Fiat Lingua

Title: mklang

Author: Isoraḳatheð

MS Date: 09-11-2017

FL Date: 06-01-2018

FL Number: FL-000051-00

Citation: Isoraḳatheð. 2017. "mklang." FL-000051-00, *Fiat Lingua*, <http://fiatlingua.org>. Web. 01 June 2018.

Fiat Lingua is produced and maintained by the Language Creation Society (LCS). For more information about the LCS, visit http://www.conlang.org/

☆△· A notice of scope ☆

In this annex we will describe how a language is created from conception to translatability, which is the end goal of languages both Purple and of the Jedic Pasaru. Since there are many guides to creating languages in general already, the annex will primarily focus on specifics of that differ from creating a language in a more "mainstream" (i.e. naturalistic) manner.

As such, it is recommended that the reader does not solely depend on the Annex to create his own language. In particular, the Annex assumes the reader already knows about certain basic and not-so-basic details about linguistics, including but not limited to:-

- Phonology / Phonological terms
- Grammatical and syntax terms
- A little bit of glossing rules
- IPA

these bits of knowledge
But also might help:
Maths, physics, programming and a little bit of geography

However in some cases basic information may be repeated to fit with a theme or because they are modified.

★△ Items required/Tools to Display

~~The~~ One of the most important part of the language creation process is to ~~ensure that a consistent~~ have a way to display it. To this end, ~~a series of~~ the following items are used:

- A4 notebooks - as many as required.
- Pens of many colours. At least 5 but 20 is ~~typically~~ better.
- ~~used~~ Other bits of stationery, such as a ruler or a calculator

A note about colours

Colours are used to emphasise, link and mark certain phrases or drawings. The default ~~text~~ colour ~~in~~ is its own; header text gets its own colour as well; all others are either used in ~~its own~~ an idiosyncratic manner or in the following groups:

Red, Yellow, Green	Orange, Sky, Lime, Pink	
Bronze, Brown	Gold, Silver	Purple, Lavender,
*also used as annotations		Violet, Magenta, Salmon
Black, Grey, Silver	Green, Teal, Lime, Aqua	

★△ Computer tools

There are a handful of tools that are purpose-written to help with digitising and organising pages on a notebook. There are also general-purpose tools that assist to this end. Below are the two that are used in such.

imagemagick

This tool is a general purpose image editor. It usually ~~cuts~~ does basic image processing such as whitening, cropping and rotating, but can also create complete PDF archives of a book when it is complete and scanned.

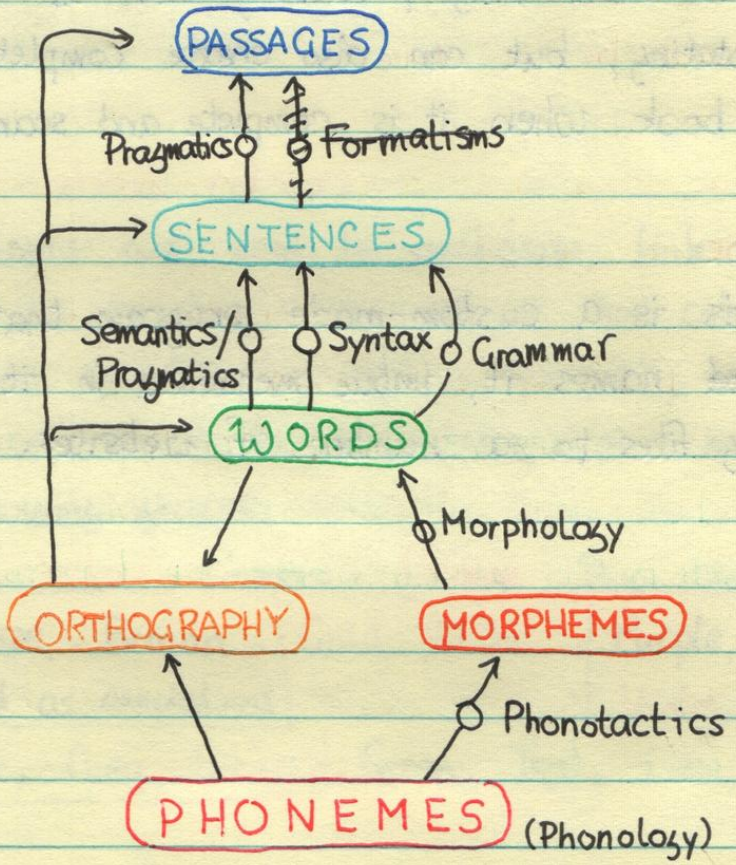
bocproc

This is a custom-made program that takes raw JPEG file, ~~and~~ names it, imbue metadata in it and posts the resulting files to a number of websites. It is written in Lisp.

★△ Anatomy of a Language I

A language, in essense, is composed of a number of features, small bits of "interesting factoids" that come together to form the language with a bit of glue.

Features are placed (or assigned) into one of the slots in the language skeletons below. All languages follow this rough structure, though details may differ.



Skeleton Slots

★△ Anatomy of a Language II

~~Lang~~ The language skeleton is a general form for all languages. Features tend to attach themselves to it, though apart from perhaps the top two parts, the bottom part and the orthography, the overall form of the languages skeleton remains roughly constant.

In detail:

Phonemes/Phonology: Usually the standard ~~IPA~~ phonology ~~like in~~ as describable using IPA, but a different substrate can be chosen (this would be a feature).

Orthography: this/these is/are alternative ways of writing out some or all of words, morphemes or phonemes. There can be more than one or none at all.

Morphemes/Morphology: Combining phonemes together using phonotactics to form basic units of meaning.

Words: Combining morphemes using morphology to form larger units. These words are sometimes identical to morphemes but the separation is usually present for traditional reasons.

Sentences: Using various semantics, syntax and grammar, ~~form~~ words come together to form sentences. These sentences need not be tied to phonology or length. ~~They are also part of~~ This group is also handles phrases.

Passages: The goal of any language here is to translate large chunks of text. To make this final step pragmatics regulate how the sentences are combined.

★△ Anatomy of a feature (I)

Features are the lifeblood of every language. And appropriately they ~~have~~ come in all shapes and sizes.

Regular vs. Irregular Features

Most features come in the form of a single sentence that can have words filled in, mad-libs style, to form an elevator pitch of sorts. For example:

"Word order is determined by $PREDICATE/~~EQUA~~COMPARABLE"
"Have (0, 1, an obnoxious amount of) $THING"
"$COMPLETELYUNRELATEDQUALITY controls $LINGUISTICDETAIL"
"Have $THING but don't have $COMMONCONSEQUENT"
(or vice versa)
"Merge (or split) $THING and $ANOTHERTHING"
"Violate $LINGUISTICUNIVERSAL"

The last of these ~~are~~ is vague enough that any number of features can fit. These usually introduce enough creativity so as to avoid being "mad-libs"-ified - these are the irregular features.

Placement

Features are "placed" into the skeleton. Any number can fit into any slot.

★△ Anatomy of a feature II

An important quality of features is that they must belong to one system or another. Not only does it keep the language thematically similar, it also allows for easy traversal between the features.

Systems, briefly

Though typically a vague notion and described implicitly rather than explicitly, a system handles parts of the language by creating a structure. For instance, a ~~*~~ word complex or a list of sentence types can comprise a system.

In this sense, a system is like an irregular feature that consists of more features.

Dependence and Mutex

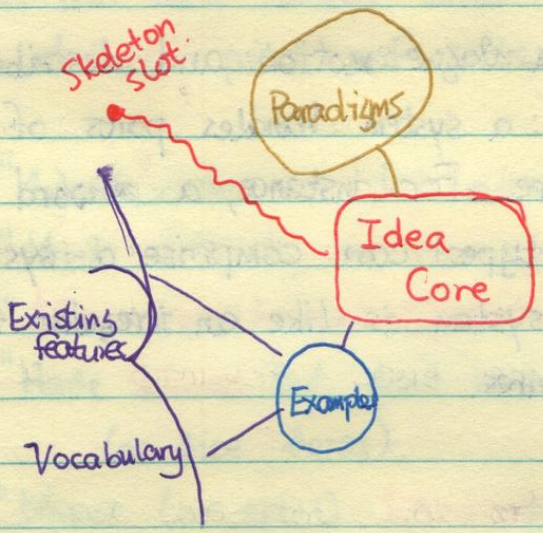
Some features depend on others to make sense. Others cannot exist in the same language as still others. This creates dependencies and mutices.

Both items create more complex situations. Dependencies increase the complexity of one language. Mutices create more languages by demanding space to accommodate them.

Anatomy of a feature III

★△Appearance and Scope

A feature tends to be pretty short and compact. Most features are simply just one line, followed by some supporting data to integrate it to the rest of the language, and then some examples. Diagrammatically:



The resulting feature would ~~tee~~ then fill 1 page of A4 paper, though smaller features can do with ½ page and particularly nasty paradigms can fill more than one — but always an integer ~~n~~number of pages.

This limits the scope of a feature to bite-sized chunks, which might seem debilitating but fortunately more features can depend one already-established ones to increase complexity. It also allows for old languages to be revisited without having to dig out the entire notebook, and a feature can in principle be reworked without disturbing the rest of the language (much).

★△ Features ~~in~~ accumulation

Inspiration for features can appear anywhere, anytime, even if language seems distantly-related at first (or at best). This is mostly passive and ~~involves~~ sticking different concepts into regular features.

Often themes are a part of the accumulation but this is not necessary. Sometimes features are just those that are mutexed from other languages. ~~Sue~~

Once about 4 or 5 features are accumulated amongst the lower levels of the skeleton a language is ready to be created. However this is not necessarily time to create it because there are other languages to tend to at the time. Note that making languages also creates features as a byproduct.

* * *

Accumulated features are either put into a waiting list and/or future pages in a book. This is informal and typically comes in the form of a title. As so it turns out virtually all features can make it through this process unchanged and survives.

★△ Features and Diachronics

Diachronics, or rather "descendent languages", are ~~us~~created somewhat differently from regular languages.

- Diachronics, in addition to the normal exchange of ~~features~~ grammar, phonology, &c., also works in the feature level.
- Most diachronics are in fact governed by features coming and going, not necessarily the ~~more~~ less "systematic" sound changes or similar.
- Except sound changes, which has a significant unstructured component.
- Features themselves come and go as ~~it~~ they please; they are sufficiently abstract that their appearance and disappearance does not pose too much of a problem.

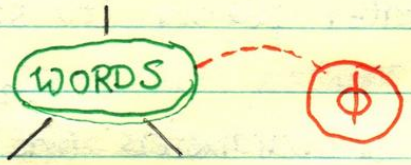
The ~~the~~ distinction between regular and diachronic languages makes for a somewhat different relation than conventional language construction would have.

★△ Features in action I

Consider this feature.

Φ = "ALL names ($THINGS) are verbs ($GRAMMATICAL CATEGORY)." (Now claimed by ⓧ△ i)

Obviously this single feature is enough - just enough - to fill up one single A4 page. This should largely consist of introducing the idea of a name, and what gets named.



This attachment to the skeleton orients it to the overall language.

Since Φ is fairly complex - that is, it affects fairly deeply into its "core" - it has a great potential to evolve into a system. The follow-on features include:

"Emphasise actions" "`essence_extract'" IRR

"Nonlinear orthography" UNREGULARISABLE

The latter two being linked also to other root nodes of the skeleton.

(The Weaving I)

★△ Features in Action IIa

With the features established (in this particular portion of the language anyway) the ~~s~~thing to do now is to give it enough dress to put it into the ~~lang~~language. This particular ~~s~~action ~~is~~ has no name, but for the nonce we'll call it weaving.

In the case of ϕ "weaving" involves describing, in approximately this order:

1> The feature, stated plainly (to some degree of "plain" anyway!)

2> ~~The~~ Any values or parameters that are to be set ~~and~~ for this language.

3> The ~~tang~~ paradigm —
a list of tables or similar that express the ~~s~~new feature in simpler (lower on the skeleton) terms.

4> Examples, if there's room.

"Approximately", is specified to allow for some of that artistic flexibility, and indeed ~~can~~ to allow for the slightly different set of challenge each feature demands.

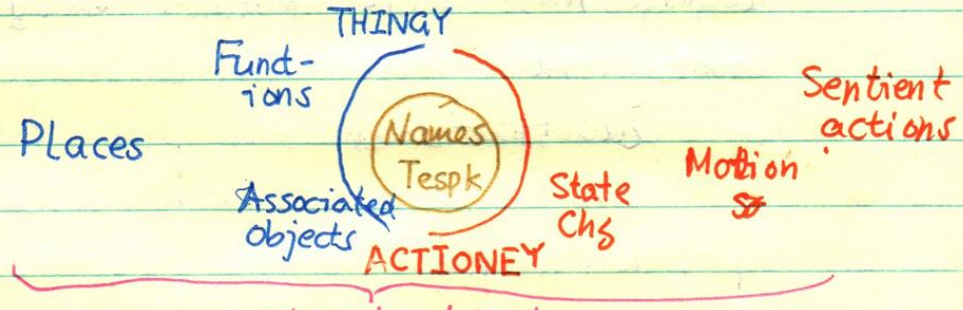
A sketch of describing ϕ which may not appear as-is in the final description of ϕ in Fs Otm is seen at right.

(The weaving II)

★△ Features in Action II b

1> In Fs Otm, names don't typically get assigned to things, they get assigned to ~~actuirons~~ actions.

2>



(Partially 3>) Places

Followed by explanations of terms

(Partially) 3> Items and ~~actions~~ that receive names, also indicate that there's some other generalisation procedure is going on. ...

Explain the procedure

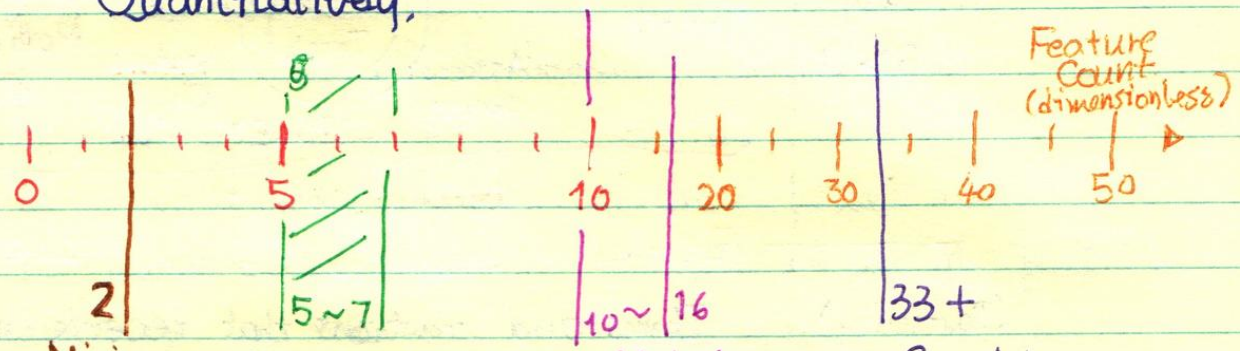
4> Examples of names

Five or six examples of each type as described above maybe 3~4 if there turns out to be quite numerous

count of features

features ~~Integers~~ A language can of course have any number of (w/in $\in \mathbb{INT}^0$, anyway). ~~The~~ In practice, older languages have more features, enough that one easily loses count.

Quantitatively,



Accumulation, of course, continues regardless of whether there is a language to stick it in, or mutex considerations. There isn't much control granted to the accumulation process, so there's correspondingly little to talk about.

★ △ Getting a bit radical I

The thing about the language skeleton is that it is correspondingly abstract (and vague). This helps it adapt to languages with a completely different "format", so's to speak, and yet still retain more-or-less relevant orientation for languages that don't even begin to resemble human.

For example, what is labelled PHONOLOGY doesn't have to involve mouths, articulations &c., but can be any system that ultimately expresses the language in one tangible medium or another. Similarly WORDS only require that the items ~~carry~~ in its purview carry some "atom" of "meaning", ~~meaning~~ with terms left for the language to define. Or not - there's reasonable defaults for almost all of the qualities.

* * *

(A notion of abstraction)

Every level in the language skeleton represents an abstraction of the previous: using the combinations of objects in the latter to create some quality of the former. This idea ~~is~~ can eventually spur on a few extra changes to the skeleton itself, as we'll see later.

~~A~~The abstractions sometimes may have its own names but those are largely obscured by conventional linguistics terminology.

★△ Getting a bit radical II

The thing about the language skeleton is that it is ~~correspondingly~~ sufficiently patterned enough that abstractions can easily be made. It is therefore extremely tempting to try and change the skeleton to allow for ever the more flexible languages.

~~Turns~~ While no existing language has gone that far the prospect ~~if~~ is definitely appealing. In particular, the upper levels of ~~each~~ the language ~~tree~~ skeleton is much easily ~~t~~changed than the lower ones, where the more influence of reality is still very much present.

One major issue with the changing of the language skeleton however is that conventional linguistic terms, already belaboured under the exoticness of the existing system, risks total breakdown and requires either a new crop of terms or a redefinition, neither of which is appealing. ~~I~~ This complicates presentation and especially translation, as it is unclear even in generalities what the output would look like.

Nevertheless this is an exciting places for languages to expand to, and the outlook ~~to~~ is promising. All it takes to bring it to life is a particularly esoteric feature that somehow refuses to fit in the skeleton.

△○ Location

★Languages & the World I

Languages are typically tied to a specific location; the speakers play a more secondary role, ~~in~~ usually by transmission through time. This may be somewhat inaccurate but the tradeoff is:-

- A more cohesive, geography & map-oriented world - making the map the territory as much as possible
- Deëmphasis of ~~a~~ small collectives and details in favour of a ~~more~~ broader, open view

Indeed, since languages take up a lot of space - especially if they're not diachronically related — the ~~re~~ geography-oriented approach makes much more sense, since the unrelatedness means we're essentially making entire language families. Otherwise, ~~tra~~related languages do take up much less space (or rather take ~~very little~~ overlapped space from the "undiachronicsed" languages) And back-projected languages may also have ~~unrelated~~ different ~~language~~ ranges.

Dialects eat up spaces inside ~~other~~ the respective language zones and cities.

Dialects

★○△ Languages ~~of~~ & the World II

Dialects are created by two primary systems —

- Areal changes Changes to a whole ~~host of~~ locale, without regard to the actual language underneath.
- Cultural changes basically everything else.

Changes can also be characterised as sporadic, (random "minor" differences), paradigmic (change in some parameter of a feature, or an entry in the paradigm) and featural. (an entire feature is added, substituted or deleted)

As in real life, what counts as standard and what dialectal is left to the prestige group to use, but from a construction standpoint it is essentially a random selection.

Diachronics affect dialects too, but they're largely in sync with dialects of the same language. The "largely" is what causes languages to drift apart over time.

(Partially)

Time-symmetric Diachronics

A simple model of diachronics is to assume changes that occur in one direction is equally likely to occur in another when considered on a feature level. This allows languages to be derived forwards and backwards in time, ~~though of course~~

Of course, Language change is not actually time-symmetric, but this doesn't change things too drastically as the main asymmetry - divergence of languages, ~~ie~~ collections of features - concerns configurations which may be time-asymmetric even as the laws governing them are time-symmetric - viz. the Laws of physics and entropy. Nevertheless care should be taken when deriving backwards to have features "merge" wherever possible.

Any language ~~is~~ can be a protolanguage; the name is given after the fact, as with conventional ~~languages~~ linguistics. Future languages must derive from past, accessible language, however ~~&~~ here "accessible" can be any language in the past and same language ~~&~~ family, due to the unique linguistic environment around the ~~region~~. J.Pasaru.

★△ Naturalism and the lack thereof

It should be at least somewhat obvious that there is a significant difference between ~~reg~~ what happens in a natural language (governed by conventional linguistics) and a language in the 1-Pasaru (governed by rules alluded to in this booklet). Facts, for once, are not on the side of the latter. But, in ~~ma~~this case this shouldn't matter. There are several reasons:

1▷ Plenty have gone into being naturalistic.

2▷ It's easier to make facts up than to keep up with real ones. This frees up effort to explore more exotic ideas and turns research into an ideas bucket rather than a rulebook.

3▷ It makes for a more mathematical approach to language. Highly abstract maths tends to invent and use completely new systems.

While of course the new set of rules would probably not be ~~entirely~~ useful in describing human languages it does make the alienness of the languages a little bit more apparent.

Utilisation or Unexpected use of ★△ Other Subjects

A consistent and recurring theme when making a language is to use and misuse ideas and concepts from other bits of knowledge ~~in a language~~ in the Library. It honestly doesn't matter what particular field one draws inspiration from, ~~Mos~~ but the less they have to do with sound directly the better. Examples include:-

GEOGRAPHY MATHEMATICS PHYSICS COOKERY
RAIL, ROAD & BOAT → RELATIONSHIPS (REAL OR FICTIONAL) – IN THE CHARACTER/I'PSNL SENSE –
DARK PATTERNS
PROGRAMMING (but this is a bit commonly used)
INDIA, CANADA & JAPAN STRATEGY IN WAR

The key is to hunt for qualities t in these fields that are particular to them but can be analogised with relative ease as features, e.g. "Roundabout sentences" or "Order words by ability to survive an earthquake". With such a ~~key~~ feature down, expanding on it by adding paramdters and a paradigm ~~in the~~ is a matter of routine, and soon we have a complete feature ready to put into ~~the~~ languages.

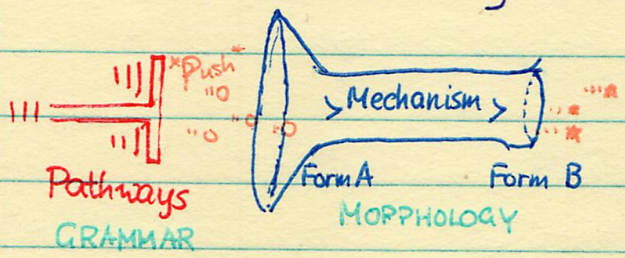
★△ Incomplete Languages

"Violate the universal that ~~claims~~ claims that a language (spoken ones, that is) will be general enough to express ~~every~~ situation and is not specialised to a particular field enough to prohibit such general-purpose communication."

This kind of ~~feature~~ is so large that it can basically support one entire language by itself (plus any supporting features). It's worthy of some further ~~notes~~:

> It's ~~fard~~ easier to have a language only do a specific thing than to forbid it from a specific thing.
> Making unremovable assumptions or "fixing" arguments in a wide class of words trades universality for extreme compactness.
> A ~~thinly~~ well-defined domain is useful but broad domains or domains based on a key object is also workable.
> Avoid using generic sentences - restricting scope via a limited dictionary does not work.

★△ Pathways and Mechanisms



Pairs of features frequently combine to allow for more interesting and structured results. The pair comes in the form as seen at ~~right~~left.

MECHANISM: A ~~φ~~ method to change things from one form to another

PATHWAY: A reason as to why the things are pushed into the mechanism in the first place.

In the language skeleton, the thing(s), mechanisms and pathway are in non-decreasing order. For example:

Pathway	Mechanism		Thing	
Grammar	Grammar	Change in rules governing	Words (Morphology)	◀ Specific grammar structure for situations
Words	Phonotactics		Phonemes	◀ Class-based phoneme changes
Sentences	Words		Morphology	◀ Taboo system of grammar enforced lexically

Multiple ~~st~~ pathways can connect to the same mechanism and vice versa, increasing the potential complexity of the language.

★Vocabulary troubles

Generally there is no need to generate or make vocabulary - just make words as needed. This is for several reasons:

- Sitting down and making things up is hard. Translating text and making things up as you go along is much easier.
- It encourages breaking up the semantic space naturally as one might forget ~~to~~ that ~~include~~ a word ~~ear~~ was included earlier to mean the same. One can also decide in advance that a word is very generic and covers a large swathe of the semantic space at once.

In general vocabulary is not governed by the feature system in specifics, but is in general. ~~For~~ The primary reason for this is that words tend to feel an unusually large attraction to reality, where the feature system doesn't work. But given enough words, ~~the~~ ~~it~~ the influence of reality starts falling off and features can start asserting themselves more confidently:

Break up the semantic space of $THINGS (somehow)
Put slots ~~regarding~~ where $LOWERLAYER and $UPPERLAYER interact
Checksum your words.

Handling Paradigms with ≥3 parameters

★△ Multidimensional shenanigans

When one has ~~a~~ a ~~function~~ paradigm with more than 3 parameters, ~~thi~~ displaying them gets hard, especially if holes are expected in it ~~(or~~ (it gets even ~~wourse~~ worse if paradigms are arranged in a convenient but $NON QUADRILLE grid.) When this occurs there are several solutions.

1. Present a formula instead of a table.
2. Nested tables generally only work with even dimension. But colour can make do with ~~a third~~ the odd one.
3. ▨ If electronics is an option, visualising complex tables via something like a pivot-table is possible.

The takeaway is that trying to present a paradigm with ≥3 parameters is always going to ~~lead~~ emphasise 2 or ~~remain~~ lose the conceptual clarity of a quadrille.

Option ⟨1⟩ however is an appealing feature in an of itself. Here you can take in any number of parameters, have them all contribute in whatever magnitude you wish, and call upon the full power of mathematics or other formalisms to produce grotesque yet compact ¢ paradigms. Of course, real languages don't do that a whole lot on account of the ursystem being totally human-unfriendly, but since when has that become an issue?

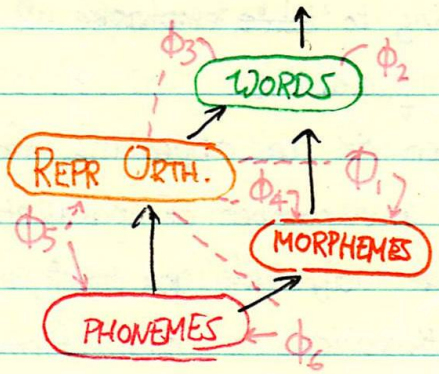
★△ Scripts, or more generally, repr

Although in general languages do not need to have any method of writing, documentation of a language and any sort of formalism will require some way to write it down anyway. So we will talk about writing languages in general, called repr.

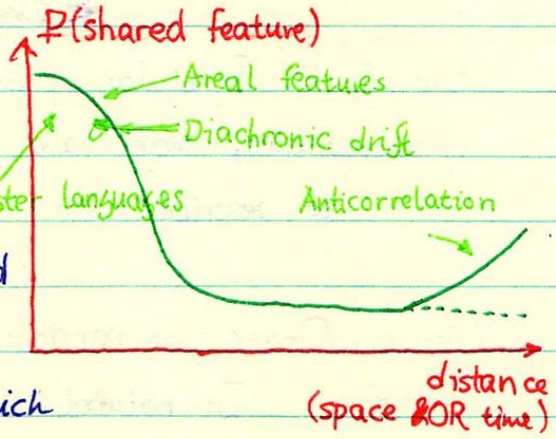
(Repr)

A repr is a special kind of script that basically is tailored for the languages or its purposes. It need not exist with the language but if not a concern might be ease of transport, in contemporary computing systems. ~~A possible repr system~~.

A repr system ~~might~~ would likely consist only of features and transparantly encode information from around it - indeed its features entirely comprise of features from nearby components of the language - though not all of ~~it~~ them for obvious reasons.



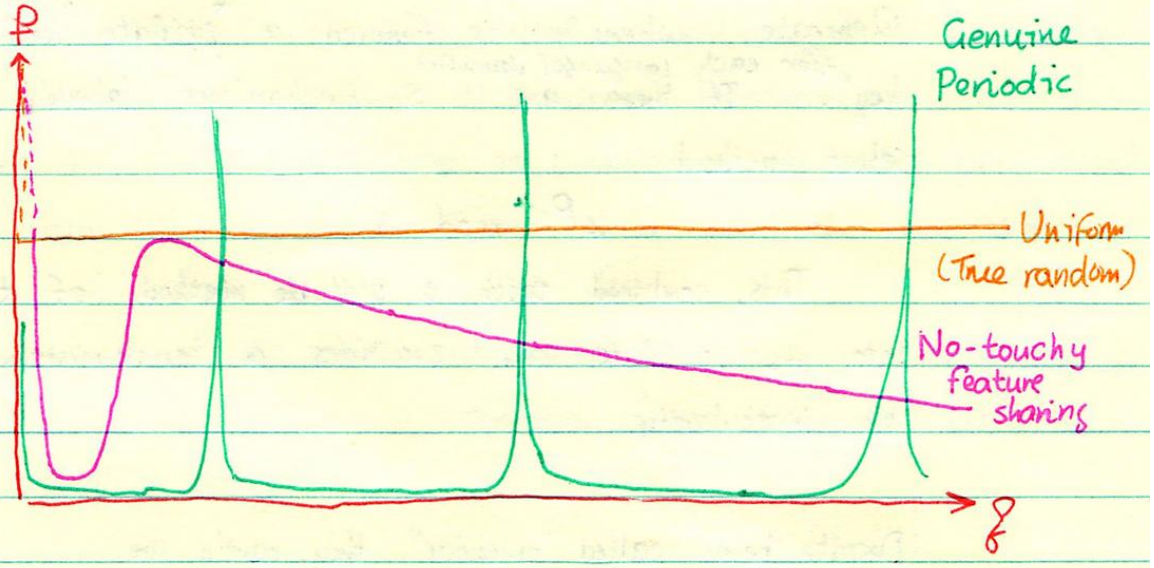
★△ Feature sharing



After your first 5 or 6 Language you might feel tempted to reuse some features in a language you created earlier. Whether you should depends on your decision of a shared feature function, ℘ an example of which is plotted at right.

An interesting thing to do might be to do a periodicity of feature recurrence in space, which is termed "anticorrelation" above. That is to say, given some definition of distances, features are more likely to crop up in evenly-spaced piles. Of course, real languages have no such compulsion.

Otherwise, a different shared feature function can result in interesting patterns:



★ Dealing with irregularities

Irregular paradigms make documentation difficult. By their nature, they have so much Kolomgrod complexity that the only way to describe them is to list them. However, there are shortcuts.

Order in randomness

In natural languages, irregularity comes from [a] old mechanisms that are no longer productive or [b] well-used words seeking efficiency in saying ~~words~~ them. The former can be simulated effectively; the latter is much harder to do but ~~ma~~ can be done "n "automatically" by...

Programmatic chaos

To create very irregular-looking examples indeed we turn to modular arithmetic and a handful of reduction criteria. Generate whenever is fancied a private key (for each word) and a public key (for each language/domain). If there are s mechanisms labelled $[0 .. s_g - 1]$ then select method

$$uP \bmod s$$

This, combined with a suitable method of transforming a word into its private key, provides a conveniently messy way to do irregularity.

Despite being called "irregular", they rarely are.

★△ The RRR system

The rules for transformation must be done in order	Ordered Random	Rules are shuffled and selected with replacement
If a rule ~~ap~~ applies, it must be	Deterministic Random	Rules have a chance of firing if called upon. Some more likely than others.
Every rule applies, exactly once.	Complete Random	Rules can apply many times or none at all. Critically each rule has a process-stop chance on top of the norm stop chance.

This system is a general replacement for any mechanism that requires an arrangement of rules and conditions. It is the Random-Random-Random system, where each "random" represents some facet of applying rules. The randomness is however usually rolled per word rather than per use, as the latter would make the new words ~~think~~ too dependent on the original form.

Variants of the system can easily be made by controlling with features, or switching ~~one~~ some of the "randoms" back to the systematic settings.

The Structure of Language

★△Conclusion I

ITEM Languages are made up of small elements called "features".
- Features are one-sentence soundbites that succiently describe a way of expressing a particular ~~me~~situation.
- They are either usable as-is or require a mad-libs style fill-in-the-blanks.
- They control a "small portion" of the language, but such portions are not well-defined.

ITEM Features can be grouped ~~to~~ into, and are interrelated by, a structure known as the "language skeleton".
- The skeleton is composed of "nodes" and "relations".
- Conventional linguistic fields are contained in both of these.
- Features can be attached to both.
- Presentation of the language is largely contained ofon the nodes.

ITEM Features are "bonded" to the skeleton by way of creating paradigms (e.g. inflection tables, ~~ti~~case lists and irregularity descriptions).

ITEM Irregularity can arise due to concerns with reality interfering with the language or by time. It can be effectively simulated using maths or other means.

ITEM Features can also play the roles of "mechanisms" and "pathways", allowing deeper interactions between them.

ITEM Features, skeletons &c. are not "realistic" and there is no claim that they can be used to describe natural languages or make naturalistic languages.

Evolution & Growth
The Context of Language
★△ Conclusion II

Languages exist in a world. # They are carried primarily by the geography of the land, with the actual speakers playing a secondary role.

Languages occupy or "fill up" some area in which other languages cannot intrude with impunity. (though it isn't impossible to do so.)

- This area is limited by borders of all sorts, natural and political. There is also a soft cap on the area.
- Dialects are allowed to fill up the space occupied by its parent language. They interact with each other in much the same way as languages do to each other.
- A language's area grows or shrinks for both obvious and random reasons.

Diachronic change is largely time-symmetric. The symmetry is broken largely by "choice" of change, rather than changes themselves.

In a similar way, change is also space-symmetric, although this manifests as having little discernable pattern in overall language change in space.

Features can simulate spatial and temporal changes:

- Spatial variation is simulated by having features ~~d~~appear in nearby languages, possibly via a shared feature function
- Temporal variation is the gain and loss of individual features over time.

Closing Remarks

~~The Evolution & Growth of Language~~

★△Conclusion III

The method of creating languages outlined in this annex is only one of many ways one can create languages. In fact this method is very obscure and to the best of my knowledge unique.

However, the main point of the notebook is not to prescribe a new way of making languages, or even describe an existing method (even though it is a goal for this ~~eom~~ to describe how I make languages, which to a large extent it does.) Rather, it is intended to produce a method of creating languages whose products are consistent with each other yet distinct from naturalistic languages, ~~&~~ in hopes that it will inspire ~~you~~ the reader to make a system that would produce similar results (or even just use this one), as an alternative to naturalism.

With that we conclude the series. I wish you luck in ~~the~~ your future creations.