

Drifting Scripts

Visual Substitution, Glyph Interpolation, and the Receptive Acquisition of New Orthographies

Flyxion

Independent Researcher

October 2026

Abstract

A writing system pairs a conventional assignment of marks to linguistic units with a perceptual structure that the assignment leaves untouched, and the second of these governs how easily the first can be changed. This essay develops that distinction through a family of constructions: visual substitution ciphers, in which each letter is replaced by one that resembles it; fonts that separate the stored character from the displayed glyph; and books whose script drifts gradually over the course of a single volume so that a reader may acquire the destination forms receptively while reading content in the ordinary way. The treatment is organised by the status of its claims. Counting results fix the size of the design space of reciprocal alphabets, about 5.3×10^{14} out of 4.0×10^{26} permutations of the Latin capitals. A computed resemblance experiment across four typefaces shows that the best global pairing is strongly font dependent, that only six of thirteen pairs survive in all four fonts, and that a pixel-overlap measure disagrees with intuition about several pairs the design literature would call close. Two geometric results concern intermediate forms: any continuous exchange of two letters by linear interpolation passes through a state in which both letters have the same glyph, and this collapse is removed by a detour through a third direction in shape space, of which Arabic letter dotting is a historical analogue. A scheduling analysis shows that the exposure a reader receives is governed by letter frequency, so that the rarest letter of a model book of eighty thousand words is seen about 278 times in total, and that starting rare letters first attains a lower bound on the length of a drift. The essay closes with formal interpretations in category theory, type theory and sheaf theory, each stated with explicit definitions, including a precise sense in which a drifting alphabet defines a compatible family of local decodings that does not glue to a global key. Successful acquisition is treated throughout as a hypothesis, and an empirical programme is specified to test it.

Keywords: writing systems, arbitrariness of the sign, substitution ciphers, variable fonts, perceptual learning, receptive learning, sheaf theory, edition provenance.

Contents

1	Introduction	4
1.1	Status of the claims	5
1.2	Plan	5
2	Visual Substitution Alphabets	6
2.1	Definitions and a worked example	6
2.2	Counting	7
2.3	Conventions that support reading	8
2.4	Cycles longer than two	8
3	Resemblance as a Quantity to Be Optimised	9
3.1	Formulation	9
3.2	A computable proxy and its results	9
3.3	What the pairings show	11
4	Arbitrariness of Assignment and Constraints on Acquisition	11
4.1	Two relations that should not be confused	11
4.2	What is known about the acquisition of letter forms	12
4.3	Levels of knowing a sign	13
4.4	Precedents	13
5	The Edition as a Composite of Independent Components	14
5.1	Four adjustable components	14
5.2	Display equivalence and the storage decision	15
5.3	Font technology	15
5.4	Contextual display	16
6	Drift: Intermediate Forms and Schedules	16
6.1	Margins and routes	16
6.2	Collision under linear exchange	17
6.3	The detour	19
6.4	Longer cycles	21
6.5	Cycles of presentation	21
6.6	Exposure and the novelty load	22
6.7	Interference and the choice of endpoint	24
7	Scripts in Contact: Latin, Arabic and Standard Galactic	24

7.1	Terminology	24
7.2	What Arabic adds	26
7.3	When substitution and rendering commute	27
7.4	Interpolation between Latin and Standard Galactic	27
8	Editable Source and the Provenance of Editions	28
8.1	Why the source is released	28
8.2	Components and their identification	28
8.3	Licensing and attribution	29
8.4	Prior work and provenance in the project’s tools	29
8.5	Accessibility	31
9	An Empirical Programme	31
9.1	Hypotheses	31
9.2	Conditions	32
9.3	Measures	32
9.4	Design considerations	33
9.5	The question to be answered	33
10	Formal Interpretations: Transformation, Typing, and Local Readability	34
10.1	Category-theoretic interpretation: transformation and composition	34
10.2	Type-theoretic interpretation: typing the publishing system	36
10.3	Sheaf-theoretic interpretation: local readability and global convention	37
10.4	What the three perspectives jointly provide	41
11	Conclusion	42
A	Counting Derivations	43
B	Optimal Pairings by Typeface	44
C	Code	44
D	A Schema for Edition Manifests	48

1 Introduction

A printed page makes two commitments at once. It commits to a linguistic content, the words and sentences that a reader is meant to recover, and it commits to a convention for turning that content into marks, the alphabet, the orthography and the particular letter shapes of a typeface. Ordinary reading hides the second commitment because the convention is learned so early and rehearsed so heavily that it ceases to be visible as a convention at all. A reader who meets a page set in an unfamiliar script is therefore in an unusual position: the content is still there, but the instrument that delivers it has to be learned first, and the usual pedagogical response is to teach the instrument separately, through an alphabet chart, a primer, or a course, before the reading proper begins.

The constructions studied in this essay start from a different question. Suppose the instrument could be changed while the content continued to be delivered, so that a reader never stopped to study a table of signs but met the new forms inside running text, a few at a time, with enough surrounding material in the familiar form to carry the argument. A book arranged in this way would drift. Its first pages would use forms the reader already knows, its later pages would introduce small departures, and its last pages would be set in a script that would have been hard to read on the first page. The book would supply part of the preparation needed to read its own ending. Whether it does so, and under what conditions, is an empirical matter, and the essay treats it as one. What can be done in advance, and what occupies most of the following pages, is to specify the objects involved precisely enough that the empirical question can be asked without ambiguity: what a visual substitution is, how resemblance between letters can be measured and optimised, how a display can be separated from a stored text, how the intermediate forms of a transition behave, how a schedule of changes interacts with the frequency of letters in running text, and what mathematical structures describe the resulting editions.

The starting point is a visual substitution cipher, a one-to-one replacement of letters in which each letter is exchanged for one that looks like it. The purpose is the reverse of the usual purpose of a cipher. The substitution is intended to be easy to see through, so that a reader gradually learns to read it without consciously decoding each letter, with spaces, punctuation and capitalisation preserved so that word shape and sentence context can support recognition where an individual replacement is ambiguous. Such an alphabet can be rendered in two quite different ways. A font can display each stored character exactly as the ordinary letter it represents, so that the stored text is substituted while the page looks normal, or it can display a shape somewhere between the stored character and its partner, so that the substitution is perceptible but the familiar structure remains available to the reader. A font of the first kind already exists in the project's font collection: the LingoJam Cipher font, which displays characters of a particular substitution cipher as ordinary Standard Galactic letters, so that the stored characters carry one representation and the font supplies the visual interpretation. The present essay generalises that separation and asks what happens when the font is not fixed but changes from one part of a book to another.

Three further motivations shape the treatment. The first is theoretical. The arbitrariness of the sign, in the sense that no necessity ties a linguistic unit to its particular written shape, is a familiar thesis, and a collection of working ciphers, typefaces and drifting editions is a way of exhibiting it in a form that can be read rather than merely asserted. Yet arbitrariness of assignment does not imply equal ease of acquisition, and the visual substitution alphabet makes the gap between the two concrete: the exchange of B and P can be learned through a perceptual bridge that the exchange of C and K lacks. The second motivation is practical. The essays in this project are released with their source in the public domain, so that readers can propose corrections and can also print any text in whatever script they choose. This turns each text into a family of editions, and the editions are at once publications and research instruments. The third is methodological. Because the editions are generated from separately identifiable components, the conditions of a transition can be varied one at a time, and the observations of how readers cope can be attached to a precise description of what they read.

1.1 Status of the claims

The essay combines results of different kinds, and the difference matters because the temptation in a subject of this sort is to let a plausible mechanism stand in for a measured outcome. Each major claim is therefore tagged below by what supports it. Statements of the first kind are proved from definitions given in the text. Statements of the second kind are computed, which means that they report the output of a program on stated inputs, the program being reproduced in the appendices, and they are exactly as strong as the inputs and the model behind them. Statements of the third kind are hypotheses about readers, and the essay does not claim to have established any of them.

1.2 Plan

Section 2 defines visual substitution alphabets and counts them. Section 3 treats resemblance as a weight to be optimised and reports a computed comparison across typefaces. Section 4 sets the arbitrariness of the sign against the constraints on acquisition. Section 5 describes an edition as a composite of independently variable components and discusses the font technology that realises it. Section 6 analyses intermediate forms and schedules of change. Section 7 considers contact between Latin, Arabic and Standard Galactic forms, with attention to terminology. Section 8 discusses the publication of editable source and the provenance of editions. Section 9 specifies an empirical programme. Section 10 gives formal interpretations in category theory, type theory and sheaf theory. Section 11 concludes, and the appendices contain the derivations, the full computational results and the code.

Section	Claim	Status
2	Size of the space of reciprocal alphabets; round-trip recovery	Proved
3	Optimal pairings differ across typefaces; overlap measure disagrees with intuition for several pairs	Computed
6	Linear exchange of two letters passes through a glyph collision; a detour removes it; three-cycles avoid it in general position	Proved
6	Exposure conservation and a lower bound on the length of a drift; rare-first ordering attains it in the cases tried	Proved (bound), computed (attainment)
7	Context-dependent display breaks the commuting of substitution and rendering unless joining classes are preserved	Proved
10	Gluing criterion for local keys; the true local keys of a swap-ending drift are compatible but have no global key for any cover	Proved
9	Drifting presentation supports receptive acquisition and transfer	Hypothesis

Table 1: Claims of the essay by kind of support.

2 Visual Substitution Alphabets

2.1 Definitions and a worked example

Let $\Sigma = \{A, \dots, Z\}$ be the Latin capitals and Σ^* the set of finite strings over Σ . A *substitution* is a bijection $\pi : \Sigma \rightarrow \Sigma$, extended letter by letter to a map on strings, $\pi(x_1x_2 \cdots x_n) = \pi(x_1)\pi(x_2) \cdots \pi(x_n)$. Characters outside Σ , namely spaces and punctuation, are left fixed, and case is preserved by applying π to the upper-case counterpart of a lower-case letter and returning the result in lower case. The substitution is *reciprocal* when $\pi \circ \pi = \text{id}$, which is the same as saying that π is an involution: it exchanges letters in pairs and leaves the others where they are. A reciprocal substitution is its own decoder, and applying it twice restores the original text.

A *visual* substitution is one chosen so that $\pi(a)$ resembles a for most letters a . The notion of resemblance is left open for the moment and is taken up in Section 3. Consider the exchanges $B \leftrightarrow P$ and $U \leftrightarrow V$ and nothing else. The word PUBLIC becomes BVPLIC: the P and B exchange, the U becomes V, and the remaining letters are untouched. BLUE becomes PLVE and LOVE becomes LOUE, and in each case a second application restores the original, so that PUBLIC passes through BVPLIC and returns. These are small examples, but they are enough to show the features that matter. The result is close to the original in shape, a sentence supplies

cues that disambiguate an individual replacement, and a reader who meets the substituted spelling a few times may come to experience it as an unfamiliar typeface or an alternative orthography rather than as a code to be solved. The same examples show a hazard of hand application: it is easy to leave the U of PUBLIC in place while exchanging the letters around it, and a substitution of even six letters is error-prone when applied by eye. This is a modest argument for generating substituted text mechanically from a stored source, a point that returns in Section 5.

2.2 Counting

The number of substitutions of the 26 capitals is $26! \approx 4.03 \times 10^{26}$. The number of reciprocal substitutions is far smaller. If $I(n)$ denotes the number of involutions on an n -element set, then letter n either is fixed, leaving $I(n - 1)$ involutions of the rest, or is exchanged with one of the other $n - 1$ letters, leaving $I(n - 2)$ involutions of the remaining $n - 2$, so that

$$I(n) = I(n - 1) + (n - 1) I(n - 2), \quad I(0) = I(1) = 1.$$

A reciprocal substitution that changes exactly $2k$ letters is determined by choosing which $2k$ letters move and then a perfect matching on them, and the number of perfect matchings on $2k$ points is $(2k - 1)!! = (2k - 1)(2k - 3) \cdots 3 \cdot 1$. Hence the number of reciprocal substitutions that exchange exactly k pairs is $\binom{26}{2k} (2k - 1)!!$, and summing over k gives $I(26)$. Running these formulas gives the figures in Table 2; the sum over k reproduces $I(26) = 532,985,208,200,576$ exactly, which serves as a check on the arithmetic. The count of reciprocal substitutions that change every letter, $25!! = 7,905,853,580,625$, is the number of perfect matchings on the full alphabet, and the count of all substitutions that change every letter is the derangement number $D(26) = 148,362,637,348,470,135,821,287,825$, which satisfies $D(n) = (n - 1)(D(n - 1) + D(n - 2))$.

Pairs exchanged k	Letters changed $2k$	Reciprocal alphabets $\binom{26}{2k} (2k - 1)!!$
1	2	325
2	4	44 850
3	6	3 453 450
5	10	5 019 589 575
8	16	10 767 019 638 375
10	20	150 738 274 937 250
13	26	7 905 853 580 625
All reciprocal alphabets, $I(26)$		532 985 208 200 576
All permutations without fixed points, $D(26)$		148 362 637 348 470 135 821 287 825
All permutations, $26!$		403 291 461 126 605 635 584 000 000

Table 2: Reciprocal substitutions of the 26 Latin capitals by the number of exchanged pairs, with the totals for all reciprocal substitutions, all substitutions without fixed points, and all substitutions.

The ratio of all permutations to reciprocal ones is about 7.6×10^{11} , so restricting to involutions discards all but a vanishing fraction of the space. Two further observations

are worth recording. First, the space is enormous in the sense relevant to cryptography and irrelevant to the present purpose, since a fixed substitution alphabet falls to frequency analysis and word-pattern analysis however the shapes are paired. The security parameter of a visual substitution is deliberately negligible, because the aim is that the reader's eye should be able to see through it. Second, the space that matters is a thin slice of the whole. A substitution that makes every letter resemble a different letter, with a resemblance high enough to be useful, is constrained by the geometry of the shapes, and the effective design space is the set of alphabets that satisfy those constraints, a small subset of the figures in the table. The interesting design problem is therefore not the size of the space but the identification of its good regions.

2.3 Conventions that support reading

A visual substitution is not meant to be read letter by letter. Skilled reading draws on word shape, on the redundancy of the language, and on the predictability of words from their context. Shannon's experiments on the prediction of printed English put the information content of ordinary text at roughly one bit per letter, against the $\log_2 26 \approx 4.7$ bits that a uniformly random letter would carry (Shannon, 1951). A reader therefore has on average a great deal of contextual help with any one letter, and a substitution that preserves spaces, punctuation and case leaves that help intact. This is the sense in which the ambiguity of an individual replacement is tolerable: the replacement of a letter by a lookalike changes a small fraction of a word's features, and the lexicon plus the sentence usually settles the rest. It is also why the quality of an alphabet should be judged on whole passages. Several individually plausible exchanges can combine into something confusing, for instance when the same word contains two exchanged letters whose joint effect is a different familiar word, and a reciprocal pairing is best treated as a first approximation to be adjusted by trying it on ordinary prose.

2.4 Cycles longer than two

Reciprocal exchange is not the only structure available. A substitution may contain longer cycles, such as $I \rightarrow L \rightarrow T \rightarrow I$, in which each letter is replaced by a different neighbour and the inverse cycle is needed for decoding. The loss relative to the reciprocal case is that encoding and decoding use different tables, and the gain is a larger set of shape-compatible alphabets, because a cycle needs only that each letter resemble its successor and not that the relation hold in both directions. A further and less obvious difference between cycles of length two and cycles of length three or more appears when the shapes are interpolated, and it is established in Section 6: a transposition cannot be effected by linear interpolation without the two letters sharing a glyph at some moment, whereas a three-cycle in general position can.

3 Resemblance as a Quantity to Be Optimised

3.1 Formulation

Suppose a weight $w(a, b) \geq 0$ is given for every pair of distinct letters, with larger values meaning that a and b look more alike, and suppose the weight is symmetric. The design of a reciprocal substitution that changes every letter and maximises total resemblance is then the problem of finding a perfect matching of maximum total weight in the complete graph on 26 vertices. This is a classical problem solvable in polynomial time by the blossom algorithm (Edmonds, 1965). If the substitution is allowed to contain longer cycles and the weight is not symmetric, the problem becomes an assignment problem with the diagonal forbidden, solved by the Hungarian method (Kuhn, 1955). If only k pairs are to be exchanged and the remaining letters left in place, the problem is a maximum-weight matching of exactly k edges, which is again polynomial. Allowing fixed points without further constraint is useless, since the identity substitution has perfect resemblance, so some such constraint, whether a number of changed letters or a quota of demonstrated arbitrariness, is needed to make the optimisation meaningful. The point of stating the problem in this form is that the optimisation machinery is independent of where the weights come from. Weights may be derived from the geometry of rendered glyphs, from feature models of letter identification (Pelli et al., 2006), or from empirical confusion matrices recording how often observers mistake one letter for another (Townsend, 1971; Gilmore et al., 1979), and the same matching procedure applies to each.

3.2 A computable proxy and its results

To make the problem concrete, a simple proxy was computed. Each capital letter was rendered as a bitmap on a 200×200 pixel canvas at an em size of 160 pixels, sharing a common baseline and centred by advance width, so that relative sizes and positions were preserved and no letter was rescaled to fill its box. The weight of a pair was the Jaccard overlap of the two bitmaps, the area of the intersection of the inked regions divided by the area of their union. This measure runs from 0 for letters with disjoint ink to 1 for identical glyphs. Four typefaces were used: DejaVu Sans, Liberation Sans, DejaVu Serif and TeX Gyre Pagella. For each, the full 26×26 weight matrix was computed and a maximum-weight perfect matching found with a standard implementation of the blossom algorithm. The procedure, with its code, is reproduced in Appendix C.

Font	Optimal total	Mean per pair	Greedy total	Weakest pair	Strongest pair
DejaVu Sans	6.689	0.515	6.616	MV (0.23)	OQ (0.92)
Liberation Sans	6.493	0.499	6.332	AW (0.27)	OQ (0.85)
DejaVu Serif	6.638	0.511	6.418	MV (0.28)	OQ (0.90)
TeX Gyre Pagella	6.443	0.496	6.102	MV (0.26)	OQ (0.81)

Table 3: Maximum-weight reciprocal pairings of the 26 capitals under the pixel-overlap measure, by typeface. The greedy column takes the best remaining pair repeatedly. Overlap values are Jaccard indices.

The totals in Table 3 correspond to mean overlaps of about one half per pair in every font, which is to say that on average the two letters of a pair share about half of the area covered by their union. The greedy procedure that repeatedly takes the best remaining pair comes within roughly one to five per cent of the optimum in total weight, and the gap is largest for the Pagella serif face, so a hand design that proceeds pair by pair and settles the awkward letters last is not far from optimal under this measure but can be improved by a global computation. The strongest pair in every font is O and Q, whose overlap lies between 0.81 and 0.92, and the weakest pair is M and V in three of the four fonts and A and W in the fourth.

Table 4 lists the overlaps for the exchanges that arise naturally in discussion, including B and P, U and V, C and G, E and F, M and N, O and Q, I and L, and M and W. Several results deserve comment. The exchange of C and G scores between 0.54 and 0.66 and that of E and F between 0.52 and 0.65, so these are good pairs by the measure. B and P score between 0.41 and 0.52, which is moderate. U and V score between 0.19 and 0.24, M and N between 0.20 and 0.26, M and W between 0.18 and 0.23, C and K about 0.2, and I and L between 0.05 and 0.12. If one asks which of these a designer would call visually related, U and V, M and W and perhaps I and L would be on most lists, and the measure ranks them low.

Pair	DejaVu Sans	Liberation Sans	DejaVu Serif	TeX Gyre Pagella
O/Q	0.92	0.85	0.90	0.81
C/G	0.54	0.62	0.66	0.58
E/F	0.60	0.52	0.65	0.56
B/P	0.41	0.52	0.47	0.49
M/N	0.26	0.21	0.20	0.21
C/K	0.22	0.21	0.19	0.19
M/W	0.23	0.19	0.20	0.18
A/H	0.22	0.21	0.22	0.19
U/V	0.24	0.21	0.23	0.19
I/L	0.05	0.05	0.12	0.09

Table 4: Pixel overlap for selected pairs, by typeface.

The reason is not mysterious. Overlap of ink area is a measure of coincidence, and it is high when two glyphs occupy the same region of the page, as O and Q do. It is not a measure of structural similarity. U and V share a topology, an open curve with two ascending ends, but the curved bottom of one and the pointed bottom of the other leave little shared area, and M and W are related by a reflection in a horizontal axis that coincidence of area does not detect. The measure is therefore a poor stand-in for the perceptual relation that a reader uses, and its results should be read as a demonstration of the procedure and a warning, not as a recommendation. A better weight could be derived from skeletons, from feature sets, or from human confusion data, and an experimental programme of the kind described in Section 9 would supply the last of these directly.

3.3 What the pairings show

Three features of the computed pairings survive the weakness of the measure, because they follow from the structure of the optimisation and not from the particular weights. First, the optimal pairing depends strongly on the typeface. Only six of the thirteen pairs are shared by all four fonts, namely A and W, C and G, D and U, H and N, M and V, and O and Q, and a further three, B and S, I and J, and T and Y, appear in three of the four. Appendix B gives the full pairings. Since the shapes of letters are properties of a design and not of an alphabet, a substitution chosen for one typeface need not be good in another, and an edition printed in a different font changes the quality of its own cipher. The implication is that the resemblance of a substitution is a property of an alphabet *together with* a typeface, and that a project which intends editions to be printed in whatever font a reader prefers has to treat the typeface as part of the experimental condition.

Second, the global constraint costs locally good pairs. B and P is an attractive exchange taken alone, with overlap up to 0.52, yet it appears in none of the four optimal pairings, where B is matched to S in three fonts and to R in the fourth. The reason is that a perfect matching must pair every letter, so a pair that is good in isolation can be displaced if its members are needed to resolve the pairing of difficult letters elsewhere. This is the situation that the earlier remark about letters such as A, H, R, W, X and Z anticipates, since these are the letters with few close neighbours, and the computed pairings agree: A is paired with W in all four fonts, at overlaps between 0.27 and 0.33 that are among the lowest in each pairing. If instead only some letters are changed, the matching is of exactly k edges and pairs such as B and P can be retained. The number of letters changed is a design variable, and the counts in Table 2 show that the space of choices over which letters to change is large at every size.

Third, the choice between changing all letters and changing a subset is not merely a matter of optimisation but of purpose. A subset substitution is easier to read and demonstrates arbitrariness less thoroughly, while a full substitution is harder to acquire and makes a stronger point about the conventionality of the shapes. The empirical programme in Section 9 treats the number of changed letters as one of its manipulated factors.

4 Arbitrariness of Assignment and Constraints on Acquisition

4.1 Two relations that should not be confused

The claim that the sign is arbitrary has a long history in linguistics. In Saussure's formulation the link between a signifier and what it signifies is not motivated by any intrinsic resemblance (de Saussure, 1983). In Peirce's classification the relation of a sign to its object may be iconic, indexical or symbolic, and the letters of an alphabet are by and large symbols, tied to what they stand for by convention and habit and not by likeness (Peirce, 1931). For writing, the thesis has a direct and checkable form. There is no necessity that the letter-shape B should serve for the sound it conventionally represents in English, the same language is written in many scripts, and a language's script can change by decree, as the replacement of an Arabic-

based script by a Latin-based one in Turkey in 1928 shows (Lewis, 1999). A small number of scripts retain iconic residues, but in the alphabets of immediate interest the assignment is conventional through and through, and the historical record of the derivation of letter shapes from one another shows convention accumulating on convention (Gelb, 1952; Daniels and Bright, 1996).

What arbitrariness of assignment does not say is that all assignments are equally easy to learn, to tell apart, to typeset, or to read in sequence. Between the set of signs and the reader there is a second structure that the assignment leaves untouched, namely the similarity relation among the shapes themselves. Two letters may be confusable, or share features, or be related by a reflection or a rotation, and this holds whatever sounds or functions are attached to them. The visual substitution alphabet makes the two relations separable. Its assignments are conventional, since there is no reason beyond the experiment's convenience for B to be replaced by P and not by some other letter. But the choice between B and P and the choice between C and K differ in the perceptual bridge they provide, and that difference is a fact about the similarity structure of the shapes and about the reader's habits, not about the assignment as such. The project's theoretical position can therefore be put as follows: the conventionality of writing is demonstrated by the existence of many workable assignments, while the conditions that make a particular convention usable are bodily, perceptual and historical, and a programme of ciphers, typefaces and drifting editions exhibits both claims within a single apparatus.

4.2 What is known about the acquisition of letter forms

The literature on how readers acquire letter forms supplies several constraints that bear directly on the design of intermediate alphabets. E. J. Gibson's account of perceptual learning treats the acquisition of an alphabet as the discovery of distinctive features, the invariant properties by which letters differ from one another, and the early experiments with letter-like forms tested which transformations children found easy to discriminate (Gibson, 1969; Gibson et al., 1962). On this view the difficulty of learning a new alphabet depends on whether its members differ from one another by features the learner is already sensitive to. A related hypothesis, the cultural recycling of cortical maps, proposes that reading draws on visual machinery that evolved for other purposes and is therefore shaped by prior biases (Dehaene and Cohen, 2007; Dehaene, 2009), and there is evidence that the contour configurations of letters across scripts resemble those found in natural scenes (Changizi et al., 2006). If these accounts are broadly right, then the set of shapes that can be learned easily is not arbitrary in the way the set of assignments is, and the freedom of a designer is bounded by the reader's perceptual priors.

One constraint is of special relevance to visual ciphers. Children and adults tend to generalise across mirror images, treating a shape and its reflection as the same object, and learning to read involves suppressing this generalisation for the letters where reflection changes identity, such as b and d or p and q (Dehaene et al., 2010). A substitution that exchanges mirror-related lower-case letters therefore recruits one of the best documented

sources of confusion in letter learning, and a designer who wishes to build resemblance into an alphabet should be aware that resemblance of this kind is not always helpful. It can be a bridge, since the reader's eye is drawn to the right neighbourhood, and it can also be an obstacle, since the reader's eye is drawn to the wrong member of the neighbourhood. Which of these happens for a given pair is an empirical question, and the relative importance of the two effects is one of the quantities the programme in Section 9 is designed to estimate.

4.3 Levels of knowing a sign

It is useful to separate several levels at which a reader may know a letter form, because the claims of the project differ by level. At the lowest level a reader recognises a form in context, resolving it through the surrounding word and sentence without being able to say what it is in isolation. At the next level the reader identifies the form when it is presented alone, which requires that its association with a letter be stable and not merely contextually inferred. At a third level the reader can name the form or decode it deliberately, as a student does with a table of signs. At the highest level the reader can produce the form, writing or drawing it correctly from memory. The distinction between receptive and productive knowledge is standard in the study of vocabulary acquisition, where recognition typically runs ahead of production (Nation, 2001), and the same asymmetry is expected here. A drifting book is designed to exercise the first level and, with luck, the second. It makes no claim to teach the third and the fourth, and any assertion that readers have acquired a script has to say at which level they have acquired it and how this was measured.

4.4 Precedents

None of this is without precedent. The Turkish alphabet reform was carried out by a state with the machinery of schooling and legislation behind it, and its theoretical interest for present purposes is that it shows a national convention replaced within a generation (Lewis, 1999). The Initial Teaching Alphabet used in British schools in the 1960s was a transitional script, intended to be learned first and then replaced by traditional orthography, and the evaluations of that programme are an obvious source of methods and cautions (Downing, 1967). Languages have long been written in scripts that are not their own: the Arabic-script writing of Romance and of several African languages in the traditions known as Aljamiado and Ajami is a standing demonstration that the same language can ride on different graphic vehicles (Daniels and Bright, 1996). Among speakers of Arabic, informal Latin-script chat orthography assigns digits to Arabic sounds by visual resemblance, a 3 for the letter ayn and a 7 for the letter ha being the commonly cited examples (Palfreyman and Al Khalil, 2003), and this is a naturally occurring visual cipher built by its users for their own purposes. At the level of the individual reader, studies of reading in geometrically transformed typography have found that practice at reading inverted or reflected text leaves traces that survive for a long time, with measurable savings when readers are tested a year later (Kolers, 1976). All of these show that conventions can be changed and that readers adapt. They differ from

the present proposal in one respect that is also the source of its novelty, namely that the acquisition is to take place inside a single volume, without a separate stage of instruction, and the conditions under which this can succeed are not settled by any of the precedents.

5 The Edition as a Composite of Independent Components

5.1 Four adjustable components

The separation that the LingoJam Cipher font exploits can be stated in general terms. A printed page results from a chain of choices, and four of them can be varied independently. The first is the *linguistic text*, the sequence of words an author writes and a corrector repairs. The second is the *character substitution*, a rule that replaces the characters of the text by other characters before anything is displayed. The third is the *glyph assignment*, a rule, realised in a font, that determines which shape is drawn for each stored character. The fourth is the *schedule*, a rule that determines how the second and third change across the positions of the text, and it is trivial in a static edition and central in a drifting one. Let $T \in \Sigma^N$ be the text, written as a sequence of N letter positions, π a substitution, and $\mathcal{G}$ a space of shapes. Each stored character $x \in \Sigma$ has a *route*, a path $\rho_x : [0, 1] \rightarrow \mathcal{G}$ from its source shape $\rho_x(0)$ to its destination shape $\rho_x(1)$, so that $\rho_x(\alpha)$ is the shape of x when it has moved a fraction α of the way along that route. The *schedule* is a vector-valued function of position,

$$\alpha(p) = (\alpha_A(p), \alpha_B(p), \dots, \alpha_Z(p)), \quad \alpha_\ell(p) \in [0, 1],$$

where p is the position in the book and $\alpha_\ell(p)$ states how far the glyph for letter ℓ has moved from its source form towards its destination form. The display of position p is the shape $\rho_x(\alpha_x(p))$ with $x = \pi(T_p)$, and the page is the sequence of these shapes. A *uniform* transition sets every component to a common value $t(p)$, which is the single-percentage picture; a *letter-by-letter* transition lets the components move separately, so that one edition may bring a single letter to its final form while leaving the others familiar, another may introduce several letters at a low fraction and strengthen them as the reader meets them repeatedly, and a third may carry every letter through ten, twenty and eventually one hundred per cent together. Figure 1 shows the arrangement. Two cautions attach to the notation. The value $\alpha_\ell(p) = 0.5$ names a position in a specified glyph transformation, and it does not imply that the reader has acquired half the letter or that the form is halfway as difficult. And the route ρ_x is itself a design choice, because the same pair of endpoints can be joined by many paths, a point developed in Section 6.

Two configurations at the extremes show what the separation permits. In a *compensated* configuration the stored text is substituted and the glyph assignment $\gamma : \Sigma \rightarrow \mathcal{G}$, a static map from stored characters to shapes (a route evaluated at a fixed fraction), undoes the substitution, so that $\gamma(\pi(x))$ is the ordinary shape of x . The page looks entirely normal, and the stored characters are the cipher. This is the arrangement of the LingoJam Cipher font, where the stored characters belong to one alphabet and the font displays them as another. In

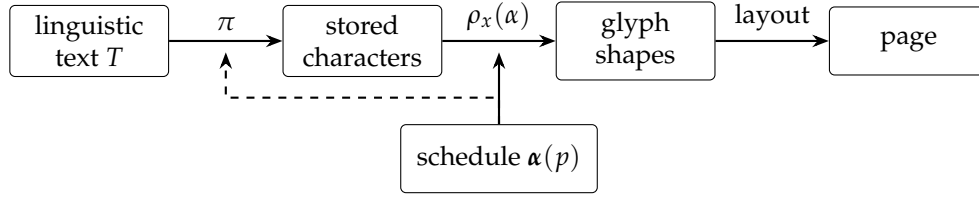


Figure 1: The composite of an edition. The text, the substitution π , the routes ρ_x and the schedule α are separately identifiable, and the page depends on all four.

an *uncompensated drifting* configuration the stored text is ordinary and π is the identity, while the routes carry each letter from its ordinary shape to the shape of its partner as the schedule advances. Here the stored characters are the plain text, and the visible forms change. Between them lie intermediate arrangements in which the font retains recognisable features of both letters of a pair, so that the substitution remains perceptible while the familiar structure supports reading, and one can also produce several grades of the same idea, moving from strongly corrective glyphs towards ordinary encoded letters, so that the reader’s familiarity with the alphabet gradually takes over the work that the font initially performed.

5.2 Display equivalence and the storage decision

A given page can be produced by many different pairs of stored text and glyph assignment. If X is a stored string and γ a glyph assignment, the display is $\gamma(X)$, and two pairs (X, γ) and (X', γ') produce the same page precisely when $\gamma(X) = \gamma'(X')$. For instance, with the exchange of B and P, the pair (stored PUB, assignment that draws P as B and B as P) and the pair (stored BUP, ordinary assignment) look identical on the page. Which representative of the equivalence class is stored is a decision with consequences, because the stored text is what search, copying, screen readers, speech synthesis, spell checking and version control act on. A compensated cipher stores gibberish that merely looks like text, and every such tool sees the gibberish. An uncompensated drifting configuration stores the author’s text, so that these tools continue to work, and places the whole burden of the transformation on the glyphs. This suggests a design principle, which will be called the stored-text principle: wherever possible the stored text should be the linguistic text, and transformations should be carried by the glyph assignment and the schedule. The principle is not a theorem but a way of making accessibility and correction survive the experiment. It means in particular that a plain edition can always be produced from the same source by selecting the ordinary glyph assignment, and that a reader who finds the drifting edition unreadable is never denied access to the text.

5.3 Font technology

The relevant mechanisms are standard but their limits matter. In an OpenType font the character map associates each stored character with a glyph, and substitution tables in the

glyph substitution table allow replacement of glyphs by alternates, in some cases depending on neighbouring glyphs (OpenType). A font whose only job is to display each character as another character’s shape is a simple case. Interpolation between shapes is available through variable fonts, introduced in OpenType 1.8, in which a font defines design axes and glyph outlines vary continuously with the axis values, provided that the master outlines are point-compatible, meaning that they have the same structure of contours and points so that corresponding points can be interpolated (OpenType). The idea of a parametrised letter is older than the standard; Knuth’s description of a meta-font treats a typeface as a program with parameters from which many related designs can be generated (Knuth, 1982).

Two limitations determine how a drifting book is built. First, a font’s layout rules operate on the text of a run and do not count occurrences across a book, so the stage of a letter, which depends on how many times that letter has been seen, cannot be computed inside the font. It has to be assigned by a preprocessing step that annotates each token with its stage, either by selecting among several static fonts that each embody one stage, or by setting the axis values of a variable font for each run of text. Second, an interpolation axis in a variable font is applied to every glyph in a run in the same way, so that per-letter schedules require either separate axes per letter, which becomes cumbersome for an alphabet, or a division of the text into runs, one for each letter and stage, at the cost of heavy markup. The sensible organisation is an edition builder, a program that reads the text, computes stages from the schedule, emits annotated text, and calls a typesetting engine with the appropriate fonts. The same builder can produce the plain edition, a static cipher, and a drifting edition from one source, which is what the reproducibility requirements of Section 8 demand.

5.4 Contextual display

The letterwise model, in which each stored character receives a shape independent of its neighbours, is adequate for the Latin alphabet in its ordinary use and fails for scripts whose letter forms depend on context. Arabic is the standard example, and Section 7 treats it. The general model is that the display function is not a map $\Sigma \rightarrow \mathcal{G}$ extended letter by letter but a map from strings of stored characters to strings of glyphs, in the simplest case computed by a finite-state transducer reading the string and emitting glyphs, possibly with ligatures that merge several characters into one glyph. All the statements of this section go through with this generalisation, but the commuting of rendering with substitution, which is automatic in the letterwise case, is not, and Proposition 7.1 below records exactly when it holds.

6 Drift: Intermediate Forms and Schedules

6.1 Margins and routes

Fix a space of shapes $\mathcal{G}$ with a distance d , for instance the Euclidean distance between the control-point vectors of point-compatible outlines, or the distance between rendered bitmaps. A drifting edition presents at each position p a letter x in the shape $\rho_x(\alpha_x(p))$, and a reader

can only use the presentation if distinct letters remain distinguishable throughout. Define the *margin* of an edition as the smallest distance between the displayed shapes of two distinct letters over all positions at which both are in play,

$$m = \min_p \min_{x \neq y} d(\rho_x(\alpha_x(p)), \rho_y(\alpha_y(p))).$$

A legibility threshold $\theta > 0$ is a property of readers and typefaces and is not derived here; the design requirement is simply $m \geq \theta$. The margin is determined by the routes ρ_x and by the schedule α jointly, and the results of this section show that the obvious choices of both can drive it to zero.

Between two fixed endpoints there are many routes. A glyph might change through the movement of its strokes, so that outlines deform continuously, through the successive replacement of its parts, so that a stem, a bowl or a terminal is exchanged for its counterpart in turn while the rest of the shape stays fixed, or through intermediate forms designed by hand specifically to preserve recognition, keyframes each of which is a good letter in its own right. These routes begin and end with identical alphabets and can differ greatly in what a reader experiences along the way. Neither the endpoints nor the schedule fixes the route, and the choice of route is a distinct design variable of the edition.

6.2 Collision under linear exchange

Let a and b be two letters whose destination shapes are exchanged, so that $\rho_a(0) = A$, $\rho_a(1) = B$, $\rho_b(0) = B$ and $\rho_b(1) = A$ for distinct shapes $A, B \in \mathcal{G}$, where $\mathcal{G}$ is a real vector space. Take the linear routes $\rho_a(\alpha) = (1 - \alpha)A + \alpha B$ and $\rho_b(\alpha) = (1 - \alpha)B + \alpha A$. Cross-dissolving two bitmaps is an example of such a route, because averaging images is linear interpolation in a vector space of images.

Proposition 6.1 (Collapse of linear exchange). *Let (α_a, α_b) be a schedule for the two letters that begins at $(0, 0)$ and ends at $(1, 1)$. (i) If the schedule is continuous, then at some position the letters a and b are displayed in the same shape. (ii) If the schedule takes only the values 0 and 1, each letter being either wholly original or wholly final, and changes one letter at a time, then the two letters are displayed in the same shape in the intermediate state.*

Proof. The shapes coincide exactly when $\rho_a(\alpha_a) = \rho_b(\alpha_b)$, that is, when $(1 - \alpha_a - \alpha_b)(A - B) = 0$, and since $A \neq B$ this holds if and only if $\alpha_a + \alpha_b = 1$. For (i), the function $f = \alpha_a + \alpha_b - 1$ is continuous with $f = -1$ at the start and $f = +1$ at the end, so it vanishes somewhere by the intermediate value theorem. For (ii), the first change of one coordinate leaves the state at $(1, 0)$ or $(0, 1)$, where $\alpha_a + \alpha_b = 1$; in the state $(1, 0)$, for example, a is already displayed in the shape B and b has not begun to move from B . $\square$

A schedule that changes in discrete steps of intermediate size can step over the collision line without landing on it. It then avoids exact coincidence only by making an abrupt jump, and its distance from coincidence at the nearest sampled position is at most $|\alpha_a + \alpha_b - 1| \|A - B\|$.

$B\|$, which is small whenever the steps are fine. Exact collision is therefore the limiting case of a general near-collision, and the simultaneous jump of both letters from $(0, 0)$ to $(1, 1)$ is the only way to avoid it altogether, at the cost of the reader seeing the exchange all at once.

At the colliding position both letters share one glyph, which is the common point $(1 - \alpha_a)A + \alpha_a B$, and in the synchronous case $\alpha_a = \alpha_b = \frac{1}{2}$ it is the midpoint $\frac{1}{2}(A + B)$. Figure 2 makes this visible for B and P by cross-dissolving rendered bitmaps. The first two rows show the stored letters B and P moving towards each other's shapes, and in the middle column the two rows are pixel for pixel identical, a fact verified by the program that drew the figure. A reader who sees this glyph cannot say from the shape alone which letter it stands for and must use context, which is exactly the loss of distinguishing information that the margin was defined to detect.



Figure 2: Cross-dissolve of B and P rendered in DejaVu Sans Bold, with both letters at the same fraction α of the way to the other's shape. In the first two rows (linear routes) the two letters have identical glyphs at $\alpha = \frac{1}{2}$. In the last two rows a mark above the stored B and below the stored P, with weight proportional to $\sin \pi\alpha$, keeps them distinct throughout and vanishes at both endpoints.

The geometry can be drawn in the unit square of the pair of schedule values (α_a, α_b) . Every schedule is a path from the lower-left corner $(0, 0)$ to the upper-right corner $(1, 1)$, and the glyphs of the two letters coincide on the anti-diagonal $\alpha_a + \alpha_b = 1$, which every such path must cross. Figure 3 shows this, and also shows the two kinds of path discussed above:

the synchronous path along the diagonal, which meets the anti-diagonal at the centre, and the one-letter-first path, which runs along the edges of the square and meets it at a corner.

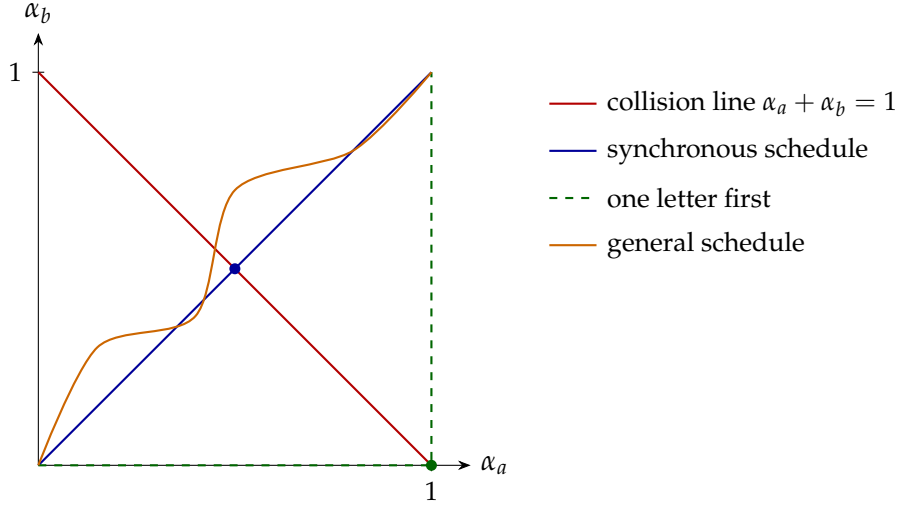


Figure 3: Schedules of two exchanged letters as paths in the unit square. Under linear routes the displayed shapes coincide on the collision line, which every path from $(0,0)$ to $(1,1)$ meets.

6.3 The detour

The collision is a consequence of confining both routes to the straight line through A and B . In a shape space of dimension at least two there is room to go around. Let $u \in \mathcal{G}$ be a vector orthogonal to $A - B$ and $\lambda > 0$, and replace the linear routes by

$$\rho_a(\alpha) = (1 - \alpha)A + \alpha B + \lambda \sin(\pi\alpha) u, \quad \rho_b(\alpha) = (1 - \alpha)B + \alpha A - \lambda \sin(\pi\alpha) u.$$

Both routes still begin and end at the original shapes, because $\sin(\pi\alpha)$ vanishes at $\alpha = 0$ and $\alpha = 1$, and in between the two letters pass on opposite sides of the line through A and B .

Proposition 6.2 (Detour). *For the routes above, and for any schedule values $(\alpha_a, \alpha_b) \in [0, 1]^2$, the displayed shapes of a and b coincide if and only if $(\alpha_a, \alpha_b) \in \{(1, 0), (0, 1)\}$. Along the synchronous path $\alpha_a = \alpha_b = \alpha$ the distance between the two shapes is*

$$\sqrt{(1 - 2\alpha)^2 \|A - B\|^2 + 4\lambda^2 \sin^2(\pi\alpha) \|u\|^2},$$

which equals $\|A - B\|$ at the endpoints and, for $\mu = \lambda \|u\| / \|A - B\| < 1/\pi$, attains its minimum $2\mu \|A - B\|$ at $\alpha = \frac{1}{2}$.

Proof. The difference of the two shapes is $d = (1 - \alpha_a - \alpha_b)(A - B) + \lambda(\sin \pi\alpha_a + \sin \pi\alpha_b)u$. Since $u \perp A - B$, $\|d\|^2 = (1 - \alpha_a - \alpha_b)^2 \|A - B\|^2 + \lambda^2(\sin \pi\alpha_a + \sin \pi\alpha_b)^2 \|u\|^2$, and this is zero if and only if $\alpha_a + \alpha_b = 1$ and $\sin \pi\alpha_a + \sin \pi\alpha_b = 0$. On $[0, 1]$ the sine is nonnegative, so the second condition forces $\sin \pi\alpha_a = \sin \pi\alpha_b = 0$, hence $\alpha_a, \alpha_b \in \{0, 1\}$, and together

with the first condition this gives the two corners. The synchronous formula follows by substitution. Writing $\alpha = \frac{1}{2} + \delta$ with $|\delta| \leq \frac{1}{2}$, the squared distance divided by $\|A - B\|^2$ is $4\delta^2 + 4\mu^2 \cos^2(\pi\delta)$, and subtracting its value $4\mu^2$ at $\delta = 0$ leaves $4\delta^2 - 4\mu^2 \sin^2(\pi\delta) = 4(|\delta| - \mu \sin \pi|\delta|)(|\delta| + \mu \sin \pi|\delta|)$. Because $\sin \pi|\delta| \leq \pi|\delta|$ and $\mu\pi < 1$, the first factor is positive for $\delta \neq 0$, so the minimum is attained uniquely at $\delta = 0$. $\square$

Figure 4 plots the synchronous separation for three values of μ . With $\mu = 0$ the separation falls to zero at the midpoint, which is the collision, and with positive μ it keeps a floor of $2\mu\|A - B\|$. The result has one caution of its own, visible in the corner case: a detour does not protect a schedule that carries one letter all the way to its destination while the other has not yet begun, because the extra term vanishes at both endpoints, so the state $(0, 1)$ shows b in the original shape of a . The schedule must therefore keep the pair away from the opposite corners, and in practice this means moving the two letters of a pair together or interleaving their steps closely.

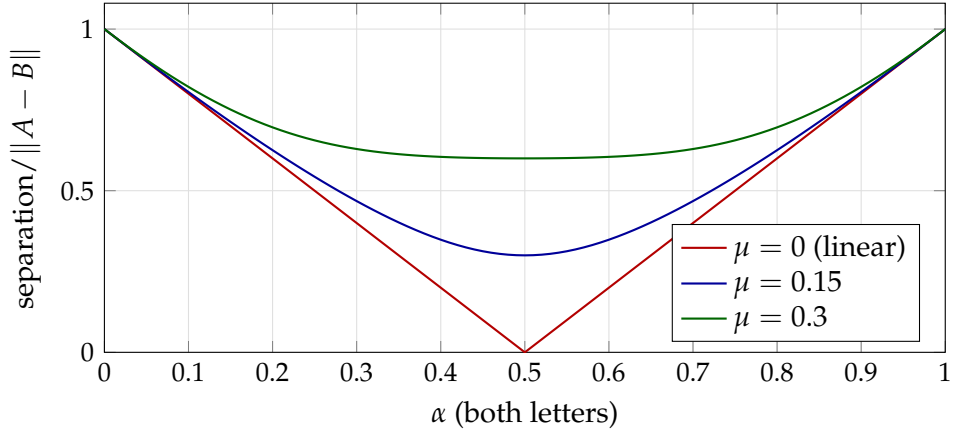


Figure 4: Distance between the displayed shapes of two exchanged letters under the detour routes of Proposition 6.2, as a fraction of the distance between the endpoint shapes. The parameter μ is a geometric parameter of the model and has no empirical status.

The detour has a historical analogue in the Arabic script. Many Arabic letters share a skeleton and are distinguished by the number and position of dots, a system in which the skeletal forms of early texts were supplemented by dotting to separate letters that the skeleton alone left ambiguous (Daniels and Bright, 1996). A mark that is orthogonal to the dimension along which two letters are confusable is exactly what the detour supplies. The last two rows of Figure 2 realise it with a dot placed above one letter and below the other, with weight proportional to $\sin \pi\alpha$. Whether such marks help or hinder reading, and whether they leave residue that interferes with the final forms, are empirical questions, and the proposition says only that the geometric obstacle can be removed at the cost of intermediate glyphs that are not letters of either endpoint alphabet.

6.4 Longer cycles

The collision of Proposition 6.1 is specific to exchange of two letters. Take a three-cycle $a \rightarrow b \rightarrow c \rightarrow a$ with distinct shapes A, B, C , so that a moves from A to B , b from B to C , and c from C to A , each along a linear route with a common parameter α .

Proposition 6.3 (Three-cycles). *If A, B, C are affinely independent, no two of the three letters are displayed in the same shape for any $\alpha \in [0, 1]$.*

Proof. The shapes at parameter α are $(1 - \alpha)A + \alpha B$, $(1 - \alpha)B + \alpha C$ and $(1 - \alpha)C + \alpha A$. The difference of the first two is $(1 - \alpha)A + (2\alpha - 1)B - \alpha C$, a combination of A, B, C whose coefficients sum to zero. By affine independence such a combination vanishes only if all coefficients vanish, but $1 - \alpha = 0$ forces $\alpha = 1$ and then $2\alpha - 1 = 1 \neq 0$. The other two differences are obtained by cyclic relabelling. $\square$

Affine independence holds for any three points not on a common line, so it is the generic case in a shape space of any realistic dimension. The consequence for design is a trade-off. Reciprocal alphabets are self-inverse and simple to document but cannot be drifted by a shared linear parameter without collision, whereas three-cycles can, at the cost of distinct tables for encoding and decoding. The alternative for reciprocal alphabets is the detour of the preceding subsection. A further remark concerns topology. The space of ordered pairs of distinct points in $\mathbb{R}^n$ is homotopy equivalent to the sphere S^{n-1} , so for $n = 2$ there are inequivalent ways of exchanging two points, clockwise and anticlockwise, while for $n \geq 3$ every route is topologically equivalent to every other. In a high-dimensional shape space, therefore, routes between the same endpoints are not distinguished by their topology, and the differences that matter between them are quantitative: the margin, the legibility of the intermediate forms, and the exposure they allow.

6.5 Cycles of presentation

Nothing in the formalism requires the schedule to be monotone or the endpoint to be final. A route can pass through a hybrid script on the way from Latin to Standard Galactic and continue back to Latin, so that forms learned in the middle of the book return in the context of the original alphabet. Formally this is a route whose parameter is a loop in $\mathcal{G}$, with the source shape of each letter also its last. Such a design changes the empirical question in an interesting way. A reader who recovers the original form after passing through the hybrid has had the opportunity to learn the association between the two while reading, and a test after the return can ask whether the intermediate forms are still recognised, which is a measure of retention independent of the final state. It also doubles the number of occasions on which the schedule must avoid collisions, since the same pairs of letters are crossing in both directions, and the discussion of exposure below applies to each leg.

6.6 Exposure and the novelty load

A schedule changes shapes as a function of position, but what a reader learns depends on the number of occasions on which each shape is seen, and these are tied to position through the frequency of letters. Let p_ℓ be the relative frequency of letter ℓ in the text and $N_\ell(p)$ the number of occurrences of ℓ before position p . A schedule that is uniform in position gives every letter the same fraction of the transition at the same point of the book and therefore gives it exposures in proportion to frequency, whereas a schedule that is driven by exposure advances each letter after a fixed number of its own occurrences, for instance

$$\alpha_\ell(p) = \min\left(1, \max\left(0, \frac{N_\ell(p) - D_\ell}{KS}\right)\right),$$

where D_ℓ is a delay in occurrences before the letter begins to change, S is a number of stages, and K is the number of exposures required per stage.

To see what position-uniform schedules imply, take a model book of 80,000 words with 4.7 letters per word on average, a modelling assumption rather than a measurement, giving $L = 376,000$ letters, and take standard English letter frequencies (Lewand, 2000). Divide the book into $S = 5$ equal stages of 75,200 letters. The letter E, with relative frequency 12.7 per cent, is seen about 9,550 times in each stage, and Z, with 0.074 per cent, about 56 times, a ratio of 172 to 1, which is the ratio of the frequencies. Table 5 gives the figures for a selection of letters, together with the length of text a letter needs to reach its final form under an exposure-driven schedule with $K = 40$ and $S = 5$, that is, 200 exposures in all. The most frequent letters complete in a few hundred words, while Z needs about 270,000 letters, 72 per cent of the model book, and this means that it spends most of the book in transition. The numbers $K = 40$ and $S = 5$ are illustrative values with no empirical standing.

Letter	Freq.	Per stage, uniform	To final stage, exposure-driven	In words
e	12.70%	9 552	1 575	335
t	9.06%	6 810	2 208	470
a	8.17%	6 142	2 449	521
n	6.75%	5 075	2 963	631
d	4.25%	3 198	4 703	1 001
m	2.41%	1 809	8 312	1 769
b	1.49%	1 122	13 405	2 852
k	0.77%	581	25 906	5 512
j	0.15%	115	130 718	27 812
q	0.10%	71	210 524	44 792
z	0.07%	56	270 268	57 504

Table 5: Exposure in a model book of $L = 376,000$ letters. Frequencies are standard English letter frequencies; the final column converts letters to words at 4.7 letters per word. Exposure-driven completion assumes $K = 40$ exposures per stage over $S = 5$ stages.

Two quantities limit what any schedule can do. The first is the total number of occurrences of the rarest letter. In the model book Z occurs about $0.00074 \times 376,000 \approx 278$ times, so a

transition of Z that requires more than 278 exposures cannot be completed inside the book, and with five stages the requirement $KS \leq 278$ allows only $K \leq 55$ per stage. The second is the amount of change a reader can be asked to absorb at once. Define the *novelty load* at a position as the fraction of letter tokens in a window around it whose letters are in an intermediate state, $0 < \alpha_\ell < 1$. A high load means that much of what the reader sees is unfamiliar, and a designer might impose a bound ν on it, with the value of ν to be fixed empirically. For a given alphabet the bound cannot be lower than the frequency of the commonest changing letter, which for E is 0.127, unless that letter is changed in sub-steps or left alone.

Model each letter ℓ as being in transition for exactly κ of its own occurrences, so for an expected κ/p_ℓ letters of text, during which it contributes load p_ℓ . Here κ is the total number of exposures a letter receives while in an intermediate state, so that in the notation above $\kappa = KS$.

Proposition 6.4 (Exposure conservation). *In this fluid model, with n changing letters, the time integral of the novelty load over the whole book is $n\kappa$ regardless of the schedule. Consequently any schedule whose load never exceeds ν requires a book of length at least $\max(n\kappa/\nu, \kappa/p_{\min})$ letters, where $p_{\min}$ is the smallest frequency among the changing letters.*

Proof. Letter ℓ contributes load p_ℓ for κ/p_ℓ letters, so its contribution to the integral is κ , independent of p_ℓ , and summing over n letters gives $n\kappa$. If the load is at most ν throughout, the integral over a book of length H is at most νH , so $H \geq n\kappa/\nu$. Separately, the rarest letter alone must remain in transition for $\kappa/p_{\min}$ letters, so $H \geq \kappa/p_{\min}$. $\square$

The proposition gives lower bounds, and the question of whether they can be attained was settled by simulation. For each combination of $\kappa \in \{20, 40, 80\}$ and $\nu \in \{0.15, 0.25, 0.40\}$ the 26 letters were scheduled in the fluid model by first-fit: each letter in turn starts at the earliest position at which adding its load keeps the total at or below ν for the whole of its interval. Two orders were compared, rarest letter first and commonest first. Table 6 gives the results. In all nine cases the rare-first order attains the bound $\kappa/p_{\min}$ exactly, which means that the capacity bound is slack in every case tested, and the common-first order takes between 2.1 and 8.4 per cent longer. Figure 5 shows the rare-first schedule for $\kappa = 40$ and $\nu = 0.25$: fifteen letters start at the first position, the eleven commonest letters begin only after the rarer ones are under way, and all of those complete within the first 4,000 letters, while the rarest letter stays in transition for 54,000.

The lesson of this analysis is not that these numbers are right, since κ and ν are unknown and the fluid model ignores the random fluctuation of letter counts, but that the structure of the problem is set by frequency. The binding constraint is the rarest letter and not the total amount of change, so a good schedule starts the slow letters early and treats the frequent letters, which learn quickly, as a late and brief affair. It also shows that a designer who wishes to give every letter the same quality of exposure must either choose an alphabet in which the rarest letters change least, so that fewer exposures are demanded of them, or accept that rare

κ	ν	Capacity bound $n\kappa/\nu$	Rare-letter bound $\kappa/p_{\min}$	Rare-first	Frequent-first
20	0.15	3 467	27 027	27 027	29 306
20	0.25	2 080	27 027	27 027	28 264
20	0.40	1 300	27 027	27 027	27 600
40	0.15	6 933	54 054	54 054	58 612
40	0.25	4 160	54 054	54 054	56 527
40	0.40	2 600	54 054	54 054	55 200
80	0.15	13 867	108 107	108 107	117 224
80	0.25	8 320	108 107	108 107	113 054
80	0.40	5 200	108 107	108 107	110 399

Table 6: Length of book, in letters, needed to complete all 26 transitions in the fluid model. κ is the number of exposures a letter spends in an intermediate state, ν the cap on the novelty load. Lower bounds are from Proposition 6.4.

letters will be known less well than common ones, in which case the transfer test of Section 9 should report performance by letter frequency.

6.7 Interference and the choice of endpoint

Two endpoint designs give different interference structures. If the destination shapes are the existing shapes of other letters, as in a visual substitution, then the end of the book reuses the shapes of its beginning with changed meanings: the shape that meant P at the start means B at the end. The reader must overwrite an old association and not merely add a new one, and the old association is among the best practised of all. If the destination shapes are new, as in a drift from Latin to Standard Galactic, there is no overlap with beginning forms and no overwriting is needed, at the price of learning an inventory of shapes with no prior foothold. Which is easier is a hypothesis for testing and not a matter of principle, and it matters for the formal analysis in Section 10, because it determines whether the local decodings of different parts of the book agree or conflict about the same shape.

7 Scripts in Contact: Latin, Arabic and Standard Galactic

7.1 Terminology

Experiments that move text between writing systems invite loose speech, and the distinctions deserve to be fixed before any claim is made about hybrids. Several different operations can be applied to a text, and they change different components of an edition. *Translation* changes the language, replacing words by words of another language that mean the same. *Transliteration* changes the script, rewriting the words of one language in the letters of another script so as to preserve pronunciation. *Substitution*, as in Section 2, changes the stored characters by a bijection of the alphabet and need not respect pronunciation at all. *Re-glyphing* changes the shapes drawn for stored characters and nothing else. *Interpolation* is re-glyphing

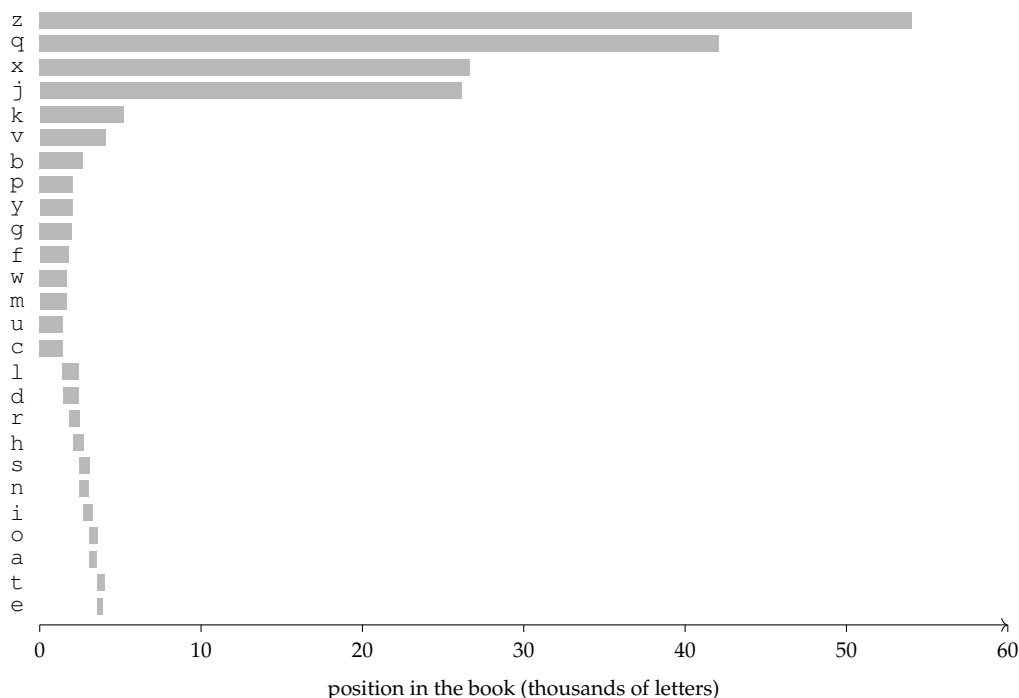


Figure 5: Rare-first schedule for $\kappa = 40$ and $\nu = 0.25$ in the fluid model. Each bar is the interval during which a letter is in an intermediate state. Rarest letters are at the top.

carried out progressively along routes by a schedule. *Re-sequencing* changes the direction or arrangement of text on the page. Table 7 summarises the effect of each on the components of an edition.

With the terms fixed, the descriptions used informally in the project can be stated exactly. A typeface with Arabic-inspired features applied to Latin letters is a re-glyphing of Latin-script text: the language is unchanged, the script is unchanged, and so is the reading direction, unless a re-sequencing is added. An Arabic-script text drawn with Latin-inspired forms retains its linguistic assignments if the assignments are preserved, and is again a re-glyphing, subject to the constraints of Arabic that are discussed below. Writing English words in Arabic letters is a transliteration, and writing Arabic words in a Latin-letter chat orthography with digits for the sounds that Latin lacks is likewise a transliteration, in which visual resemblance has been used as the principle of assignment. Interpolations between Latin and Standard Galactic are a different and especially clean case, since Standard Galactic is a letter-for-letter substitution of the Latin alphabet that appeared in video games of the early 1990s, so that the correspondences are given in advance and the whole of the design effort goes into the route between the shapes.

Operation	Changes	Example
Translation	language and its words	English text rendered in Spanish
Transliteration	script and stored characters, aiming at pronunciation	Arabic words written in Latin letters
Substitution	stored characters, by a bijection	PUBLIC stored as BVPLIC
Re-glyphing	glyph shapes only	Latin letters redrawn with Arabic-inspired strokes
Interpolation	glyph shapes, progressively	Latin drifting towards Standard Galactic
Re-sequencing	direction or arrangement of text	Latin text set right to left

Table 7: Operations on a text and what each changes.

7.2 What Arabic adds

Arabic cannot be treated as a Latin alphabet with different shapes, and a hybrid design that ignores the differences will fail as a reading system. The script is written from right to left, is cursive, and joins most of its letters to their neighbours. A letter has up to four forms depending on whether it joins to the letter before it, the letter after it, both, or neither, conventionally called the isolated, initial, medial and final forms. Not all letters join in both directions: six letters join to a preceding letter but not to a following one, and a word containing one of them is broken into several connected pieces, so that the joining behaviour of a letter is a property of the letter and not only of the typeface ([Unicode Consortium](#)). Some letter sequences, most famously the sequence of lam and alif, are written as obligatory ligatures. In digital typography these behaviours are implemented by shaping, which assigns forms to letters from their context, and by glyph substitution features that realise the isolated, initial, medial and final forms and ligatures ([OpenType](#)). Finally, many Arabic letters share a skeletal shape and differ in the number and placement of dots, which, as noted in [Section 6](#), is the system that the detour mimics.

These facts have three consequences for the present project. The first is that for scripts of this kind the display function cannot be a map from stored characters to shapes applied one character at a time, and the model of [Section 5](#) has to be read in its general form, in which display is a function of the string. The second is that the joining structure offers resources for design that the Latin alphabet lacks, and also constraints: a hybrid that draws Arabic letters with Latin-inspired outlines has to decide whether and how joining is retained, and a hybrid that gives Latin letters connecting strokes has changed the orthographic behaviour of the Latin script and not merely its decoration. The third is a matter of respect for a living writing system. Imitation of Arabic lettering in other scripts has often been produced by designers who cannot read the script, with results that are illegible to those who can, and a

project that treats Arabic as a resource and not as an ornament should involve Arabic readers in the design and testing of any hybrid forms.

7.3 When substitution and rendering commute

The context dependence of Arabic gives a precise sense to a question that is trivial in the Latin case. In the letterwise model it does not matter whether one first substitutes characters and then renders them with a glyph map g , or renders the original text with the glyph map $g \circ \pi$: the glyph sequences are identical. Under shaping this need not hold. Let each letter x have a joining class $J(x) \in \{D, R, N\}$, dual-joining, right-joining or non-joining, and let the form of the letter at position i of a string X be a function of $J(x_{i-1})$, $J(x_i)$ and $J(x_{i+1})$ only: a letter is joined to its predecessor if the predecessor is dual-joining and the letter is dual- or right-joining, and joined to its successor if the letter is dual-joining and the successor is dual- or right-joining, and its form is isolated, final, initial or medial according as it is joined to neither, the predecessor only, the successor only, or both. Write $\text{sh}_g(X)$ for the glyph sequence obtained by giving position i the glyph $g(x_i, \text{form}_i(X))$.

Proposition 7.1 (Commuting under shaping). *If π is a substitution that preserves joining classes, $J(\pi(x)) = J(x)$ for all x , then for every string X and every glyph map g , $\text{sh}_g(\pi(X)) = \text{sh}_{g \circ \pi}(X)$, where $(g \circ \pi)(x, f) = g(\pi(x), f)$.*

Proof. The form of position i depends only on the joining classes of x_{i-1} , x_i and x_{i+1} , and replacing each letter by its image under π leaves these classes unchanged by hypothesis, so $\text{form}_i(\pi(X)) = \text{form}_i(X)$. Then position i of $\text{sh}_g(\pi(X))$ is $g(\pi(x_i), \text{form}_i(X)) = (g \circ \pi)(x_i, \text{form}_i(X))$, which is position i of $\text{sh}_{g \circ \pi}(X)$. $\square$

The hypothesis is not idle. A substitution that exchanges a dual-joining letter such as nun with a right-joining letter such as ra, two shapes that share a descending curve, changes the joining of the letter that follows, so that a letter that was in final form becomes isolated, and the two routes to a page produce different pages. For Latin letters the question does not arise, since all letters belong to the single non-joining class and every substitution therefore preserves it. The proposition shows that a visual substitution for Arabic is not merely a matter of finding resembling shapes: the joining classes must be matched, or the equivalence of the two routes to a page is lost. It also illustrates a general moral about the formal treatment of editions, developed in Section 10: the property that two routes give the same page is a property to be proved under stated hypotheses, and the hypotheses carry design content.

7.4 Interpolation between Latin and Standard Galactic

Standard Galactic has no distinction of case, so the Latin-to-Galactic correspondence sends both the upper-case and the lower-case form of a letter to one shape, and the map on forms is not injective. Under the stored-text principle this does no harm, since the stored text retains the case and the glyph assignment merely ignores it, but a reader of the final script loses

capitalisation as a cue and the design should note this. The shape spaces of the two alphabets are otherwise unrelated, and straight interpolation of unrelated outlines yields forms that belong to neither. A more promising route is a sequence of hand-designed keyframes, with Latin at one end, Standard Galactic at the other, and between them one or two hybrid forms each of which shares a feature with each neighbour, and with interpolation only between adjacent keyframes. The route is then piecewise, the margins can be checked at each keyframe by the build-time test described in Section 10, and the hybrid forms can be tested separately for recognisability. Because the endpoint shapes are new, the interference problem discussed at the end of Section 6 does not arise in the same form, and the empirical comparison of the two kinds of endpoint is among the first experiments proposed in Section 9.

8 Editable Source and the Provenance of Editions

8.1 Why the source is released

The essays of the project are released with their source in the public domain, and the release serves two purposes that the preceding sections make precise. The first is correction. A text that is public and editable can be repaired by anyone who finds an error in an argument or a citation, and under the stored-text principle of Section 5 a correction to the linguistic text propagates to every edition, however presented. The second purpose is variation. Because the substitution, the routes and the schedule are separate from the text, any reader can print the same essay in a cipher of their choosing, in a typeface of their choosing, on a schedule of their choosing, and the project's own ciphers are then a small sample of a space that readers can extend. A text released in this way is not a single publication but a family of editions, and each edition is simultaneously a way of reading the argument and an experimental stimulus in the study of how readers acquire scripts.

8.2 Components and their identification

For the family to be usable as a research instrument, an edition has to record what it says and how it presents it, and the record must separate the components so that they can be varied one at a time. A suitable organisation keeps five things separately identifiable: the linguistic text with its correction history; the substitution table; the glyph designs, that is the master shapes and the routes between them; the schedule, including its parameters and any random seed; and the builder, the program that turns these into a typeset page. An edition is then described by a manifest that names each component and its version. Appendix D gives a schema. The point of a manifest is to make three kinds of change distinguishable that would otherwise be confused. An *authorial revision* changes the linguistic text along its own lineage. An *experimental edition* changes the substitution, the routes, the schedule or the builder while leaving the text untouched, and should be labelled as such, so that no one mistakes a variant presentation for the author's. A *derivative text* changes the words and is a different work with its own lineage, which may cite its parent.

The builder should be deterministic: given the manifest it must produce identical output, so that an edition can be reproduced from its description and so that observations made on one reader's copy can be attached to a precise stimulus. Two subtleties arise. Corrections to the text change the number of occurrences of letters and therefore shift the positions at which an exposure-driven schedule advances, so the schedule has to be recomputed from the corrected text and the edition receives a new identifier; no harm results, but a record of results has to say which version of the text was read. And a pseudo-random component of a schedule, if used, needs a recorded seed.

8.3 Licensing and attribution

A public-domain dedication covers what its author dedicates. Glyph designs drawn from scratch can be dedicated in the same way as the text. Outlines derived from existing typefaces carry the licences of those typefaces, many of which permit modification and redistribution under conditions, and a project that wishes its fonts to be freely reusable should either design its masters from scratch or begin from fonts whose licences allow it. The same applies to hybrids with other writing systems: Arabic letter forms are an inheritance of a living tradition and not a stock of free material, and the credit for shapes derived from particular designers belongs with the designers. A concrete case in the project's own font collection is the monospaced Standard Galactic face, which is a modified version of the open-licensed NovaMono typeface. The SIL Open Font License under which NovaMono is released requires every modified version to remain under that licence and forbids a modified version from keeping the reserved font name, so the derivative can be neither public-domain nor GPL, and it should be renamed. A single repository-wide licence is therefore the wrong instrument for a collection that mixes original and borrowed outlines, and a per-file statement, as in the manifest of Appendix D, is the right one.

8.4 Prior work and provenance in the project's tools

The constructions of this essay were preceded by working tools in the project's public repositories, and the relation between the tools and the formal treatment is worth stating, both to credit the earlier work and to mark where the essay goes beyond it.¹

The font collection. The collection contains several Standard Galactic faces with different purposes and not a single font: a large face of about fourteen thousand glyphs (`Sga-Regular`), a cursive face, a monospaced face with box-drawing characters, a small face that displays a substitution cipher (the LingoJam Cipher font, thirty-five glyphs), a logical-symbol face used for a symbol substitution, and Systada. The essay's phrase "Standard Galactic" therefore names a family, and any claim about a route to Standard Galactic is a claim about a

¹The repository is `standardgalactic/alphabet`, branch `core`, whose `fonts/` directory and top level hold the files named here. The statements in this subsection about file contents were checked against the files; statements about how the fonts were made are the author's and are not recorded in the repository.

route to one of these faces. Systada is the terminal-friendly face. It was built for a terminal that could reach only the early, ASCII part of the character range, so that the stored text had to stay inside a small code range while the font supplied the shapes. This is the stored-text principle of Section 5 arising from a hardware constraint and not from a design preference: the font was made small because the stored alphabet was small. Scripts in the repository (`transform.py` and its later versions) take Systada as input and write ASCII-restricted variants, including versions with box and scaling adjustments.

Glyph provenance. A number of the Systada glyphs were drawn by hand with a marker on paper, scanned, and converted to a font with the Calligraphr web service, whose template sheets and a screenshot are kept in the repository; the remaining glyphs were taken from a free typeface. The repository does not record which glyphs fall under which source, nor the name and licence of the free typeface. This matters for the discussion of licensing above, because the hand-drawn glyphs can be dedicated to the public domain by their author, while the borrowed glyphs carry the licence of their source, and the repository’s declared licence applies to both only to the extent that the source licence permits. The manifest of Appendix D has a licence field for this reason, and for these fonts it should be filled glyph-group by glyph-group once the source of the borrowed outlines has been identified. Until then the honest statement is that the provenance of part of the font is undocumented.

The interpolation notebook. The repository contains a notebook that renders each letter of the Latin alphabet, upper and lower case, in two faces (`Sga-Regular` and a `Nova Mono` face), blends the two raster images with weight $\alpha = k/10$ for $k = 0, \dots, 10$, and saves one eleven-frame animation per letter; a final cell sets a phrase by chaining the animations. Later cells adjust the relative size and vertical offset of the second face until the two renderings align. In the vocabulary of this essay, the notebook is a drift with a single global schedule, since every letter has the same α at the same step, and its route for each letter is the straight line in image space between two rasterised shapes, a cross-dissolve and not an interpolation of outlines. Three consequences follow. First, because each Latin letter is sent to its own Galactic shape and not exchanged with another letter, the collision of Section 6 does not arise; it would arise in the same notebook if the pair of faces were replaced by a pair of exchanged letters, as the cross-dissolve figure of that section shows for B and P. Second, a cross-dissolve keeps both shapes visible at every intermediate step, which is a different perceptual regime from the outline routes of Section 6, and whether it supports recognition better or worse than an outline route is an empirical question of the kind listed in Section 9. Third, the global schedule is the case of the schedule vector $\alpha(p)$ in which all components are equal, and the exposure analysis of Section 6 shows why that case is costly for rare letters: a common α moves every letter at the same pace whatever its frequency. The notebook is thus the uniform baseline against which per-letter schedules can be compared. One cell of the notebook also attempts to use Unicode mathematical italic letters as a second endpoint; the character ranges used there do not correspond to the italic block, so that experiment did not test what it was meant to.

Symbol substitution. A short script in the repository maps the lowercase Latin letters to symbols associated with the logical-symbol face, folding case on the way. The mapping is arbitrary and not visual, so it is a useful contrast to the visual alphabets of Section 2. It also illustrates the round-trip condition of that section in a concrete way: several of its target symbols, among them the plus sign, the hyphen, the slash, the percent sign, the exclamation mark and the ampersand, are ordinary punctuation, so a text that already contains them cannot be decoded unambiguously, and the capital letters are lost by case folding. Recovery of the original is guaranteed only for texts over the lowercase alphabet with no other occurrences of the target symbols, which is exactly the type-level restriction on the domain of encoding stated in Section 10.

8.5 Accessibility

An edition that drifts is a poor medium for a reader who needs the text in a form they can already read, and this is a reason for the stored-text principle and for always providing a plain edition alongside any experimental one. The reasons are practical as much as ethical. Screen readers and speech synthesis read the stored text, and in a compensated cipher they read the cipher. Search and copying work on stored text. Readers with low vision or dyslexia are among the people for whom some intermediate forms may be harder than for others, and an experimental programme that wishes to say anything about acquisition has to include them or to state plainly that it does not.

9 An Empirical Programme

The constructions above would be of limited interest if they could not be tested, and the central proposal, that a book can teach its own script, is exactly the kind of statement that is easy to find plausible and hard to confirm. This section specifies what the tests would be. It reports no results and is not a pilot. Its purpose is to state the hypotheses, conditions and measures in enough detail that a result, including a negative one, could be interpreted, and that the same design could be run by others with their own material.

9.1 Hypotheses

The first group of hypotheses concerns whether acquisition happens at all. **H1** says that readers of a drifting edition recognise the final-stage forms in isolation more accurately than readers who encounter the same forms for the first time, so that the book has taught something; this is the minimal claim, and it distinguishes acquisition from contextual guessing. **H2** says that this recognition transfers, in that readers of the drifting edition read a fresh passage, not part of the book, set in the final-stage script with higher accuracy and speed than control readers. The second group concerns the comparison of ways of presenting the transition. **H3** says that a drifting presentation yields better transfer than presenting the final script from the outset, with or without an alphabet table, at equal reading time. **H4** says that a drift whose

schedule is driven by exposure, in the sense of Section 6, yields better transfer for rare letters than a drift whose schedule is uniform in position. **H5** says that routes matter: for the same endpoints and the same schedule, routes with designed intermediate keyframes preserve comprehension better during the transition than routes that interpolate outlines linearly. The third group concerns durability and the limits of the claim. **H6** says that recognition survives a delay of days or weeks, with the possibility that the pattern of retention depends on how exposures were spaced (Cepeda et al., 2006; Bjork, 1994). **H7** says that the receptive gains of H1 and H2 are not accompanied by comparable gains in production, which would confirm that the book teaches the first level of knowing a sign and not the highest. None of these is assumed to be true, and the design is intended to produce interpretable results if all are false.

9.2 Conditions

The conditions to be compared fall naturally into a small number of arms, each defined by a manifest in the sense of Section 8. A *plain* arm presents the text in ordinary letters and serves as a baseline for comprehension and reading speed. A *static* arm presents the final script from the first page without assistance, and a *static with table* arm presents it with an alphabet table. An *abrupt* arm switches from ordinary to final forms at a single point. The drifting arms differ along the dimensions identified earlier: uniform against exposure-driven schedules, linear against keyframe routes, reciprocal pairs with and without detour marks, new-shape endpoints against swapped-shape endpoints, and monotone drifts against cycles. A complete factorial design is far too large, and the programme proceeds in stages, with the earliest experiments comparing a few arms on a single alphabet and later ones varying one dimension at a time from the best-performing arm.

9.3 Measures

The primary measures are chosen to correspond to the levels of knowing a sign distinguished in Section 4. Comprehension is measured by questions about the content of the passage, scored independently of the script, and reading speed by words per unit time, which together show whether a presentation preserves the ability to follow an argument, the criterion by which any transition must be judged. Recognition in isolation is measured by presenting single letter forms, drawn from the final alphabet, and recording accuracy and response time, by letter, so that frequency effects can be examined. Transfer is measured by a fresh passage in the final script. Retention is measured by repeating these tests after a delay. Production is measured separately, by asking participants to write a given word in the final script. Subjective ratings of difficulty are collected but are secondary, since readers are poor judges of what they have learned. For the transitions themselves, performance can also be recorded as a function of position in the book, which would show where in a drift comprehension dips, and this connects the empirical record to the margin and novelty-load quantities of Section 6.

9.4 Design considerations

Several features of the setting complicate inference. Participants differ in prior exposure: a reader who knows Standard Galactic from games, or who reads Arabic, begins from a different place than one who does not, and these must be measured and balanced. The specific alphabet confounds the effect of presentation, so the substitution should be counterbalanced across participants, with different pairings used for different readers, or else the study estimates the difficulty of one alphabet and not of a method. The same text should not be seen by the same participant in more than one arm, so that memory for content is not confused with learning of script. Participants should be assigned to arms at random, the exposures each letter receives should be recorded so that the dependence on exposure can be analysed directly, and analysis should use models that treat participants and letters as sampled units. The hypotheses, the exclusion rules and the analysis plan should be recorded before data are collected. The effect sizes are not known in advance, so the size of the first studies would be set from pilots, and the design is meant to report intervals and not merely tests of significance.

There is also precedent for the kind of effect sought. Readers asked to read transformed typography show substantial practice effects, and some of the benefit appears to persist long after the practice (Kolars, 1976). The work on skilled reading offers a framework for what a measure of reading speed and comprehension can show about the processes underlying it (Rayner et al., 2012). A careful reading of the history of the Initial Teaching Alphabet, in which a transitional script was designed to give way to the conventional one, would supply cautions about transfer from a special script back to the usual one (Downing, 1967).

9.5 The question to be answered

The programme can be summarised in one question: which routes preserve comprehension while producing recognition that survives outside the particular book that taught it? The two clauses can come apart, and the distinction between them is the reason for distinguishing contextual assistance from acquisition. A book may be perfectly readable throughout its drift because the reader is guessing from context, and may leave nothing behind. A book may teach the forms but at a cost to comprehension so great that no one finishes it. The target is a route on which both conditions hold, and the empirical programme exists to find out whether such a route exists, for which alphabets, and for which readers. If it does not, that is a result, and the essay's claim that the arbitrariness of the sign is separable from the constraints on its acquisition is unaffected by it, since that claim is precisely about how much of the difficulty lies in the assignment and how much in the structure of the shapes and the reader.

10 Formal Interpretations: Transformation, Typing, and Local Readability

The preceding sections have used mathematical language where it was needed, and this section collects the structure into three formal perspectives, category theory (Mac Lane, 1998), type theory (Martin-Löf, 1984; Pierce, 2002) and sheaf theory (Mac Lane and Moerdijk, 1992), each of which answers a different question about editions. Category theory distinguishes transformations and their compositions, and in particular asks when two ways of reaching the same page, or of travelling between the same endpoint alphabets, are equivalent. Type theory distinguishes the kinds of entity being transformed, and asks what a publishing system should be forced to state about an edition before it is allowed to build it. Sheaf theory asks whether interpretations acquired in different passages of a book remain compatible as letters change, at different rates, from one passage to the next. These are presented as distinct perspectives with explicit definitions, and a claim is made only where it is proved; where a perspective is used as an organising metaphor, that is said.

10.1 Category-theoretic interpretation: transformation and composition

Alphabets, strings and rendering. Let **Alph** be the category whose objects are finite alphabets and whose morphisms are functions between them. A substitution is an automorphism $\pi : \Sigma \rightarrow \Sigma$ in this category, and a reciprocal substitution is an automorphism with $\pi \circ \pi = \text{id}_\Sigma$, an element of order at most two in the group $\text{Aut}(\Sigma)$. The assignment $A \mapsto A^*$ of the set of finite strings, with the extension f^* of a letter map f to strings, is a functor from **Alph** to the category of monoids, the free-monoid functor. A glyph assignment is a morphism $g : A \rightarrow \mathcal{G}$, where $\mathcal{G}$ is regarded as an alphabet of shapes, and rendering a string is the morphism $g^* : A^* \rightarrow \mathcal{G}^*$. The statement that rendering a substituted text is the same as rendering the original text with a reassigned font is now a statement about composites.

Proposition 10.1 (Letterwise commuting). *For every substitution $\pi : A \rightarrow A$, glyph assignment $g : A \rightarrow \mathcal{G}$ and string $X \in A^*$, $g^*(\pi^*(X)) = (g \circ \pi)^*(X)$.*

Proof. Functoriality of $(\cdot)^*$ gives $(g \circ \pi)^* = g^* \circ \pi^*$, and applying both sides to X gives the claim. $\square$

This is the composition-level content of the equivalence stated informally in Section 5, and its proof is a single line because letterwise rendering is a functor. The line also shows exactly where the statement fails to generalise. A shaping function sh_g of the kind used for Arabic is a function $A^* \rightarrow \mathcal{G}^*$ that is not a monoid homomorphism, because the forms at the boundary of a concatenation XY depend on both X and Y , so $\text{sh}_g(XY) \neq \text{sh}_g(X) \text{sh}_g(Y)$ in general. The category of monoids is therefore too coarse for contextual display, and one must work in a category whose morphisms are functions between sets of strings, with Proposition 7.1 supplying a hypothesis, preservation of joining classes, under which the square commutes anyway.

Presentations and equivalence of presentations. Define a *presentation* to be a pair (X, g) of a string $X \in A^*$ and a glyph assignment $g : A \rightarrow \mathcal{G}$, and let a morphism $f : (X, g) \rightarrow (X', g')$ be a letter map $f : A \rightarrow A'$ with $f^*(X) = X'$ and $g' \circ f = g$. Then $g'^*(X') = g'^* f^*(X) = (g' \circ f)^*(X) = g^*(X)$, so the displayed page is invariant along morphisms. The two presentations of the example of Section 5, the stored text PUB drawn with a font that draws P as B and B as P, and the stored text BUP drawn with the ordinary font, are isomorphic: the isomorphism is the substitution π itself, which is why one may be said to *be* the other seen through a change of stored representation. Compensated and uncompensated configurations are in this sense isomorphic presentations, and the stored-text principle is a recommendation about which object of an isomorphism class to store.

Three notions of sameness. Three different relations between presentations have been in use and should not be conflated. The first is *equality of displays*, $g^*(X) = g'^*(X')$, which is preserved by morphisms of presentations. The second is *recoverability*, the existence of a decoder $d : \mathcal{G}^* \rightarrow A^*$ with $d \circ g^* = \text{id}$. This holds if and only if g is injective on A , which is to say if and only if no two letters share a glyph, and Proposition 6.1 can be rephrased as the statement that a linear exchange of two letters necessarily passes through a presentation whose rendering map is not injective. The third is *equivalence of reader performance*, the claim that readers do equally well on two editions in comprehension, recognition or transfer. This is not a relation definable in the categories above, because it is a relation between outcomes of experiments and not between the structures that the morphisms preserve. A successful character conversion establishes the first relation and, when the glyph assignment is injective, the second, and it establishes none of the psychological ones by itself.

Routes as paths, and why topology does not distinguish them. Consider the shape space $\mathcal{G}$ with its topology. Routes are paths, and paths in a space form a category, in which objects are points, morphisms are paths, and composition is concatenation, associative up to reparametrisation. A cycle of presentation, such as Latin to hybrid to Standard Galactic and back, is the concatenation of routes into a loop. Two paths with the same endpoints are different morphisms of this category, and become identified only after passing to homotopy classes. In a vector space of shapes every two paths with the same endpoints are homotopic, so homotopy cannot distinguish routes of the kind discussed in Section 6, and the relevant structure is the *constrained* space of whole-alphabet configurations,

$$\mathcal{C}_\theta = \{(y_x)_{x \in A} \in \mathcal{G}^A : d(y_x, y_{x'}) \geq \theta \text{ for all } x \neq x'\},$$

in which distinct letters are kept at least θ apart. The joint route of an edition is a path in $\mathcal{C}_\theta$ from the source configuration to the destination configuration, and two such routes are equivalent as routes if they are homotopic *within* $\mathcal{C}_\theta$, which is a strictly finer relation than homotopy in $\mathcal{G}$. For two letters and a shape space that is a plane, the space of pairs at distance at least $\theta > 0$ is homotopy equivalent to a circle, so there are inequivalent ways of exchanging two letters, and an exchange followed by an exchange back traces the generating loop if

the two exchanges turn the same way and a contractible loop if they turn in opposite ways. In shape spaces of three or more dimensions the corresponding spaces of distinct points are simply connected and every route is equivalent to every other, so the differences that remain are metric: whether a path with margin at least θ exists at all, which depends on the size of θ relative to the spacing of the shapes and, in a bounded region of legible forms, can fail, and how large the margin is along the path. The existence of a margin- θ route is the formal form of the claim that a drift preserving distinctness exists. Different routes between the same endpoints are thus not interchangeable, for three reasons that correspond to three equivalence relations of increasing strength: they can differ in whether they exist within $\mathcal{C}_\theta$, in their margins, and in the reader performance they produce, and of these the first two can be settled by mathematics and computation and the third only by experiment.

10.2 Type-theoretic interpretation: typing the publishing system

A weakness of informal descriptions of editions is that different kinds of object are all called text. A type-theoretic treatment declares the kinds and forces conversions to state their inputs and outputs. Let `Alphabet` be the type of finite alphabets and, for $A : \text{Alphabet}$, let $\text{Text}(A)$ be the type of finite lists over A . Linguistic text, encoded character sequences, glyph sequences and laid-out pages are given different types, and a conversion is a function between them: for a substitution π , $\text{encode}_\pi : \text{Text}(A) \rightarrow \text{Enc}_\pi(A)$ and $\text{decode}_\pi : \text{Enc}_\pi(A) \rightarrow \text{Text}(A)$, where $\text{Enc}_\pi(A)$ is a copy of $\text{Text}(A)$ indexed by the substitution that produced it, so that decoding with a substitution other than the one used for encoding is a type error and not a silent corruption. A rendering function `render` takes an element of $\text{Enc}_\pi(A)$ and a font for A to a glyph sequence and does not count as a translation; a translation is a function between texts of different languages and has no input in common with it. A cipher does not count as a proof of readability, because readability is not a type that any of these functions inhabits.

Dependent structure. Stronger contracts need dependent types. An *edition* can be declared as a dependent record, a Σ -type, with fields

- an alphabet $A : \text{Alphabet}$ and a text $T : \text{Text}(A)$;
- a substitution $\pi : A \rightarrow A$ together with a proof that π is a bijection, and, if reciprocal editions are required, a proof that $\pi \circ \pi = \text{id}$;
- routes $\rho_x : [0, 1] \rightarrow \mathcal{G}$ for each $x : A$, with endpoints as declared;
- a schedule α , which in the exposure-driven case is a function of the text, $\alpha : (T : \text{Text}(A)) \rightarrow \text{Schedule}(T)$, so that the edition type genuinely depends on the text it is made from; and
- a *margin certificate*: a proof that for the declared routes and schedule the margin m of Section 6 is at least the declared θ .

The last field is a finite computation, since the margin is a minimum over the positions at which the schedule changes and the pairs of letters, and for linear and piecewise-linear routes

the minimum can be computed exactly. A builder that demands the certificate will refuse to typeset an edition containing the collision of Proposition 6.1, and the repair, a detour or a keyframe, becomes a condition of construction and not an afterthought. This is what Section 8 meant by checking the hypotheses at build time. That the schedule is a function of the text also gives a typed account of a point made there: when a correction changes the text, the schedule type changes with it, so an edition made from the corrected text is a different object, and a record of observations must name which.

Proposition 10.2 (Round trip). *For every bijection $\pi : A \rightarrow A$ and every $X : \text{Text}(A)$, $\text{decode}_\pi(\text{encode}_\pi(X)) = X$, where decode_π applies π^{-1} letter by letter; and if $\pi \circ \pi = \text{id}$ then $\pi^{-1} = \pi$, so that the underlying maps on lists of encode_π and decode_π coincide once the index π is forgotten. (As typed functions they are not equal, since they have different domains and codomains.)*

Proof. By induction on the list X . For the empty list both sides are empty. If $X = x :: X'$, then $\text{decode}_\pi(\text{encode}_\pi(x :: X')) = \pi^{-1}(\pi(x)) :: \text{decode}_\pi(\text{encode}_\pi(X')) = x :: X'$ by the hypothesis of the induction. If $\pi \circ \pi = \text{id}$ then $\pi^{-1} = \pi$ and the two letterwise maps on lists coincide. $\square$

Three kinds of claim. The proposition is a claim of the first kind: a theorem about the encoder, inhabited by a proof that is a finite derivation. The margin certificate is a claim of a second kind, decidable by computation on the data of the edition. A third kind concerns readers. A statement about readers can itself be represented in type theory, for instance as a proposition about a recorded data set of outcomes, or as a statistical statement over a model of those outcomes, so the difficulty is not that such statements are inexpressible. It is that neither the proof about the encoder nor the margin certificate inhabits a type of that kind: no derivation about the structure of an edition establishes that readers acquire its script. The statement is supported instead by *evidence*, records with provenance (the participants, the arm, the measures, the version of the edition read), and the typing discipline consists in keeping that evidence attached to the edition it concerns and distinct from the theorems and certificates, so that none of the three substitutes for another. This is the type-theoretic form of the discipline of Table 1.

10.3 Sheaf-theoretic interpretation: local readability and global convention

Setting. A reader meets a book locally. Glyphs are recognised in one word, inferred in another passage, and met in modified form several chapters later. To make precise what it means for such local interpretations to cohere, let $X = \{1, \dots, N\}$ be the positions of the book, let Γ be the finite set of distinct shapes that appear, each intermediate shape counted as a shape in its own right, and let $D : X \rightarrow \Gamma$ be the display, so that $D(p)$ is the shape at position p . For $U \subseteq X$ write $\Gamma(U) = D(U)$ for the set of shapes appearing in U . A *key* on U is a function $k : \Gamma(U) \rightarrow \Sigma$, a reader's working assignment of letters to the shapes occurring there. Let $\text{Key}(U)$ be the set of keys on U , and for $V \subseteq U$ let the restriction be $k \mapsto k|_{\Gamma(V)}$. This makes Key a presheaf on the subsets of X , every subset being regarded as open. A

window cover is a family $\mathcal{U} = \{U_i\}$ of subsets covering X , for instance runs of consecutive positions that overlap. The sheaf condition asks that every family of keys $k_i \in \text{Key}(U_i)$ that agree on overlaps, $k_i|_{\Gamma(U_i \cap U_j)} = k_j|_{\Gamma(U_i \cap U_j)}$, glue to a unique key on $\bigcup_i U_i$ restricting to each k_i .

Two observations are needed before the main results. The shapes of an overlap, $\Gamma(U_i \cap U_j)$, can be a proper subset of the shapes common to the two windows, $\Gamma(U_i) \cap \Gamma(U_j)$, because a shape can occur in both windows without occurring in the part they share. Agreement on overlaps is thus weaker than agreement on shared shapes. And the sheaf condition is a condition on *all* compatible families of keys, not only on the true one, so it can fail even for a book in which every shape has a definite meaning.

When the condition holds. For a shape $\gamma \in \Gamma$ let G_γ be the graph whose vertices are the windows containing γ , with an edge between two windows when their intersection contains a position at which γ is displayed.

Proposition 10.3 (Gluing criterion). (i) *Gluing is unique whenever it exists.* (ii) *If every graph G_γ is connected, then every compatible family of keys glues.* (iii) *If some G_γ is disconnected and $|\Sigma| \geq 2$, there is a compatible family of keys that does not glue, so Key is not a sheaf for $\mathcal{U}$.*

Proof. (i) Every shape of $\bigcup_i \Gamma(U_i)$ lies in some $\Gamma(U_i)$, and the glued key must agree there with k_i . (ii) Given a compatible family, define $k(\gamma) = k_i(\gamma)$ for any window U_i containing γ . If two windows are joined by an edge of G_γ , their intersection contains an occurrence of γ , so compatibility gives equal values; by connectedness the value does not depend on the window chosen, so k is well defined, and by construction it restricts to each k_i . (iii) Suppose G_γ has a nonempty proper union C_1 of components, with complement C_2 among the windows containing γ , and choose distinct letters $\sigma_1, \sigma_2 \in \Sigma$ and any fixed $\sigma_0 \in \Sigma$. Let $k_i(\gamma) = \sigma_1$ for windows in C_1 and σ_2 for windows in C_2 , and let every other shape and every window not containing γ be assigned σ_0 . If two windows overlap in a position displaying γ , they are adjacent in G_γ , hence in the same component and given the same value. If they do not, their overlap does not display γ , and the keys agree on all other shapes. So the family is compatible; but any glued key would need $k(\gamma) = \sigma_1$ and $k(\gamma) = \sigma_2$. $\square$

The criterion has an immediate interpretation for rare letters. A shape that occurs only a few times, far apart, and never in the overlap of two windows has a disconnected graph, so that a reader's evidence about it in one passage constrains nothing about its meaning in another. For such shapes the global key is a convention held by the reader, not a consequence of local compatibility, and this is a formal counterpart of the observation in Section 6 that the rarest letters are the ones whose acquisition is least supported by the book.

Two kinds of failure. The presheaf Key over all subsets of X fails the sheaf condition in a way that has nothing to do with drift. Cover X by singletons: the overlaps are empty, so every family of keys is compatible, and if a shape occurs at two positions the two singleton keys may assign it different letters, so no global key restricts to both. This happens for an ordinary

static script, and Proposition 10.3(iii) records it as the disconnection of the occurrence graph G_γ for the cover chosen. The sheaf condition is therefore a property of a presheaf together with a cover, and for covers fine enough to separate occurrences it is not expected to hold. The coherence statement below is deliberately different. It concerns one specific family, the true keys, and asks whether that family extends to a global key. For a static script it does, for *every* cover, since each shape has one meaning throughout. For a swap-ending drift it does not for *any* cover, since the meanings differ at two positions and every position lies in some window. The first failure is an artefact of how the book is cut into windows; the second is an obstruction in the edition itself, and it is only the second that bears on whether a reader can hold one convention for the whole book.

True local keys and global coherence. Suppose each window is small enough that every shape in it has a single meaning there, and let $\kappa_U \in \text{Key}(U)$ be the *true key* of a window U , the assignment of each displayed shape to the letter it stands for.

Proposition 10.4 (Coherence). *The family of true keys (κ_{U_i}) is compatible on overlaps. It glues to a key on X if and only if every shape has the same meaning at all of its occurrences.*

Proof. If γ is displayed in $U_i \cap U_j$, then at that position γ has a single meaning, and by the assumption on windows $\kappa_{U_i}(\gamma)$ and $\kappa_{U_j}(\gamma)$ are both this meaning; so the true keys agree on overlaps. If every shape has one meaning throughout, the function assigning that meaning is a key on X restricting to each κ_{U_i} . Conversely, if a shape has different meanings at two positions, then taking windows containing them gives $\kappa_{U_i}(\gamma) \neq \kappa_{U_j}(\gamma)$ and no single k can restrict to both. $\square$

The proposition applies directly to the two endpoint designs of Section 6. A drift to new shapes, as from Latin to Standard Galactic along routes whose intermediate shapes are distinct from the endpoints and from one another, gives each shape a single meaning throughout, and the true keys glue to a global key. A drift that ends in swapped shapes does not. In the exchange of letters a and b with detour routes, the pure shape A means a at the start of the book and b at the end, while in the middle, where both letters are in intermediate forms, the pure shape A does not occur at all. If the windows are shorter than the middle stretch, no window contains both an early and a late occurrence of A and none of the overlaps contains A at all, so the true keys of early and late windows are compatible with those of the middle windows and nevertheless have no common extension. The book is *locally coherent and globally incoherent*: no two adjacent windows disagree about any shape, so the rate at which meanings change between neighbouring windows is zero, and yet the global assignment does not exist. This formalises a feature of the swap design that the margin and exposure analyses of Section 6 do not capture. Coherence is a property of the whole book, not of its neighbourhoods, and what the reader must acquire in the swap design is a key that varies with the phase of the book and that is, at the level of single windows, invisible.

Refinement by context. The incoherence can be removed by refining the shapes. Let $\Gamma' = \Gamma \times \{\text{early}, \text{late}\}$ and let the display carry a phase marker, so that the pure shape A in the early part is a different element of Γ' from the pure shape A in the late part. With this refinement every element of Γ' has a single meaning and the true keys glue by Proposition 10.4. In practice the phase marker would be a cue that the reader can track: a colour, a running footer, a typographic device that changes at the transition. This is a design suggestion that follows from the formal analysis and is untested, and it amounts to making the context in which a shape is read part of what the reader learns about the shape.

Contextual and conventional readability. The key presheaf concerns conventions, assignments of letters to shapes that hold across a window. Reading by context is more flexible: a reader may decode the same shape as one letter in one word and another letter in the next, because the sentence makes the difference clear. This is captured by the presheaf Tok of *token decodings*, in which a section over U is a function $d : U \rightarrow \Sigma$ assigning a letter to each position. It is a sheaf for every cover, since functions on positions glue trivially. Let $\text{Tok}_{\text{adm}}(U)$ be the set of token decodings under which every complete word lying inside U is a word of the lexicon, and let $\text{Dec}(U) \subseteq \text{Key}(U)$ be the set of keys k for which $k \circ D$ lies in $\text{Tok}_{\text{adm}}(U)$. Restricting an admissible decoding to a smaller window keeps it admissible, since fewer words lie inside the smaller window, so both are presheaves and Dec is a subpresheaf of Key . A decoding d comes from a key exactly when it is constant on positions that display the same shape, $D(p) = D(q) \Rightarrow d(p) = d(q)$. Call a window *contextually readable* if $\text{Tok}_{\text{adm}}(U) \neq \emptyset$ and *conventionally readable* if $\text{Dec}(U) \neq \emptyset$. Conventional readability implies contextual readability and the converse fails, and the failure is informative: a window in which a reading is available through context although no single assignment of letters to shapes will serve is one in which the text can be read only by decoding the same shape differently at different positions, with no stable convention available to the reader. The collision of Proposition 6.1 produces exactly such windows, since a shared glyph for two letters can be read correctly only by decoding the same shape differently at different positions. The distinction is the formal counterpart, on the side of the edition, of the difference between understanding a passage through contextual assistance and holding a convention stable enough to read unfamiliar material; whether a reader holds such a convention is a further matter, taken up next.

Availability, acquisition and comprehension. The definitions above are about what an edition makes available, and three things that are easily run together need to be kept apart. A key is *available* on a window U if $\text{Dec}(U) \neq \emptyset$: some assignment of letters to the shapes of U turns every complete word of U into a word of the lexicon. This is a property of the edition and the lexicon and says nothing about any reader. A key is *acquired* on U by a reader R if the assignment κ_R that the reader actually holds, a partial function from shapes to letters whose domain and content are empirical quantities, is defined on $\Gamma(U)$ and its restriction lies in $\text{Dec}(U)$. Acquisition is a state of the reader at a time, measurable in the way recognition in isolation is measured in Section 9. *Comprehension* is a third thing, an outcome such as accuracy on questions about the passage or reading speed, which can occur without

acquisition, through contextual decoding, and can fail with it, through inattention or the difficulty of the content. Acquisition on U implies availability on U , since the acquired key is then an admissible one, and nothing else among the three follows formally from anything else. In particular, exposure does not appear in the model as a premise from which acquisition is derived. The connection between exposure and acquisition, that a shape seen often enough comes to be held, is a hypothesis of the empirical programme and is not a theorem about editions.

Transfer. The same machinery states what transfer to a new book requires. Let the learned book have final-stage true key k_S , defined on the shapes Γ_S that it displayed in final form, and let a new book display shapes Γ' from the same script. The key glues across the two books, in the sense of Proposition 10.3, exactly on $\Gamma_S \cap \Gamma'$, and the shapes of Γ' outside Γ_S have no section to extend. Transfer to unfamiliar material is therefore limited, at most, to shapes that the learned book displayed in final form, a necessary condition for any reader to hold their keys on the strength of that book. How many of those keys a given reader actually acquired is the empirical question of the preceding paragraph, and by Section 6 the shapes displayed often enough to make acquisition plausible are predominantly the frequent ones.

The formal question. The perspective yields a precise form of the question with which the programme began: under what conditions can changing local representations support a coherent, transferable interpretation? Within the model the conditions on the edition are four. The displays of small windows must give each shape a single meaning (local consistency). Either no shape changes meaning across the book, as in a new-shape endpoint with collision-free routes, or the changes of meaning are indexed by a context that the reader can track (global coherence). Every window must have an admissible key available, $\text{Dec}(U) \neq \emptyset$ (local availability). And for transfer the shapes of the target material must lie among those the learned book displayed in final form (coverage). These conditions are necessary for the formal coherence of the interpretation, and they are not sufficient for psychological acquisition: that the sheaf-theoretic structure of an edition is coherent says that acquisition is not ruled out by the structure, and the empirical programme of Section 9 is what could show that it occurs.

10.4 What the three perspectives jointly provide

Table 8 sets the three perspectives side by side. Together they distinguish the composition of transformations (what produces what, and which routes to a page agree), the kinds of entity being transformed (what must be stated and checked before an edition is built), and the compatibility of interpretations across changing contexts (what a reader must hold, and what the book can and cannot ensure about it). Each leaves something unsaid. The category perspective is silent on readers, the type perspective separates the three kinds of claim without evaluating any of the third kind, and the sheaf perspective is a model of interpretation whose psychological adequacy is not established by its mathematics.

Perspective	Objects	Main results here	Silent on
Category	alphabets, presentations, routes as paths	letterwise commuting; presentations isomorphic under substitution; routes distinguished by margin, not topology; failure of commuting under shaping	reader performance
Type	texts, encodings, glyphs, editions with certificates	indexed encodings; round trip; margin certificate checked at build time; three kinds of claim	whether certificates suffice for legibility
Sheaf	keys on windows, token decodings	gluing criterion; coherence of true keys; swap endings locally coherent but not global; contextual versus conventional readability	whether coherent keys are acquired

Table 8: The three formal perspectives.

11 Conclusion

The argument of the essay can be stated in a few sentences. A writing system combines a conventional assignment of marks to linguistic units with a perceptual structure among the marks, and the second is not arbitrary even when the first is. Visual substitution alphabets make the difference tangible, because their assignments are interchangeable in principle and their resemblances are not, and the design of such an alphabet is an optimisation whose answer depends on the typeface, on the notion of resemblance used, and on how many letters are changed. A font that separates stored characters from displayed shapes, together with routes between shapes and a schedule over the positions of a book, turns a single text into a family of editions, and the drifting edition, in which the script changes while the text is read, is the member of the family that asks whether a reader can acquire a script receptively through sustained reading.

The analysis of such editions produced results of three kinds. Some are mathematical: the sizes of the spaces of alphabets, the collapse of linear exchange and its removal by a detour, the avoidance of collision by cycles of length three, the conservation of exposure and the lower bound it gives on the length of a drift, the failure of substitution and shaping to commute unless joining classes are preserved, and the gluing criterion for local keys. Some are computational and exactly as strong as their inputs: the comparison of pairings across four typefaces and the simulation of schedules in a fluid model. And some are hypotheses about readers, which the essay leaves as hypotheses and for which it specifies an empirical programme. The pattern across all of them is that the structure of the problem is governed by things one might first overlook: the dependence of resemblance on the typeface, the inevitability of a shared glyph when two letters exchange along a straight path, the

dominance of the rarest letter in the length of a drift, and the possibility of a book that is locally coherent at every point and has no global key.

Several lines of work follow. The weights of the pairing problem should be replaced by measured confusion data, and the optimisation redone. The detour marks and keyframe routes should be designed and tested for recognisability. The exposure model should be extended from its fluid approximation to the actual random sequence of letters in a particular text, with the schedule computed from that text. The coherence conditions of Section 10 suggest an experiment with and without a visible phase marker for swap-ending drifts. And the formal treatment should be pushed where it is thinnest, in the relation between the sheaf-theoretic model of conventions and measurable properties of readers. Because the source and the builder are public, each of these can be pursued by others on their own texts and compared, and the editions that result will be part of the evidence for or against the proposition with which the essay began: that a publication can be designed as a passage between representational conventions, and that the reader of such a publication can be carried across.

A Counting Derivations

Involutions. Let $I(n)$ be the number of involutions on $\{1, \dots, n\}$. Classify an involution by what it does to the element n . If n is fixed, the rest is an involution on $n - 1$ elements, giving $I(n - 1)$ possibilities. If n is exchanged with $j \neq n$, there are $n - 1$ choices of j and the remaining $n - 2$ elements carry an involution, giving $(n - 1)I(n - 2)$. Hence $I(n) = I(n - 1) + (n - 1)I(n - 2)$ with $I(0) = I(1) = 1$. Iterating to $n = 26$ gives $I(26) = 532,985,208,200,576$.

Involutions with exactly k exchanges. Choose the $2k$ moved letters in $\binom{26}{2k}$ ways. A fixed-point-free involution on $2k$ letters is a perfect matching; the number of these is $(2k - 1)!!$, since the smallest letter can be matched with any of $2k - 1$ others and the remaining $2k - 2$ letters matched in $(2k - 3)!!$ ways. Summing $\binom{26}{2k}(2k - 1)!!$ over $k = 0, \dots, 13$ reproduces $I(26)$ exactly, which was checked by computer. The values for $k = 4, 6, 7, 9, 11, 12$, omitted from Table 2, are 164,038,875; 100,391,791,500; 1,305,093,289,500; 53,835,098,191,875; 205,552,193,096,250; and 102,776,096,548,125. The largest single term is at $k = 11$, so most reciprocal alphabets change 22 of the 26 letters.

Derangements. The number $D(n)$ of permutations of n elements without fixed points satisfies $D(n) = (n - 1)(D(n - 1) + D(n - 2))$ with $D(0) = 1$ and $D(1) = 0$, and $D(26) = 148,362,637,348,470,135,821,287,825$. The ratio $26!/I(26) \approx 7.57 \times 10^{11}$.

Font	Optimal reciprocal pairing (overlap in parentheses)
DejaVu Sans	AW (0.32), BS (0.59), CG (0.54), DU (0.54), EL (0.44), FP (0.56), HN (0.63), IJ (0.72), KR (0.38), MV (0.23), OQ (0.92), TY (0.40), XZ (0.41)
Liberation Sans	AW (0.27), BS (0.59), CG (0.62), DU (0.58), EK (0.43), FL (0.29), HN (0.60), IT (0.59), JZ (0.32), MV (0.41), OQ (0.85), PR (0.62), XY (0.31)
DejaVu Serif	AW (0.31), BR (0.64), CG (0.66), DU (0.48), EF (0.65), HN (0.47), IJ (0.56), KX (0.49), LP (0.49), MV (0.28), OQ (0.90), SZ (0.28), TY (0.43)
TeX Gyre Pagella	AW (0.33), BS (0.44), CG (0.58), DU (0.53), EL (0.71), FP (0.56), HN (0.38), IJ (0.72), KR (0.46), MV (0.26), OQ (0.81), TY (0.34), XZ (0.33)

Table 9: Maximum-weight reciprocal pairings of the 26 capitals under the pixel-overlap measure, with the overlap of each pair in parentheses.

B Optimal Pairings by Typeface

Pairs present in all four optimal pairings are A/W, C/G, D/U, H/N, M/V and O/Q. B/S, I/J and T/Y occur in three of four, E/L, F/P, K/R and X/Z in two, and the remaining pairs in one. The weight matrices were computed from bitmaps rendered with PIL at an em size of 160 pixels on a 200×200 canvas, with the baseline at 75 per cent of the canvas height and glyphs centred by advance width, and the pairing was found with the maximum-weight matching routine of NetworkX with maximum cardinality enforced.

C Code

The programs below produced the computed results of the essay. They are reproduced so that the results can be rerun with other fonts, other weights and other parameters.

Glyph overlap and optimal reciprocal pairing.

```
import sys, json, itertools
import numpy as np
from PIL import Image, ImageDraw, ImageFont
import networkx as nx

FONTS = {
    "DejaVu Sans": "/usr/share/fonts/truetype/dejavu/DejaVuSans.ttf",
    "Liberation Sans": "/usr/share/fonts/truetype/liberation/LiberationSans-Regular.ttf",
    "DejaVu Serif": "/usr/share/fonts/truetype/dejavu/DejaVuSerif.ttf",
    "TeX Gyre Pagella": "/usr/share/texmf/fonts/opentype/public/tex-gyre/texgyrepagella-regular.otf",
}
LET = [chr(c) for c in range(ord('A'), ord('Z')+1)]
SIZE = 160 # em size in px
```

```

CANVAS = 200

def render(font, ch):
    im = Image.new("L", (CANVAS, CANVAS), 0)
    d = ImageDraw.Draw(im)
    # anchor on baseline, centred horizontally by advance width
    adv = font.getlength(ch)
    d.text(((CANVAS-adv)/2, CANVAS*0.75), ch, font=font, fill=255, anchor="ls")
    a = np.array(im) > 127
    return a

def iou(a, b):
    inter = np.logical_and(a, b).sum()
    uni = np.logical_or(a, b).sum()
    return inter/uni if uni else 0.0

out = {}
for name, path in FONTS.items():
    try:
        font = ImageFont.truetype(path, SIZE)
    except Exception as e:
        print("skip", name, e); continue
    bm = {c: render(font, c) for c in LET}
    W = {}
    for a, b in itertools.combinations(LET, 2):
        W[(a,b)] = iou(bm[a], bm[b])
    G = nx.Graph()
    for (a,b), w in W.items():
        G.add_edge(a, b, weight=w)
    M = nx.max_weight_matching(G, maxcardinality=True)
    pairs = sorted(tuple(sorted(p)) for p in M)
    total = sum(W[p] for p in pairs)
    top = sorted(W.items(), key=lambda kv: -kv[1])[:12]
    # greedy baseline: repeatedly take best remaining pair
    used=set(); greedy=[]
    for (a,b), w in sorted(W.items(), key=lambda kv: -kv[1]):
        if a in used or b in used: continue
        greedy.append((a,b,w)); used.update([a,b])
    gtotal = sum(w for _, w in greedy)
    named = {f"{a}{b}": W[(a,b)] for a,b in [("B","P"), ("U","V"), ("C","G"), ("E",
"F"), ("M","N"), ("O","Q"), ("I","L"), ("C","K"), ("M","W"), ("A","H")]}
    out[name] = dict(pairs=[(a,b,W[(a,b)]) for a,b in pairs], total=total,
                    mean=total/13, greedy_total=gtotal, top=[(a,b,w) for (a,b),
w in top], named=named)
    print("==", name)
    print(" optimal reciprocal derangement total IoU = %.3f (mean %.3f per pair)"
    " % (total, total/13))
    print(" greedy total = %.3f" % gtotal)
    print(" pairs:", " ".join(f"{a}{b}:{W[(a,b)]:.2f}" for a,b in pairs))
    print(" top12:", " ".join(f"{a}{b}:{w:.2f}" for (a,b),w in top))
    print(" named:", " ".join(f"{k}:{v:.2f}" for k,v in named.items()))
json.dump(out, open("out/glyph_similarity.json", "w"), indent=1)

```

Fluid-model schedules.

```
import json
LEW = dict(zip("etaoinshrdlcumwfgypbvjkxqz",
[12.702, 9.056, 8.167, 7.507, 6.966, 6.749, 6.327, 6.094, 5.987, 4.253, 4.025,
  2.782, 2.758, 2.406, 2.360, 2.228, 2.015, 1.974, 1.929, 1.492, 0.978,
  0.772, 0.153, 0.150, 0.095, 0.074]))
tot=sum(LEW.values()); p={k:v/tot for k,v in LEW.items()}
n=26
def simulate(order, K, nu):
    """Fluid model: letter l is in transition for K/p_l letters, adding load
    p_l.
    Start each letter, in the given order, at the earliest time at which total
    load
    stays <= nu for its whole interval. Returns makespan (letters) and
    intervals."""
    ivs=[] # (start,end,load)
    def load_ok(s,e,pl):
        pts=sorted({s}|{a for a,b,_ in ivs if s<=a<e}|{b for a,b,_ in ivs if s<
        b<=e})
        for t in pts:
            cur=sum(1 for a,b,l in ivs if a<=t<b)
            if cur+pl>nu+1e-12: return False
        return True
    out={}
    for ch in order:
        d=K/p[ch]; pl=p[ch]
        cands=sorted({0.0}|{b for a,b,_ in ivs})
        for s in cands:
            if load_ok(s,s+d,pl):
                ivs.append((s,s+d,pl)); out[ch]=(s,s+d); break
        else:
            raise RuntimeError("no slot for "+ch)
    return max(e for s,e in out.values()), out
def main():
    freq_desc=sorted(p,key=lambda c:-p[c]); freq_asc=freq_desc[::-1]
    res={}
    print("lower bounds: n*K/nu and K/pmin")
    for K in (20,40,80):
        for nu in (0.15,0.25,0.40):
            lb1=n*K/nu; lb2=K/min(p.values())
            h_asc,_=simulate(freq_asc,K,nu); h_desc,_=simulate(freq_desc,K,nu)
            print("K=%2d nu=%.2f LB nK/nu=%8.0f LB K/pmin=%8.0f rare-first
            =%8.0f frequent-first=%8.0f (letters)" % (K,nu,lb1,lb2,h_asc,h_desc))
            res[f"{K}_{nu}"]=dict(K=K,nu=nu,lb_capacity=lb1,lb_rare=lb2,
            rare_first=h_asc,frequent_first=h_desc)
    json.dump(res,open("out/schedule_sim.json","w"),indent=1)
    # detail for K=40, nu=0.2 rare-first: starts
    h,iv=simulate(freq_asc,40,0.25)
    print("detail K=40 nu=0.25 rare-first:")
    for ch in freq_asc: print(ch, "start %8.0f end %8.0f" % iv[ch])
    json.dump({ch:iv[ch] for ch in freq_asc},open('out/gantt_K40_nu25.json','w')
    )
```

```
if __name__=='__main__':
    main()
```

Exposure-driven staging. The script below assigns to each occurrence of an alphabet letter a stage from 0 to S , advancing one stage after each K further occurrences once a letter-specific delay has passed. Run on the text of the working conversation from which this essay was developed (about 9,700 letters) with $K = 5$ and $S = 3$ and the pairs B/P, U/V, C/G, E/F, M/N and O/Q, it reports the character positions at which each of the twelve letters first reaches its final stage: E at 152, O at 247, N at 409, C at 444, U at 496, P at 602, B at 744, M at 762, G at 921, V at 1647, F at 2449 and Q at 8134. The order follows letter frequency, and the rare letter Q, with 21 occurrences in the text, reaches its final stage only after 8,134 characters.

```
"""Exposure-driven drift staging for a visual substitution alphabet.
Stage 0 shows a letter in its familiar form; stage S shows its substituted
partner.
A letter advances one stage after every K further occurrences, once its own
delay
(counted in occurrences of that letter) has passed."""
import sys
PAIRS = ["BP", "UV", "CG", "EF", "MN", "OQ"]
PI = {}
for a, b in PAIRS:
    PI[a] = b; PI[b] = a

def stages(text, K, S, delay):
    seen, out = {}, []
    for ch in text:
        up = ch.upper()
        if up in PI:
            c = seen.get(up, 0); seen[up] = c + 1
            st = max(0, min(S, (c - delay.get(up, 0)) // K))
            out.append((ch, st))
        else:
            out.append((ch, None))
    return out

def show(tokens, S):
    """Discrete rendering: final stage shows the partner letter; earlier stages
    the original."""
    res = []
    for ch, st in tokens:
        if st == S:
            p = PI[ch.upper()]
            res.append(p if ch.isupper() else p.lower())
        else:
            res.append(ch)
    return "".join(res)
```

```

def involution_ok():
    return all(PI[PI[x]] == x for x in PI)

if __name__ == "__main__":
    assert involution_ok()
    text = open(sys.argv[1], encoding="utf-8").read()
    K, S = int(sys.argv[2]), int(sys.argv[3])
    delay = {ch: int(sys.argv[4]) * i for i, ch in enumerate("BPUVCGEFMNOQ")}
    if len(sys.argv) > 4 else {}
    tok = stages(text, K, S, delay)
    print(show(tok, S)[:0], end="")
    # report, per letter, the character position where stage S is first reached
    first = {}
    for pos, (ch, st) in enumerate(tok):
        if st == S and ch.upper() not in first:
            first[ch.upper()] = pos
    print("first position at final stage:", dict(sorted(first.items(), key=
lambda kv: kv[1])))
    print("tokens at final stage in last 1000 chars: %d of %d letters in the
alphabet" %
        (sum(1 for _, s in tok[-1000:] if s == S), sum(1 for _, s in tok
[-1000:] if s is not None)))

```

D A Schema for Edition Manifests

The following schema lists the fields an edition manifest would carry. It is a specification and not a record of any existing edition, and the values shown are symbolic.

```

edition:
  id:          <hash of the fields below>
  kind:        authorial | experimental | derivative
  text:
    work:      <identifier of the work>
    version:   <version of the linguistic text>
    parent:    <identifier of the preceding version, if any>
  alphabet:    <identifier, e.g. Latin capitals and lower case>
  substitution:
    table:     <identifier of the bijection pi>
    reciprocal: true | false
  glyphs:
    masters:   <identifiers of the master outlines>
    routes:    <per-letter route definitions, endpoints and keyframes>
    license:   <licence of the outlines>
  schedule:
    kind:      uniform | exposure-driven | letter-by-letter | cyclic
    parameters: {K, S, delays D_l, novelty cap nu}
    seed:      <recorded if any randomness is used>
  margin:
    threshold: theta
    certificate: <result of the build-time check>

```

builder:	
version:	<version of the edition builder>
evidence:	<links to observations made on this edition, if any>

References

- Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. In J. Metcalfe and A. Shimamura (Eds.), *Metacognition: Knowing about Knowing*, pp. 185–205. MIT Press.
- Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., and Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3), 354–380.
- Changizi, M. A., Zhang, Q., Ye, H., and Shimojo, S. (2006). The structures of letters and symbols throughout human history are selected to match those found in objects in natural scenes. *The American Naturalist*, 167(5), E117–E139.
- Daniels, P. T. and Bright, W. (Eds.) (1996). *The World's Writing Systems*. Oxford University Press.
- Dehaene, S. (2009). *Reading in the Brain*. Viking.
- Dehaene, S. and Cohen, L. (2007). Cultural recycling of cortical maps. *Neuron*, 56(2), 384–398.
- Dehaene, S., Nakamura, K., Jobert, A., Kuroki, C., Ogawa, S., and Cohen, L. (2010). Why do children make mirror errors in reading? Neural correlates of mirror invariance in the visual word form area. *NeuroImage*, 49(2), 1837–1848.
- Downing, J. (1967). *Evaluating the Initial Teaching Alphabet*. Cassell.
- Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17, 449–467.
- Gelb, I. J. (1952). *A Study of Writing*. University of Chicago Press.
- Gibson, E. J. (1969). *Principles of Perceptual Learning and Development*. Appleton-Century-Crofts.
- Gibson, E. J., Gibson, J. J., Pick, A. D., and Osser, H. (1962). A developmental study of the discrimination of letter-like forms. *Journal of Comparative and Physiological Psychology*, 55(6), 897–906.
- Gilmore, G. C., Hersh, H., Caramazza, A., and Griffin, J. (1979). Multidimensional letter similarity derived from recognition errors. *Perception & Psychophysics*, 25(5), 425–431.
- Knuth, D. E. (1982). The concept of a meta-font. *Visible Language*, 16(1), 3–27.
- Kolers, P. A. (1976). Reading a year later. *Journal of Experimental Psychology: Human Learning and Memory*, 2(5), 554–565.
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 83–97.
- Lewand, R. E. (2000). *Cryptological Mathematics*. Mathematical Association of America.
- Lewis, G. (1999). *The Turkish Language Reform: A Catastrophic Success*. Oxford University Press.

- Mac Lane, S. (1998). *Categories for the Working Mathematician*, 2nd edition. Springer.
- Mac Lane, S. and Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.
- Martin-Löf, P. (1984). *Intuitionistic Type Theory*. Bibliopolis.
- Nation, I. S. P. (2001). *Learning Vocabulary in Another Language*. Cambridge University Press.
- Microsoft Corporation. *OpenType Specification*, including the font variations introduced in version 1.8. Microsoft Typography documentation.
- Palfreyman, D. and Al Khalil, M. (2003). “A funky language for teenzz to use”: Representing Gulf Arabic in instant messaging. *Journal of Computer-Mediated Communication*, 9(1).
- Peirce, C. S. (1931–1958). *Collected Papers of Charles Sanders Peirce*. Harvard University Press.
- Pelli, D. G., Burns, C. W., Farell, B., and Moore-Page, D. C. (2006). Feature detection and letter identification. *Vision Research*, 46(28), 4646–4674.
- Pierce, B. C. (2002). *Types and Programming Languages*. MIT Press.
- Rayner, K., Pollatsek, A., Ashby, J., and Clifton, C. (2012). *Psychology of Reading*, 2nd edition. Psychology Press.
- de Saussure, F. (1983). *Course in General Linguistics*, translated by R. Harris. Duckworth. (Original work published 1916.)
- Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell System Technical Journal*, 30(1), 50–64.
- Townsend, J. T. (1971). Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics*, 9(1), 40–50.
- Unicode Consortium. *The Unicode Standard*, chapter on Arabic. Unicode Consortium.