Motivating questions about formal models of phrase structure

Desiderata: projection, phrases/hierarchical structure, feature relations (agreement, licensing, selection), etc.

Claim: set-theory not well-suited for describing these desiderata (& spatial structure, if we are careful, is)

Traditional “spatial” categories, and their connection to syntax

Informal sketch of formal model proposed in paper

Some linguistic insights
MOTIVATION

- We start with some motivating questions:
  - How is feature combination like phrase-building, or
  - What is the “ambient” category of structured objects where these things can live together? i.e. what formal objects have properties like hierarchical structure, projection, feature relations...
  - Current answer: Chomsky’s *Categories and Transformations*, *Minimalist Inquiries*, *On Phases*, *Problems of Projection*, Fukui’s *Merge and Bare Phrase Structure*, Hornstein’s *A Theory of Minimal Syntax* all use set-theoretic objects, implying that set-theoretic structure is this “ambient” category
  - I argue in the paper that this structure is not “good” for describing basic syntactic properties, like projection, feature-relations, substructures (constituents), etc.
Underlying mathematical modeling questions:

- When are two phrase markers isomorphic?
- What is a substructure of a phrase marker (i.e. how is a constituent a substructure of a BPS phrase marker)?
- Have not yet motivated “isomorphisms”; “substructures” are clearly motivated for describing phrase structure, as well as conditions like “extension” and “monotonicity” (output should have prior structure as substructure)

- What about the workspace? Do we need a new “layer” of formalism to describe it, like in Collins and Stabler’s *A Formalization of Minimalist Syntax*?
Doesn’t set theory already handle hierarchy, substructures, features, etc. with elementhood?

Not quite

Collins and Stabler 2011 already recognized that set-theoretic structure itself does not encode things like occurrences of elements

e.g. Let $A=\{x,y\}$; $B=\{u,v\}$, $C=\{x,z\}$, $K=\{A,\{C,\{A,B\}\}\}$

It does not make sense to ask set-theoretically if $A$ is a substructure of $K$ in one or two ways; simply, $A$ is an element of $K$, and also of $\{A,B\}$

Similarly, it does not make sense to say that there are two occurrences of feature $x$ set-theoretically; it is simply the case that $x$ is a feature of $A$ and $C$
Collins and Stabler have rediscovered a quirk of the relation between sets and graphs, presented in detail in Azcel’s *Non-well-founded sets*.

Non-isomorphic labeled graphs unambiguously refer to the same set \( \{\{\text{see,John}\},\text{John}\} \).

In other words, there is not a one-to-one correspondence between well-founded graphs and well-founded sets.

A tree representation requires a *choice* of representation that is not determined by the set theory.
Such a choice matters and goes beyond the structure of sets.

- In the left figure, there is only one graph-theoretic inclusion from \{John\} into \(K\), but in the right figure there are two distinct inclusions.

- Similarly for features.

This additional structure seems necessary, since constituents are neither subsets nor submodels (of the ambient set theory), but might be subgraphs (with canonical choice of graph representation).
Suppose we fix the “canonical tree graph-theoretic” representation.

The choice of translation of sets into graphs helps with general occurrence problem for pure sets, but...

Additional choices must be made for how to handle the urelements (elements which are not sets - i.e. the features).

Suppose we “check” features within our set-theoretic syntactic object - i.e. an element representing a feature changes when satisfied/marked invisible at interface/etc.

E.g. a in A changes to a’ when merging A and B.
Why (Not) Sets? Part II - Substructures & Isomorphisms

- Represent by $A'$ the set which is exactly $A$, with $a_k$ replaced by $a_k'$
- Extension: $A$ should be a substructure of $\{A', B\}$

A not element of $\{A', B\}$; not subset, etc.

- Must assume some structural isomorphism; e.g. subgraph

- I.e. there is some monomorphism including $A$ into $\{A', B\}$; otherwise, $A$ is not a substructure of $\{A', B\}$  
  Extension: $A$ should be a substructure of $\{A', B\}$

- Above subgraph inclusion doesn’t preserve the "identity/names" of the features
Identity of features too strong, or $A$ and $A'$ are related by no morphisms (a fortiori, they are not isomorphic), so there can be no way in which $A$ is a substructure of $\{A', B\}$

However, the fact that the feature is distinct from others determines graph-structure, so seems important

Relative equality of elements in a phrase marker might be structurally important

- e.g. $\{X, \{X, Y\}\}$ and $\{A, \{A, B\}\}$ isomorphic (in one way!); neither isomorphic to $\{L, \{M, N\}\}$
• Interpretation of “=” (including identity of elements), determined independently of interpretation of elementhood

• Many possible distinct treatments of identity of elements as structure

• None determined by set-theoretic structure

• In fact, set-theoretic structure cannot tolerate many of these notions - translation into a category of related objects (like labeled graphs) is necessary
How to determine labeling?

New option: minimal search

- would need to make use of “occurrence” structure to distinguish different “occurrences” of feature in different lexical items, identical sets, etc.
- i.e. to define which occurrences count as being “within” sister to the head

Old solution: “The item that projects is the item that selects” Core Syntax /“Set-Merge typically has an inherent asymmetry” Minimalist Inquiries

- But, in our example, \{A',B\}, A' is not “A having been modified”, but rather, just a different set
- Extra-set-theoretic annotation required (such as annotated feature calculus, “trigger” function (Collins and Stabler), etc.); i.e. the set \{A',B\} doesn’t determine label by itself
Old-old solution (C&T): mark explicitly \{X,\{X,Y\}\}

- complications defining which sets are terms and which are labels, and what to say about sets like \{X,Y\}

- uses the "extra" structure - = as a predicate, occurrence-sensitive representation, etc.

- questionable that the predicate = is sufficient; if features are "updated", and a feature of X is "deleted/checked/etc." should be \{X',\{X,Y\}\}, which has different properties
General quirks:

- Lexical items must have only urelements as members, since containing a set as a member will make it non-terminal.

- How to represent adjunction? e.g. if we use pair-merge \(<a, b>\), but this is to be represented set-theoretically, it will be (depending on choice of representation), e.g. \({\{a\}, \{a, b\}}\), which is ambiguous.

- i.e. \(\text{Adjoin}(a, b) = \text{Merge}(\text{Merge}(a, a), \text{Merge}(a, b)))\)
How are feature relations represented? i.e. how do we know when two features agree, etc.?

What about the workspace?

- e.g. a single set-theoretic object \{X\} has two distinct syntactic interpretations: a workspace with one object Merge(X,X) and the syntactic object Merge(Merge(X,X), Merge(X,X))
Set-theoretic structure represents only elementhood; this by itself is not good for defining:

- occurrences
- substructures (constituents, extension, monotonicity)
- isomorphisms (like A isomorphic to a substructure of \{A',B\})
- structured lexical items
- Adjunction vs. merge
- relations between features
- labels
Finite Partial Orders are Finite Sober Spaces

- Trees are special cases of graphs, but also special cases of partial orders.
- Partial orders: a set $X$ with a relation $\leq$, which is
  - reflexive $x \leq x$
  - transitive $a \leq b$, $b \leq c$ imply $a \leq c$
  - antisymmetric $a \leq b$ and $b \leq a$ imply $a = b$
A monotone map $f: X \rightarrow Y$ between partial orders is a set function where $x \leq x'$ implies that $fx \leq fx'$

- e.g. "+1" on the natural numbers monotonically "slides them over"
Finite partial orders, and hence trees, are special cases of topological spaces.

A topological space is a set $X$ together with a collection of subsets of $X$, denoted $O(X)$, called the open sets, such that:

- If $U$ and $V$ are in $O(X)$, then their intersection is in $O(X)$.
- The union of any family of sets in $O(X)$ is in $O(X)$.
- $X$ and the empty set are in $O(X)$.
- These sets form a FDL.

A partial order $X$ can be turned into a finite topological space, by considering a subset $U$ of $X$ open iff for each $x$ in $U$ and every $y$ in $X$ such that $x \leq y$, $y$ is also in $U$. 
ORDER THEORY

- In a tree with the root as minimal element, and leaves as maximal, each constituent $K$ is an open subset of the space.

- A forest is a finite partial order where the collection of elements below any particular element is a linear order.

- A tree is a forest with a unique minima.
When X is sober (when no “double vision”), the lattice O(X) is “good enough” to determine the set X and the space.

The lattice of opens of a partial order
A function \(1 \rightarrow X\) picks out a point of \(X\)

\(O(1)\) is the “true false” lattice

Lattice homomorphisms from a lattice \(D\) to \(O(1)\) correspond to “points”

The points return the original space

FDL and FPos are dually equivalent

i.e. for every \(f: X \rightarrow Y\) monotone; \(F: O(Y) \rightarrow O(X)\) FDL morphism, and vice-versa
Notice that a "point" corresponds in the FDL to "places where that point occurs; smallest correspond to dominated constituent"
A morphism of finite lattices preserves $\land$, $\lor$

- corresponds to arbitrary monotone functions

Every FDL also has a $\to$ operation; an FDL morphism is "Heyting" if it preserves $\to$

- corresponds to open monotone functions

A map between partial orders $f: X \to Y$ is called an open map if the image $f(U)$ of any open subset $U$ of $X$ is open in $Y$

A function $f$ between trees is constituent preserving iff it is open

We call a function between trees $f: X \to Y$ constituent-preserving iff every constituent $K$ of $X$ is taken to a constituent $f(K)$ of $Y$, and constituent inclusion ordering is preserved
What does the $\rightarrow$ operation do?

We call $a \rightarrow 0$, where 0 is the bottom element, the pseudocomplement, and it is denoted $\neg a$.

Viewing an FDL as a space, and an object $a$ as an open subset, $\neg a$ is the largest open subset disjoint from $a$.

For $X$ a tree, $K$ a constituent, $\neg K$ is an open subset of $X$, and its connected components are exactly the constituents c-commanding $K$ using the definition:

A constituent $V$ c-commands a constituent $K$ if every node properly containing $V$ properly contains $K$. 
A constituent-preserving map (open map) preserves c-command “up to the image”; formally:

- If $X$ and $Y$ are trees, and $f:X \to Y$ is an open map, and $V$ is a constituent c-commanding $K$ in $Y$, then the connected components of $f^{-1}(V)$ are exactly the constituents whose images are contained in $V$ which c-command some connected component of $f^{-1}(K)$.
The preimage of \{him\} is the open set corresponding to “your friend”, which has just one connected component.

The connected components of the preimage of each constituent c-commanding “him” c-commands “your friend”
In this way, the category of finite partial orders with open maps captures constituents and c-command structurally (are structures preserved under open maps)

Constituents are substructures

"Coproducts" of partial orders are (universally defined) generalizations of "disjoint unions" of partial orders.

Intuitively, the coproduct $X+Y$ of $X$ and $Y$ simply "sets the two partial orders next to each other", where elements in $X+Y$ are ordered as they are in the subspaces, with no order relations between elements of $X$ and $Y$

Captures notions like "workspace" nicely
The set of nodes projected from (the same occurrence of) the same head form an interval. We are usually interested in trees where these intervals partition the tree, such that each interval has a leaf.

Any partition on a space induces a quotient map (a universal construction on that space)
A picture of a quotient

The quotient of a tree by projection orders the heads by “domain” (as in “minimal domain”)

The minimal domain is (roughly) the nodes immediately dominated by a node

- e.g. [the] is in the minimal domain of [feed], but [dog] is not

Slogan version: the tree of dependencies (domains) is the constituency tree “modulo projection”
DERIVATIONS: BIG PICTURE

- Partial orders capture dependency and constituency nicely
- Can add data to keep track of projection
- Would need to add data if features were added to “group” the features and label into “chunks”
- We would like to instead come up with a generalization of partial orders, which already has many of these properties
  - and analysis of structural relations between features and phrases (and projection, (minimal) domains, etc.) is straightforward
- The generalization is that we should make partial orders “dynamic” (objects viewed as continuously varying)
The idea is that a “continuously variable” partial order captures the notion of a derivation - if a partial order is a dependency structure, a derivation is a varying dependency structure.

How?: recall that FDL and FPos are dually equivalent.

- A finite partial order $X$ can be viewed as a family of inclusions $O(X)$ between subsets (which themselves are finite partial orders).

- A derivation will also be described from the “dual” perspective: a lattice $O(\Delta)$ of finite partial orders, with not only inclusions $U \rightarrow V$ between them, but (in principle, general) continuous functions between them.
DERIVATIONS

- $O(\Delta)$ is defined to have abstracted versions of "intersection" and "union"
  - "unions" generalized to "colimits" - a formal (universal) method of "gluing up" a collection of objects connected by morphisms
  - "intersections" generalized to "pullbacks" - a formal (universal) method of detecting how morphisms $X \rightarrow Z$ and $Y \rightarrow Z$ "overlap in $Z$"
- Derivations form spatial objects this way; they have generalizations of continuous functions between them and form a category
The category of finite partial orders embeds into the category of derivations fully and faithfully (they are the ‘static’ derivations)

There are best ways to ”reverse” this inclusion on the left and right (find best approximate ‘static’ derivations)
A point $1 \rightarrow \Delta$ is an element of the space $\text{pt}\Delta$.

Just like with finite partial orders, a point $x:1 \rightarrow \Delta$ corresponds with an (upward-closed) family of partial orders $X_i$ of $O(\Delta)$, with a specified element $x_i$ of $X_i$, for each stage in the filter.

This can be thought of as “the steps containing the abstract point $x$ and all of its images under operations in $O(\Delta)$”.

If $y \leq x$ are two points in $\text{pt}\Delta$, then necessarily the collection $Y_i$ of “stages where (an image of) $y$ occurs” is a subset of the collection $X_i$.

When each $y_i$ in the space $Y_i$ is exactly the $x_i$ in the corresponding $Y_i = X_i$, we say that $y$ is a projection of $x$. 
There is a natural “closeness” of points if they are in the same “step” of the derivation (“same space”), and it is straightforward to determine the derivation morphisms which preserve this “closeness” (which shows how “steps” are encoded in the derivation).

Similarly, we can generalize roots of a tree/forest (minima) to “local minima” of “close” parts of the derivation (points of the derivation which exist in the same step).

This gives a notion of label or term (root of a “step”).

We can also generalize open maps from the case of finite partial orders, to define open maps between derivations.

Gives a notion of “constituent-preserving”
A visualization of a closeness-preserving, label preserving, open morphism
It turns out that there are universal constructions corresponding to “adjoin this root to another”

An identical universal condition can also be used to abstractly “adjoin one feature to another”, or similarly, “identify (check)” two features
A picture of “phrasal merge/adjoin” and “feature license/adjoin”

In the paper, we construct canonical functors which take a derivation $\Delta$ with “current stage” $D$, which we would like to extend along an “operation” (continuous map) $f:D \rightarrow Z$; surjectivity of $f$ can be thought of as “inclusiveness”, and is induced in the cases constructed in the paper.

We functorially construct a derivation $\Delta^f$, which is “extensive” as it has $\Delta$ as an open subderivation, and $Z$ is the new “current stage”
Similar universal constructions can be used to describe phrasal merge/adjunction, but also feature agreement (local and long-distance), checking, licensing, etc.

These have “nice” spatial properties - e.g. specification is really move/re-merge+agree, where identically long-distance agree is like specification, without the movement (involves identical feature configuration requirements)

Once we extend a derivation along an operation, we can interrogate how “connected” the (projections of the) features have become
At a stage $Z$, we can ask which lexical terms map into $Z$, giving a picture of which lexical items have been merged up to the stage $Z$

For each abstract point $x$ in $Z$, we can find $U_x$, the stage where $x$ was inserted, and $U^Z_x$, the projection of $x$ at $Z$.

We think of these as the minimal and maximal projections of $x$ relative to the stage $Z$. 
We can form a 2x2 comparison of minimal and maximal projections of two lexical items $x$ and $y$ at $Z$.

 Leads to 6 potential degrees of connectivity.

 When $x$ is in the minimal domain of $y$ at $Z$, then $x$ is at least adjunct-connected to $y$.

 Additional licensing - specifier.

 Selection - complements.

 (can talk about selected specifiers in ? period)
- Workspaces fall out straightforwardly

- The category of derivations has coproducts, which are like generalized disjoint unions

  - E.g. if we want to merge $\Delta$ and $\Gamma$ by extending them to a new stage $Z$ (induced universally by some set of conditions), we can form their coproduct $\Delta + \Gamma$, followed by an extension $(\Delta + \Gamma)^f$

  - Then, if we have a derivation $\Delta + \Gamma + \Xi$, and the inclusion into an extension $\Delta + \Gamma \rightarrow (\Delta + \Gamma)^f$, there is an induced map $\Delta + \Gamma + \Xi \rightarrow (\Delta + \Gamma)^f + \Xi$
WHAT DID WE LEARN?

- Derivations are partial orders which have been generalized to include the notion of “change”

- This is sufficient to define projection, terms, substructures, isomorphisms, coproducts (workspaces), etc.

- Phrases and features can live in the same geometric world; phrasal and featural “combinations” are induced as universal solutions to similar geometric “problems” (generalized “adjunction” and “identification”)

- The inner geometry of derivations not only describes terms and phrases, but also the relations between features; these relations together determine headness, specifier/adjunct/complement relations, agreement relations, etc.
Set theory struggles with similar constructions, and requires idiosyncratic choices of representation of sets as related structured objects to define most of the basic notions.
There is a relation between colimits (generalized coproducts and coequalizers) and sheaves, sets whose values vary over parts of the object. These sheaves can be thought of as “variable sets of indices” and they are associated to spaces with “glued together copies of parts of an object.” This gives a notion of “copies.” Derivations actually form a 2-category - there are morphisms between morphisms; in particular, there are deformations of self-maps of derivations which can be used to represent “closure operator-like” endomorphisms which describe phases, etc.