Mathematical structure of syntactic Merge

Matilde Marcolli

lecture at Utrecht University, June 2023

Lecture based on:

1. Matilde Marcolli, Noam Chomsky, Robert C. Berwick, *Mathematical structure of syntactic Merge*, arXiv:2305.18278
2. Matilde Marcolli, Robert C. Berwick, Noam Chomsky, *Old and new Minimalism: a Hopf algebra comparison*, preprint 2023

Generative Linguistics (a very quick outline for the mathematicians)

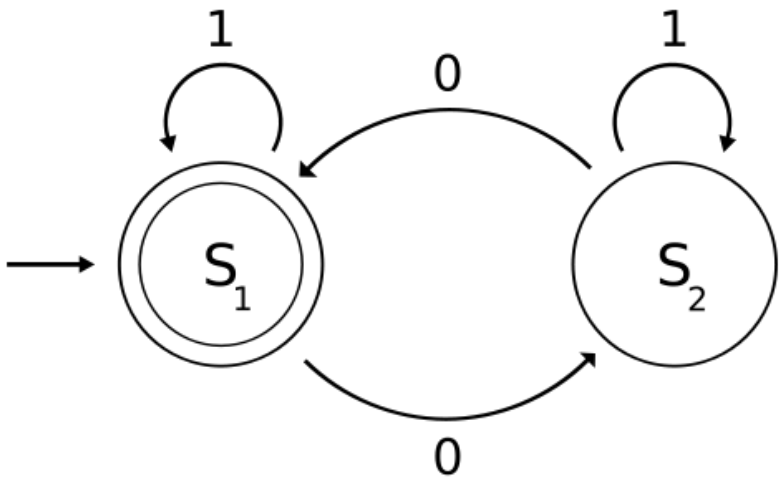
- Chomsky from mid 1950s onward
- key idea: syntax as a computational process
- Early phase: formal languages types of grammars (regular, context-free, context-sensitive, recursively enumerable); non-terminal/terminal symbols and production rules; *Chomsky hierarchy*: recognizable by automata (finite state, pushdown stack, linear-bounded Turing, Turing)
- issues: class of actual natural languages? involved production rules even for small modules of syntax, etc
- need for simpler, more natural generative models
- other problem with formal languages: strings versus structures $\Rightarrow$ Minimalism

Original picture: Syntax as recursive structure

- key idea from the *formal languages* approach: investigate the recursive structures that can be generated by a given grammar/automaton
- *example*: not all human languages are context-free grammars (Huijbregts, Shieber): cross-serial dependencies

- in general *syntactic parse trees* of sentences are a recursive structure: can see formal languages as generating ordered lists (sentences) but also (better) as generating trees

Formal languages describe properties of ordered sequences (strings)



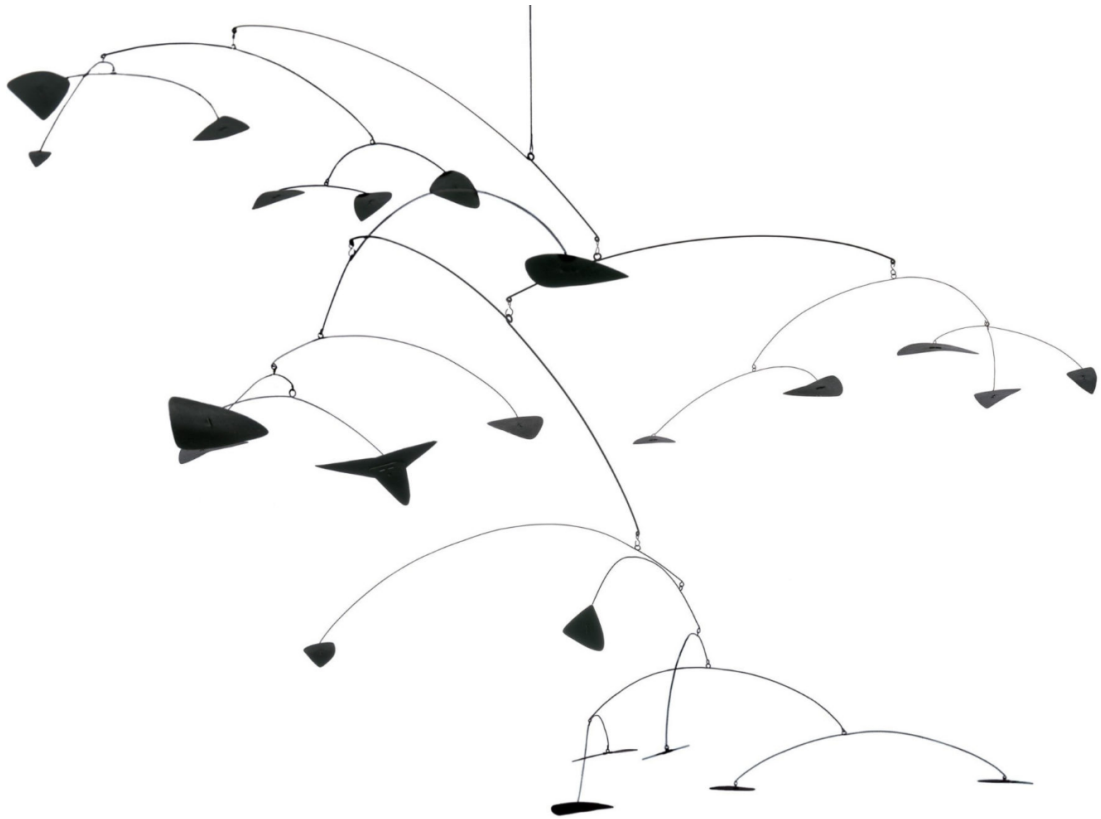
time-ordered sequence of transition in an automaton that computes the language, oriented paths in a graph

but is this what languages look like?

- what languages appear to look like

$$00220100211120001212100220000200211\ldots$$

- what languages actually look like



The Minimalist Model: Merge as fundamental structure

- introduced by Chomsky in the early '90s; various versions and revisions over the years; New Minimalism developed by Chomsky starting around 2013
- Jumping directly to the most recent formulation (comparison with old Minimalism in our second paper)
- key idea: syntax as a computational process depends on a single key operation called Merge
- Two step process: core computational structure (Merge) + externalization (language variation, syntactic parameters, word order/planar embedding of trees, etc)

Goals of this work:

- The core computational structure of Merge (syntactic objects) has a very simple mathematical description
- The action of Merge on workspaces is related to a Hopf algebra structure
- Desirable linguistic properties of Merge follow directly from this algebraic structure (they don't have to be imposed as additional requirements)
- The core structure is the same as the core structure underlying Dyson–Schwinger equations in physics (fundamental nature and optimality of Merge can be seen as analogous to the "least action principle" of physics)
- Any higher n-arity Merge would both massively undergenerate and overgenerate with respect to binary (Huijbregts): here quantifiable and directly following from algebraic structure

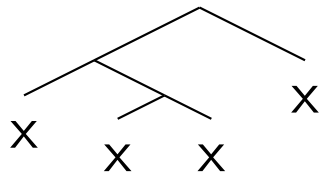
Core computational structure of Merge

- free non-associative commutative magma $\mathfrak{T}$
- elements are balanced bracketed expressions in a single variable x, with the binary operation (binary set formation)

$$(\alpha, \beta) \mapsto \mathfrak{M}(\alpha, \beta) = \{\alpha, \beta\}$$

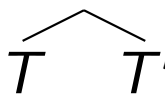
  where α, β are two such balanced bracketed expressions
- equivalent description: elements are finite binary rooted trees (without any choice of planar structure!)

$$\{\{x\{xx\}\}x\} \longleftrightarrow$$



- operation on trees

$$\mathfrak{M}(T, T') =$$



Generative process of $\mathfrak{T}$

- a formal trick: take vector space $\mathcal{V}(\mathfrak{T})$ (say over $\mathbb{Q}$) spanned by elements of $\mathfrak{T}$ (convenient for writing a list of possibilities as a sum)
- note for the mathematicians: the magma operation $\mathfrak{M}$ on $\mathfrak{T}$ identifies $\mathcal{V}(\mathfrak{T})$ with the free commutative non-associative algebra generated by a single variable x (free algebra over the quadratic operad freely generated by the single commutative binary operation)
- assign a grading (a weight, measuring size) to the binary rooted trees by the number of leaves, $\ell = \#L(T)$, so the vector space decomposes $\mathcal{V}(\mathfrak{T}) = \oplus_\ell \mathcal{V}(\mathfrak{T})_\ell$
- in a formal infinite sum $X = \sum_\ell X_\ell$ of variables X_ℓ in $\mathcal{V}(\mathfrak{T})_\ell$

$$X = \mathfrak{M}(X, X)$$

fixed point equation

- the equation $X = \mathfrak{M}(X, X)$ can be solved recursively by degrees

$$X_n = \mathfrak{M}(X, X)_n = \sum_{j=1}^{n-1} \mathfrak{M}(X_j, X_{n-j})$$

- solution $X_1 = x$, $X_2 = \{xx\}$,
  $X_3 = \{x\{xx\}\} + \{\{xx\}x\} = 2\{x\{xx\}\}$,
  $X_4 = 2\{x\{x\{xx\}\}\} + \{\{xx\}\{xx\}\}$, and so on
- coefficients: $\{x\{xx\}\}$ and $\{\{xx\}x\}$ same abstract tree (while two different planar embeddings)
- recursive solution describes the generative process of $\mathfrak{T}$ through the Merge operation $\mathfrak{M}$

Sneak preview

- special fundamental case of Dyson–Schwinger equations

$$X = \mathfrak{B}(P(X))$$

with $X = \sum_\ell X_\ell$ by degrees, $P(X)$ a polynomial function (here a single quadratic term) and $\mathfrak{B}$ a type of (possibly n-ary) Merge operation

- recursive construction of solutions of equations of motion in quantum field theory (more on this later!)

Next step: Syntactic Objects

- extend the core computational structure to the generative process of syntactic objects
- an assigned (finite) set $\mathcal{SO}_0$ of *lexical items* and *syntactic features* such as $N, V, \ldots$
- set $\mathcal{SO}$ of syntactic object obtained as

$$\mathcal{SO} = \mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$$

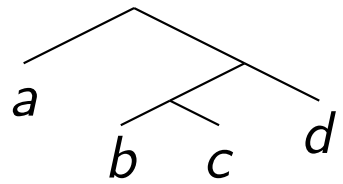
free, non-associative, commutative magma over the set $\mathcal{SO}_0$

- small technical note: both in the core magma $(\mathfrak{T}, \mathfrak{M})$ and in the magma $(\mathcal{SO}, \mathfrak{M})$ of syntactic objects it's convenient to formally add an "empty tree" as multiplicative unit 1 with $\mathfrak{M}(T, 1) = T = \mathfrak{M}(1, T)$

- equivalent description:

$$\mathcal{SO} \simeq \mathfrak{T}_{\mathcal{SO}_0}$$

finite binary rooted trees with leaves labelled by elements of $\mathcal{SO}_0$ (with no assigned planar embedding)

$\{a, \{\{b, c\}, d\}\} \leftrightarrow$ 

- operation on trees

$$\mathfrak{M}(T, T') = \overbrace{T \quad T'}$$

Next step: Workspaces

- material (lexical items and syntactic objects) available for computation
- Merge operation updates workspace for the next step of structure formation
- mathematically workspaces are forests (finite disjoint unions of binary rooted trees with leaves labelled by $\mathcal{SO}_0$, no planar structure)
- set of all workspaces is set $\mathfrak{F}_{\mathcal{SO}_0}$ of binary rooted forests with no assigned planar structure
- given a workspace $F \in \mathfrak{F}_{\mathcal{SO}_0}$, the material in F that is accessible for computation consists of the lexical items and all the trees that were obtained through previous applications of Merge
- inductive definition encoded in the notion of accessible terms

Accessible terms

- tree $T \in \mathfrak{T}_{\mathcal{SO}_0}$ and $v \in V(T)$: subtree T_v rooted at v
- $V_{int}(T)$ non-root vertices of T
- accessible terms of T

$$Acc'(T) = \{T_v \mid v \in V_{int}(T)\} \text{ and } Acc'(T) = \{T_v \mid v \in V(T)\}$$

- workspace $F \in \mathfrak{F}_{\mathcal{SO}_0}$ with $F = \sqcup_{a \in \mathcal{I}} T_a$

$$Acc(F) = \bigsqcup_{a \in \mathcal{I}} Acc'(T_a)$$

- What structure do accessible terms correspond to?
- answer: Workspaces form a Hopf algebra

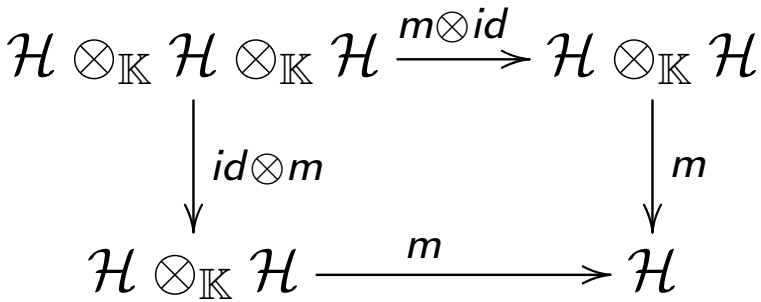
What is a Hopf algebra?

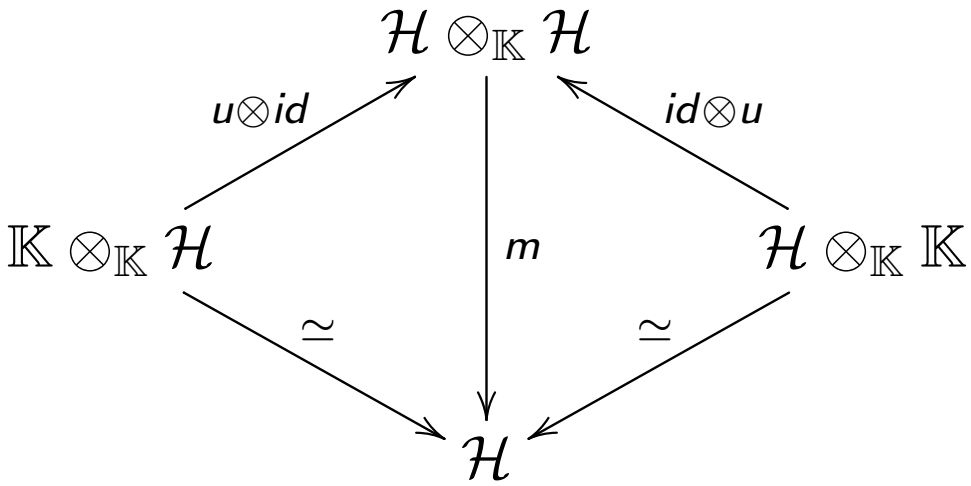
- mathematical method of describing composition–decomposition
- product: an "assemble operation" (two inputs one output) for how to assemble different objects together
- coproduct: a "decomposition operation" (one input two outputs) listing all possible ways of decomposing an objects into parts
- compatibility between these two operations

A formal definition of Hopf algebra

- Hopf algebra $\mathcal{H}$ is a vector space over a field $\mathbb{K}$, endowed with
  - multiplication $m : \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \to \mathcal{H}$;
  - unit $u : \mathbb{K} \to \mathcal{H}$;
  - comultiplication $\Delta : \mathcal{H} \to \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H}$;
  - counit $\epsilon : \mathcal{H} \to \mathbb{K}$;
  - antipode $S : \mathcal{H} \to \mathcal{H}$
- multiplication is associative
- comultiplication is coassociative
- u is multiplicative unit and ϵ is comultiplicative counit
- S relates m and Δ and u and ϵ
- all this expressed by diagrams
- these formal requirements ensure a good pair of composition/decomposition operations

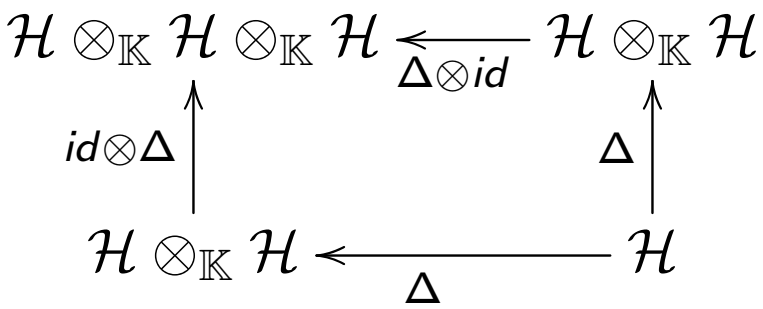
multiplication: associativity and unit

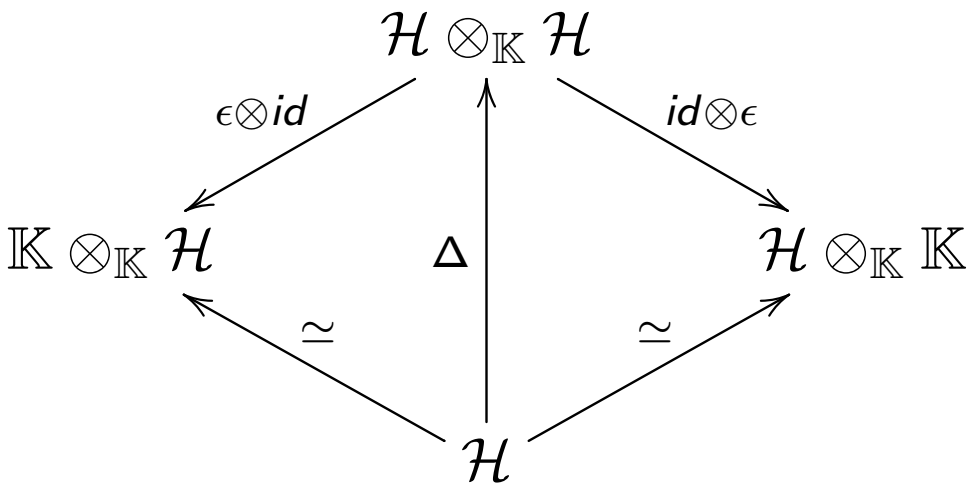




commutativity of these diagrams

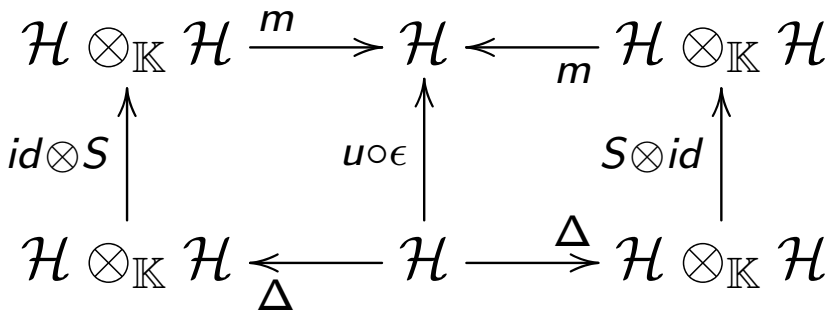
comultiplication: coassociativity and counit





commutativity of these diagrams

antipode: compatibility: commutativity of diagram



Note: if the Hopf algebra is graded $\mathcal{H} = \oplus_{\ell \geq 0} \mathcal{H}_\ell$ with $\mathcal{H}_0 = \mathbb{K}$ and m, Δ compatible with grading, antipode comes for free

$$S(X) = -X - \sum S(X')X''$$

inductively for $\Delta(X) = X \otimes 1 + 1 \otimes X + \sum X' \otimes X''$ with X', X'' terms of lower degree

Workspaces as a Hopf algebra

- same trick as before: form vector space $\mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0})$ to write possibilities as sums (grading by number of leaves as for trees)
- product on forests simply given by disjoint union $\sqcup$ (put two forests together to form a new one)
- unit of product is formal "empty tree"
- coproduct is the interesting part: extract accessible terms on one side, and remove them on the other side (i.e. what's left)

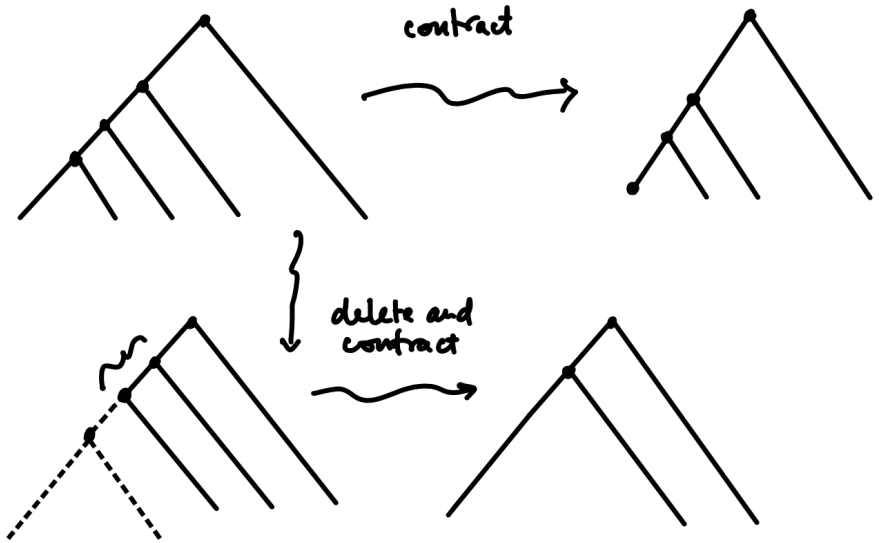
$$T_v \otimes T/T_v$$

- a few delicate points here: sum over all possibilities, how to take quotient T/T_v, coassociativity

taking quotients: T/T_v

Two possibilities: contracting or deleting

- contracting: T/T_v obtained by shrinking subtree T_v to its root vertex
- deleting: T/T_v obtained by removing T_v and taking the unique maximal binary tree obtained from $T \setminus T_v$ by shrinking edges



difference: $T_v = T$ first case quotient • single vertex, second case quotient empty tree; linguistic relevance: in first case remaining root vertex needs labeling, need a labeling algorithm for internal vertices (projection)

coproduct and coassociativity

- $T \in \mathfrak{T}_{\mathcal{SO}_0}$, subforest $F_{\underline{v}} \subset T$ union of disjoint subtrees $T_{v_1}, \dots, T_{v_k}$ for $\underline{v} = (v_1, \dots, v_k)$, quotient given by

$$T/F_{\underline{v}} = (\cdots (T/T_{v_1})/T_{v_2} \cdots)/T_{v_k}$$

- coproduct on trees

$$\Delta(T) = \sum_{\underline{v}} F_{\underline{v}} \otimes (T/F_{\underline{v}})$$

on forests $F = \sqcup_a T_a$ by $\Delta(F) = \sqcup_a \Delta(T_a)$

- coproduct is coassociative $(\mathrm{id} \otimes \Delta) \circ \Delta = (\Delta \otimes \mathrm{id}) \circ \Delta$
- parts of coproduct

$$\Delta(T) = \sum_{n \geq 2} \Delta_{(n)}(T) \quad \text{with} \quad \Delta_{(2)}(T) = \sum_{v} T_v \otimes T/T_v$$

$\Delta_{(2)}$ needed for single application of binary Merge

- compatibility of product and coproduct

$$\Delta \circ \sqcup = (\sqcup \otimes \sqcup) \circ \tau \circ (\Delta \otimes \Delta)$$

  τ flips middle two terms of $\mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H}$

- Note: cancellation of copies via T/T_v terms: automatically distinguishes *copies* from *repetitions*; only copies (that will be the ones produced by Internal Merge) get cancellation

- Size of workspaces different measures of size
  - $b_0(F)$ number of connected components (trees) of the forest F (number of syntactic objects in the workspace)
  - $\alpha(F) = \#Acc(F)$ number of accessible terms
  - $\sigma(F) = \#V(F) = b_0(F) + \alpha(F)$ number of vertices (components plus accessible terms)
  - $\hat{\sigma}(F) = b_0(F) + \#V(F)$
  - $\ell(F)$ number of leaves (grading of Hopf algebra)

  How these change under Merge action on workspaces

Next step: Action of Merge on Workspaces

- pair of syntactic objects $S, S' \in \mathcal{SO}$

$$\mathfrak{M}_{S,S'} = \sqcup \circ (\mathfrak{B} \otimes \mathrm{id}) \circ \delta_{S,S'} \circ \Delta$$

- use all Hopf algebra structure Δ, $\sqcup$ but not algebra/coalgebra morphism
- if simultaneously look at all the possible Merge operations on a given workspace just take

$$\sqcup \circ (\mathfrak{B} \otimes \mathrm{id}) \circ \Delta$$

Unpacking the expression of $\mathfrak{M}_{S,S'}$ in a more explicit form...

$$\mathfrak{M}_{S,S'} = \sqcup \circ (\mathfrak{B} \otimes \mathrm{id}) \circ \delta_{S,S'} \circ \Delta$$

1. Δ produces the list of accessible terms along with the quotients that represent cancellation of deeper copies
2. $\delta_{S,S'}$ searches in the list of accessible terms for matching copies of S and S'
3. if not found, replaces coproduct terms $T_v \otimes T/T_v$ with $1 \otimes T$ so workspace remains unchanged
4. use of $\mathfrak{B}$ for Merge of syntactic objects just because

$$\begin{array}{ccc} \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0}) \otimes \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0}) & \xrightarrow{\mathfrak{M}} & \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0}) \\ & {\scriptstyle \sqcup} \searrow \quad \nearrow {\scriptstyle \mathfrak{B}} & \\ & \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0}) & \end{array}$$

5. if found $\mathfrak{B} \otimes \mathrm{id}$ merges the two terms creating a new component $\mathfrak{M}(S, S')$ and keeps cancellation of deeper copies
6. $\sqcup$ assembles back the new workspace

Cases of Merge (too many forms of Merge?)
The modified part of work space looks like

$$\mathfrak{M}(T_v, T_w) \sqcup (T_a/T_v) \sqcup (T_b/T_w)$$

Various cases $\mathfrak{M}(\alpha, \beta)$

1. $\alpha = T_i$ and $\beta = T_j$ with $T_i, T_j \in F$ and $i \neq j$;
2. $\alpha = T_i \in F$ and $\beta \in Acc(T_j)$ for some $T_j \in F$, with two sub-cases:
   a) $i = j$
   b) $i \neq j$
3. $\alpha \in Acc(T_i)$ and $\beta \in Acc(T_j)$ for some $T_i, T_j \in F$, with two sub-cases:
   a) $i = j$
   b) $i \neq j$

(1) External Merge; (2a) Internal Merge; (2b) and (3b) Sideward Merge; (3a) Countercyclic Merge

Comment on Internal Merge

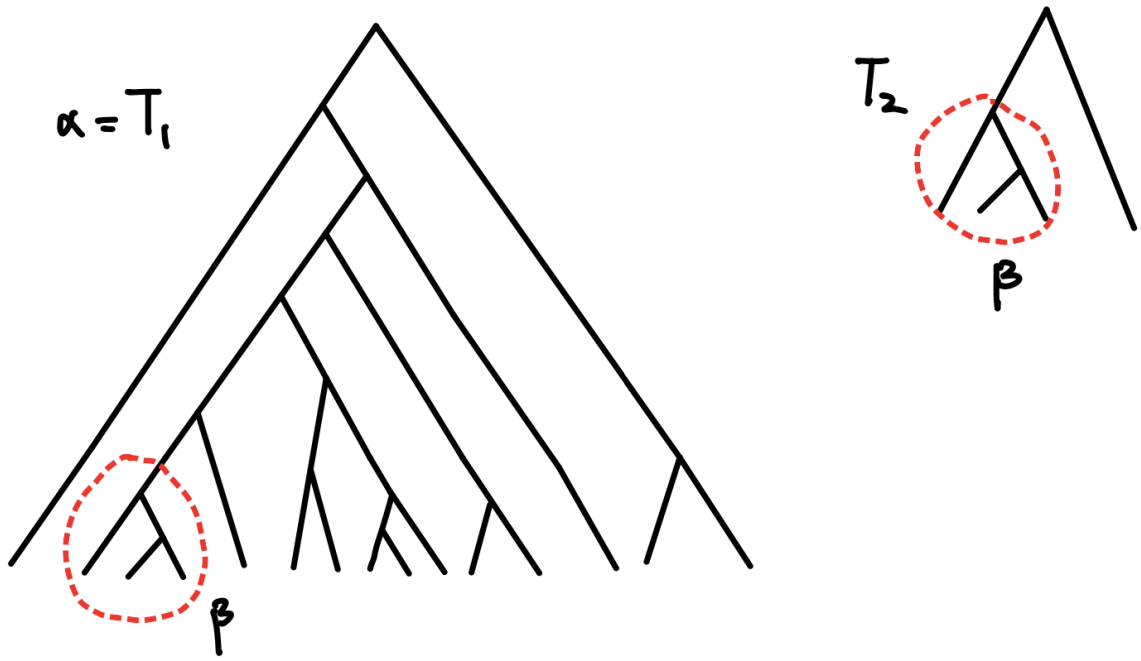
- We have a formal empty tree $\mathbf{1}$ which satisfies

$$\mathfrak{M}(T, \mathbf{1}) = \mathfrak{M}(\mathbf{1}, T) = T$$

- Note: language does not incorporate arithmetic
  - a unary Merge is needed in the first step of the successor function of Peano arithmetic $\emptyset \mapsto \{\emptyset\}$
  - then von Neumann description of the nonnegative integers is just (Internal) Merge $\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset, \{\emptyset\}\}\}, \ldots$
  - but in syntactic objects (as binary rooted trees) merging with the empty tree just leaves T unchanged
- If $\beta \in Acc(T_a)$ for a component T_a of workspace F the Merge $\mathfrak{M}_{\beta,\mathbf{1}}$ produces a new workspace with the component T_a replaced by $\beta \sqcup T_a/\beta$; External Merge on this workspace then gives $\mathfrak{M}(\beta, T_a/\beta)$ giving Internal Merge
- Note: we'll see that $\mathfrak{M}_{\beta,\mathbf{1}}$ does not "exist in isolation" only in composition as Internal Merge

Minimal Search

- Problem: Sideward and Countercyclic Merge have undesirable linguistic properties
- in Minimalism these undesirable forms of Merge are eliminated on a principle of Minimal Search
- key idea: efficient search for matching terms (our $\delta_{S,S'}$) would first look for matching components of workspace (External Merge) then for a single component and an accessible term of the same (Internal Merge): everything else is a less efficient search
- How to formalize Minimal Search in our Hopf algebra setting?
- Minimal with respect to what? Cost function? Leading order term in an expansion?



Need a cost function / order degree justifying why finding the copy of β deep inside T_1 is "more efficient" than finding the copy of β high up into another component (regardless of number of components of F and of depth of T_1)

Minimal Search as Leading Order Extraction

- introduce degrees in the coproduct

$$\Delta^{(\epsilon,\eta)}(T) = \sum_{\underline{v}} \epsilon^{d_{\underline{v}}} F_{\underline{v}} \otimes \eta^{d_{\underline{v}}}(T/F_{\underline{v}})$$

  $\underline{v} = \{v_1, \ldots, v_n\}$ take $d_{\underline{v}} = d_{v_1} + \cdots + d_{v_n}$ with d_v the distance of a vertex v to the root of T

- Merge action correspondingly weighted

$$\mathfrak{M}^{\epsilon}_{S,S'} = \sqcup \circ (\mathfrak{M}^{\epsilon} \otimes \mathrm{id}) \circ \delta_{S,S'} \circ \Delta^{(\epsilon,\epsilon^{-1})}$$

$$\mathfrak{M}^{\epsilon}(\epsilon^{d}\alpha, \epsilon^{\ell}\beta) = \epsilon^{|d+\ell|}\, \mathfrak{M}(\alpha, \beta)$$

- Take the leading term for $\epsilon \to 0$
- only Merge derivations that survive in the limit are compositions of only External and Internal Merge

Another proposal in Minimalism: get rid of unwanted Merges by their effect on Workspace size functions

Good operations should not increase the number of components of workspace (derivation converges) and not decrease number of accessible terms (no syntactic information gets destroyed)

- External and Internal Merge

	b_0	$\#Acc$	σ	$\hat{\sigma}$
External	-1	$+2$	$+1$	0
Internal	0	0	0	0

External: components decrease by one in $\mathfrak{M}(T, T')$ and two roots become new accessible terms

Internal: same components and because of how quotient defined

$$\#Acc(T_v) + \#Acc(T/T_v) + 2 = \#Acc(T)$$

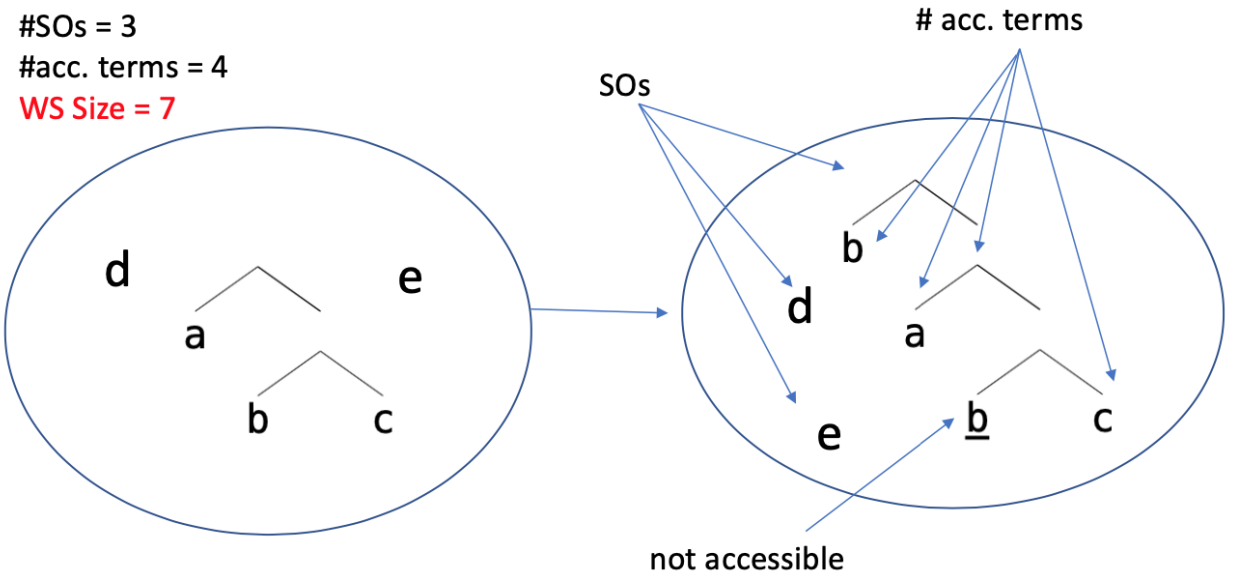
in T/T_v all the vertices of T_v and one additional vertex of T removed

This counting of size agrees with the counting for Internal Merge in S. Fong, R. Berwick, J. Ginsburg, *The combinatorics of merge and workspace right-sizing*, Evolinguistics Workshop, 2019

- Internal Merge of b and {a, {b, c}}

Minimal Search: exclude lower copies from set of accessible terms

- Copy of b is indicated as b



#SOs = 3
#acc. terms = 4
WS Size = 3 + 4
WS Size = 7

IM keeps the # of SOs and accessible terms the same.

IM does not change WS Size

i.e. same way of taking "quotient tree"

- Merge $\mathfrak{M}_{\beta,\mathbf{1}}$ ruled out except in composition to Internal Merge

	b_0	$\#Acc$	σ	$\hat{\sigma}$
$\mathfrak{M}_{S,1}$	$+1$	-2	-1	0

- Sideward and Countercyclic Merge

	b_0	$\#Acc$	σ	$\hat{\sigma}$
Sideward (3b)	$+1$	0	$+1$	$+2$
Sideward (2b)	0	$+1$	$+1$	$+1$
Countercyclic (3a)	$+1$	$\#Acc(T_{a,w_a})$	$\sigma(T_{a,w_a})$	$\sigma(T_{a,w_a})+1$
Countercyclic (3a)	$+1$	-2	-1	0

Possible constraints on workspace size functions

- number of accessible terms non-decreasing ($\Delta\alpha \geq 0$)
- number of syntactic objects non-increasing ($\Delta b_0 \leq 0$)
- size of the workspace not decreasing and not increasing more than one ($0 \leq \Delta\sigma \leq 1$)
- $\hat{\sigma}$ is a conserved quantity ($\Delta\hat{\sigma} = 0$)

Internal and External Merge satisfy all of these; other cases:

	$\Delta b_0 \leq 0$	$\Delta\alpha \geq 0$	$0 \leq \Delta\sigma \leq 1$	$\Delta\hat{\sigma} = 0$
Sideward (3b)	N	Y	Y	N
Sideward (2b)	Y	Y	Y	N
Countercyclic (3a)	N	Y	N	N
Countercyclic (3a)	N	N	N	Y

Observation (by Riny Huijbregts): quotient by contraction only
Internal/External Merge increase accessible terms by exactly one

Why is Merge binary?

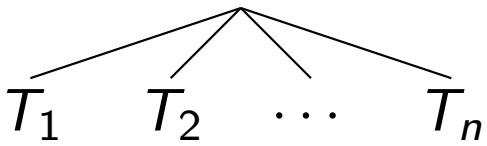
- Also an optimization: a Merge with any other $n \geq 3$ arity would both undergenerate and overgenerate with respect to binary Merge (observed by Riny Huijbregts)
- syntactic objects of a hypothetical n-ary Merge

$$\mathcal{SO}^{(n)} = \mathrm{Magma}^{(n)}_{na,c}(\mathcal{SO}_0, \mathfrak{M}_n)$$

- rooted n-ary trees (without planar structure)

$$\mathcal{SO}^{(n)} \simeq \mathfrak{T}^{(n)}_{\mathcal{SO}_0}$$

$$(T_1, \ldots, T_n) \mapsto \mathfrak{M}(T_1, \ldots, T_n) =$$

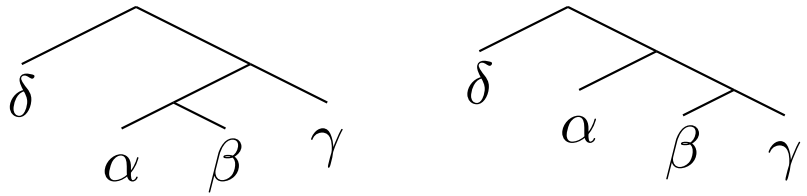


- by number of leaves

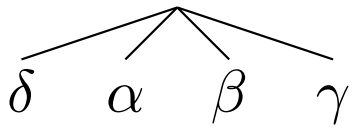
$$\mathcal{SO}^{(n)} = \bigsqcup_{k \geq 1} \mathcal{SO}^{(n)}_{k(n-1)+1}$$

two forms of undergeneration

- achievable lengths only $\ell = k(n-1)+1$ for $k \geq 1$ (excludes examples like *it rains*)
- ambiguities are not detected: example



(ambiguity of *I saw someone with a telescope*) become unambiguous



- undergeneration depends on syntactic objects, overgeneration depends on action on workspaces

action on workspaces of n-ary Merge and overgeneration

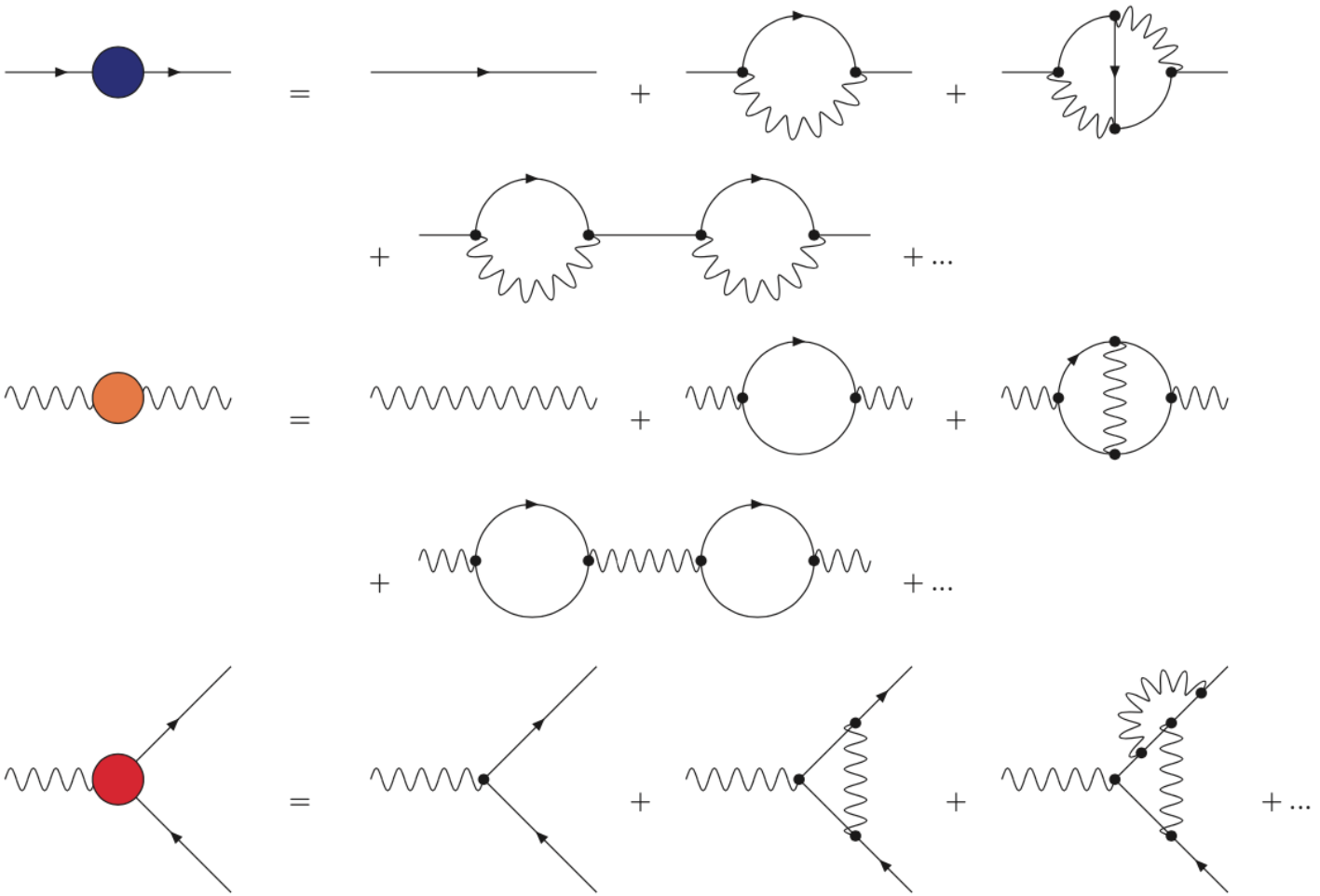
- workspaces are n-ary forests $F \in \mathfrak{F}^{(n)}_{\mathcal{SO}_0}$, same form of product and coproduct
- but for n-ary trees need to take quotients as contraction (so problem with labels reappears)
- Merge operations depending on an n-tuple of n-ary syntactic objects (with n-ary $\mathfrak{B}$)

$$\mathfrak{M}_{S_1,\dots,S_n} = \sqcup \circ (\mathfrak{B} \otimes \mathrm{id}) \circ \delta_{S_1,\dots,S_n} \circ \Delta$$

- overgeneration: example (by Riny Huijbregts) with $n = 3$ and $F = \{\alpha, \beta, \gamma\} \sqcup \delta \sqcup \eta$ $S = (S_1, S_2, S_3)$ given by $S_1 = \alpha$, $S_2 = \beta$, and $S_3 = \{\alpha, \beta, \gamma\}$ gives new workspace $\{\alpha, \beta, \{\alpha, \beta, \gamma\}\} \sqcup \delta \sqcup \eta$ and further application with $S_1 = \delta$, $S_2 = \eta$, and $S_3 = \{\alpha, \beta, \gamma\}$ gives $\{\delta, \eta, \{\alpha, \beta, \gamma\}\}$ (responsible for examples like **peanuts monkeys children will throw*)
- can count explicit amount of undergeneration and of overgeneration as a function of size of trees (number of leaves)

Linguistic Merge versus Physical DS equations: a useful parallel

- in quantum field theory generative process involving graphs (Feynman graphs)
- can be described in terms of formal languages (using graph grammars)
- however not the best way to think of Feynman graphs
- Hopf algebra structure: product $\sqcup$, coproduct $\Delta(\Gamma) = \sum \gamma \otimes \Gamma/\gamma$ subgraphs and quotient graphs
- better for factorization problems (extraction of meaningful physical values = renormalization) with consistency across subgraphs
- better for recursive solutions of equations of motion $X = \mathfrak{B}(P(X))$ Dyson–Schwinger equation
- known in QFT that solutions of DS are quantum implementation of "least action principle" for classical solutions: optimization



Examples: recursive solutions of Dyson–Schwinger equations in quantum electrodynamics

- the formalism of Hopf algebras and extraction of finite parts was adapted to the theory of computation (Manin, 2009) as extraction of computable parts from undecidable problems
- "extraction of meaning" (finite values from divergent integrals in physics; computable parts of non-computable functions in theory of computation) via the formalism of renormalization (factorization of maps from Hopf algebras to Rota–Baxter algebras)
- suggests a possible strategy to extend the computational model of syntax to a computational model of the syntactic-semantic interface

Final step: Externalization

- core computational mechanism of syntax works on structures and does not require planar embeddings (linear ordering)
- realization of language through speech is time-oriented (ordered set) requires planarization
- different languages have different word-order structure
- language specific non-canonical choice of planarization of the binary rooted trees (depending on syntactic parameters)
- externalization happens *after* all Merge computations have taken place

Observation: morphisms of magmas

- free symmetric Merge: free commutative non-associative magma of syntactic objects

$$\mathcal{SO} = \mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$$

- also have a free non-commutative, non-associative magma on the same set $\mathcal{SO}_0$

$$\mathcal{SO}^{nc} := \mathrm{Magma}_{na,nc}(\mathcal{SO}_0, \mathfrak{M}^{nc})$$

- it generates the planar binary rooted trees with leaves labelled by $\mathcal{SO}_0$

$$\mathcal{SO}^{nc} \simeq \mathfrak{T}^{pl}_{\mathcal{SO}_0}$$

  write these as T^π (with T for abstract tree, π for choice of planar embedding)
- there is a morphism if magmas $\mathcal{SO}^{nc} \to \mathcal{SO}$ that forgets the planar embedding $T^\pi \mapsto T$

- but... there is no morphism of magmas going the other way from $\mathcal{SO}$ to $\mathcal{SO}^{nc}$
- because since $(\mathcal{SO}, \mathfrak{M})$ is commutative it should map to a commutative sub-magma of $(\mathcal{SO}^{nc}, \mathfrak{M}^{nc})$
- but $(\mathcal{SO}^{nc}, \mathfrak{M}^{nc})$ does not have nontrivial commutative sub-magmas: if a nonempty planar tree T^{π} is in a commutative sub-magmas then $\mathfrak{M}^{nc}(T^{\pi}, T^{\pi})$ also is but this contradicts commutativity since

$$\mathfrak{M}^{nc}(T^{\pi}, \mathfrak{M}^{nc}(T^{\pi}, T^{\pi})) \neq \mathfrak{M}^{nc}(\mathfrak{M}^{nc}(T^{\pi}, T^{\pi}), T^{\pi})$$

- linguistic consequence: you can either have a symmetric Merge acting *before* assignments of planar structure (linear ordering) as in new Minimalism, or you can have asymmetric Merge acting on planar trees (old Minimalism) but *not both at the same time consistently*

Externalization: first step

- observed languages have a linear ordering (time-ordering in speech) and have different word-order structures (syntactic parameters)
- in the new Minimalism model all Merge derivations happen at the level of free symmetric Merge (syntactic objects and action on workspaces)
- assignment of linear ordering (planar structure of trees) happens only *after* all Merge computations have taken place
- assignment of planar structures to trees is seen as a step necessary to make core syntactic computations compatible with the need for linear ordering due to the mechanics of speech (externalization)
- so assignment of planar structure is a non-canonical section (non-unique and language-dependent through syntactic parameters) of the projection $\mathcal{SO}^{nc} \to \mathcal{SO}$
- this section is not a morphism of magmas

Externalization: second step

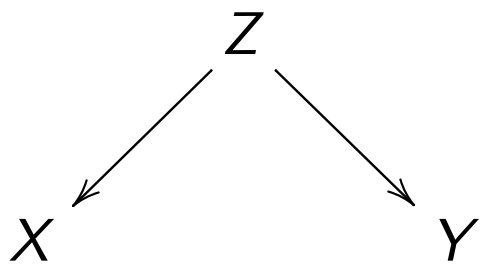
- planarization σ_L via a language-dependent non-unique section of the projection

$$\mathfrak{T}^{pl}_{\mathcal{SO}_0} \underset{\text{proj}}{\overset{\sigma_{\mathcal{L}}}{\rightleftarrows}} \mathfrak{T}_{\mathcal{SO}_0} ,$$

- only requirement on $\sigma_{\mathcal{L}}$ is compatibility with word-order parameters of given language $\mathcal{L}$
- obtain in this way a planar tree $T^{\pi_{\mathcal{L}}} = \sigma_{\mathcal{L}}(T)$ for every syntactic object $T \in \mathcal{SO}$ with no further restriction (for instance no restriction on assignment of labels in $\mathcal{SO}_0$ at the leaves)
- so need further elimination of those objects $T^{\pi_{\mathcal{L}}} \in \mathcal{SO}^{nc}$ that violate linguistic constraints (more syntactic parameters) of language $\mathcal{L}$: quotient map $\Pi_{\mathcal{L}} : \mathcal{V}(\mathfrak{T}^{pl}_{\mathcal{SO}_0}) \to \mathcal{V}(\mathfrak{T}^{pl,\mathcal{L}}_{\mathcal{SO}_0})$
- two-step externalization: section of a projection followed by another projection ... correspondence

correspondences

- in mathematics the simplest way of describing transformation is through functions $f : X \to Y$ single valued assignments $x \mapsto f(x)$
- sometimes functions are not the best way of going from X to Y and a better notion is correspondences



climbing one arrow "the wrong way" then going down the other one (includes the case of multivalued functions)

What happens to action on workspaces in externalization?

- since all Merge operations happen with symmetric Merge before externalization it seems one cannot see at all this action after externalization (because magma structure not preserved by planarization $\sigma_{\mathcal{L}}$)
- but one can still see part of it
- $\mathcal{A}_{na,c} = (\mathcal{V}(\mathfrak{T}_{\mathcal{S}\mathcal{O}_0}), \mathfrak{M})$ non-associative commutative algebra
- representations for a non-associative algebras $\mathcal{A}$ are just *linear maps* (not algebra hom) $\rho : \mathcal{A} \to \mathrm{End}(\mathcal{V})$ on vector space $\mathcal{V}$
- fix an argument of Merge: $\mathfrak{M}^T(T') := \mathfrak{M}(T, T')$
- then representation (in the above sense) from action on workspaces $F = \sqcup_a T_a$

$$\rho(T)(F) = \sqcup \circ (\mathfrak{M}^T \otimes 1) \circ \Delta(F) = \sqcup_a (\mathfrak{M}(T, T_{a,v}) \sqcup T_a / T_{a,v})$$

suffices to determine full action if known for all T

- the projection part is compatible with action of asymmetric Merge
- but section $\sigma_{\mathcal{L}}$ is not a magma morphism so only projection in the other direction is compatible with Merge action

$$\begin{array}{ccc}
 & & \mathcal{A}_{na,nc,\mathcal{L}} \otimes \mathcal{V}(\mathfrak{F}^{pl,\mathcal{L}}_{\mathcal{SO}_0}) \xrightarrow{\rho^{pl,\mathcal{L}}} \mathcal{V}(\mathfrak{F}^{pl,\mathcal{L}}_{\mathcal{SO}_0}) \\
 & \overset{\Pi_{\mathcal{L}}\otimes\Pi_{\mathcal{L}}}{\nearrow} & \overset{\Pi_{\mathcal{L}}}{\nearrow} \\
\mathcal{A}_{na,nc} \otimes \mathcal{V}(\mathfrak{F}^{pl}_{\mathcal{SO}_0}) & \xrightarrow{\rho^{pl}} & \mathcal{V}(\mathfrak{F}^{pl}_{\mathcal{SO}_0}) \\
\downarrow{\scriptstyle \Pi\otimes\Pi} & & \downarrow{\scriptstyle \Pi} \\
\mathcal{A}_{na,c} \otimes \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0}) & \xrightarrow{\rho} & \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0})
\end{array} .$$

- but climbing up the projection Π with the section $\sigma_{\mathcal{L}}$ leads to only a partially defined Merge action on the image
- indeed in old Minimalism, with Merge after planarization, partially defined with specific conditions on domains

Discussing alternatives to externalization

- proposal of passing from free symmetric merge $(\mathcal{SO}, \mathfrak{M})$ to planar trees via a *canonical* choice of planarization (language independent) determined by additional data on trees (head)
- Richard Kayne's LCA (Linear Correspondence Axiom): "linearization" here means assignment of planar embedding to binary rooted trees
- what is expected in terms of magma structure?

$$\mathfrak{T}^{pl}_{\mathcal{SO}_0} \underset{\text{proj}}{\overset{\sigma^{LCA}}{\rightleftarrows}} \mathfrak{T}_{\mathcal{SO}_0} ,$$

1. either define a merge on the image as

$$\mathfrak{M}^{LCA}(\sigma^{LCA}(T_1), \sigma^{LCA}(T_2)) := \sigma^{LCA}(\mathfrak{M}(T_1, T_2))$$

   but then just an isomorphic copy of $(\mathcal{SO}, \mathfrak{M})$ with no effect
2. or σ^{LCA} simply a section as in externalization with Merge only acting before planarization

problem with head-driven planarization

- vertex v in T c-commands another vertex w if neither dominates the other and the lowest vertex that dominates v also dominates w; asymmetrically c-commands if c-commands and not sister vertices
- LCA procedure: assign ordering v precedes v' iff
  1. v asymmetrically c-commands v' or
  2. a maximal projection w dominating v (T_w not in any larger tree with same head) c-commands v'
- difficulty: cannot have σ^{LCA} defined on all of $\mathcal{SO}$ (even if just a section that is not a morphism of magmas as in externalization)
- to see this: abstracting the formal properties of the syntactic *head*

- *head function* $h_T : V^o(T) \to L(T)$ from non-leaf vertices to leaves
- if $T_v \subseteq T_w$ and $h_T(w) \in L(T_v) \subseteq L(T_w)$, then $h_T(w) = h_T(v)$
- write $h(T)$ for value of h_T at root of T
- for a pair (T, h_T) and $(T', h_{T'})$, there are two possible $h_{\mathfrak{M}(T,T')}$: marking one or the other of the two edges attached to new root
- i.e. choices of $h(\mathfrak{M}(T, T')) = h(T)$ or $h(\mathfrak{M}(T, T')) = h(T')$
- so total of $2^{\#V^o(T)}$ possible head functions on a tree T
- *head* of a subtree $T_v \subset T$ is leaf $h_T(v)$ reached by following path of only marked edges (that determine h_T) from v

head functions and planar embeddings

- a head function $h_T : V^o(T) \to L(T)$ determines a planar embedding $T^{\pi^{h_T}}$ of T: put the marked edge below each vertex to the left of the other
- question of a special (canonical) choice of planarization σ^{LCA} becomes a canonical choice of a head function

$$\mathfrak{T}_{\mathcal{SO}_0} \ni T \overset{h^{LCA}}{\mapsto} h_T$$

- this mapping h^{LCA} should be determined by the labels $\lambda(\ell) \in \mathcal{SO}_0$
- note that in $\mathfrak{T}_{\mathcal{SO}_0}$ leaves labels are arbitrary (only in quotient map $\Pi_{\mathcal{L}}$ step of externalization some are ruled out)
- so to have σ^{LCA} defined on all $\mathcal{SO}$ should be able to choose one of the two $h_{\mathfrak{M}(T,T')}$ based on $\lambda(h_T(T))$ and $\lambda(h_{T'}(T'))$
- Problem: if $\lambda(h_T(T)) = \lambda(h_{T'}(T'))$ cannot distinguish two possible $h_{\mathfrak{M}(T,T')}$ even if $\mathcal{SO}_0$ were totally ordered

additional material: old forms of Minimalism and Hopf algebras

- constructing I/E Merge directly on planar trees
- including labeling and domains for applicability based on labels
- Hopf algebras of planar binary rooted trees (Loday–Ronco Hopf algebra)
- role of I/E Merge in terms of LR Hopf alg
- different structures for Internal and External Merge (not coming from same operation)
  - Internal Merge determines system of right-ideal coideals (weak notion of quotient)
  - External Merge determines partially defined operated algebra

planar binary rooted trees

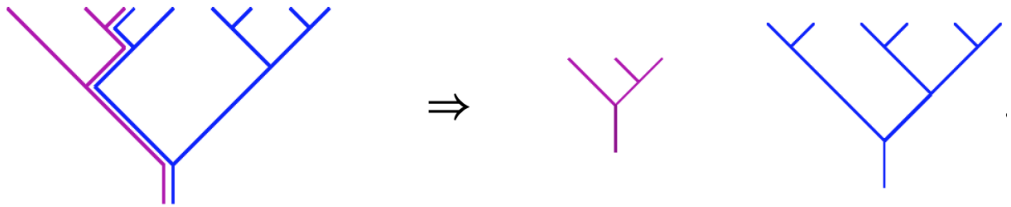
- $\mathcal{V} = \oplus_{k \geq 0} \mathcal{V}_k$ vector space spanned by planar bunary rooted tree, $k =$ number of non-leaf vertices ($k+1$ leaves)
- now will have labeling of internal vertices also D_V set of labels
- for $d \in D_V$ grafting operator $\wedge_d$

$$\wedge_d : \mathcal{V} \otimes \mathcal{V} \to \mathcal{V}, \quad T_1 \otimes T_2 \mapsto T = T_1 \wedge_d T_2 = \quad \begin{array}{c} d \\ \diagup \quad \diagdown \\ T_1 \quad T_2 \end{array}$$

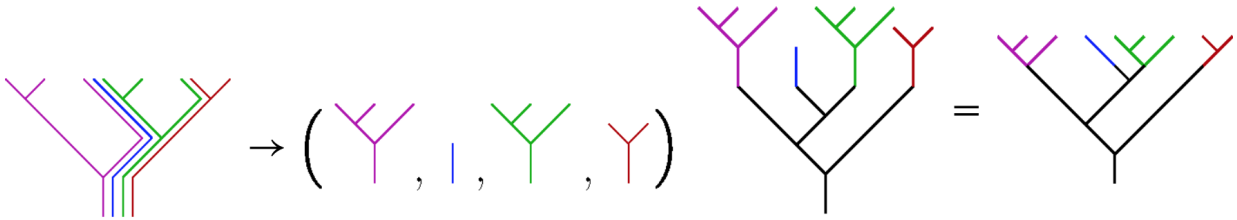
- $S \backslash T$ (S under T) grafting root of T to rightmost leaf of S
- T / S (S over T) grafting the root of T to leftmost leaf of S
- $T_1 \wedge_d T_2 = T_1 / S \backslash T_2$ with S planar binary tree with single non-leaf vertex decorated by $d \in D_V$
- each planar rooted tree is $T = T_\ell \wedge_d T_r$ (left and right subtrees below root)

Loday–Ronco Hopf algebra of planar binary rooted trees $\mathcal{H}_{LR}$

- product and coproduct defined inductively by degrees
- can also see graphically
  - coproduct sum $\Delta(T) = \sum T' \otimes T''$ over all decompositions of tree along paths from one of leaves to root



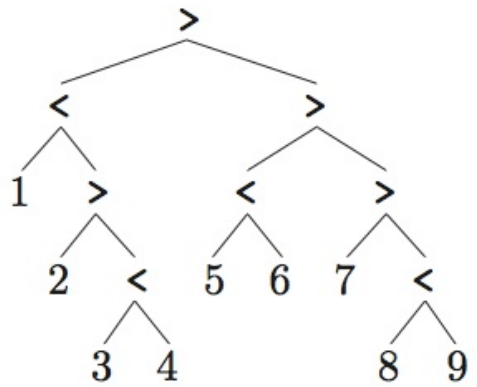
  - product $T \star T' = \sum_{(T_0,\ldots,T_n)} \gamma(T_0,\ldots,T_n;T')$ using same decompositions of first tree into as many pieces as leaves of second tree then grafting to leaves



  - antipode inductively constructed by degrees

Stabler's Computational Minimalism

- example of old formulation of Minimalism (also relation to formal languages)
- planar binary rooted trees with labels:
  - leaves labelled by lexical items and syntactic features $X \in \{N, V, A, P, C, T, D, \ldots\}$
  - also "selector" features σX for head selecting a phrase XP
  - can also have labels that are strings (ordered finite sets) $\alpha = X_0 X_1 \cdots X_r$ of syntactic features
  - labels "licensor" ω and "licensee" $\bar{\omega}$
  - internal vertices labelled by $\{>, <\}$ following *head* of subtree



External and Internal Merge: combinatorial structure

- External Merge

$$\mathcal{E}(T_1 \otimes T_2) = \begin{cases} \bullet \wedge T_2 & T_1 = \bullet \\ T_2 \wedge T_1 & \text{otherwise,} \end{cases}$$

- Internal Merge

$$\mathcal{I}(T) = \pi_C(T) \wedge \rho_C(T)$$

C elementary admissible cut of T with $\rho_C(T)$ pruned tree containing root of T and $\pi_C(T)$ part severed by cut (elementary cut: tree not forest)

External Merge: domain

- $T[\alpha]$ for tree where head label starts with α
- domain of External Merge

$$\mathrm{Dom}(\mathcal{E}) = \mathrm{span}_{\mathbb{Q}}\{(T_1[\beta], T_2[\alpha]) \,|\, \beta = \sigma\alpha\}$$

- for $\alpha = X_0X_1\cdots X_r$ or $\alpha = \sigma X_0X_1\cdots X_r$ take $\hat{\alpha} = X_1\cdots X_r$
- External Merge

$$\mathcal{E}(T_1[\sigma\alpha], T_2[\alpha]) = \begin{cases} T_1[\widehat{\sigma\alpha}] \wedge_< T_2[\hat{\alpha}] & |T_1| = 1 \\ T_2[\hat{\alpha}] \wedge_> T_1[\widehat{\sigma\alpha}] & |T_1| > 1 \end{cases}$$

Internal Merge: domain

- tree $T[\alpha]$ where $\alpha = X_0 \cdots X_r$ or $\alpha = \sigma X_0 \cdots X_r$ or $\alpha = \omega X_0 \cdots X_r$ or $\alpha = \bar{\omega} X_0 \cdots X_r$
- domain of Internal Merge

$$\mathrm{Dom}(\mathcal{I}) = \mathrm{span}_{\mathbb{Q}} \left\{ T[\alpha] \,|\, \exists\, T_1[\beta] \subset T[\alpha], \text{with} \begin{array}{l} \beta = \bar{\omega} X_0 \hat{\beta}, \\ \alpha = \omega X_0 \hat{\alpha} \end{array} \right\}$$

- Internal Merge (Stabler's notation and admissible cuts notation)

$$\mathcal{I}(T[\alpha]) = T_1^M[\hat{\beta}] \wedge_> T\{T_1[\beta]^M \to \emptyset\} = \pi_C(T) \wedge_> \rho_C(T)$$

Note: there are issues with EM and IM in this form (unlabelable exocentric constructions)

domains under iteration (compounding problem)

- iterations of the internal merge, $\mathcal{D}_{N+1} \subset \mathcal{D}_N$ with $\mathcal{D}_N := \mathrm{Dom}(\mathcal{I}^N)$
  - N (complete) subtrees $T_1, \ldots, T_N$ in T
  - $T_1^M, \ldots, T_N^M$ maximal projections of subtrees (also complete subtrees)
  - subtrees T_i^M are disjoint.

$$\mathrm{Dom}(\mathcal{I}^N) = \left\{ T[\alpha] \,\middle|\, \exists T_1[\beta^{(1)}], \ldots, T_N[\beta^{(N)}] \right.$$

$$\text{with} \left\{ \begin{array}{l} (1), (2), (3) \text{ are satisfied} \\ \beta_0^{(1)} = \bar{\omega} X_0, \ldots, \beta_0^{(N)} = \bar{\omega} X_{N-1} \\ \alpha = \omega X_0 \omega X_1 \cdots \omega X_{N-1} \cdots \end{array} \right\} \Bigg\}$$

$$\mathcal{I}^{\#C}(T[X]) = \bigwedge^{1+\#C} \left(\pi_C(T)[\hat{\mathbb{Y}}] \; \rho_C(T)[\hat{X}^N] \right)$$

$$\pi_C(T)[\hat{\mathbb{Y}}] = T_N^M[\hat{\beta}^{(N)}] \cdots T_1^M[\hat{\beta}^{(1)}]$$

label $[\hat{\alpha}^N]$ of tree $\rho_C(T)$: what remains of original label X after removing initial terms $\omega X_0 \omega X_1 \cdots \omega X_{N-1}$

Internal Merge and coproduct and product in $\mathcal{H}_{LR}$

- coproduct $\Delta(T) = \sum T' \otimes T''$ decompositions: in each one side contains head of tree T (both if on boundary line of cut)
- if head of T and head of $\pi_C(T)$ same side then that side is in $\mathrm{Dom}(\mathcal{I})$
- so pieces of the coproduct are in $\mathrm{Dom}(\mathcal{I}) \otimes \mathcal{H}_{ling} + \mathcal{H}_{ling} \otimes \mathrm{Dom}(\mathcal{I})$
- other terms (different sides) are in $\mathcal{H}_{ling} \otimes \mathcal{H}_{ling}$
- T in $\mathrm{Dom}(\mathcal{I})$ and C elementary admissible determined by label condition; set of partitions

$$\mathcal{P}_{\mathcal{I}}(T) = \{ T = (T', T'') \mid \begin{array}{l} (h(T) \in T' \text{ and } h(\pi_C(T)) \in T') \text{ or} \\ (h(T) \in T'' \text{ and } h(\pi_C(T)) \in T'') \end{array} \}$$

- modify coproduct

$$\Delta_{\mathcal{I}}(T) := \sum_{(T',T'') \in \mathcal{P}_{\mathcal{I}}(T)} T' \otimes T''$$

and remains same outside of $\mathrm{Dom}(\mathcal{I})$

- with this modified coproduct $\mathrm{Dom}(\mathcal{I})$ is a *coideal* of the coalgebra $\mathcal{H}_{ling}$

$$\Delta_{\mathcal{I}}(\mathrm{Dom}(\mathcal{I})) \subset \mathrm{Dom}(\mathcal{I}) \otimes \mathcal{H}_{ling} + \mathcal{H}_{ling} \otimes \mathrm{Dom}(\mathcal{I})$$

- modified product $\star_{\mathcal{I}}$ on $\mathcal{H}_{ling}$: for trees T, T' where T' has $n+1$ leaves: decompositions where head of T in component grafted to head of T'

$$\mathcal{P}_{\mathcal{I}}(T, T') := \{(T_0, \ldots, T_n) \,|\, h(T) \text{ and } h(\pi_C(T)) \in T_{h(T')}\}$$

$$T \star_{\mathcal{I}} T' = \sum_{(T_0,\ldots,T_n)\in\mathcal{P}_{\mathcal{I}}(T,T')} \gamma(T_0, \ldots, T_n; T')$$

- $h(T \star_{\mathcal{I}} T') = h(T)$ as head of each $\gamma(T_0, \ldots, T_n; T')$ same as the head of T
- component $T_{h(T')}$ is in $\mathrm{Dom}(\mathcal{I})$ when $T \in \mathrm{Dom}(\mathcal{I})$ so $T \star_{\mathcal{I}} T'$ also in $\mathrm{Dom}(\mathcal{I})$
- $\mathrm{Dom}(\mathcal{I})$ right-ideal of algebra $(\mathcal{H}_{ling}, \star_{\mathcal{I}})$

$$\mathrm{Dom}(\mathcal{I}) \star_{\mathcal{I}} \mathcal{H}_{ling} \subset \mathrm{Dom}(\mathcal{I})$$

- not left-ideal

- form of Internal Merge $\mathcal{I}(T \star_{\mathcal{I}} T') =$

$$\sum_{(T_0,\ldots,T_n)\in\mathcal{P}_{\mathcal{I}}(T,T')} \pi_C(T_{h(T')}) \wedge_> \gamma(T_0,\ldots,\rho_C(T_{h(T')}),\ldots,T_n; T')$$

- internal merge $\mathcal{I}$ defines a right $(\mathcal{H}_{ling}, \star_{\mathcal{I}})$-module given by the cosets

$$\mathcal{M}_{\mathcal{I}} := \mathrm{Dom}(\mathcal{I})\backslash\mathcal{H}_{ling}$$

- combined with iteration of domains: $\mathcal{D}_{N+1}\backslash\mathcal{D}_N$ determines a coideal in the coalgebra $\mathcal{D}_{N+1}\backslash\mathcal{H}_{ling}$
- this gives a projective system of right-module coalgebras

$$\mathcal{M}_{\mathcal{I}^N} := \mathrm{Dom}(\mathcal{I}^N)\backslash\mathcal{H}_{ling}$$

- quotient right-module coalgebras or "generalized quotients" of Hopf algebras: suitable notion of quotients in the case of noncommutative Hopf algebras
- Conclusion: the implementation of Merge at the level of planar trees introduces significant complications in algebraic structure compared to free symmetric Merge