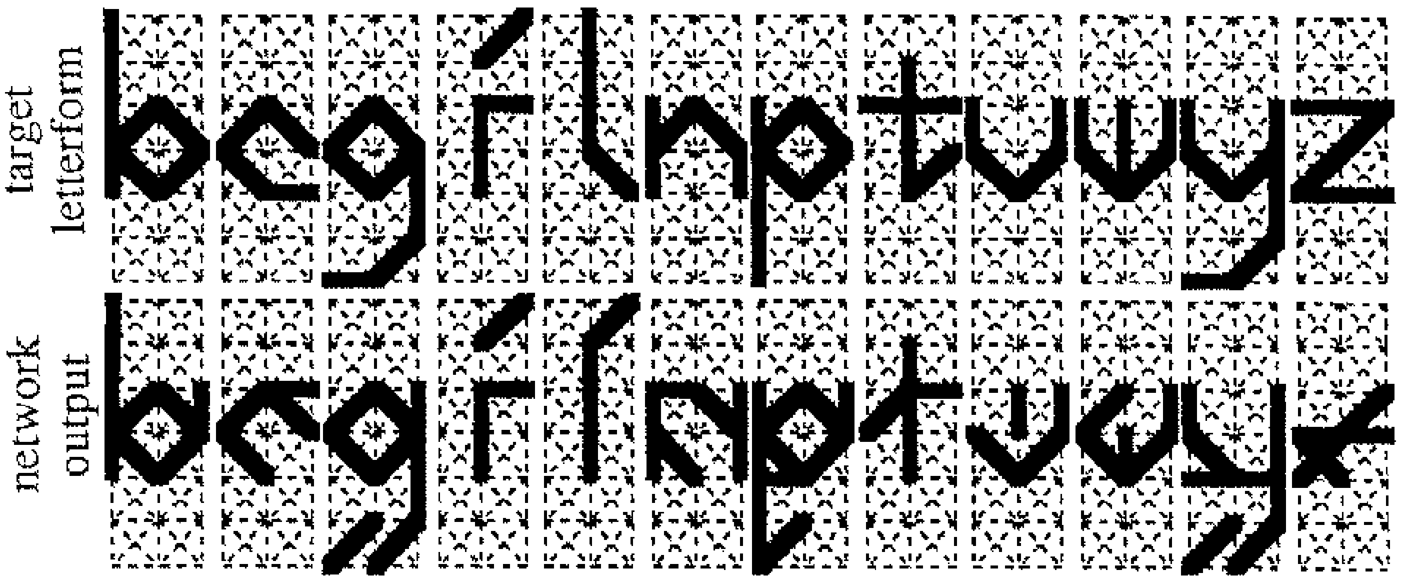

Figure 8: Target letterforms from *hunt four* (top row), and the output of the GridFont network (bottom row). This figure is taken directly from [Grebert 1991a].

The GridFont network's results were not completely terrible, especially considering that it had never been trained on the letters it was meant to create — but they were not good either. The network was trained on so many letters that it had been exposed to virtually all of the key stylistic features of that font.[5] Even with all this information, it still came up with some very bizarre letters, including one that was completely unrecognizable (Figure 8).

[4]See also [Grebert et al., 1991b], [Grebert et al., 1991c], and [Grebert et al., 1992].
[5][Grebert et al., 1991c] selected their training set specifically in order to have this characteristic (see p. 387).

Since the GridFont network does nothing more than pattern association, it has no conceptual foundation for design. It simply creates letters based on statistical similarities that it has "generalized out" of its training set. If concepts were merely statistical beasts, this might work, but such is not the case. Because GridFont has no abstract *idea* about what it means to be a given letter, it does a number of very naive things. For example, it often adds weird extra jogs to its letters. The network has no *explicit* concepts that it can refer to and manipulate during the design process. GridFont has no awareness of the internal structures and boundaries of its "concepts".

The most serious problem with the GridFont approach is the complete lack of temporal interaction between different letters of the emerging alphabet. Simply put, GridFont has no central feedback loop whatsoever. In Letter Spirit, any design decisions made during the creation of one letter will have a large amount of influence on the design of all the other letters in the gridfont (even the "finished" ones). The GridFont network has no such property. All of GridFont network's letters are created in parallel at the same instant — they are all pushed out in one feedforward gush.[6] GridFont's creators seem to have lost sight of the fact that temporality and interaction are central aspects of large-scale acts of creation.

8.2 Prior work on letter recognition

There are various approaches to letter recognition by machines, depending on the type of recognition performed [Mantas, 1986]. The approach most relevant to the Examiner and Adjudicator is known as Optical Character Recognition (OCR). OCR involves converting documents printed in a variety of fonts into machine-readable form. Although there are papers on OCR with impressive titles like "On the Recognition of Printed Characters of Any Font and Size"[7], and although OCR hardware and software are widely available commercially, a survey of the literature shows that the problem of OCR has not by any means been satisfactorily solved. A system that can recognize letters in many typefaces that it has never seen (or been trained on) has yet to be developed. In [Kurzweil, 1990], OCR pioneer Raymond Kurzweil states, "While machines exist today that can accurately recognize many type styles in common usage, no machine can successfully deal with the level of abstraction required by . . . ornamental forms." (The types of letters he is referring to are those shown in our Figure 1.)

Cognitive psychologists studying letter perception emphasize the building-up of higher-level structures from visual stimuli [Palmer, 1977][Palmer, 1978][Treisman and Gelade, 1980], a view that harmonizes with the Letter Spirit approach to letter perception. It is generally agreed that both bottom-up chunkings of syntactic features and top-down contextual pressures are vital in letter recognition and categorization.

We feel it is critical to build and test alternative recognition architectures in order to assess the relative strengths and weaknesses of the Letter Spirit approach to recognition. Two on-going letter-recognition projects at CRCC using architectures radically different from the Examiner — NetRec and DumRec[8] — are described in Appendix B.

9 Conclusion

Letter Spirit is meant to model the central mechanisms of creativity. We feel that if AI and cognitive science are to understand the workings of the human mind, they must focus more attention on the fluidity of concepts. Letter Spirit attempts to model a large-scale creative act using statistically emergent fluid concepts. Its architecture explores new ground in both AI and cognitive science. A complete implementation of Letter Spirit will be of interest both as an experiment in imparting creativity to a machine and as a model of human creativity.

Letter Spirit will test the applicability of the parallel-terraced-scan architecture to large-scale creative tasks. The parallel stochastic processing mechanisms of Letter Spirit fall under the rubric of *emergent computation*, wherein complex high-level behavior emerges as a statistical consequence of many small computational actions. Like Copycat and Tabletop, Letter Spirit will occupy a level of cognitive modeling somewhere between connectionism and symbolic AI — the level we feel is the most useful for the understanding of high-level perception, the fluidity of concepts, and creativity.

[6] In actuality, they are created one at a time since only one letter-node can be clamped during a cycle, but the design of one letterform has no effect on the design of the others — thus, conceptually, a whole font is designed in parallel, without inter-letter interactions.

[7] Actually, [Kahan et al., 1987] tested their model on six fairly standard fonts.

[8] More information on the two projects is available in [McGraw, 1990] and [McGraw, 1992].

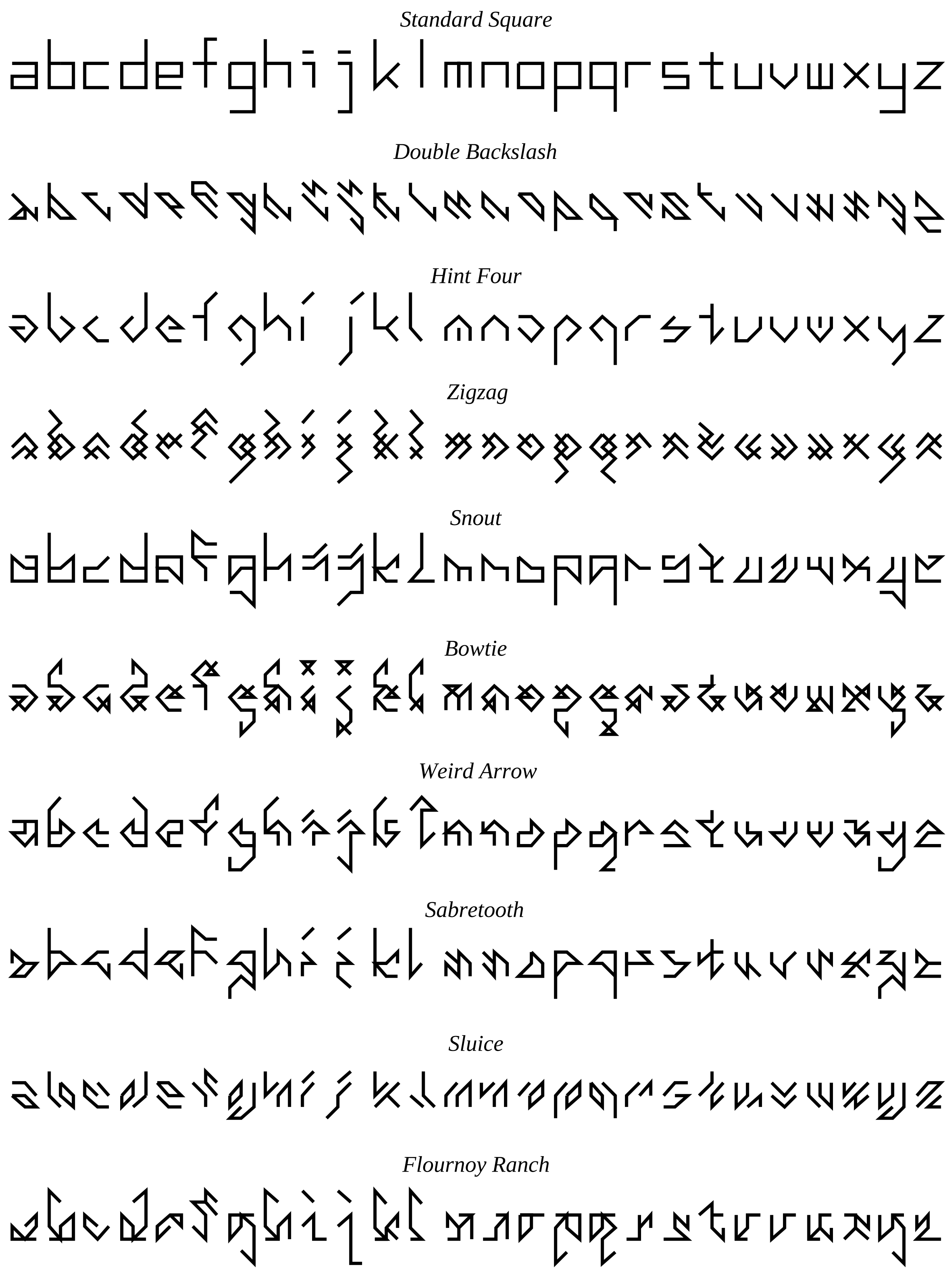

Figure 9: This figure, like Figure 2, can be thought of as illustrating both the vertical problem (categorical sameness) and the horizontal problem (stylistic sameness) for which Letter Spirit is named.

Appendix A: Processing in the Examiner

The Examiner operates roughly as follows, although the following outline may give the impression that processing is more serial and well-ordered than it really is. In reality, processing occurs in a more parallel manner, with various aspects described below often proceeding concurrently.

- Quanta are probabilistically bonded together (by local perceptual micro-agents) with different amounts of "glue". (For example, more glue tends to be deposited at straight junctions than at angles.) The glue-depositing agents execute completely bottom-up syntactic operations.
- When enough glue has been deposited, the glued shape is metaphorically "shaken". This amounts to probabilistically breaking the glued shape into parts at weak joints, resulting in a set of chunks (usually made up of between two and four quanta) called *parts.* Since no concepts have been invoked up to this point, parts are still purely syntactic entities.
- Each part is scanned by multiple syntactic micro-agents that probabilistically attach *syntactic labels* to the part. Labels denote very simple properties of parts like curviness, length, width, location, and so on, and in no way involve the set of categories into which these stimuli will eventually be channeled.
- The presence of a particular label on a given *part* serves as a cue that tends to lightly activate one or more *roles* with which the label is associated (*i.e.,* roles of which the given label is at least somewhat diagnostic). For instance, the labels "left-diagonal", "straight", and "in central zone" would tend to activate one of the "slash" roles of 'x'. It is important to understand that even a label such as "straight" is probabilistic, in the sense that a not totally-straight part *might* get that label, with probability diminishing with its non-straightness.
- The various light activations sum up to a *total* activation-level for each role. Any role with sufficient activation will attempt to "mate" with the specific part whose labels activated it (here is where syntax and semantics meet). Roles compete with each other for a given part. Usually the role that best matches a part will win a fight, but once again, role-matching is probabilistic. A complementary way of describing the process is to say a given part competes for the attention of various roles.
- As roles and parts attempt to couple, a given part may need to be slightly altered in order to be a good mate for a given role. A quantum or two may need to be stolen from one or more neighboring parts to make the part in question more attractive to a suitor. Likewise, small pieces that seem to make a part ugly in the eyes of possible role-mates may need to be given away or simply detached. (Note that only the parts are undergoing regrouping; the underlying quanta defining the letterform remain the same.) The resulting structures composed of quanta are now results of the combined influence of bottom-up and top-down processing. As such, these parts are no longer totally syntactic entities, and we call them "(semantically) adjusted parts."
- Roles compete for parts throughout the letter, adjusting the parts as they go. When this adjustment-and-mating phase is over, there should be a fairly strong match-up between roles and semantically-adjusted parts.
- Each instantiated role will have a few tags attached, stating *how well* the given part instantiates the role. This information will focus on how the part deviates from various norms associated with the role. For example, a 't' whose spine is too tall or bent over at the top, or whose crossbar is too short, too high, or tilted will have tags stating such things.
- Each realized role begins to alert one or more *wholes* (*i.e.,* full letter-concepts such as 'a') for which it provides evidence, in the sense of fitting a particular conceptualization of that letter. Role/whole coupling is analogous to part/role coupling, only it occurs at a higher (more semantic) level. Particular letter-conceptualizations will be activated according to how strongly their component roles are realized in the actual grid letter.
- Different wholes thus become activated to different extents. Each sufficiently-activated whole attempts to perceive the shape on the grid as an instance of itself.
- If there is a clear leading contender among the wholes, it is deemed the winner. If there is a close race between several, the letterform is deemed ambiguous and therefore unacceptable. In a borderline case between clear-winner and close-race, a probabilistic decision is made that chooses between the two courses of action.

Appendix B: Two Rival CRCC Approaches to Letter Recognition

NetRec

We have completed several experiments in gridfont letter recognition using two- and three-layer feed-forward connectionist networks. The NetRec networks are trained (using backpropagation) on a variety of gridfonts and then tested on fonts they have not yet seen. Results show that connectionist networks can perform fairly well on recognition tasks. Backpropagation is especially good at finding statistical regularities among what seem to be disparate category members. Thus if a network is trained on a large number of zany gridfont 'z's, it will discover many of the statistical similarities they share, and exploit this to recognize 'z's it has not yet encountered.

One problem with a connectionist approach is that although a connectionist network can classify letter-forms as strong or weak category members, it cannot judge them in terms of style. It is possible that the stylistic aspects of a proposed 'a' are completely wrong (with respect to other already-designed letters) even though the shape itself is a reasonable member of the category 'a'. The only way around this would be to train another network on the given style and somehow coordinate the two. But since an entire gridfont in the target style does not yet exist, there is no way of talking about what that style is; not enough information exists for style training (as evidenced by Grebert's connectionist GridFont network discussed above). What is needed is *a network that does analogy.* However, analogy has yet to be successfully tackled in the connectionist paradigm.

DumRec

This more traditional AI program compares a novel letter with a large number of stored letters, counting agreements and disagreements of all sorts (which are computed with various levels of complexity), and making a decision based on a weighted comparison of micro-features. DumRec's database of stored letters can include any number of different "training letters" for a given category. DumRec has a graphical interface that makes understanding its results, entering mystery letters, and seeing what a data set looks like fairly straightforward. We are still playing with DumRec in order to determine its strengths and weaknesses. Interestingly, it is fairly good at gridfont letter recognition, most of the time either guessing the correct letter or picking a "reasonable" wrong one. Comparing DumRec's performance to the performance of the Examiner of Letter Spirit will be both informative and interesting.

Bibliography

[Altshuller, 1988] Altshuller, G. (1988). *Creativity as an Exact Science: The Theory of the Solution of Inventive Problems.* Studies in Cybernetics. Gordon and Breach Science Publishers, New York.

[Anderson, 1983] Anderson, J. (1983). *The Architecture of Cognition.* Harvard University Press, Cambridge, Mass.

[Andersson, 1988] Andersson, H. (1988). Man, machine and creativity. *AI & Society: The Journal of Human and Machine Intelligence*, 3:155–158.

[Blesser, 1973] Blesser, B. (1973). Character recognition based on phenomenological attributes. *Visible Language*, 7(3).

[Boden, 1990] Boden, M. A. (1990). *The Creative Mind: Myths and Mechanisms.* Basic Books, New York.

[Bongard, 1970] Bongard, M. (1970). *Pattern Recognition.* Hayden Book Co, Spartan Books, Rochelle Park, NJ.

[Bozinovic and Srihari, 1989] Bozinovic, R. and Srihari, S. (1989). Off-line cursive script word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):68–83.

[Burr, 1988] Burr, D. (1988). Experiments on neural net recognition of spoken and written text. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1162–1168.

[Carbonell, 1990] Carbonell, J., editor (1990). *Machine Learning: Paradigms and Methods.* MIT Press, Cambridge, Mass.

[Cash and Hatamian, 1987] Cash, G. and Hatamian, M. (1987). Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, 39:291–310.

[Chalmers et al., 1992] Chalmers, D., French, R., and Hofstadter, D. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal for Experimental and Theoretical Artificial Intelligence*, 4(3):185–211.

[Chalmers, 1992] Chalmers, D. J. (1992). Chapter 2: Subsymbolic computation and the chinese room. In Dinsmore, J., editor, *The Symbolic and Connectionist Paradigms: Closing the Gap*, pages 25–49. Lawrence Erlbaum.

[Chamberlain, 1984] Chamberlain, W. (1984). *The Policeman's Beard is Half-Constructed: Computer Prose and Poetry.* Warner Software/Books, New York.

[Collins and Loftus, 1975] Collins, A. and Loftus, E. (1975). A spreading activation theory of semantic memory. *Psychological Review*, 82:407–428.

[Crozier and Chapman, 1984] Crozier, W. and Chapman, A., editors (1984). *Cognitive Processes in the Perception of Art.* Advances in Psychology (19). North-Holland, Amsterdam.

[Dehn, 1981] Dehn, N. (1981). Story generation after tale-spin. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence.*

[Epstein, 1988] Epstein, S. (1988). Learning and discovery: One system's search for mathematical knowledge. *Computational Intelligence*, 4(1):42–53.

[Erman et al., 1980] Erman, L., Hayes-Roth, F., Lesser, V., and Raj Reddy, D. (1980). The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2):213–253.

[Fauconnier, 1985] Fauconnier, G. (1985). *Mental Spaces: Aspects of Meaning Construction in Natural Language.* MIT Press, Cambridge, Mass.

[Finke, 1990] Finke, R. (1990). *Creative Imagery.* Lawrence Erlbaumn Associates, Hillsdale, NJ.

[Frank and Gilovich, 1989] Frank, M. and Gilovich, G. (1989). Effect of memory perspective on retrospective causal attributions. *Journal of Personality and Social Psychology*, 57:399–403.

[French, 1990] French, R. (1990). Subcognition and the limits of the turing test. *Mind*, 99(393):53–65.

[French, 1992] French, R. (1992). *Tabletop: An emergent stochastic computer model of analogy-making.* PhD thesis, University of Michigan, Ann Arbor, Michigan.

[Frey and Slate, 1990] Frey, P. and Slate, D. (1990). Letter recognition using holland-style adaptive classifiers. Unpublished report.

[Fromkin, 1980] Fromkin, V., editor (1980). *Errors in Linguistic Performance: Slips of the Tongue, Ear, Pen, and Hand.* Academic Press, New York.

[Fukushima, 1989] Fukushima, K. (1989). Analysis of the process of visual pattern recognition by the neocognitron. *Neural Networks*, 2:413–420.

[Fukushima, 1992] Fukushima, K. (1992). Character recognition with neural networks. *Neurocomputing*, 4:221–233.

[Gentner and Forbus, 1990] Gentner, D. and Forbus, K. (1990). A note on "creativity and learning in a case-based explainer". *Artificial Intelligence*, 44:373–375.

[Grebert et al., 1991a] Grebert, I., Stork, D., Keesing, R., and Mims, S. (1991a). Connectionist generalization for production: An example from gridfont. In *Proceedings of the 1991 International Joint Conference on Neural Networks (IJCNN).*

[Grebert et al., 1991b] Grebert, I., Stork, D., Keesing, R., and Mims, S. (1991b). Connectionist generalization for production: An example from gridfont. Technical Report CRC-TR-91-20, Ricoh California Research Center, Menlo Park, CA.

[Grebert et al., 1991c] Grebert, I., Stork, D., Keesing, R., and Mims, S. (1991c). Network generalization for production: Learning and producing styled letterforms. In *Proceedings of the Neural Information Processing Systems Conference (NIPS) 1991.*

[Grebert et al., 1992] Grebert, I., Stork, D., Keesing, R., and Mims, S. (1992). Connectionist generalization for production: An example from gridfont. *Neural Networks*, 5.

[Guyon et al., 1989] Guyon, I., Poujaud, I., Personnaz, L., and Dreyfus, G. (1989). Comparing different neural network architectures for classifying handwritten digits. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-89)*, pages II, 127–132, Washington, D.C.

[Hammond, 1989] Hammond, K. (1989). Chapter 6: Chef. In Riesbeck, C. and Schank, R., editors, *Inside Case-Based Reasoning*, pages 165–211. Lawrence Erlbaum and Associates, Hillsdale, New Jersey.

[Haugeland, 1987] Haugeland, J., editor (1987). *Mind Design: Philosophy, Psychology, Artificial Intelligence.* MIT Press, Cambridge, Mass.

[Hinton et al., 1992] Hinton, G., Williams, K., and Revow, M. (1992). Adaptive elastic models for hand-printed character recognition. Presented at Cogsci 1991, and posted to Neuroprose.

[Hofstadter, 1979] Hofstadter, D. (1979). *Gödel, Escher, Bach: an Eternal Golden Braid.* Basic Books, New York.

[Hofstadter, 1983] Hofstadter, D. (1983). The architecture of jumbo. In *Proceedings of the International Machine Learning Workshop*, Monticello, IL.

[Hofstadter, 1985a] Hofstadter, D. (1985a). *Metamagical Themas*, chapter 26: Waking up from the Boolean dream: Subcognition as computation, pages 631–665. Basic Books, New York.

[Hofstadter, 1985b] Hofstadter, D. (1985b). *Metamagical Themas*, chapter 24: Analogies and roles in human and machine thinking. Basic Books, New York.

[Hofstadter, 1985c] Hofstadter, D. (1985c). *Metamagical Themas*, chapter 10: Parquet deformations: A subtle, intricate art form. Basic Books, New York.

[Hofstadter, 1985d] Hofstadter, D. (1985d). *Metamagical Themas*, chapter 12: Variations on a theme as the crux of creativity, pages 232–259. Basic Books, New York.

[Hofstadter, 1985e] Hofstadter, D. (1985e). *Metamagical Themas*, chapter 13: Metafont, metamathematics, and metaphysics: Comments on Donald Knuth's article "The concept of a meta-font", pages 232–259. Basic Books, New York.

[Hofstadter, 1985f] Hofstadter, D. (1985f). *Metamagical Themas*, chapter 23: On the seeming paradox of mechanizing creativity, pages 547–603. Basic Books, New York.

[Hofstadter, 1985g] Hofstadter, D. (1985g). *Metamagical Themas: Questing for the Essence of Mind and Pattern.* Basic Books, New York. See especially Chapters 10, 12, 13, 23, 24, and 26.

[Hofstadter, 1987a] Hofstadter, D. (1987a). Fluid analogies and human creativity. Technical Report 16, Center for Research on Concepts and Cognition, Indiana University, Bloomington, IN.

[Hofstadter, 1987b] Hofstadter, D. (1987b). Introduction to the letter spirit project and to the idea of "gridfonts". Technical Report 17, Center for Research on Concepts and Cognition, 510 N. Fess, Bloomington, IN, 47408.

[Hofstadter and McGraw, 1993] Hofstadter, D. and McGraw, G. (1993). Letter spirit: An emergent model of the perception and creation of alphabetic style. Technical Report 68, Center for Research on Concepts and Cognition, 510 N. Fess, Bloomington, IN 47405.

[Hofstadter and Mitchell, 1988] Hofstadter, D. and Mitchell, M. (1988). Concepts, analogies, and creativity. In Goebel, R., editor, *CSCSI-88: Proceedings of the Seventh Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 94–101. Ablex, Edmonton, Alberta, Canada.

[Hofstadter and Mitchell, 1991] Hofstadter, D. and Mitchell, M. (1991). The copycat project: A model of mental fluidity and analogy-making. Technical Report 58, Center for Research on Concepts and Cognition, Indiana University, Bloomington, IN. To appear in Barnden, J. and Holyoak, K. *Advances in Connectionist and Neural Computation Theory. Volume 2: Analogical Connections.*

[Hofstadter and Moser, 1989] Hofstadter, D. and Moser, D. (1989). To err is human; to study error-making is cognitive science. *Michigan Quarterly Review*, XXVIII(2):185–215.

[Hofstadter, 1987c] Hofstadter, D. R. (1987c). *Ambigrammi: Un Microcosmo Ideale per lo Studio della Creatività.* hopefulmonster, Firenze. Published in Italian.

[Hofstadter, 1992] Hofstadter, D. R. (1992). Some ideas and questions about creativity for the "mechanisms of creativity" workshop. (revised edition for the second workshop). Center for Research on Concepts and Cognition, 510 N. Fess, Bloomington, IN, 47408.

[Holland, 1975] Holland, J. (1975). *Adaption in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI. Reissued by MIT Press, 1992.

[Holland et al., 1986] Holland, J., Holyoak, K., Nisbett, R., and Thagard, P. (1986). *Induction.* MIT Press, Cambridge, Mass.

[Holland, 1986] Holland, J. H. (1986). Chapter 20: Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In Michalski, R., Carbonell, J., and Mitchell, T., editors, *Machine Learning: An Artificial Intelligence Approach*, volume 2, pages 593–623. Morgan Kauffmann Publishers.

[Johnson-Laird, 1987] Johnson-Laird, P. (1987). Reasoning, imagining, and creating. *Bulletin of the British Psychological Society*, 40:121–129.

[Johnson-Laird, 1988] Johnson-Laird, P. (1988). Chapter 8: Freedom and constraint in creativity. In Sternberg, R., editor, The Nature of Creativity, pages 202–219. Cambridge University Press, New York.

[Kahan et al., 1987] Kahan, S., Pavlidis, T., and Baird, H. (1987). On the recognition of printed characters of any font and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):274–288.

[Kahneman and Miller, 1986] Kahneman, D. and Miller, D. (1986). Norm theory: comparing reality to its alternatives. *Psychological Review*, 93:136–153.

[Kim, 1991] Kim, S. (1991). Notes from the first "mechanisms of creativity" workshop. Held in March, 1991 at the Center for Research on Concepts and Cognition, 510 N. Fess, Bloomington, IN, 47408.

[Kim, 1992] Kim, S. (1992). Notes from the second "mechanisms of creativity" workshop. Held in March, 1992 at the Center for Research on Concepts and Cognition, 510 N. Fess, Bloomington, IN, 47408.

[Kim, 1990] Kim, S. H. (1990). *Essence of Creativity: A Guide to Tackling Difficult Problems.* Oxford University Press, New York.

[Knuth, 1982] Knuth, D. (1982). The concept of a meta-font. *Visible Language*, XVI(1):3–27.

[Knuth, 1986] Knuth, D. (1986). *The METAFONT Book.* Addison-Wesley, Reading, MA.

[Kolodner and Penberthy, 1990] Kolodner and Penberthy (1990). A case-based approach to creativity in problem-solving. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pages 978–985. Lawrence Earlbaum Associates.

[Kurzweil, 1990] Kurzweil, R. (1990). *The Age of Intelligent Machines.* MIT Press, Cambridge. MA.

[Lakoff, 1987] Lakoff, G. (1987). *Women, Fire, and Dangerous Things.* University of Chicago Press, Chicago, IL.

[Langley et al., 1987] Langley, P., Simon, H., Bradshaw, G., and Zytkow, J. (1987). *Scientific Discovery: Computational Explorations of the Creative Process.* MIT Press, Cambridge, Mass.

[Langton, 1989] Langton, C., editor (1989). *Artificial Life*, volume 6 of *Santa Fe Institute Studies in the Sciences of Complexity.* Addison-Wesley.

[Langton et al., 1992] Langton, C., Taylor, C., Farmer, J. D., and Rasmussen, S., editors (1992). *Artificial Life*, volume 10 of *Santa Fe Institute Studies in the Sciences of Complexity.* Addison-Wesley.

[Lee and Oldham, 1990] Lee, M. and Oldham, W. (1990). Font recognition by a neural network. *International Journal of Man-Machine Studies*, 33:41–61.

[Lenat, 1982] Lenat, D. (1982). Am: Discovery in mathematics as heuristic search. In Davis, R. and Lenat, D., editors, *Knowledge-Based Systems in Artificial Intelligence*, pages 1–225. McGraw-Hill, New York.

[Lenat, 1983a] Lenat, D. (1983a). Eurisko: A program that learns new heuristics and domain concepts. *Artificial Intelligence*, 21(1,2):61–98.

[Lenat, 1983b] Lenat, D. (1983b). Theory formation by heuristic search. *Artificial Intelligence*, 21(1,2):31–59.

[Leyton, 1988] Leyton, M. (1988). A process-grammar for shape. *Artificial Intelligence*, 2(34):213–247.

[Mantas, 1986] Mantas, J. (1986). An overview of character recognition methodologies. *Pattern Recognition*, 19(6):425–430.

[Mantas, 1987] Mantas, J. (1987). Methodologies in pattern recognition and image analysis — a brief survey. *Pattern Recognition*, 20(1):1–6.

[McClelland and Rumelhart, 1981] McClelland, J. and Rumelhart, D. (1981). An interactive-activation model of context effects in letter perception: Part 1, an account of basic findings. *Psychological Review*, 88(5):375–407.

[McCorduck, 1991] McCorduck, P. (1991). *Aaron's Code: Meta-art, Artificial Intellgence and the Work of Harold Cohen.* Freeman, New York.

[McDermott, 1981] McDermott, D. (1981). Chapter 5: Artificial intelligence meets natural stupidity. In Haugeland, J., editor, *Mind Design.* MIT Press, Cambridge, Mass.

[McGill, 1989] McGill, A. (1989). Context effects in judgements of causation. *Journal of Personality and Social Psychology*, 57:189–200.

[McGraw, 1990] McGraw, G. (1990). Dumrec lives — birth of a project with multiple programalities. Center for Research on Concepts and Cognition, Indiana University, 501 North Fess, Bloomington, IN 47405 (internal document).

[McGraw, 1992] McGraw, G. (1992). Letter spirit: Recognition and creation of letterforms based on fluid concepts. Technical Report 61, Center for Research on Concepts and Cognition, 510 North Fess, Bloomington, IN, 47408.

[McGraw and Hofstadter, 1993] McGraw, G. and Hofstadter, D. (1993). Perception and creation of alphabetic style. Technical report, AAAI. Selected papers from the 1993 AAAI Symposium on Creativity and Artificial Intelligence. (in press).

[Meehan, 1976] Meehan, J. (1976). *The Metanovel: Writing Stories by Computer.* PhD thesis, Yale University, New Haven, Connecticut.

[Meehan, 1978] Meehan, J. (1978). Tale-spin, an interactive program that writes stories. *Natural Language*, 5:91–98.

[Meredith, 1986] Meredith, M. (1986). *Seek-Whence: A Model of Pattern Perception.* PhD thesis, Indiana University, Bloomington, IN. Also available as Technical Report No. 214, Computer Science Department, Indiana University.

[Meredith, 1991] Meredith, M. (1991). Data modeling: A process for pattern induction. *Journal for Experimental and Theoretical Artificial Intelligence*, 3:43–68.

[Minsky, 1985] Minsky, M. (1985). *The Society of Mind.* Simon and Schuster, New York.

[Mitchell, 1990] Mitchell, M. (1990). *Copycat: A Computer Model of High-level Perception and Conceptual Slippage in Analogy Making.* PhD thesis, University of Michigan, Ann Arbor, Michigan.

[Mitchell, 1993] Mitchell, M. (1993). *Analogy-making as Perception.* MIT Press/Bradford Books (in press), Cambridge, MA.

[Mori et al., 1984] Mori, S., Yamamoto, K., and Yasuda, M. (1984). Research on machine recognition of handprinted characters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):386–405.

[Mueller, 1987] Mueller, E. (1987). *Daydreaming and Computation: A Computer Model of Everyday Creativity, Learning, and Emotions in the Human Stream of Thought.* PhD thesis, University of California at Los Angeles, Los Angeles, California.

[Nanard et al., 1986] Nanard, M., Nanard, J., Gandara, M., and Porte, N. (1986). Declarative approach for font design by incremental learning. Technical report, Centre de Recherche en Informatique de Montpellier, Montpellier, France.

[Newell, 1990] Newell, A. (1990). *Unified Theories of Cognition.* Harvard University Press, Cambridge, Mass.

[Norman, 1981] Norman, D. (1981). Categorization of action slips. *Psychological Review*, 88(1):1–15.

[Palmer, 1977] Palmer, S. (1977). Hierarchical structure in perceptual representation. *Cognitive Psychology*, 9:441–474.

[Palmer, 1978] Palmer, S. (1978). Structural aspects of visual similarity. *Memory & Cognition*, 6(2):91–97.

[Podgorny and Garner, 1979] Podgorny, P. and Garner, W. (1979). Reaction time as a measure of inter- and intra-object visual similarity: Letters of the alphabet. *Perception & Psychophysics*, 26(1):37–52.

[Quillian, 1968] Quillian, M. (1968). Semantic memory. In Minsky, M., editor, *Semantic Information Processing.* MIT Press, Cambridge, Mass.

[Regier, 1991] Regier, T. (1991). Learning spatial concepts using a partially-structured connectionist architecture. Technical Report TR-91-050, International Computer Science Institute, Berkeley, California.

[Rieger, 1975] Rieger, C. (1975). Conceptual memory and inference. In Schank, R., editor, *Conceptual Information Processing.* North-Holland, Amsterdam.

[Ritchie and Hanna, 1990] Ritchie, G. and Hanna, F. (1990). Am: A case study in ai methodology. In Partridge and Wilks, editors, *The Foundations of AI: A sourcebook.* Cambridge University Press, New York.

[Rosch, 1976] Rosch, E. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8:382–439.

[Rosch and Lloyd, 1978] Rosch, E. and Lloyd, B. (1978). *Cognition and Categorization.* Lawrence Erlbaum Associates, Hillsdale, NJ.

[Rowe, 1991] Rowe, J. (1991). *Emergent Creativity: A Computational Study.* PhD thesis, Computer Science Department, University of Exeter, UK.

[Rowe and Partridge, 1991a] Rowe, J. and Partridge, D. (1991a). Creativity: A survey of ai approaches. Technical Report R 214, Department of Computer Science, University of Exeter, UK.

[Rowe and Partridge, 1991b] Rowe, J. and Partridge, D. (1991b). Creativity: Learning flexible representations through emergent memory. Technical Report R 215, Department of Computer Science, University of Exeter, UK.

[Rowe and Partridge, 1991c] Rowe, J. and Partridge, D. (1991c). Creativity: Two theories modelled and compared. Technical Report R 213, Department of Computer Science, University of Exeter, UK.

[Schank and Riesbeck, 1981] Schank and Riesbeck, C. (1981). *Inside Computer Understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ.

[Schank, 1983] Schank, R. (1983). *Dynamic Memory.* Cambridge University Press, Cambridge, England.

[Schank, 1986] Schank, R. (1986). *Explanation Patterns: Understanding Mechanically and Creatively.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[Schank and Childers, 1988] Schank, R. and Childers, P. (1988). *The Creative Attitude: Learning to Ask and Answer the Right Questions.* MacMillan, New York.

[Schank and Leake, 1989] Schank, R. and Leake, D. (1989). Creativity and learning in a case-based explainer. *Artificial Intelligence*, 40(1-3):353–385. Also in Carbonelle, J. (ed.), *Machine Learning: Paradigms and Methods*, MIT Press, Cambridge, MA, 1990.

[Steedman, 1984] Steedman, M. J. (1984). A generative grammar for jazz chord sequences. *Music Perception*, 2:52–77.

[Sternberg, 1988] Sternberg, R. J., editor (1988). *The Nature of Creativity: Contemporary Psychological Perspectives.* Cambridge University Perss, Cambridge.

[Treisman and Gelade, 1980] Treisman, A. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(12):97–136.

[Turing, 1950] Turing, A. (1950). Computing machinery and intelligence. *Mind*, 52(236):433–460.

[Tversky, 1977] Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:372–352.

[Tversky and Hutchinson, 1986] Tversky, A. and Hutchinson, J. (1986). Nearest neighbor analysis of psychological spaces. *Psychological Review*, 93(1):3–22.

[Vernon, 1970] Vernon, P. (1970). *Creativity.* Penguin Press.

[Vickery, 1983] Vickery, R. (1983). *Sharing Architecture.* University of Virginia Press, Charlottesville, VA.

[Wallace and Gruber, 1989] Wallace, D. B. and Gruber, H. E. (1989). *Creative People at Work.* Oxford University Press, New York.

[Wang and Suen, 1984] Wang, Q. and Suen, C. (1984). Analysis and design of a decision tree based on entropy reduction and its applicatoin to large character set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):406–417.

[Yazdani, 1984] Yazdani, M. (1984). Creativity in men and machines. In Torrance, S., editor, *The Mind and the Machine: Philosophical Aspects of Artificial Intelligence*, pages 177–181. Ellis Horwood Limited, Chichester.

[Zuckerman, 1992] Zuckerman, J. (1992). *The Scheme/X/Motif Manual.* Motorola, Inc., first edition.