We propose a category $\mathbf{Der}$ of generalizations of partial orders which we use to model syntactic derivations. The objects of $\mathbf{Der}$ are ‘variable’ partial orders which keep track of derived syntactic objects (SOs), parameterized over a sequence of steps. This gives information relating the structure of each derived SO to each subsequent SO, i.e. the ‘structural changes’. $\mathbf{Der}$ characterizes isomorphisms between derivations (correspondence between steps, derived objects, and dependencies introduced), embeddings of derivations (with inclusions of constituents as a special case), and the relationship between derivations and their derived objects. We show that this category has many good properties, including the existence of disjoint unions of derivations (used to model workspaces and structural changes induced by syntactic operations) and products of derivations (which can be used to model relations on derivations). We also give a form of pushout methods for describing syntactic operations and contexts of application, which we use to recursively construct derivations. We formalize ‘overlap’ or ‘degree of connectivity’ introduced by operations, and we give interpretations of this data which characterize grammatical relations. Finally, we construct toy grammars as collections of derivations closed under some set of operations, and we define equivalence and isomorphism of grammars in terms of the structure of the lexical items they have and derivations they can construct. We then briefly discuss difficulties other mainstream computational methods have in capturing similar properties, and ways that derivations allow for more general constructions like feature geometry and feature sharing.

Contents

1. Informal discussion of derivations 3
   1.1. The objects ........................................... 3
   1.2. Morphisms of derivations .......................... 6
   1.3. $\mathbf{Der}$ as a category ............................ 8
       1.3.1. Structural properties of $\mathbf{Der}$ .......... 8
   1.4. Spatial structure: constituency and c-command ..... 12
1.5. Applications: giving computations articulated structure .......................... 15
  1.5.1. Grammatical operations .................................................... 15
  1.5.2. Recursive construction of derivations and equivalence of grammars .. 17
  1.6. Grammatical relations .......................................................... 18

2. Formal theory of derivations ...................................................... 20
  2.1. Formal definition of Der ...................................................... 21
  2.2. Qualitative properties of derivations and derived objects ................. 22
  2.3. Der as a concrete category ................................................... 25
  2.4. Separation .............................................................................. 35
     2.4.1. Forests and trees in FPos ................................................. 35
     2.4.2. Forests and trees in Der .................................................. 38
  2.5. Constituency ........................................................................... 45
  2.6. Summary .................................................................................. 48

3. Formal theory of grammars and syntactic operations .............................. 49
  3.1. Syntactic operations ............................................................... 51
     3.1.1. Motivating example .......................................................... 51
     3.1.2. Formal theory of operations .............................................. 56
  3.2. Equivalence of grammars ......................................................... 65
  3.3. Grammatical relations ............................................................. 66
  3.4. Toy grammatical rules ............................................................. 74
  3.5. Compilations of rules .................................................................. 83
     3.5.1. Categories of sequences ..................................................... 84
     3.5.2. Sequences of operations and compilations ......................... 85
     3.5.3. The rule generated by a sequence of rules ......................... 87
     3.5.4. A note on sequences .......................................................... 88
     3.5.5. Examples ........................................................................... 88
  3.6. Summary ................................................................................... 90

4. Copying ......................................................................................... 91
  4.1. Copying with trees - toy example .............................................. 93
  4.2. Copies in Der ............................................................................ 94
  4.3. Generation, compilation, and properties of rules for 1Der ................ 96
  4.4. Copy-data and chains ............................................................... 101

5. Tethering objects of Der to Minimalist Grammars .................................. 102

6. OTHER EXAMPLES? ....................................................................... 102

A. Proofs .......................................................................................... 102
1. Informal discussion of derivations

We give an overview of derivations, their properties, and linguistic analysis which can be performed with them.

1.1. The objects

We define a derivation as a finite set together with a partial ordering and a kind of ‘representation functor’ or ‘atlas’ \((X, \leq, \omega)\). We write \(\Delta\) for a derivation and \(|\Delta|\) for its underlying set \(X\).

\(|\Delta|\) consists of every feature and label of each step of the derivation. The partial ordering \(x \leq y\) is read ‘\(x\) derivationally dominates \(y\)’ or ‘\(x\) (derivationally) depends on \(y\)’. The relationship between \(\leq\) and \(\omega\) subdivide \(\leq\) into some special cases which have especially intuitive meaning.

All relations in the following list are reflexive and antisymmetric.

1. \(x \preceq y\), read ‘\(y\) is close to \(x\)’, which roughly means that \(x\) and \(y\) are elements in the same step, and \(x\) dominates \(y\) in the traditional sense in that step.

2. \(x \sqsubseteq y\), read ‘\(x\) is a projection of \(y\)’, which roughly means that \(x\) is just the occurrence of \(y\) at some (potentially) later stage. This relation is transitive.

3. General derivational dominance can be seen as a sort of combination of the two. Roughly \(x \leq y\) if \(y\) goes on to be dominated by \(x\) at some stage. This relation is transitive.

See Fig. 1.
Figure 1: $x \preceq y$, $x \sqsubseteq y$, and general $x \leq y$, respectively. The boxes loosely represent steps of the derivation, and arrows between them maps relating prior objects to later ones.

To make sense of $\omega$, it is necessary to use an equivalence between finite partial orders and certain topological spaces. Call a subset $U \subset |\Delta|$ of a derivation open if it is closed under derivational dominance in that $x \in U$ and $x \leq y$ imply $y \in U$. The collection of open sets of $X$ and the partial ordering $\leq$ determine each other and give rise to an equivalence between partial orders and topological spaces of this form.\footnote{In fact, for finite sets $X$, topologies on $X$ and preorderings (transitive, reflexive relations) on $X$ are in exact correspondence. There is actually an isomorphism of categories between finite preorders and finite topological spaces, in that order-preserving functions correspond exactly to continuous functions, using the correspondence between preorders and topologies. The partial orders naturally correspond to spaces meeting a separation condition known as sobriety.} See Fig. 2.
\( \omega \) assigns an order-preserving surjection \( U \to \top_U \) to each open subset \( U \subset |\Delta| \) in a compatible way. We call \( \top_U \) the ‘derived syntactic objects (SOs) at \( U \)’. \( \top_U \) can be thought of as a dependency tree between the labels and features of the lexical items (or their images under syntactic operations performed in the derivation). \( U \to \top_U \) takes each point \( x \in U \) to the element corresponding to it in the derived object at that step. This must meet two compatibility conditions.

1. For each inclusion \( V \subset U \) of open subsets of \( |\Delta| \), there exists an order-preserving function \( \top_V \to \top_U \), thought of as representing the ‘net structural change’ in the derived SOs from the portion \( V \) of the derivation to \( U \), such that the following diagram commutes.

\[
\begin{array}{ccc}
V & \longrightarrow & \top_V \\
\downarrow & & \downarrow \\
U & \longrightarrow & \top_U
\end{array}
\]

Such a map, if it exists, is determined uniquely. For any point \( x \in V \), the point \( x_V \in \top_V \) representing \( x \) in the derived SO \( \top_V \) must get mapped to \( x_U \in \top_U \), the point representing it in the SOs at the ‘later’ stage \( U \) if the diagram is to commute. We require that this mapping is well-defined and order-preserving.

2. If \( \{U_i\}_{i \in I} \) are open subsets such that \( \bigcup_{i \in I} U_i = U \) (which is always open), associated to derived SOs \( \top_{U_i} \), the derived SOs \( \top_U \) at \( U \) can be recovered as an ‘amalgamation’ of the prior steps \( \top_{U_i} \).

In \( \S 2.1 \), we naturally characterize both of these conditions as saying that \( \omega \) is a sheaf on the space \( (|\Delta|, \leq) \). There, we will make condition (2) precise.
While the general (families of parts of) derived objects $\top_U$ might be somewhat abstract, many of them have very concrete interpretations. For any point $x \in |\Delta|$, we denote by $U_x$ the set of $x$ and all points it derivationally dominates. The derived SO $\top_{U_x}$ is a partial order with root $x_{U_x}$, and can be thought of as the partial order of points dominated by $x$ in the derived SO at the step $x$ occurs in. Condition (2) tells us that it is sufficient to define $\omega$ on just open sets of the form $U_x$, since they determine $U \to \top_U$ for all $U$. For any $U$, $\top_U$ can be constructed as an amalgamation of the derived constituents $\top_{U_x}$ at each point $x \in U$.

In §2.4, we classify derivations which are ‘tree-like’. See Fig. 4. Not only does the internal structure of a derivation $\Delta$ partition it into steps if it is separated, but the ordering between steps is determined by the derivational ordering of elements within each step. In this way, we will see morphisms $\phi : \Delta \to \Gamma$ which preserve relative stephood as mapping steps to steps in an order-preserving way, but we do not need to state these steps and their ordering explicitly as they are intrinsic to the structure of $\Delta$ and $\Gamma$.

More generally, much of the basic theory of derivations developed in §2.1 characterizes many linguistically useful properties such as projection, constituency, and grammatical relations in a manner which is invariant under isomorphisms of derivations. Moreover, the theory of derivations allows us to do this for ‘dependency partial orders’ and ‘dependency-preserving structural changes’ in general, regardless of the particular formulations of grammatical operations or structure of primitive and derived SOs. This will allow for a general theory of rules, generalizing over phrase-building, agreement, and other ‘dependency-introducers’, which we give in §3.

1.2. Morphisms of derivations

We equip the class of derivations with morphisms $\phi : \Delta \to \Gamma$. We write $\Delta = (|\Delta|, \leq_{\Delta}, \omega_{\Delta})$ and $\Gamma = (|\Gamma|, \leq_{\Gamma}, \omega_{\Gamma})$. A morphism is a function $|\phi| : |\Delta| \to |\Gamma|$ which preserves derivational dominance and the ‘structure sheaves’ $\omega_{\Delta}$ and $\omega_{\Gamma}$.

This first condition simply means that if $a \leq_{\Delta} b$, then $\phi(a) \leq_{\Gamma} \phi(b)$. We say that the second condition is that $\phi$ is ‘locally continuous’. While this has an elegant abstract definition, we initially describe an important property is has with respect to comparing the SOs derived in $\Delta$ and $\Gamma$.

Let $U \subset \Delta$ and $V \subset \Gamma$ be open subsets such that $\phi(U) \subset V$. In other words, all the points occurring in the derivation $\Delta$ ‘up to $U$’ are being compared to points in the derivation $\Gamma$ within $V$. Both $U$ and $V$ are associated to derived SOs $\top_U$ and $\top_V$ via order-preserving surjections $U \to \top_U$ and $V \to \top_V$ by their respective ‘atlases’ $\omega$ and $\tau$. Notice that if there is a ‘completion’ of the following diagram, it is unique:

$$
\begin{array}{ccc}
U & \xrightarrow{\phi} & V \\
\downarrow & & \downarrow \\
\top_U & \xrightarrow{\text{-----}} & \top_V
\end{array}
$$
For any point \( p \) in \( \top_U \), there is some \( x \in U \) such that \( x_U = p \). If the diagram is to commute, the map \( \top_U \to \top_V \) must take \( x_U \) to \( \phi(x)_V \). We again require that this gives a well-defined function and that it is order-preserving. We say that a function between \( \Delta \) and \( \Gamma \) with this property is ‘locally continuous’. We will sometimes write the induced map as \( \phi^V_U \).

Again by the sheaf condition (2), it is sufficient to check that such maps exist just for the constituents. That is, it is sufficient to check that for every \( x \in \Delta \), there exists an order-preserving function \( \top_U x \to \top_U \phi(x) \) completing the diagram:

\[
\begin{array}{ccc}
U_x & \xrightarrow{\phi} & U_{\phi(x)} \\
\downarrow & & \downarrow \\
\top_U x & \longrightarrow & \top_U \phi(x)
\end{array}
\]

Together, we say that \( \phi \) is a morphism if it preserves derivational dominance and is locally continuous.\(^2\) A derivation morphism is then a function \( \phi : \Delta \to \Gamma \) between derivations which preserves derivational dependence that naturally induces piecewise comparisons between derived constituents.

However, they also preserve much more structure. For example, if \( a \sqsubseteq b \) is a projection relation, then \( \phi(a) \sqsubseteq \phi(b) \) is a projection relation. In fact, this is a special case of much richer structure which is preserved by morphisms. Let \( U \subset U' \) be open subsets of a derivation \( \Delta \) corresponding to derived objects \( \top_U \) and \( \top_{U'} \) together with the ‘net structural change map’ \( \top_U \to \top_{U'} \) and similarly for \( \top_V \to \top_{V'} \) in \( \Gamma \). Then the following diagram commutes.

\[
\begin{array}{ccc}
\top_U & \xrightarrow{\phi^V_U} & \top_V \\
\downarrow & & \downarrow \\
\top_{U'} & \xrightarrow{\phi^V_{U'}} & \top_{V'}
\end{array}
\]

For example, \( \top_U \) and \( \top_V \) might be derived noun phrases, while \( \top_{U'} \) and \( \top_{V'} \) are derived determiner phrases, with the net structural changes being the maps taking the noun phrases into the complement position of the determiner phrases. See Fig. 3.

\(^2\)From the sheaf-theoretic viewpoint, this is the typical definition of a morphism of spaces with structure sheaves, such as smooth mappings in differential geometry, or morphisms of schemes.
1.3. Der as a category

A composition of derivation morphisms is itself a derivation morphism, and identity functions are identity morphisms with respect to composition, and so Der, consisting of derivations and derivation morphisms, forms a category. This category has many good and useful properties, which we describe in sequence.

An immediate one is that we get a notion of isomorphism. An isomorphism is a morphism \( \phi : \Delta \to \Gamma \) such that there is a morphism \( \psi : \Gamma \to \Delta \) such that \( \phi \circ \psi \) is the identity on \( \Gamma \) and \( \psi \circ \phi \) is the identity on \( \Delta \). Isomorphisms of derivations are bijections which exactly preserve derivational dominance and give isomorphisms between derived objects at each step with isomorphic relations between steps.

1.3.1. Structural properties of Der

The functor \( |·| : \text{Der} \to \text{FSet} \) taking \( \Delta \) to \( |\Delta| \) allows us to view derivations as ‘structured sets’. Der internally has many good properties, many of which have set-theoretic correlates. The major result is that this category is finitely complete and cocomplete. This induces many good properties. Among them:
1. For any pair of derivations $\Delta, \Gamma$, there exists a product derivation $\Delta \times \Gamma$. The underlying set of points of this derivation are pairs $(d, g)$ with $d \in |\Delta|$ and $g \in |\Gamma|$. The projection maps $\pi_\Delta : \Delta \times \Gamma \to \Delta$ and $\pi_\Gamma : \Delta \times \Gamma \to \Gamma$ sending $(d, g)$ to $d$ and $g$, respectively, are derivation morphisms, and if $a : \Pi \to \Delta$ and $b : \Pi \to \Gamma$ is any pair of derivation morphisms from a fixed derivation to $\Delta$ and $\Gamma$, there is an induced function $f : \Pi \to \Delta \times \Gamma$ taking $p \in |\Pi|$ to $(ap, bp)$, and this function is a morphism. Furthermore, $(a, b) \leq (a', b')$ iff $a \leq a'$ in $\Delta$ and $b \leq b'$ in $\Gamma$; $(a, b) \sqsubset (a', b')$ iff $a \sqsubset a'$ and $b \sqsubset b'$; and $(a, b) \preceq (a', b')$ iff $a \preceq a'$ and $b \preceq b'$. Such products can be used to describe relations on derivations in terms of derivational structure.

2. For any pair of derivations $\Delta, \Gamma$, there exists a sum derivation $\Delta + \Gamma$. A sum of derivations can just be thought of as the derivation consisting of the two ‘sitting next to each other’. The underlying set of points of this derivation is the disjoint union of points of $\Delta$ and $\Gamma$. There are coprojection subderivation embeddings $\rho_\Delta : \Delta \to \Delta + \Gamma$ and $\rho_\Gamma : \Gamma \to \Delta + \Gamma$ given by the subset inclusions into the disjoint union. If $a : \Delta \to \Pi$ and $b : \Delta \to \Pi$ is any pair of morphisms to a fixed derivation $\Pi$, there is an induced function $f : \Delta + \Gamma \to \Pi$ sending $d \in \Delta \subset \Delta + \Gamma$ to $a(d)$ and $g \in \Gamma \subset \Delta + \Gamma$ to $b(g)$, and this function is a morphism. Furthermore, $x \leq y$ iff $x, y \in \Delta \subset \Delta + \Gamma$ and $x \leq y$ in $\Delta$, or $x, y \in \Gamma$ and the relation holds there; similar results hold for closeness and projection.

Sums of derivations are useful for representing workspaces and the behavior of rules. In the simplest case of partial orders, we can think of a map $f : A + B \to Z$ (especially when $f$ is surjective) as an operation taking in the syntactic objects $A$ and $B$, and mapping them to a syntactic object $Z$. By definition, such a map corresponds to a pair of maps $f_A : A \to Z$ and $f_B : B \to Z$, indicating how the structure of $A$ and $B$ are each mapped into the SO produced, $Z$. When $T_i$ are all trees, the sum $\bigoplus_i T_i = T_a + T_b + \ldots$ can be thought of as a workspace consisting of the SOs $T_i$, and it uniquely decomposes back into those trees as its connected components. Similar results hold for derivations.

3. For any subset $S \subset |\Delta|$, there is a unique ‘subderivation structure’ on $S$. This means that there is a unique derivation $S$ with underlying set $S$ such that the inclusion $S \hookrightarrow |\Delta|$ gives a morphism $S \hookrightarrow \Delta$, and any function $f : \Gamma \to S$ whose composite $\Gamma \to S \hookrightarrow \Delta$ is a morphism must itself be a morphism. In this sense, on any subset $S$ of a derivation $\Delta$, there is an induced structure on $S$ thought of as ‘restricting the structure of $\Delta$ down to $S$’. For each point $x \in S$, denoting $S_x = \{y \in S \mid x \leq y\}$ and $U_x = \{y \in \Delta \mid x \leq y\}$, the morphism $S \hookrightarrow \Delta$ induces $\top_{S_x} \twoheadrightarrow \top_{U_x}$, an embedding of partial orders.
While this characterization is ‘concrete’ (referring to properties of the underlying set function), we prove that it can also be described internal to the category $\text{Der}$ as an effective monomorphism.

These substructures are also well-behaved categorically: for example, a substructure embedding composed with a substructure embedding is itself a substructure embedding. They also give a factorization system for homomorphisms: every derivation morphism $\phi : \Delta \to \Gamma$ factors as a surjective morphism onto the set-theoretic image, followed by the substructure embedding, i.e. $\Delta \to \text{im}(\Delta) \hookrightarrow \Gamma$, and this factorization is essentially unique.

Figure 4: Informal picture of a derivation representing a DP.
Fig. 5: We give three examples of subderivations whose associated inclusions into the derivation in Fig. 4 are substructure embeddings in Der.

Fig. 5 gives examples of subderivations of the derivation in Fig. 4. In the top row, the left derivation is a subderivation which we call a ‘derivational constituent’, given by a constituent-preserving embedding. It represents all steps up to some ‘complete step’ in the containing derivation.

The right derivation is a more general ‘open subderivation’. It represents some constituent of a derived object, and all parts of prior steps which went into building it. For example, in the final step of the original derivation, the ‘dog’-phrase has been selected for its N feature by a feature in the lexical item ‘the’. The associated open subderivation will represent this by showing that the selectional feature participated in the construction of this derived constituent. We will look at some applications of having these subobjects to describing copying of portions of derived objects, along the lines of the ideas in
The bottom derivation is a subderivation in a more abstract way. The intended inclusion sends the lexical items ‘the’ and ‘dog’ isomorphically to the lexical items in the derivation in Fig. 4, while the final derived object in this subderivation is to be viewed as a subobject of the final derived object in Fig. 4. This is a subderivation, but not an open one - we have ‘skipped steps’. The idea is that we can represent in what sense the sequence of steps for building ‘the dog’ is a subsequence of steps for building ‘the red dog’, skipping adjunction of ‘red’. However, it is slightly more nuanced than this - each derived object in the subderivation is associated to a subobject of the derived objects in the larger one. The inner structure of steps in the subderivation also correspond to operations in the larger one - it is still true for example that the nominal feature of ‘dog’ is selected by a feature of ‘the’. The subderivation embedding represents this fact.

While this last example is a more ‘intuitive’ example of a subderivation which is not open, \textbf{Der} has subderivation structures on arbitrary subsets, which can be used to describe where certain configurational relationships hold between a set of points in a derivation.

More computational models such as Stabler’s Minimalist Grammars have a method for describing a relation between a derivation and a ‘subderivation’ of the first kind only (being an ‘operand’). We show in SECTION that in general, the constituency relation usually defined in these models is ill-behaved. It is not possible to express the other kinds of relationships between derivations and their more general subderivations in these models. We also show that similar issues hold for set-theoretic Bare Phrase Structure.

1.4. Spatial structure: constituency and c-command

We have referenced an equivalence between partially ordered sets and certain topological spaces, given by associating each partially ordered set \((P, \leq)\) to the space given by the set \(P\) and collection of upward-closed sets of \(P\). This actually gives rise to an equivalence of categories between \textbf{FPPos} of finite partial orders and order-preserving maps and \textbf{FSSpc} of finite sober spaces and continuous functions, a restriction of an equivalence between \textbf{FProset} of finite preorders and \textbf{FSpc} of all finite topological spaces. This roughly means that any structural statement about a space corresponds to a statement about it as an order, and vice-versa.

An interesting subclass of continuous functions are the open maps. A map \(f : X \to Y\) between spaces is open if the image \(f(U)\) of every open set \(U \subset X\) is itself open in \(Y\). When \(X\) and \(Y\) are trees, this is equivalent to saying that every constituent \(K \subset X\) is taken surjectively to a constituent \(f(K)\) and a containment of constituents \(K \subset C\) in \(X\) implies a containment of constituents \(f(K) \subset f(C)\) in \(Y\). Such maps directly compare the constituents of \(X\) to those of \(Y\). We give a result that c-command is a kind of invariant under such constituent-preserving maps between trees, and generalize it to tree-like derivations.

We call a derivation morphism \(\phi : \Delta \to \Gamma\) open if its underlying order-preserving func-
tion is an open map. Such a map still associates to each derived constituent of \( \Delta \) a derived constituent of \( \Gamma \). Concretely, for any open \( D \subset \Delta \), its direct image \( \phi(D) \) is open, and the associated map between derived objects \( T_\Delta \rightarrow T_{\phi(D)} \) is a surjection. The open embeddings correspond exactly to the inclusions of families of constituents of derived SOs into a derivation. An example of an open embedding which is not a ‘derivational constituent’ was given by the upper right example in Figs. 4 and 5.

We will define a ‘closeness’ relation in §2.1 between points which is invariant under isomorphism, where \( x \preceq y \) is close if \( y \) is essentially in the same derived object as \( x \) and is dominated by it. We call a derivation morphism \( \phi : \Delta \rightarrow \Gamma \) coherent if it preserves closeness, i.e. if \( x \preceq y \) occur in the same step, then they are still kept ‘close’ in \( \Gamma \). We can define the labels of derivations structurally as minima with respect to closeness, indicating that they are the minimal object in some step (i.e. the ‘root’ of it). We call these elements terms, and a morphism term-preserving if it takes terms to terms. Finally, we call a map constituent-preserving if it is open, coherent, and term-preserving. An example of a map which is term-preserving and coherent but not open is given by the inclusion of the derivation of ‘the dog’ into the derivation of ‘the red dog’ given in Figs. 4 and 5. An example of an open embedding which is not term-preserving is given by the derivation in the upper right corner.

An example of a constituent-preserving map which is not just an embedding is given by the ‘pronoun substitution’ map in Fig. 6. We can view comparisons of derivations with ‘complete’ constituents to versions with variables or empty categories similarly.
Figure 6: An informal picture of a constituent-preserving map showing some features

The constituent-preserving embeddings into a derivation are exactly the inclusions of families of derivational constituents. In tree-like cases, the constituent-preserving embeddings of ‘connected’ derivations into a derivation \( \Delta \) are exactly its derivational constituents. In each case, we can capture various notions of maps which preserve properties of constituents ‘on the nose’, and the embeddings of that kind describe the constituents of that kind for any given derivation. These roughly break into two kinds - whole portions of the derivation up to some complete step, or all parts of the derivation feeding into some derived constituent at some step. The first is clearly a special kind of the second.

All of these ‘special’ morphisms (term-preserving, coherent, open, embedding, isomorphism) can be given abstract treatment internal to \( \textbf{Der} \), meaning they can be described using only morphisms (without reference to the underlying set or order structure) and hence can translated into any category equivalent to \( \textbf{Der} \), and the notions are preserved by that equivalence.
1.5. Applications: giving computations articulated structure

The ‘good’ categorical properties of $\text{Der}$ ensure that we can get reasonable information about isomorphisms of derivations, their subparts, their points, and constituents, and we also have many standard constructions on them such as products and sums. These constructions are useful for building new derivations out of existing ones. We have already given the example of constructing a workspace as a sum $W = \coprod_{i \in I} \Delta_i = \Delta_n + \Delta_b + \ldots$. Each summand $\Delta_i$ embeds into this workspace as ‘constituent’ of it. If the $\Delta_i$ are each ‘connected’, this $W$ uniquely decomposes back into them.

1.5.1. Grammatical operations

We suggested that an $n$-ary operation on SOs could be thought of as a morphism $f : A_1 + \ldots + A_n \to Z$, where $Z$ is the output object. The defining property of coproducts is that this morphism $f$ corresponds exactly to a family of morphisms $f_i : A_i \to Z$, corresponding to $A_i \hookrightarrow \coprod A_i \to Z$, which shows the structural ‘inclusion’ of $A_i$ into $Z$. Thinking of a syntactic operation on a tuple of objects as a morphism makes it easy to describe qualitative structural properties of them. For instance, requiring that $f$ be surjective can be thought of as requiring that it is inclusive (each element of output object arises from element of some input object); asking that each $f_i$ is an embedding says that the operation only introduces dependencies between separate objects, and is a weakened form of the extension condition. We will also give some methods for analytically deconstructing operations into component parts using these structural properties and comparing structural changes they induce. For example, merger of an XP into an agreement position might be decomposed as involving both ‘agreement’ and ‘phrasal attachment’; long-distance and local agreement can be seen as inducing the same structural change, but with different structural descriptions of where they can be applied.

We construct an example similar to the merge operation on dependency trees described in [Boston et al., 2010], viewed structurally. For any pair of trees $(T, R)$, there is an operation $f : T + R \to Z$ attaching the root of $R$ to the root of $T$. $f$ is inclusive in that every point of $Z$ comes from a point of $T$ or $R$. While $f$ is not a substructure embedding (there are order relations in $Z$ which are not in $T + R$), each map $f_T : T \to Z$ and $f_R : R \to Z$ is, showing that structure of each input object is embedded into the output object.

However, a morphism $f : A_1 + \ldots + A_n \to Z$ only describes a particular operation on a particular collection of objects. We are often more interested in how to assign these operations for any relevant tuple $(A_1, \ldots, A_n)$. Further, many syntactic rules, like AGREE, may actually take more than one value on a given pair of object $(\Delta, \Gamma)$ if applied to different context. Consider the following scenario: $X^0$ is merging with YP, such that $X$ has a probe, and YP has multiple equidistant goal features which are all closest to $X$ amongst potential goals. Distinct SOs can be formed by AGREE for each goal.

So rather, for a binary rule, what we would like is a subcategory $\mathcal{C} \subset \text{Der} \times \text{Der}$ of ‘contexts’ a rule can apply in and a functor $G : \mathcal{C} \to \text{Set}$ which assigns to each relevant
pair of derivations \((\Delta, \Gamma)\) in \(C\) a set \(G(\Delta, \Gamma)\) of morphisms \(h : \Delta + \Gamma \to Z\) which we consider ‘instances’ of the operation \(G\). Such morphisms correspond to maps \(\top_h : \top_\Delta + \top_\Gamma \to Z\) which take the sum of derived objects to a new object \(Z\). We will primarily be interested in rules \(G\) such that there is a basic ‘template’ pair of objects \((M_1, M_2) \in C\) and operation \(r : M_1 + M_2 \to X\) in which some sense ‘generates’ \(G\). We can think of this ‘generating rule’ as analogous to a phrase structure rule \(\phi \to \psi\) which generates many particular instances of replacements \(\ldots \phi \ldots \to \ldots \psi \ldots\). We think of \(C\) as putting a ‘geometric’ context restriction on the application of \(r\). In other words, the domain of the generating operation \(r\), together with the category \(C\), gives a presentation of the ‘structural description’ of configurations in SOs where the rule can apply. \(r\) itself gives the ‘structural change’ instruction.

In the given example, \(C\) can be a category of pairs \((\Delta, \Gamma)\) of some class of derivations such that \(\Delta\) and \(\Gamma\) (and hence \(\top_\Delta\) and \(\top_\Gamma\)) are both rooted, where morphisms \((a, b) : (\Delta, \Gamma) \to (\Pi, \Psi)\) are pairs of morphisms such that both preserve the respective root (and hence induce order-preserving functions \(\top_a : \top_\Delta \to \top_\Pi\) and \(\top_b : \top_\Gamma \to \top_\Psi\) which preserve the roots as well). The generating function \(r : \{a\} + \{b\} \to \{a < b\}\) attaches \(b\) to \(a\). A context for \(r\) is the unique \(C\)-morphism \(\{!a, !b\} : (\{a\}, \{b\}) \to (\Delta, \Gamma)\) associated to functions \(\{a\} \to \top_\Delta\) and \(\{b\} \to \top_\Gamma\) picking out the root of each. We use another colimit construction to apply \(r\) in the context \((!a, !b)\) called a \textit{pushout} which assigns to \((\Delta, \Gamma)\) the function \(h : \Delta + \Gamma \to Z\) associated to the function \(\top_h : \top_\Delta + \top_\Gamma \to Z\) attaching the root of \(\top_\Gamma\) to the root of \(\top_\Delta\). See Fig. 7.

The only generalization is that if there is more than one \(C\)-context to apply an operation \(r : M_1 + M_2 \to X\) in in an object \((\Delta, \Gamma)\) - i.e. more than one \(C\)-morphism \((M_1, M_2) \to (\Delta, \Gamma)\) - then there may be more than one operation \(h : \Delta + \Gamma \to Z\) associated to the operation \(r\). It is also possible that there are no good \(C\)-contexts to apply an operation \(r\) in in a given object, in which case the set of \(r\)-operations in context will be empty.

Figure 7: Example rule attaching one root to another.
1.5.2. Recursive construction of derivations and equivalence of grammars

We have given a method for describing \( n \)-ary (non-copying) grammatical rules. In particular, a binary rule can be thought of as a context \( C \subset \text{Der} \times \text{Der} \) together with a generating operation \( r : M_1 + M_2 \to X \) on a given \( C \)-object \((M_1, M_2)\). It induces a functor \( G_r : C \to \text{Set} \) which assigns to each \((\Delta, \Gamma) \in C\) a set of morphisms \( h : \Delta + \Gamma \to Z \) thought of as ‘\( r \) applied in some admissible context’, where \( Z \) is the ‘new’ SO produced.

The collection of morphisms from derivations to ‘new’ SOs can be made into a category \( \text{Der}/\text{FPos} \). We construct a functor \( \text{ext} : \text{Der}/\text{FPos} \to \text{Der} \) which takes any operation \( h : \Delta \to Z \) to a derivation \( \Delta^h \) which contains the original derivation as an open subderivation, but whose final derived object is \( Z \). Additionally, the open embedding \( \Delta \hookrightarrow \Delta^h \) is associated to the morphism \( \top_h : \top_\Delta \to Z \) in \( \Delta^h \). We call this ‘the extension of \( \Delta \) along \( h \).

Given a base set \( \text{LEX} \) of lexical items - objects of \( \text{FPos} \) - and a collection of rules \( G_r : C \to \text{Set} \), it becomes straightforward to define a class of ‘derivable objects’ \( \mathcal{L}_G \) - the closure of \( \text{LEX} \) under the rules.

\( \mathcal{L}_G \) can be turned into a category by considering it a full subcategory of \( \text{Der} \), meaning that the set of morphisms between any pair of objects of \( \mathcal{L}_G \) is just the set of all derivation morphisms between them.

Since we can view the objects of \( \mathcal{L}_G \hookrightarrow \text{Der} \) as ‘structured sets’ for any grammar \( G \), we can define ‘extensional equivalence’ between grammars in terms of the structure of derivations it can build. We would like this to be weaker than isomorphism, in that if two languages just differ in the number of isomorphic constructions they have, they should not be considered inequivalent. For example, if we have grammars \( G \) and \( G' \) with the same rules where \( \text{LEX} \subset \text{LEX}' \) such that every lexical item in \( \text{LEX}' \) is isomorphic to one in \( \text{LEX} \), then the derivations producible by \( G' \) will each be isomorphic to one producible by \( G \). Such a description is important in capturing how a language does not essentially change syntactically if words are introduced with identical distribution to existing words which engage in the same grammatical relations. This also guarantees that our choice of representation of derived objects does not affect the essential structure of the language.\(^3\)

Such equivalences also have simple categorical characterization. Given the inclusions \( i : \mathcal{L}_G \hookrightarrow \text{Der} \) and \( j : \mathcal{L}_{G'} \hookrightarrow \text{Der} \), we say that they are equivalent subcategories if there exist functors \( F : \mathcal{L}_G \to \mathcal{L}_{G'} \) and \( G : \mathcal{L}_{G'} \to \mathcal{L}_G \) such that \( jF \approx i \) and \( j \approx iG \) are naturally isomorphic, which implies that \( FG \approx 1_{\mathcal{L}_{G'}} \) and \( GF \approx 1_{\mathcal{L}_G} \) are naturally isomorphic. We can strengthen this to isomorphism of languages by asking that \( FG = 1_{\mathcal{L}_{G'}} \) and \( GF = 1_{\mathcal{L}_G} \), which requires that the correspondence be bijective.

We will compare this to the method outlined in [Keenan and Stabler, 2003] for describing isomorphism of grammars. In particular, we will show that the method given in

\(^3\)See the discussion of free extensions in [Keenan and Stabler, 2003] for a method which handles extensions of grammars by new lexical items. However, their definition cannot describe more general cases of equivalence.
Keenan and Stabler, 2003 compares languages in terms of where rules can apply, not in terms of any internal structure of the derivations or SOs. In particular, there are inclusions of grammars which do not preserve isomorphisms: SOs which are ‘nonisomorphic’ by the definition in Keenan and Stabler, 2003 in some language may be isomorphic in a subgrammar simply by virtue of having fewer objects to combine with. In general, their notion of constituency is not well-behaved with respect to ‘isomorphism’ either. Two SOs having identical constituency will not imply isomorphism between them in those models (while identical constituency definitionally implies isomorphism of derivations in Der).

1.6. Grammatical relations

We think of a grammatical operation acting on a pair of objects as a map \( h : \top_\Delta + \top_\Gamma \to Z \). This data is equivalent to giving a pair of maps \( h_1 : \top_\Delta \to Z \) and \( h_2 : \top_\Gamma \to Z \) which show how each input object maps into the new SO \( Z \), and can be seen as the effect of an instruction to ‘glue’ the two objects together in some way. A ‘grammatical relation’ is then some measurement of how much \( \top_\Delta \) and \( \top_\Gamma \) come to overlap in \( Z \).

Consider an analogous situation in a Stabler Minimalist Grammar (MG) in Fig. 8. We set aside issues concerning in what sense there is a particular tree uniquely associated with an SO in MGs, and assume there is some tree-theoretic way to represent the derivation of a SO. The position of the tree occupied by ‘Pierre::D’ which is usually associated with a ‘complement’ position actually represents the lexical item ‘Pierre’ prior to its merger with ‘praises’. The parent of ‘Pierre’ - ‘praises Pierre:=D V’ - represents the derived VP, where ‘Pierre’ has become the complement. However, the lexical items ‘praises::=D =D V’ and ‘Pierre::D’ are not literally substructures of the derived VP, within which they bear a Head-Comp relationship. Rather, the Head-Comp relationship is defined with respect to the operation mapping the two lexical items into this later state in a particular way. Presumably, in some weaker sense, ‘Pierre’ remains the complement of ‘praises’ in all later derived SOs.
This ‘weaker’ notion - how do two lexical items come to relate in some later stage - may be more appropriate in certain situations. Consider adjunction to a \( T' \), in between the T head and [Spec,TP], such as in the sentence ‘Dave reluctantly left.’ An ‘X-bar’ tree is depicted in Fig. 9. The relative specifier relationship between ‘Dave’ and the T head at the step TP is essentially the same as if the adjunct were absent (though the derivational structure different). We express this in the derivational model by saying that the ‘degree of overlap’ between ‘Dave’ and T at the TP step is the same, regardless of the presence of the adjunct.

Dual to the way we represent operations as built by a \textit{pushout} or ‘gluing construction’, sending a pair of objects \( A \) and \( B \) into \( Z \), we describe grammatical relations as a \textit{pullback} or ‘generalized intersection’. In particular, we look at the lexical items of a derivation and their projections at some given stage \( Z \) which they occur in. The pullbacks between the head and phrasal projections of a fixed pair of lexical items in a derivation which occur in some stage \( Z \) describe the dependencies which have been introduced between them by the stage \( Z \). Whether these pullbacks are empty or not indicates whether a dependency of a certain
form has been introduced, and we use this property to characterize complementation,
specification, and adjunction. Furthermore, there is a series of implications regarding
‘degree of overlap’, which orders these relationships with complementation being the most
overlapping, specification less so, and adjunction the least.

Immediate dominance in some SO implies one of these three relationships holds between
them. The method also generalizes to allow descriptions of ‘indirect’ grammatical related-
ness, such as two specifiers sharing a single licensing feature (e.g. a tuple of *wh*-words in
[Spec,CP] in a language which allows multiple *wh*-fronting). While these definitions will
have their most intuitive linguistic meaning in derivations constructed recursively by the
procedures laid out in the previous sections, they make sense in arbitrary derivations, as
they only make use of the primitive spatial properties, and always describe a degree of
‘dynamic overlap’.

2. Formal theory of derivations

May of the results about derivations, how to represent them, and how to compute with them
can be proven straightforwardly due to the fact that finite partial orders can be represented
algebraically as finite distributive lattices. There is in fact a duality between FPos, the
category of finite partial orders and order-preserving maps, and FDL, the category of finite
distributive lattices and lattice homomorphisms. Since this equivalence is used in almost
every aspect of the formal model, we present the duality explicitly.

**Birkhoff Duality.** For any finite partially ordered set \((X, \leq)\), the collection of upward-
closed sets (upsets) of \(X\) is a finite distributive lattice. In fact, the category of finite
distributive lattices and lattice homomorphisms FDL is dually equivalent to FPos. We
construct an equivalence concretely.

We define \(O : \text{FPos} \to \text{FDL}^{\text{op}}\) to be the functor taking a finite partial order \((X, \leq)\) to
its lattice of open sets \(O(X)\). For any set function \(f : X \to Y\) and any subset \(S \subset Y\), let
\(f^{-1}S\) denote the set \(\{x \in X \mid f(x) \in S\}\). A function between partial orders \(f : X \to Y\)
is order-preserving if and only if for every open subset \(U \subset Y\), \(f^{-1}U\) is an open subset of
\(X\). \(O\) then associates to each order-preserving function \(f : X \to Y\) a homomorphism of
lattices \(f^{-1} : O(Y) \to O(X)\), taking \(U \in O(Y)\) to \(f^{-1}U\).

Conversely, we construct a functor \(\text{pt} : \text{FDL}^{\text{op}} \to \text{FPos}\). Let 1 denote any one-point
partial order with the only possible ordering, and let \(O(1)\) be its open set lattice, which
we write \(\{\top, \bot\}\) for 1 and \(\emptyset\) respectively. Since the elements of a partial order are in
correspondence with the order-preserving maps \(1 \to X\), we then define the set of points
associated to a finite distributive lattice \(D\) to be the set of homomorphisms \(\text{FDL}(D, O(1))\).
Note that in a finite partial order, \(x \leq y\) if and only if every open set containing \(x\) also
contains \(y\). We then turn the set \(\text{FDL}(D, O(1))\) into a partial order by giving an ordering
\(p \leq q\), where \(p, q : D \to O(1)\) are homomorphisms, if and only if for all \(d \in D\), \(p(d) = \top\)
implies \(q(d) = \top\); i.e. \(p(d) \leq q(d)\) for all \(d \in D\). We denote this partial order \(\text{pt}(D)\). The
construction can be made functorial: to any homomorphism of lattices $h : D \to C$, we associate the function $\mathbf{pt}(h) : \mathbf{pt}(C) \to \mathbf{pt}(D)$ which takes a point $p : C \to \mathcal{O}(1)$ to the point $ph : D \to C \to \mathcal{O}(1)$. $\mathbf{pt}(h)$ is in fact order-preserving.

These two functors give an equivalence of categories. Namely, for any finite partial order $X$, $k_X : X \overset{\sim}{\to} \mathbf{pt}(\mathcal{O}(X))$ sending $x \in X$ to $x^{-1} : \mathcal{O}(X) \to \mathcal{O}(1)$ is an isomorphism of finite partial orders, and for any finite distributive lattice $l_D : D \overset{\sim}{\to} \mathcal{O}(\mathbf{pt}(D))$ sending $d$ to $\{p : p(d) = \top\}$ is an isomorphism of lattices.

We define a prime filter of a finite distributive lattice to be a subset $P \subset D$ which is exactly the set of elements mapping to ‘true’ under some homomorphism $p : D \to \mathcal{O}(1)$.

Two useful facts are immediately implied by the equivalence. Sublattice inclusions correspond exactly to surjective order-preserving maps between partial orders, and surjective lattice homomorphisms correspond exactly to subspace embeddings of finite partial orders.

Take any homomorphism of finite distributive lattices $h : D \to C$. This can be considered a functor between partial order categories, and it has a left adjoint $h_0 : C \to D$. This adjoint takes an element $c \in C$ to the smallest element $d \in D$ such that $c \leq h(d)$. We can always view $h$, $D$, and $C$ as arising from an order-preserving map between partial orders. We write the associated order-preserving map as $f : X \to Y$ and $h$ as $f^{-1} : \mathcal{O}(Y) \to \mathcal{O}(X)$. We write the left adjoint $f_1 : \mathcal{O}(X) \to \mathcal{O}(Y)$ which takes each $U \in \mathcal{O}(X)$ to the smallest open in $\mathcal{O}(Y)$ containing the image $f(U)$. Such a smallest open exists and can be computed as the intersection of all open subsets $V$ of $Y$ such that $U \subset f^{-1}V \iff f(U) \subset V$.

### 2.1. Formal definition of Der

**Definition 1.** For any space $X$, we call a functor $F : \mathcal{O}(X)^{op} \to \text{FDL}$ a sheaf of finite distributive lattices if the presheaf $UF : \mathcal{O}(X)^{op} \to \text{Set}$ where $U$ is the forgetful functor from FDL to Set, is a sheaf of sets.

Since $U$ creates limits, this coincides with the usual definition of sheaves in a category of ‘structured sets’, such as sheaves of rings or abelian groups in algebraic geometry or differential geometry.

For any finite space $X$, there is an associated functor $\mathcal{O}_X : \mathcal{O}(X)^{op} \to \text{FDL}$ which takes each open subset $U \in \mathcal{O}(X)$ to the lattice $\mathcal{O}(U)$; and each inclusion $V \to U$ to a homomorphism $\mathcal{O}(U) \to \mathcal{O}(V)$ taking $W \in \mathcal{O}(U)$ to $W \cap V \in \mathcal{O}(V)$. This functor is a sheaf of distributive lattices in the sense above.

**Definition 2.** For a finite partial order $X$, we define a local topology on $X$ to be a subsheaf (of distributive lattices) inclusion $\omega \subset \mathcal{O}_X$.

**Definition 3.** A derivation is a finite partial order $X$ together with a local topology $\omega \subset \mathcal{O}_X$.

**Definition 4.** A morphism of derivations on derivations $\Delta = (X, \leq, \omega)$ and $\Gamma = (Y, \leq \tau)$ is an order-preserving function $\phi : \Delta \to \Gamma$ such that if $V \in \tau U$, then $f^{-1}V \in \omega(f^{-1}U)$.  

21
A morphism of derivations is then a continuous function \( f : X \to Y \) between base spaces together with an induced morphism \( f^\#: \tau \to \omega f^{-1} \) of sheaves on \( Y \), bringing it in line with morphisms from other ‘geometric’ categories like morphisms of (locally) ringed spaces such as schemes from algebraic geometry or manifolds from differential geometry.

All of the other properties and categorical constructions like products, sums, and sub-derivations follow from these axioms.

**Claim 1.** The above abstract definition of a derivation is equivalent to giving the following data:

1. A finite set \( X \)
2. A partial ordering \( \leq \) on \( X \)
3. For each open subset \( U \subset X \), a sublattice inclusion \( \omega U \subset \mathcal{O}(U) \) such that:
   
   a) If \( V \subset U \) and \( W \in \omega U \), then \( W \cap V \in \omega V \)
   
   b) If \( U_1 \cup U_2 = U \) and \( V_1 \in \omega U_1 \) and \( V_2 \in \omega U_2 \) are any two open subsets such that \( V_1 \cap U_2 = V_2 \cap U_1 \), then \( V_1 \cup V_2 \in \omega U \).

The category presented above is equivalent to the category informally presented in §1. We simply associate to each inclusion \( \omega U \hookrightarrow \mathcal{O}(U) \) the map \( U \to \top_U \) by duality. Every derivation in the sense of §1 can be obtained this way, up to isomorphism.

### 2.2. Qualitative properties of derivations and derived objects

We now give some results which formally relate derivations and their derived objects.

The first result describes the relation between a derivation and its derived objects with respect to the objects of \( \text{Der} \).

**Claim 2.** To each \( U \in \mathcal{O}(\Delta) \), we have an associated finite distributive lattice \( \omega U \) which by duality corresponds to a finite partial order \( \top_U \). Similarly, for any inclusion \( U' \subset U \), we have a homomorphism \( \omega U \to \omega U' \), which by duality corresponds to an order-preserving map of partial orders \( \top_U \to \top_U' \).

This follows immediately from the definitions of \( \text{Der} \) and the dual equivalence between \( \text{FDL} \) and \( \text{FPos} \).

We have a corresponding relationship at the level of morphisms of derivations.

**Claim 3.** Let \( \phi : \Delta \to \Gamma \) be any morphism of derivations. Let \( U \subset \Delta \) be any open subset, and let \( V \subset \Gamma \) be any open subset such that the image \( \phi(U) \) is contained in \( V \). We have an induced morphism of lattices \( \tau V \to \omega(\phi^{-1}V) \) taking \( K \in \tau V \) to \( \phi^{-1}K \), which by duality corresponds to an order-preserving map of orders \( \top_{\phi^{-1}V} \to \top_V \). Since \( U \subset \phi^{-1}V \), we have an associated order-preserving map of derived objects \( \top_U \to \top_{\phi^{-1}V} \), which we compose with the above map to get an order-preserving map \( \top_U \to \top_V \), which we denote \( \phi_U^V \).
If \( U' \subset U \) in \( \Delta \) is associated to the morphism of derived SOs \( k : \tau U' \to \tau U \), then the induced morphism \( \phi^V_{U'} : \tau U' \to \tau V \) is simply \( \phi^V_{U'} = \phi^V_{U'} \circ k : \tau U' \to \tau U \to \tau V \). Dually, if \( V \subset V' \) in \( \Gamma \) is associated to the morphism of derived SOs \( j : \tau V \to \tau V' \), then the induced morphism \( \phi^V_{U'} : \tau U \to \tau V' \) is \( j \circ \phi^V_{U'} \).

This again follows directly from the definition of morphisms and the condition that \( \tau V \to \omega(\phi^{-1} V) \) be a morphism of lattices.

We now show what the sheaf condition means for the derived SOs.

**Claim 4.** We say that a collection open sets \( U = (U_i)_{i \in I} \) is an open cover of \( U \) if \( \bigcup_{i \in I} U_i = U \) and each \( U_i \subset U \) is an open subset. We say that an open cover is saturated if for any \( U_i \in U \) and \( V \subset U_i \) open, \( V_i \in U \). Clearly every open cover has a unique saturation which doesn’t change the union.

Let \( U \) be any open cover of \( U \). We associate to \( U \) a directed diagram of finite partial orders, consisting of the partial orders \( \tau_{U_i} \) for each \( U_i \in U \) together with the maps \( \tau_{U_i} \to \tau_{U_j} \), associated to inclusions \( U_i \subset U_j \).

We claim that \( \tau U \) together with the morphisms \( u_i : \tau_{U_i} \to \tau U \) associated to inclusions \( U_i \subset U \) turn \( \tau U \) into a colimit of this diagram. This means that for any object \( Z \) together with a compatible family of maps \( f_i : \tau_{U_i} \to Z \) from the directed diagram, there is a unique order-preserving map \( k : \tau U \to Z \) such that \( f_i = k \circ u_i \) for each \( U_i \). A family of maps \( f_i : \tau_{U_i} \to Z \) is compatible if for any \( U_j \subset U_i \) in \( U \), \( f_j : \tau U_i \to \tau U_j \to Z \) is equal to \( f_i : \tau_{U_i} \to Z \), where \( ! : \tau U_i \to \tau U_j \) is the function associated to the inclusion \( U_i \subset U_j \).

**Proof.** The sheaf condition says that if \( U \) is a saturated cover of \( U \), then the diagram \( \omega U \), consisting of the lattice \( \omega U_i \) for each \( U_i \in U \) and lattice homomorphism \( \omega U_i \to \omega U_j \) has limit \( \omega U \). It follows immediately by duality that the associated partial order diagram is a colimit diagram. \( \square \)

**Sections in terms of matching families.** It is an immediate consequence that the partial orders \( \tau U \) are determined by the stalks \( \omega U_x \). If we take the collection \( U_x \) for each \( x \in U \), this must be a cover of \( U \). We call a collection \( V_x \in \omega U_x \) a matching family on this cover if for any \( x, y, V_x \cap U_y = V_y \cap U_x \). The elements \( V \in \omega U \) correspond to matching families on covers. To each \( V \in \omega U \), the collection \( V \cap U_x \in \omega U_x \) is a matching family on the \( U_x \); conversely, for any matching family, the union \( \bigcup_{x \in U} V_x = V \) is an element of \( \omega U \) by the sheaf condition. Matching families on this (or any) cover then determine the elements of \( \omega U \) and hence \( \tau U \) by duality.

We now give a relation between the points of a derivation and the points of derived SOs where they occur.

**Claim 5.** Let \( x \in \Delta \) by any point of \( \Delta \). It is associated to a prime filter \( h : \mathcal{O}(\Delta) \to \mathcal{O}(1) \) taking \( U \) to \( \top \) iff \( x \in U \), and \( \bot \) otherwise. Denote this prime filter by \( P_x \). If \( U \) is any open subset such that \( x \in U \), then the set \( P_x \cap \omega U \) is a prime filter in \( \omega U \), and hence corresponds to a point of \( \tau U \), which we denote \( x_U \).

23
Proof. In appendix.

Definition 5. Let \( x, y \in \Delta \). We say that \( x \) is a projection of \( y \), written \( x \sqsubset y \), if \( x \leq y \), and for any open \( U \) containing \( x \), the point \( x_U \in \top_U \) is equal to \( y_U \).

For a point \( x \in \Delta \), we write \( U_x \) for the open subset of points \( \{ y \in \Delta \mid x \leq y \} \).

Claim 6. \( x \sqsubset y \) iff \( \top_U y \to \top_U x \) takes \( y_U \) to \( x_U \).

Pf. FILL IN WITH PICTURE

Claim 7. If \( \phi : \Delta \to \Gamma \) is any morphism, and \( x \sqsubset y \) in \( \Delta \), then \( \phi(x) \sqsubset \phi(y) \).

Pf. FILL IN WITH PICTURE

Definition 6. We say that \( y \) is close to \( x \), written \( x \preceq y \), if \( U_y \in \omega U_x \).

Definition 7. If \( \phi : \Delta \to \Gamma \) is a morphism such that \( x \preceq y \) implies \( \phi(x) \preceq \phi(y) \), we say that \( \phi \) is coherent.

Definition 8. We say that an element \( t \in \Delta \) is a term if it is minimal with respect to closeness. That is, if \( x \preceq t \) implies \( x = t \).

Definition 9. We say that \( \phi : \Delta \to \Gamma \) is term-preserving if whenever \( t \) is a term of \( \Delta \), then \( \phi(t) \) is a term of \( \Gamma \).

Definition 10. An order-preserving function \( f : X \to Y \) between partial orders is an open map if each open subset \( U \subset X \) has image \( f(U) \subset Y \) which is open in \( Y \).

Accordingly, we say that a derivation morphism \( \phi : \Delta \to \Gamma \) is open if the underlying map of partial orders is open.

Definition 11. We call \( \phi : \Delta \to \Gamma \) a constituent-preserving map if it is term-preserving, coherent, and open.

We now characterize the adjoint functorial relationships between \( \text{FPos} \), thought of as derived SOs, and \( \text{Der} \), thought of as derivations of SOs.

Claim 8. Let \( \text{pt} : \text{Der} \to \text{FPos} \) be the functor ‘forgetting’ the local topology on a derivation \( \Delta = (|\Delta|, \leq \omega) \), sending it to \( (|\Delta|, \leq) \).

This functor has a left adjoint \( i : \text{FPos} \to \text{Der} \), sending \( (X, \leq) \) to \( (X, \leq, \mathcal{O}_X) \). The counit \( \text{pt} \circ i \to 1_{\text{FPos}} \) is a natural isomorphism. The unit \( i \circ \text{pt} \to 1_{\text{Der}} \) on each derivation takes the underlying partial order of points to the derivation, acting by identity on the underlying points \( \text{pt}(\Delta) \to \Delta \). This can be compared with the map \( \delta|X| \to X \) on any finite partial order \( X \), sending the underlying set with discrete topology into \( X \).

The functor \( i \) has a further left adjoint \( \top(-) \). We can compute the values of this functor explicitly. For any derivation \( \Delta \), we take the open set \( |\Delta| \) corresponding to the whole
space. The designated collection $\omega(\Delta)$ is a finite distributive lattice, and we write its associate partial order $\top_{\Delta}$. Often we abuse notation and write $\omega(\Delta)$ as $\omega \top_{\Delta}$. The unit $1_{\text{Der}} \to i \circ \top_{(-)}$ is the morphism $\Delta \to \top_{\Delta}$ collapsing each point of the derivation to its representative in $\top_{\Delta}$. This is given by the dual of the inclusion of lattices $\omega \top_{\Delta} \hookrightarrow \mathcal{O}(\Delta)$. The counit $\top_{(-)} \circ i \to 1_{\text{FPos}}$ is a natural isomorphism.

Proof.

Claim 9. Let $1$ denote any derivation with underlying one-point set. There is only one such derivation up to isomorphism. The set of points $|\Delta|$ underlying a derivation are in a canonical bijection with the set of morphisms $1 \to \Delta$. Another way to say this is that $1$ represents the forgetful functor, in that $|\cdot| \cong \text{Der}(1, -)$ are naturally isomorphic.

2.3. Der as a concrete category

We first give a series of definitions which make sense in any category. They are the standard abstractions of bijections, injections, surjections, and substructure inclusions.

Definition 12. A morphism $f : A \to B$ in a category $C$ is . . .

1. An isomorphism if there is a morphism $g : B \to A$ where $gf = 1_A$ and $fg = 1_B$, where $1_A$ and $1_B$ are the identity morphisms.

2. A monomorphism if for any morphisms $x, y : Z \to A$, $fx = fy$ implies $x = y$. This is often called a left-cancellable morphism.

3. An epimorphism if for any morphisms $x, y : A \to Z$, $xf = yf$ implies $x = y$. This is often called a right-cancellable morphism.

4. A regular monomorphism if there exist morphisms $a, b : B \to C$ such that $af = bf$ and for any $g : Y \to B$ such that $ag = bg$, there exists a unique morphism $u : Y \to A$ such that $g = fu$.

It is straightforward from these definitions that an isomorphism is a monomorphism, an epimorphism, and a regular epimorphism in any category.

For example, in $\text{FPos}$, a morphism $f : A \to B$ is . . .

1. An isomorphism iff it is bijective and its inverse function $f^{-1}$ is also order-preserving

2. A monomorphism iff its underlying set-function is injective (if $f(a) = f(a')$, then $a = a'$)

3. An epimorphism iff its underlying set-function is surjective (if for every $b \in B$, there is some $a \in A$ such that $f(a) = b$)
4. A regular monomorphism iff \( f \) is an injection such that \( a \leq a' \) iff \( f(a) \leq f(a') \).

A category is called a **construct** if there is a faithful functor \( U : C \to \text{Set} \). \( \text{Der} \) is a construct by using \( |\cdot| : \text{Der} \to \text{Set} \) taking each derivation to its underlying set of points. In any construct, it makes sense to define substructures.

**Definition 13.** In a construct \( U : C \to \text{Set} \), a morphism \( m : A \to B \) is called an **embedding** if \( Um : UA \to UB \) is an injective function, and for any object \( C \) of \( C \) and any function \( f : UC \to UA \), if \( Um \circ f : UC \to UA \to UB \) is a morphism, then there is a (unique) morphism \( k : C \to A \) with underlying function \( f \).

The idea is that \( A \) just has the structure of \( B \) ‘restricted down to the subset \( UA \). Any morphism into \( B \) whose set-theoretic image is contained in \( UA \) must factor through \( A \) as a morphism, indicating that \( A \) represents the structure on \( UA \) ‘inside’ \( B \). In \( \text{FP} \), the embeddings are exactly the regular monomorphisms. We will give a similar result for derivations.

**Claim 10.** In \( \text{Der} \)

1. The isomorphisms are exactly bijections \( \phi : \Delta \to \Gamma \) whose inverse function is also a morphism. In particular, \( a \leq \Delta b \) iff \( \phi(a) \leq_{\Gamma} \phi(b) \) and \( \top_U \cong \top_{\phi(U)} \) for each open.

2. The monomorphisms are exactly morphisms \( \phi : \Delta \to \Gamma \) whose underlying function is injective

3. The epimorphisms are exactly morphisms \( \phi : \Delta \to \Gamma \) whose underlying function is surjective

4. The regular monomorphisms \( \phi : \Delta \to \Gamma \) are exactly the embeddings.

Proof of the first two claims are straightforward. If \( \phi : \Delta \to \Gamma \) is an isomorphism with inverse \( \psi \), then \( \phi \) and \( \psi \) must be bijections to compose to the identities. If \( \phi \) is monomorphic, then it must be injective by the representability of \( |\cdot| \) by \( 1 \). If the function underlying \( \phi \) is injective, we show that \( \phi \) is monomorphic. Let \( a, b : \Gamma \to \Xi \) be any morphisms such that \( \phi a = \phi b \) and \( \phi \) injective. Then, for all points \( d \in \Delta \), \( \phi(a(d)) = \phi(b(d)) \), and by injectivity \( a(d) = b(d) \). But then \( a = b \), so \( \phi \) is monomorphic.

The most straightforward way to show the last two conditions is by using more general constructions in \( \text{Der} \). The existence of these constructions will follow from next claim.

**Claim 11.** \( \text{Der} \) is finitely complete and cocomplete.

We break this down into pieces. Finite completeness means that \( \text{Der} \) has

1. A **terminal object** \( 1 \)
2. For any pair of derivations $\Delta, \Gamma$ a **product** derivation $\Delta \times \Gamma$

3. For any pair of morphisms $a, b : \Delta \Rightarrow \Gamma$ an equalizer $s : S \to \Delta$

A terminal object is a derivation $1$ such that for every derivation $\Delta$, there is exactly one morphism $! : \Delta \to 1$. The terminal derivations are singletons with the only possible derivation structure.

A product derivation is a derivation $\Delta \times \Gamma$ together with projection morphisms $\pi_{\Delta} : \Delta \times \Gamma \to \Delta$ and $\pi_{\Gamma} : \Delta \times \Gamma \to \Gamma$ which are **universal** with respect to maps into the two objects. This means that for any object $\Pi$ and pair of morphisms $s : \Pi \to \Delta$ and $t : \Pi \to \Gamma$, there is a unique morphism $u : \Pi \to \Delta \times \Gamma$ such that $s = \pi_{\Delta} \circ u$ and $t = \pi_{\Gamma} \circ u$. Product derivations have points $(d, g)$ with $d \in \Delta$ and $g \in \Gamma$, and $\Delta \times \Gamma$ has the structure of $\Delta$ and $\Gamma$ in each coordinate.

An equalizer is a derivation $S$ together with a morphism $s : S \hookrightarrow \Delta$ such that $as = bs$, and $s$ is **universal** with respect to making $a$ and $b$ equal. It is a fact in any category that an equalizer is a monomorphism, and we call a monomorphism which arises in this way a **regular monomorphism**. We will prove that the regular monomorphisms and substructure embeddings in $\text{Der}$ are the same.

Finite cocompleteness means that $\text{Der}$ has

1. An **initial object** $0$

2. For any pair of derivations $\Delta, \Gamma$ a **coproduct** or **sum** derivation $\Delta + \Gamma$

3. For any pair of morphisms $a, b : \Delta \Rightarrow \Gamma$ a **coequalizer** $s : \Gamma \to \tilde{\Gamma}$

The definitions are dual to those above. An initial object has exactly one morphism $\epsilon : 0 \to \Delta$ to any other object. The coproduct has coprojections $\kappa_{\Delta} : \Delta \to \Delta + \Gamma$ and $\kappa_{\Gamma} : \Gamma \to \Delta + \Gamma$ which are universal with respect to pairs of maps out of $\Delta$ and $\Gamma$. This means that for any object $\Xi$ and pair of morphisms $s : \Delta \to \Xi$ and $t : \Gamma \to \Xi$, there is a unique morphism $u : \Delta + \Gamma \to \Xi$ such that $u \circ \kappa_{\Delta} = s$ and $u \circ \kappa_{\Gamma} = t$. A coequalizer of two morphisms $a, b$ is a universal morphism $s$ such that $sa = sb$. This means that if $t : \Gamma \to \Xi$ is a morphism such that $ta = tb$, then there is a unique morphism $u : \tilde{\Gamma} \to \Xi$ such that $us = t$.

In $\text{Der}$, the initial object will be the empty derivation, sums will act like disjoint unions, and coequalizers like ‘quotients by a congruence’.

We start with the initial and terminal objects.

**Lemma 11.1.** $\text{Der}$ has initial object with underlying set $\emptyset$ and terminal object with underlying set $1$, where $1$ is any singleton.

**Proof.** In appendix.
**Lemma 11.2.** \( \text{Der} \) has coproducts, where the sum \( \Delta + \Gamma \) has underlying set \(|\Delta| + |\Gamma|\) which is the disjoint union of the underlying sets of the summands. The ordering \( a \leq b \) in \( \Delta + \Gamma \) iff \( a \leq b \) in \( \Delta \) or \( \Gamma \) gives \( \Delta + \Gamma \) the open set structure \( \mathcal{O}(\Delta) \times \mathcal{O}(\Gamma) \) where a pair \((D, G)\) corresponds to the open set \( D + G \) which is upward-closed in \( \Delta + \Gamma \) since each of \( D \) and \( G \) are. Writing the local topologies as \( \omega \) and \( \tau \) respectively, we define a local topology on \( \Delta + \Gamma \) given by \((\omega \times \tau)(D, G) = \{(A, B) \mid A \in \omega D \text{ and } B \in \tau G\} \).

**Proof.** In appendix.

**Lemma 11.3.** We construct coequalizers. Let \( a, b : \Delta \to \Gamma \) be any two parallel morphisms of derivations. We construct \( \mathcal{Q} \) and morphism \( q : \Gamma \to \mathcal{Q} \). We let \( \mathcal{O}(\mathcal{Q}) \) be the set of elements \( U \in \mathcal{O}(\Gamma) \) such that \( a^{-1}U = b^{-1}U \). We claim that this is a finite distributive lattice given the ordering \( U \leq V \) in \( \mathcal{O}(\mathcal{Q}) \) iff the relation holds in \( \mathcal{O}(\Gamma) \). We give \( \mathcal{O}(\mathcal{Q}) \) a local topology \( \tau \) such that for \( U, V \in \mathcal{O}(\mathcal{Q}) \), \( V \in \tau U \) iff \( V \in \tau U \), where \( \tau \) is the local topology on \( \Gamma \). The sublattice inclusion \( \mathcal{O}(\mathcal{Q}) \hookrightarrow \mathcal{O}(\Gamma) \) leads to a map of partial orders \( \Gamma \to \mathcal{Q} \) by duality, and we claim that this is a derivation morphism. Moreover, \( qa = qb \) and \( q \) is universal with respect to coequalizing \( a \) and \( b \).

**Proof.** In appendix.

Having proven that \( \text{Der} \) has coproducts, coequalizers, and an initial object, it follows that it has all colimits. In particular, it has pushouts, which we will use to prove concrete properties of subderivations and epimorphisms. We will now show that \( \text{Der} \) has all limits.

**Lemma 11.4.** Given a pair of derivations \( \Delta \) and \( \Gamma \), we define a derivation \( \Delta \times \Gamma \). This derivation has set of points \( (d, g) \) for \( d \in \Delta \) and \( g \in \Gamma \) and \( (d, g) \leq (d', g') \) iff \( d \leq d' \) and \( g \leq g' \). This space has open sets \( \bigcup_{i \in I} D_i \times G_i \), where \( D_i \) and \( G_i \) are open in \( \Delta \) and \( \Gamma \). We define stalks of a sheaf on \( \Delta \times \Gamma \), which uniquely determine a sheaf on \( \Delta \times \Gamma \). For each \( (d, g) \in \Delta \times \Gamma \), we take \((\omega \otimes \tau)(U_d \times U_g) = \{ \bigcup_{i \in I} D_i \times G_i \mid D_i \in \omega U_d 	ext{ and } G_i \in \tau U_g \} \). We claim that this is a derivation, and that the obvious projection functions \( \pi_\Delta : \Delta \times \Gamma \to \Delta \) and \( \pi_\Gamma : \Delta \times \Gamma \to \Gamma \) mapping \( (d, g) \mapsto d \) and \( g \) respectively are morphisms. Further, \( \Delta \times \Gamma \) with these projections is a product of \( \Delta \) and \( \Gamma \) in \( \text{Der} \).

**Proof.** In appendix.

**Lemma 11.5.** We construct equalizers. Let \( a, b : \Delta \to \Gamma \) be any parallel pair of morphisms of derivations. We construct a derivation \( \mathcal{S} \) and morphism \( m : \mathcal{S} \to \Delta \). Let the underlying set of points of \( \mathcal{S} \) consist of the set \( S = \{ d \in \Delta \mid ad = bd \} \) with order \( d \leq d' \) iff the relation holds in \( \Delta \). We define a sheaf of lattices by giving their values on stalks. For each point \( x \in S \), we define \( \omega_S S_x = \{ A \cap S \mid A \in \omega U_x \} \), where \( S_x \) is the collection \( \{ y \in S \mid x \leq y \} \). We claim this gives rise to a sheaf of finite distributive lattices on \( S \) which turns \( \mathcal{S} \) into a derivation, and the subset inclusion \( m : \mathcal{S} \hookrightarrow \Delta \) into a morphism. Moreover, \( am = bm \), and \( m \) is universal with respect to equalizing \( a \) and \( b \).

28
Proof. In appendix.

This completes the proof of Claim 11. We look at a corollary of Claim 11 to complete the proof of Claim 10.

Claim 12. As a corollary of Claim 11, every diagram
\[
\begin{array}{ccc}
\Delta & \xrightarrow{a} & \Gamma \\
\downarrow{b} & & \downarrow{t} \\
\Xi & \xrightarrow{s} & \Sigma \\
\end{array}
\]

Has a pushout \( \Sigma \) together with morphisms \( s : \Xi \to \Sigma \) and \( t : \Gamma \to \Sigma \) such that \( ta = sb \) which is universal with respect to this property. This means for any \( \Pi \) together with morphisms \( x : \Xi \to \Pi \) and \( y : \Gamma \to \Pi \) such that \( ya = xb \), there is a unique morphism \( u : \Sigma \to \Pi \) such that \( ut = y \) and \( us = x \).

This pushout can be computed as follows. It has underlying open set lattice isomorphic to \( O(\Gamma) \times O(\Delta) \times O(\Xi) = \{(G, X) \in O(\Gamma) \times O(\Xi) \mid a^{-1}G = b^{-1}X\} \). This lattice determines a finite partial order up to isomorphism by duality. We define a local topology on this space, where \( (H, Y) \in \zeta(G, X) \) iff \( H \in \omega G \) and \( Y \in \tau X \). The inverse image maps are given by projection, taking \( (G, X) \mapsto G \) and \( X \), respectively. These are lattice homomorphisms, and hence give rise to order-preserving functions \( \Xi \to \Xi + \Delta \Gamma \) and \( \Gamma \to \Xi + \Delta \Gamma \) which are morphisms. That these are morphisms and are universal with respect to completion of the square can be proven using methods identical to those in the proof of Claim 11.

In particular, by computation, we have the following claim.

Claim 13. If the following diagram of derivations is a pushout diagram in \( \mathbb{D}er \)
\[
\begin{array}{ccc}
\Delta & \xrightarrow{a} & \Gamma \\
\downarrow{b} & & \downarrow{t} \\
\Xi & \xrightarrow{s} & \Sigma \\
\end{array}
\]

Then the following diagram of partial orders is a pushout diagram in \( \mathbb{F}Pos \).
\[
\begin{array}{ccc}
\text{pt}\Delta & \xrightarrow{\text{pt}(a)} & \text{pt}\Gamma \\
\downarrow{\text{pt}(b)} & & \downarrow{\text{pt}(t)} \\
\text{pt}\Xi & \xrightarrow{\text{pt}(s)} & \text{pt}\Sigma \\
\end{array}
\]

Proof. Immediate from the construction of pushouts in \( \mathbb{D}er \) and \( \mathbb{F}Pos \). 

In particular, when \( m : \mathcal{S} \to \Delta \) is a regular monomorphism, we will be interested in the pushout of \( m \) along itself. Such a pushout \( \Sigma \) together with the pair of maps \( s, t : \Delta \to \Sigma \) is usually called the cokernel of \( m \). It is a fact that in every category, if \( m \) is a regular monomorphism, if \( m \) has a cokernel, then \( m \) is the equalizer of this cokernel. Since \( \mathbb{D}er \)
has pushouts, every regular monomorphism will arise as the equalizer of its cokernel. This is essentially the statement “in \( \text{Der} \), all regular monomorphisms are effective.”

We now establish a concrete claim about subsets which will help complete the proof of Claim 10.

**Claim 14.** Let \( \Delta \) be a derivation and \( S \subseteq |\Delta| \) any subset. We turn \( S \) into a partial order by considering it a subspace of \( \text{pt}(\Delta) \). We then get a morphism of partial orders \( S \hookrightarrow \text{pt}(\Delta) \) which extends to a morphism of derivations \( S \hookrightarrow \text{pt}(\Delta) \rightarrow \Delta \). Call this inclusion \( i \). The pushout of \( i \) along itself exists, and the induced morphisms \( s, t : \Delta \rightrightarrows \Delta +_S \Delta \) have the property that \( s(d) = t(d) \) if and only if \( d \in S \).

**Proof.** In appendix.

We now complete the proof of Claim 10.

**Lemma 14.1.** A morphism of derivations \( \phi : \Delta \rightarrow \Gamma \) is epimorphic iff it is surjective.

**Proof.** \( \text{Surj} \Rightarrow \text{Epi} \) Suppose that \( \phi \) is surjective. Let \( a, b : \Gamma \rightarrow \Sigma \) be any pair of morphisms such that \( a\phi = b\phi \). For any point \( g \in \Gamma \), we can find some \( d \in \Delta \) such that \( \phi(d) = g \) by surjectivity. Then \( a\phi(d) = b\phi(d) \) implying that \( a(g) = b(g) \) for all \( g \in \Gamma \), so \( a = b \).

\( \neg \text{Surj} \Rightarrow \neg \text{Epi} \) Suppose that \( \phi \) is not surjective. Then there is some point \( g \in \Gamma \) such that there is no \( d \in \Delta \) where \( \phi(d) = g \). Construct the subset \( |\Gamma| - \{g\} \equiv S \). By Claim 14, we can construct morphisms \( s, t : \Gamma \rightrightarrows \Gamma +_S \Gamma \) with the property that \( s(x) = t(x) \) iff \( x \in S \), i.e. iff \( x \neq g \). Then \( s\phi = t\phi \), since for all \( d \in \Delta \), \( \phi(d) \neq g \). But by construction, \( s \neq t \), since \( s(g) \neq t(g) \).

**Lemma 14.2.** \( m : S \rightarrow \Delta \) is a regular monomorphism if and only if it is an embedding.

**Proof.** \( \text{Reg} \Rightarrow \text{Emb} \) If \( m \) is regular, then there exist morphisms \( a, b : \Delta \rightrightarrows \Gamma \) such that \( m \) is their equalizer. We apply \( |\cdot| \) to get a diagram of sets. Now suppose we have a derivation \( \Sigma \) and function \( f : |\Sigma| \rightarrow |S| \) such that \( |m| f \) underlies a morphism.

\[
\begin{array}{ccc}
|\Sigma| & \xrightarrow{|m|} & |\Delta| \\
\downarrow f & & \xrightarrow{[a]} & \downarrow \text{Emb} & \xrightarrow{[b]} & |\Gamma| \\
|\Sigma| & \xrightarrow{|m| f} & |S|
\end{array}
\]

Now, \( |a|(|m| f) = (|a||m|)|f \) and \( |b|(|m| f) = (|b||m|)|f \). But \( |a||m| = |b||m| \), so \( |a|(|m| f) = |b|(|m| f) \), and \( |m| f \) is a morphism equalizing \( a \) and \( b \). So there must be a unique morphism \( u : \Sigma \rightarrow \mathcal{S} \) such that \( mu = |m| f \). We apply the \( |\cdot| \) functor to obtain the equality \( |m||u| = |m| f \) of set functions. But \( |m| \) is injective, so \( |u| = f \), showing that \( f \) underlies a morphism.

\( \text{Emb} \Rightarrow \text{Reg} \) Let \( m : S \rightarrow \Delta \) be an embedding. We construct the pushout of \( m \) along itself to obtain morphisms \( s, t : \Delta \rightrightarrows \Delta +_S \Delta \). By Claim 14, \( s(d) = t(d) \) iff \( d \in S \). Let \( \Sigma \) be any derivation and \( h : \Sigma \rightarrow \Delta \) any morphism whose set-theoretic image fits in \( \mathcal{S} \). \( h \) has
image inside $S$ iff $sh = th$, i.e. if it equalizes $s$ and $t$. Since $S$ is an embedding, $h$ factors as a morphism $g : \Sigma \to S$ such that $mg = h$. This shows that $m$ equalizes $s$ and $t$, and that it is universal with respect to equalizing $s$ and $t$.

This completes the proof of Claim 10.

We now prove some concrete properties of the product, coproduct, substructure, and ‘quotient’ (coequalizer) constructions.

**Claim 15.** In Der

1. $\Delta \times \Gamma$ has set of points $|\Delta| \times |\Gamma|$. $(d, g) \leq (d', g')$ iff $d \leq d'$ in $\Delta$ and $g \leq g'$ in $\Gamma$. $(d, g) \sqsubset (d', g')$ iff $d \sqsubset d'$ in $\Delta$ and $g \sqsubset g'$ in $\Gamma$. $(d, g) \preceq (d', g')$ iff $d \preceq d'$ in $\Delta$ and $g \preceq g'$ in $\Gamma$.

2. $\Delta + \Gamma$ has set of points $|\Delta| + |\Gamma|$. $x \leq y$ iff $x \leq y$ in $\Delta$ or $\Gamma$. $x \sqsubset y$ iff $x \sqsubset y$ in $\Delta$ or $\Gamma$. $x \preceq y$ iff $x \preceq y$ in $\Delta$ or $\Gamma$.

3. On any subset $S \subset |\Delta|$, there is a unique induced subderivation structure $S$. For all $x, y \in S$, $x \leq y$ in $S$ iff in $\Delta$; and $x \sqsubset y$ in $S$ iff in $\Delta$. If $x \leq y$ in $\Delta$, this implies $x \leq y$ in $S$.

    For any subderivation embedding $m : S \hookrightarrow \Delta$, at each point $x \in S$, the induced map from the derived object $\top_S x$ at $x$ in $S$ to the derived object $\top_U x$ at $x$ in $\Delta$ is an embedding of partial orders.

4. Every regular epimorphism $q : \Gamma \to Q$ arises as a ‘quotient’ of $\Gamma$ by a congruence relation $\sim$ on $\Gamma$, whose underlying partial order is computed as the coequalizer of $\sim$ on $\text{pt}(\Gamma)$ in $\text{FPos}$.

5. The classes $\text{Epi}$ and $\text{RegMon}$ form a factorization system for Der. That is, every morphism $\phi : \Delta \to \Gamma$ factors as an epimorphism followed by a regular monomorphism $\Delta \to \text{im}(\phi) \hookrightarrow \Gamma$ in an essentially unique way.

We explain these last two claims.

**Quotients and congruences.** For a set $X$ together with an equivalence relation $\sim$ on $X$, we define the quotient of $X$ by $\sim$ to be a function $q : X \to \bar{X}$, where $\bar{X}$ is the set of equivalence classes of points in $X$, and $q$ is the function mapping each point $x \mapsto [x]$ to its equivalence class.

More generally, in a concrete category, given an object $X$ and equivalence relation $\sim$ on $|X|$, we can talk about the quotient of $X$ by $\sim$ (if it exists) as a morphism $q : X \to \bar{X}$ where $x \sim y$ implies $q(x) = q(y)$ which is universal with respect to this property. That is, if $r : X \to Y$ is any morphism such that $x \sim y$ implies $r(x) = r(y)$, then there is a unique morphism $u : \bar{X} \to Y$ such that $uq = r$.

31
Definition 14. In any category, a pullback of two morphisms \( f : A \to B \) and \( g : C \to B \), if it exists, is an object \( P \) together with morphisms \( x : P \to A \) and \( y : P \to C \) such that \( fx = gy \) and \( (P, x, y) \) is universal with respect to this property. That is, if \( s : D \to A \) and \( t : D \to C \) are any morphisms such that \( fs = gt \), then there is a unique \( u : D \to P \) such that \( xu = s \) and \( yu = t \).

In any category, the pullback of a morphism \( f : A \to B \) with itself, if it exists, is called the kernel of \( f \). In the category of sets, for any function \( f : A \to B \), the kernel of \( f \) is the equivalence relation \( x \sim y \) if \( f(x) = f(y) \) on \( A \), together with the two projections into \( A \). Since \( \text{Der} \) is finitely complete, every map \( \phi : \Delta \to \Gamma \) has a kernel. The underlying set of a pullback of derivations can be computed as a pullback of underlying sets.\(^4\) In particular, for any morphism \( \phi : \Delta \to \Gamma \), the underlying set of \( \ker(\phi) \) is the set of points \( \{(d, d') \in |\Delta \times \Delta| \mid \phi(d) = \phi(d')\} \), which is always an equivalence relation on \(|\Delta|\). As in any finitely complete category, this kernel can be computed as a regular subobject of \( \Delta \times \Delta \) (formally similar to relation). We call an equivalence relation arising from the kernel of a map a congruence relation, and we show that every regular epimorphism (morphism coequalizing some pair) arises as the ‘quotient’ of a derivation by a congruence relation - namely, (the underlying set of) its kernel.

Factorization systems. We simplify the definitions in [Adámek et al., 2004] for a \((\text{Epi}, \text{RegMon})\) factorization system.

Definition 15. Let \( \mathcal{C} \) be any category in which the composite of any two regular monomorphisms is again a regular monomorphism. We say that \((\text{Epi}, \text{RegMon})\) is a factorization system for \( \mathcal{C} \) if:

1. For any morphism \( f : c \to d \) in \( \mathcal{C} \), there is a factorization of \( f = m \circ e \)

\[
\begin{array}{ccc}
  c & \xrightarrow{f} & d \\
  \downarrow{e} & \nearrow{m} & \\
  k & &
\end{array}
\]

as an epimorphism followed by a regular monomorphism.

2. For any epimorphism \( e : a \to b \) and regular monomorphism \( m : c \to d \) and pair of arbitrary morphisms \( f : a \to c \) and \( g : b \to d \) such that \( ge = mf \), there is a unique diagonal \( u : b \to c \) such that \( ue = f \) and \( mu = g \).

\[
\begin{array}{ccc}
  a & \xrightarrow{e} & b \\
  \downarrow{f} & \nearrow{u} & \downarrow{g} \\
  c & \xrightarrow{m} & d
\end{array}
\]

\(^4\)Since \( pt \) is a right adjoint, it preserves pullbacks. The forgetful functor \( U : \text{FPos} \to \text{FSet} \) has the discrete functor \( \delta : U \) as its right adjoint, so \( U \) also preserves pullbacks.
The above axioms will guarantee that a factorization is essentially unique. If we have any two epi-regular mono factorizations \( a \xrightarrow{e} k \xrightarrow{m} b \) and \( a \xrightarrow{e'} k' \xrightarrow{m'} b \), then there is a unique isomorphism \( u : k \cong k' \) which is the diagonal of the associated square.

We claim that \((\text{Epi}, \text{RegMon})\) is a factorization system for \( \text{Der} \). Concretely, this will mean that every morphism factors as a surjection onto its image given the subderivation structure, followed by an embedding.

**Proof of Claim 15.** We start with the first claim. We have already shown by construction that \( |\Delta \times \Gamma| \cong |\Delta| \times |\Gamma| \) and \((d, g) \leq (d', g')\) iff \( d \leq d' \) in \( \Delta \) and \( g \leq g' \) in \( \Gamma \). By Claim 7, any derivation morphism preserves projection, so if \((d, g) \sqsubseteq (d', g')\), then \( d \sqsubseteq d' \) and \( g \sqsubseteq g' \).

We then only have to show that if \( d \sqsubseteq d' \) and \( g \sqsubseteq g' \), then \((d, g) \sqsubseteq (d', g')\) for the projection claim.

**Lemma 15.1.** If \( d \sqsubseteq d' \) and \( g \sqsubseteq g' \), then \((d, g) \sqsubseteq (d', g')\).

**Proof.** We look at the morphism \( \top_{U(d', g')} \to \top_{U(d, g)} \). Since \( U(d, g) \) and \( U(d', g') \) are rooted with roots \((d, g)\) and \((d', g')\), so are their associated SOs. The largest element of \((\omega \times \tau)(U_d \times U_g)\) is \( U_d \times U_g \), and \( U_d \times U_g \cap (U_d' \times U_g') = U_d' \times U_g' \), so \( \top_{U(d', g')} \to \top_{U(d, g)} \) takes the root to the root.

**Lemma 15.2.** \((d, g) \leq (d', g')\) iff \( d \leq d' \) and \( g \leq g' \).

**Proof.** By construction,
\[
\Rightarrow U_d' \times U_g' \in (\omega \times \tau)(U_d \times U_g) \implies U_d' \in \omega U_d \text{ and } U_g' \in \tau U_g \text{ (as these sets cannot be ‘factored’ as a union of proper open subsets)}.
\]
\[
\Leftarrow U_d' \in \omega U_d \text{ and } U_g' \in \tau U_g \implies U_d' \times U_g' \in (\omega \times \tau)(U_d \times U_g)
\]

All the claims in (2) follow immediately from the construction similarly.

**Lemma 15.3.** On any subset \( S \subset |\Delta| \), there is a unique subderivation structure on \( S \).

**Proof.** Construct the subspace \( S \subset \text{pt}(\Delta) \) and hence the morphism \( i : S \to \Delta \), and construct the cokernel \( a, b : \Delta \rightrightarrows \Delta + S \Delta \). By Claim 14 the equalizer of \( a \) and \( b \) is a subderivation of \( \Delta \) with underlying set \( S \).

By construction \( x \leq y \) in \( S \) iff in \( \Delta \). Since any morphism is projection-preserving, \( x \sqsubseteq y \) in \( S \) implies \( x \sqsubseteq y \) in \( \Delta \).

For the projection claim, we must only show that \( x \sqsubseteq y \) in \( \Delta \) implies \( x \sqsubseteq y \) in \( S \).

**Lemma 15.4.** If \( m : S \rightrightarrows \Delta \) is a subderivation embedding, \( x, y \in S \), \( x \sqsubseteq y \) in \( \Delta \) implies \( x \sqsubseteq y \) in \( S \).

**Proof.** Follows immediately from \( S \rightrightarrows \Delta \) being a subderivation and Lemma 15.6.
We now prove the claim that closeness in a derivation restricts to closeness in a subderivation.

**Lemma 15.5.** If \( m : S \hookrightarrow \Delta \) is a subderivation embedding, \( x, y \in S \), \( x \preceq y \) in \( \Delta \) implies \( x \preceq y \) in \( S \).

**Proof.** \( x \preceq y \) in \( \Delta \) means \( U_y \in \omega U_x \). Then, by construction, \( S_y = U_y \cap S \in \omega_S(U_x \cap S) = \omega_S(S_x) \).

We give a brief example of why the reverse doesn’t hold (nor should it).

![Diagram of a subderivation structure on \{a, a', b\} is incoherent.](image)

If we look at the subderivation on \{a, a', b\} in Fig. 10, leaving out \( b' \), the element \( b \) will become close to the element \( a' \), despite not being close in the super-structure. If \( b \) and \( a \) remained in ‘separate steps’, this would force \( b \) to project to \( a' \) (as \( a' \) would be the only element in the final step). Luckily, the induced substructure on the subset does not do this: we can still see \( b \) as a dependent of \( a \) which does not project to it - it is brought ‘closer’ to \( a' \) to represent this dependency.

We have the notion of ‘coherent substructure’ to indicate when step-hood is preserved ‘on the nose’ in the substructure.

**Lemma 15.6.** Let \( m : S \hookrightarrow \Delta \) be a subderivation embedding. For \( x \in S \), denote by \( S_x \) the set of points derivationally dominated by \( x \) in \( S \), and denote by \( U_x \) the set of points derivationally dominated by \( x \) in \( \Delta \). The induced morphism \( m_{S_x}^{U_x} : \top_{S_x} \to \top_{U_x} \) is an embedding of partial orders.

**Proof.** The induced map is given dually by a homomorphism of lattices \( \omega U_x \to \omega_S(U_x \cap S) = \omega_S(S_x) \). By construction, this map is surjective. By Lemma 2, §IX.4 in [Mac Lane and Moerdijk, 1992], the associated map of partial orders is an embedding.

Proof of the fourth claim is almost immediate. Since \textbf{Der} has all pullbacks, every regular monomorphism is effective (the coequalizer of its kernel pair) by [Borceux, 1994a] 2.5.7. The fact that the kernel gives an equivalence relation follows immediately from the fact that \( \text{pt} : \textbf{Der} \to \textbf{FPos} \) and \( U : \textbf{FPos} \to \textbf{FSet} \) both have left adjoints, and hence preserve
limits. We can compute the kernel on the underlying sets, and this must be an equivalence relation.

Finally, we prove that \((\text{Epi}, \text{RegMon})\) is a factorization system for \(\text{Der}\).

**Lemma 15.7.** \((\text{Epi}, \text{RegMon})\) is a factorization system for \(\text{Der}\)

**Proof.** Let \(\phi : \Delta \to \Gamma\) be any morphism. Denote by \(\text{im}(\phi)\) the set of points \(g \in \Gamma\) such that there is some \(d \in \Delta\) where \(\phi(d) = g\). As a subset of \(\Gamma\), we can give this the unique subderivation structure. By the universal property of embeddings, \(\phi\) will factor as a set-function through \(\text{im}(\phi)\), and hence must factor through it as a morphism. This morphism is clearly surjective, so by Claim 10, it is epimorphic. So, every morphism factors as an epimorphism followed by a regular monomorphism.

Now, let \(e : \Delta \to \Gamma\) be any epimorphism (surjection), and let \(m : \Xi \to \Pi\) be any regular monomorphism, and let \(f : \Delta \to \Xi\) and \(g : \Gamma \to \Pi\) be any morphisms such that \(ge = mf\). We construct the requisite unique diagonal \(d : \Gamma \to \Xi\) by a diagram chase.

Choose any element \(b \in \Gamma\). Since \(e\) is epimorphic, by Claim 10 it is surjective, so we can find some element \(a \in \Delta\) such that \(e(a) = b\). We tentatively assign \(b\) to \(f(a)\), and we must make sure that this is well-defined.

Fixing \(b \in \Gamma\), choose any two elements \(a, a' \in \Delta\) such that \(e(a) = e(a') = b\). Clearly, \(g(e(a)) = g(e(a'))\) and hence \(m(f(a)) = m(f(a'))\). But since \(m\) is injective, \(f(a) = f(a')\). So choice of a mapping to \(b\) does not matter, and the function \(d : |\Gamma| \to |\Xi|\) mapping \(b\) to \(f(a)\) for any \(a\) such that \(e(a) = b\) is well-defined.

By construction, for any element \(a \in \Delta\), \(d(e(a)) = f(a)\). For any \(b \in \Gamma\), \(d(b)\) is equal to \(f(a)\) for any \(a\) mapping to \(b\). Choose any such \(a\), and notice that \(g(b) = g(e(a)) = m(f(a)) = m(d(b))\). So \(de = f\) and \(g = md\).

We must now only prove that \(d\) is a morphism. But \(g = md\), and \(g\) is a morphism. Since \(m\) is an embedding, \(d\) must be a morphism.

This completes the proof of Claim 15.

### 2.4. Separation

We now classify subclasses of derivations which correspond to particularly simple ‘hierarchical structures’, much like certain subclasses of partial orders (‘trees’ and ‘forests’) do.

#### 2.4.1. Forests and trees in \(\text{FPos}\)

We first consider the simpler case of trees and forests in \(\text{FPos}\).

**Definition 16.** A partial order \(P\) is a **linear order** iff for every pair of elements \(x, y \in P\), either \(x \leq y\) or \(y \leq x\).
Definition 17. A finite partial order $P$ is a **forest of trees** if for every $x \in P$, $\{y \in P \mid y \leq x\}$ is a linear order.

Definition 18. A forest of trees $P$ is a **tree** if it has a minimum $z \in P$, such that $z \leq x$ for all $x \in P$.

There is an intuitive sense in which all forests are made up of trees - each ‘connected’ as a subspace - while ‘disconnected’ from each other. Describing connectedness is easier in topological terms, though (by equivalence) the following definitions have order-theoretic characterizations as well.

We first define connected spaces in general.

Definition 19. For a topological space $X$ and subset $S \subset X$, the **subspace topology** on $S$ is the topology given by $\mathcal{O}(S) = \{U \cap S \mid U \in \mathcal{O}(X)\}$. When $S$ is open, we call the resulting subspace an **open subspace**.

Definition 20. A subset $S \subset X$ given the subspace topology is **connected** if there is no open cover of $S$ where the open sets in the cover are pairwise disjoint.

In particular, an open subspace $U \subset X$ is connected iff there are no disjoint open subsets $W \cap V = \emptyset$ such that $W \cup V = U$.

Definition 21. The **connected components** of a space $X$ are the largest connected subspaces of $X$.

For any space $X$, the connected components are closed (the complement of an open set), and are always disjoint. When $X$ is finite, they are also open.

When $X$ is a finite partial order, a subspace $S$ is connected iff it is possible to ‘zig-zag’ up or down between any two elements $s, s' \in S$, where we form a sequence of elements $z_i \in S$, where $s \leq z_1$ or $s \geq z_1$, and similarly $z_i \leq z_{i+1}$ or $z_i \geq z_{i+1}$ for each step in the sequence such that eventually we get to $s'$.

![Figure 11: A disconnected partial ordering on the set \{a, b, c, d, e\}. It has two connected components: the subspaces corresponding to \{a, b\} and \{c, d, e\}. Since $X$ is finite and every connected open subspace is prime, $X$ is a forest. Each connected component of $X$ is then a tree.](image)

For this proof, and a description of the relationship between topological connectivity and paths, see [May, 2003].
Claim 16. If \( P \) is a finite forest, the connected components of \( P \) are the open subspaces \( T_i \subset P \), each a tree, such that any pair \( T_i \cap T_j = \emptyset \) is disjoint if \( i \neq j \), and \( \bigcup_{i \in I} T_i = P \).

We write a union of disjoint sets as sums, so the above proposition can be stated that every finite forest \( P \) factors uniquely as the disjoint union \( P = T_1 + \ldots + T_n \), up to rearrangement of the summands.

Applying this statement to trees, any open subset \( U \subset T \) of a tree is a forest, and hence factors into constituents of \( T \), with \( U = K_1 + \ldots K_n \).

To define constituents formally, we introduce a notion of irreducibility for open sets.\(^6\)

**Definition 22.** An open subset \( P \subset X \) of a topological space \( X \) is prime if \( P \) cannot be expressed as the union of two proper open subsets \( U \cup V = P \).

When \( X \) is a partial order, primality will correspond to an order-theoretic property.

**Definition 23.** Call a partial order \( X \) rooted if it has a unique minimum element.

Claim 17. An open subset \( U \subset X \) of a finite partial order \( X \) is rooted iff it is prime. We write \( U_x \) for the prime open subset with root \( x \in X \).

**Note 1.** If \( X \) is a finite partial order, every prime open subset \( U \) is connected.

We can now give open-set descriptions of forests and trees using topological properties.

**Claim 18.** The following are equivalent for a finite partial order \( X \)

1. \( X \) is a forest of trees
2. Every open connected subspace \( U \subset X \) is prime
3. The intersection of any two prime open subsets \( P, Q \), \( P \cap Q \) is either \( P \), \( Q \), or \( \emptyset \).

(2) is the most ‘topological’ statement, with (3) showing how prime opens can be treated essentially the same as points (the order-theoretic statement is: any two incomparable points must not lie above a common point).

**Note 2.** A tree is a connected forest.

**Note 3.** If \( X \) is a tree, a prime open subset (or, equivalently, connected open subset) \( U \subset X \) is called a constituent.

Open subspaces are special subspaces where the inclusion picks out an open subset of the superset. In trees, these open inclusions closely correspond to picking out families of constituents, and decompose into constituents in a unique way.

\(^6\)Both irreducibility and primality are formal generalizations of a property of prime numbers. For integers, \( p \) is prime if \( p \mid ab \) implies \( p \mid a \) or \( p \mid b \). The condition \( U \cup V = P \) with \( U, V \), and \( P \) all open implies \( U = P \) or \( V = P \) is the same as the condition above. We could make a total analogy, by replacing the partial ordering of natural numbers ‘by divisibility’ with the partial ordering in the lattice, and multiplication with \( \cup \). Then the usual definition of primality amounts to \( P \subset A \cup B \) implies \( P \subset A \) or \( P \subset B \). By the distributive law in finite distributive lattices, this is equivalent to the condition just stated.
Figure 12: Since partial orders have a direction, rootedness does as well. We write smaller elements higher on the page - e.g. e and f are maximal elements in this picture, while a and b are minimal elements. The lefthand partial order is prime, since there are no properly contained opens which union to it. The righthand tree is not prime - the two open sets \{a, c\} and \{b, c\} union to it. The righthand tree is hence also not a forest (or tree), since it is connected, but not prime.

Figure 13: The subspace corresponding to the circled set is open. As a forest, this open set decomposes uniquely into connected components corresponding to \{b, c\} and \{c, e, f\}. Each of these components is connected, and since X is a tree, also prime (rooted), thereby corresponding to a constituent.

2.4.2. Forests and trees in Der

The terms of a derivation - and the constituents they are associated with - give a spatial correlate of the derivational notion of `step`. A `step` of a derivation is a constituent which cannot be simply included in a subspace of another constituent. However, objects of Der in general do not have to be cleanly separated into `disjoint steps`, and we would like to distinguish those which are from those which are not. We give an example.

**Example of a non-separated derivation.** A set function \( f : X \to Y \) puts no conditions on whether or not \( X \) and \( Y \) have any elements in common. For example, we may have a subset \( U \subset X \cap Y \) which is fixed by \( f \), in that \( f(u) = u \) for all \( u \in U \). If \( X \) and \( Y \) are spaces, \( f \) continuous, and \( U \) a shared open subset, then \( U \) will openly belong to two `different` spaces.

In particular, \( X \) and \( Y \) may each have a unique largest element, making them terms. Then the subset \( U = \{a\} \) will be part of two distinct terms.
Figure 14: A derivation with a stage $U = \{a\}$ belonging openly to distinct terms.

It is easily shown that this is a derivation. $\{y < a\}$ has three opens (itself, $\{a\}$ and $\emptyset$), and $\omega U_y \rightarrow \omega U_x$ takes $\{y < a\}$ to $\{x < a\}$, fixing $\{a\}$ and $\emptyset$. There are no covers other than the trivial ones, and each collection of sections $\omega Z$ is a finite distributive lattice.

The obvious points $x$ and $y$ are also terms; $y$ is not dominated by anything, and $x$ only by $y$, but $\{x < a\} \not\in \omega \{y < a\}$.

From the general point of view, there is nothing strange about this - it is expressing the case where the domain and codomain of a function $f$ share a subset fixed by $f$. This may even be useful for other theories of syntactic structure; however, here we restrict attention to a simpler case. We would like the derivation to be cleanly separated into terms, steps of the derivation, such that each projection of each feature belongs to exactly one step.

**Steps of a derivation.** Consider again the derivation we have been representing informally by the ‘tree’ in Fig. 15.

![Diagram of a derivation](image)

Figure 15: Informal picture of separated derivation
The underlying partial order is actually the one depicted in Fig. 16.

![Figure 16: Underlying partial order of Fig. 15](image)

Now, \( a' \preceq b' \) is the only closeness relation. In this derivation, closeness breaks the poset apart into 3 partitions: \( \{a\} \), \( \{b\} \) and \( \{a', b'\} \). Drawing boxes around each equivalence class, the partition blocks themselves become naturally ordered as depicted in Fig. 15, inheriting their ordering from the ordering of elements inside the partitioning. There is an ordering \( B \preceq C \) between blocks if there is some ordering \( b \preceq c \) between elements \( b \in B \) and \( c \in C \) in the blocks, or if \( B \preceq C \) can be obtained in the transitive closure of these relations. We would like this ordering of blocks to be a partial order. However, such an ordering is not well-defined for arbitrary partitions (consider the partition \( \{a, b'\}, \{a', b\} \)). We recapitulate the theory of regular partitions on finite partial orders from [Codara, 2009], which describe when a partition is ‘compatible’ with the underlying ordering. The description is quite simple: a regular partition is one arising as the fibers of a regular epimorphism, and the blocks inherit ordering from the quotient.

We will characterize derivations whose closeness relation naturally partitions them apart into steps. From this definition, it will be straightforward to characterize those derivations whose steps are arranged into ‘trees’ and ‘forests’.

**Regular partitions.** In \( \text{FSpc} \cong \text{FProset} \), the category of finite preorders, a regular epimorphism \( f : B \rightarrow A \) is exactly a quotient of \( B \) by some equivalence relation with the quotient topology. Given an equivalence relation \( \sim \) on a preorder \( B \), its quotient can be constructed as follows: (i) the underlying set is the set of equivalence classes of \( B \); (ii) which we endow with smallest preorder containing the relations \([a] \leq [b] \) whenever \( a \leq b \) in \( B \).

Regular epimorphisms in \( \text{FPpos} \) abstract the situation from sets and spaces. Given an equivalence relation \( \sim \) on a partial order \( B \), the quotient of \( B \) by \( \sim \) can be computed by first constructing the quotient preorder, then ‘soberifying’. The soberification of a preorder \( P \) takes the quotient of \( P \) by the relation \( x \sim y \) if \( x \leq y \) and \( y \leq x \). The resulting preorder is always antisymmetric, and hence a partial order.

The regular epimorphisms in \( \text{FPpos} \) are exactly those arising as a quotient. Given a regular epimorphism \( f : B \rightarrow A \) of finite partial orders, we can construct a canonical equivalence relation associated to \( f \): its kernel. This is defined to be the relation \( a \sim b \).
iff \( f(a) = f(b) \). The quotient of \( B \) by the kernel of \( f \) is an order-preserving function \( k : B \to \hat{B} \) isomorphic to \( f \).

We call an equivalence relation arising as the kernel of a quotient of partial orders a **regular partition** of \( B \).

[Codara, 2009] gives the following results for regular morphisms in the category \( \text{FPos} \) (Codara uses ‘poset’ to mean ‘finite poset’).

**Definition 2.7** (Blockwise order). Let \((P, \leq)\) be a poset and let \( \pi = \{B_1, B_2, \ldots, B_k\} \) be a partition of the set \( P \). For \( x, y \in P \), \( x \) is blockwise under \( y \) with respect \( \pi \), written

\[ x \preceq \pi y, \]

if and only if there exists a sequence

\[ x = x_0, y_0, x_1, y_1, \ldots, x_n, y_n = y \in P \]

satisfying the following conditions:

1. for all \( i \in \{0, \ldots, n\} \), there exists \( j \) such that \( x_i, y_i \in B_j \),
2. for all \( i \in \{0, \ldots, n-1\} \), \( y_i \leq x_{i+1} \).

**Definition 2.8** (Fibre-coherent map). Consider two partially ordered sets \((P, \leq_P)\) and \((Q, \leq_Q)\). Let \( f : P \to Q \) be a function, and let \( \pi_f = \{f^{-1}(q) \mid q \in f(P)\} \) be the set of fibres of \( f \). We say \( f \) is a fibre-coherent map whenever for any \( p_1, p_2 \in P \), \( f(p_1) \leq_Q f(p_2) \) if and only if \( p_1 \preceq_{\pi_f} p_2 \).

Codara gives the following result using these definitions.

**Proposition 2.4** In \( \text{FPos} \), regular epimorphisms are precisely fibre-coherent surjections.

Codara characterizes regular partitions as follows.

**Definition 3.4** A regular partition of a poset \( P \) is a poset \((\pi_f, \preceq)\) where \( \pi_f \) is the set of fibres of a fibre-coherent surjection \( f : P \to Q \), for some poset \( Q \), and \( \preceq \) is the partial order on \( \pi_f \) defined by

\[ f^{-1}(q_1) \preceq f^{-1}(q_2) \text{ if and only if } q_1 \leq q_2, \]

for each \( q_1, q_2 \in Q \).

Finally giving the following result:
Theorem 3.2. If \( P \) is a poset, \( (\pi = \{B_1, B_2, \ldots, B_k\}, \preceq) \) is a regular partition of \( P \) if and only if \( \pi \) is a partition of the underlying set \( P \), and \( \preceq \) is a partial order on \( \pi \) such that for each pair \( B_i, B_j \) of blocks of \( \pi \), and for all \( x \in B_i \), \( y \in B_j \),

\[
x \preceq \pi y \text{ if and only if } B_i \preceq B_j,
\]

where \( \preceq \pi \) is the blockwise quasiorder [preorder] induced by \( \pi \).

By Def 3.4, this ordering of blocks is actually the partial order \( P/\sim \), with each equivalence class standing for the block. Summarizing, \( f : P \to Q \) is a regular epimorphism of finite posets if and only if for the partition \( \pi \) on \( P \) of fibres of \( f \), the blockwise ordering of elements in \( P \) coincides with the ordering of the image of the blocks they are contained in under \( f \).

This has the consequence that we can translate between an induced ordering between blocks (subsets) of a partial order \( P \) and blockwise ordering between elements in those blocks. We can use this to define a good notion of a space of points being divided into terms, coherent with the ordering of elements in \( \Delta \).

**Separation by terms, derivational forests and trees.** General derivations are quite abstract, so that \( \text{Der} \) is a well-behaved and fairly simple category to manipulate. However, one consequence is that, despite common sense, closeness is not generally transitive. All objects of \( \text{Der} \) which we will use to model syntactic derivations will however have a transitive closeness relation, and it will in fact separate the derivation into steps.

**Definition 24.** We say that a derivation is **transitive** if \( x \preceq y \) and \( y \preceq z \) imply \( x \preceq z \).

**Claim 19.** Let \( \Delta \) be a transitive derivation. Every point \( x \in \Delta \) is close to some term.

**Proof.** Suppose that \( x \) is a term. Then \( x \preceq x \), and we are done. If \( x \) is not a term, then there must be some point \( y \neq x \) such that \( y \preceq x \). If \( y \) is a term, we are done. Otherwise, there must be some \( z \neq y \) such that \( z \preceq y \), and by transitivity \( z \preceq x \). If \( z \) is a term, then we are done, otherwise we can produce a \( w \neq z \) such that \( w \preceq x \). This process must eventually terminate in a term since \( \Delta \) is finite. We can continually produce smaller points \( x \) is close to, and we will either hit a non-minimal element which is a term, or a minima, which must be a term. \( \square \)

**Claim 20.** We define a relation \( xRy \) if there is a term \( t \) such that \( t \preceq x \) and \( t \preceq y \). For any derivation, this relation is reflexive and symmetric. If \( \Delta \) is transitive, \( R \) is transitive (and hence an equivalence relation) if and only if for each point \( x \in \Delta \), there is a unique term \( t \) which \( x \) is close to.
Proof. $\Rightarrow$) Suppose $R$ is transitive. Let $x$ be any point, and let $t, s$ be any two terms such that $t, s \preceq x$ (such a term must always exist by Claim 19). Then $tRx$ (since $t \preceq t$ and $t \preceq x$) and $xRs$. By the transitivity of $R$, $tRs$, so that there is some term $r$ such that $r \preceq t, s$. But since $t$ and $s$ are terms, $r = t = s$.

$\Leftarrow$) Suppose that for every point $x$, there is exactly one term $t$ such that $t \preceq x$. Now suppose that $xRy$ and $yRz$. The first relation means there is a term $t \preceq x, y$, and the second means there is a term $s \preceq y, z$. Applying the assumption to $y$, $t = s$. Then $t \preceq x, y, z$, so $R$ is transitive.

Definition 25. A derivation $\Delta$ is separated if $\Delta$ is transitive, and the above relation is a regular partition of $\text{pt}(\Delta)$.

$R$ being an equivalence relation means each point is close to exactly one term - i.e. each element of $\Delta$ is in exactly one ‘step’, and this step is rooted. This partition is regular iff $x \preceq_{\pi} y$ is blockwise ordered by terms iff $f(x) \leq f(y)$, where $f$ is the function taking each point to its equivalence class under the quotient map.

$\Delta$ is separated if we can partition the points into steps of the derivation, where the natural ordering between points translates into an ordering between term-blocks in a particularly ‘nice’ way. We can then treat the derivation as if it is broken apart into ‘chunks.’

The blockwise ordering translates the derivational ordering to an ordering between ‘blocks’ corresponding to steps. The fact that such a translation is possible - i.e. the the blockwise ordering is induced by the derivational ordering - is exactly the constraint that the partition be regular.

Claim 21. A coherent morphism $\phi : \Delta \to \Gamma$ between separated derivations preserves the equivalence relation $R$ above, in that $xRy$ in $\Delta$ implies $\phi(x)R\phi(y)$ in $\Gamma$.

Proof. Suppose $xRy$ in $\Delta$. Then there is a unique term $t$ such that $t \preceq x, y$. Since $\phi$ is coherent, $\phi(t) \preceq \phi(x), \phi(y)$. By Claim 19, there is a unique term $s$ of $\Gamma$ such that $s \preceq \phi(t)$. By transitivity, $s \preceq \phi(x), \phi(y)$, so $\phi(x)R\phi(y)$ in $\Gamma$.

Definition 26. A separated derivation $\Delta$ is a forest of terms if for any two terms $t$ and $s$ of $\Delta$, $U_t \cap U_s$ is either equal to $\emptyset$, $U_t$, or $U_s$.

Definition 27. We call a derivation a tree of terms if it is a forest of terms with a unique term $U_r$ such that for all terms, $U_r \cap U_t = U_t$, the root term.

Claim 22. Any forest of terms $\Delta$ in the sense of Def. 26 factors as a coproduct $\Delta = \bigsqcup_{i \in I} \Delta_{/U_{r_i}}$, where $U_{r_i} \in \mathcal{O}($Def. $\Delta)$ are the constituents associated to the minimal terms of $\Delta$, and $\Delta_{/U_{r_i}}$ are the open subderivations associated to these constituents. Additionally, each $\Delta_{/U_{r_i}}$ is a tree of terms, in the sense of Def. 27, and each inclusion $\Delta_{/U_{r_i}} \hookrightarrow \Delta$ is a constituent-preserving embedding.
Proof. In appendix.

A tree of terms is a derivation which separates into steps which are blockwise ordered as a tree, and analogously for a forest of terms. Claim 22 gives an analog of Claim 16 with respect to the blocks of steps in a separated derivation. As with order-theoretic forests, the decomposition of a forest into disjoint trees is essentially unique, up to ordering of the summands, and isomorphism in each summand. This makes it easy to model workspaces which break down into component objects, and to describe operations acting on a sum of trees (which uniquely decomposes back into trees).

Note 4. The condition that \( \Delta \) is a tree of terms does not at all imply that for a stage \( X \in O(\Delta) \), the syntactic object \( T_X \) represented is a tree or forest. It only implies that it is rooted by the term, and that these ‘blocks’ are ordered as a tree. This leaves room for feature-geometry and feature-sharing.

Suppose that \( (X, \sim_X) \) and \( (Y, \sim_Y) \) are partial orders with equivalence relations on the underlying sets and \( q_X : X \to \bar{X} \) and \( q_Y : Y \to \bar{Y} \) are their quotients. Let \( f : X \to Y \) be an order-preserving function which preserves the relation, in the sense that if \( x \sim_X x' \), then \( f(x) \sim_Y f(x') \). Then there is always an induced map \( q_f : \bar{X} \to \bar{Y} \) between the quotients such that \( q_Y f = q_f q_X \). This is simply a consequence of the fact that \( \bar{X} \) is ‘universal’ amongst continuous maps out of \( X \) where the equivalence relation becomes equality. Since \( f \) is order and equivalence-preserving, and \( q_Y : Y \to \bar{Y} \) takes \( \sim_Y \) similar elements to equal elements, \( q_Y \circ f : X \to Y \to \bar{Y} \) is an order-preserving map taking \( \sim_X \) similar elements of \( X \) to equal elements. By the universal property of quotients (i.e. of coequalizers), there is then a map \( q_f : \bar{X} \to \bar{Y} \).

Definition 28. If \( \Delta \) is a separated derivation, let \( R \) denote the equivalence relation ‘is in the same term’, and call the quotient \( t_\Delta : pt(\Delta) \to pt(\Delta)/R \equiv T(\Delta) \) the term-quotient of \( \Delta \).

Claim 23. If \( \Delta \) is a forest of terms (resp. tree of terms), then its term-quotient is a forest of terms (resp. tree of terms).

These follow immediately from the fact that the partitions are regular, and the blocks are ordered as forests (resp. trees). There is one point in \( T(\Delta) \) for each ‘step’ of the derivation. Thought of as equivalence classes (the fibers of the quotient), each such point corresponds to a block of elements in \( \Delta \) all close to the same term \( t \).

Claim 24. If \( \phi : \Delta \to \Gamma \) is a coherent morphism between separated derivations, then there is a unique order-preserving function \( q_\phi : T(\Delta) \to T(\Gamma) \) between the term-quotients of \( \Delta \) and \( \Gamma \) making the following diagram commute.
This follows immediately from the universal property of quotients. This simply says that coherent morphisms between separated derivations lead to maps between their associated term-quotients.

**Note 5.** A finite partial order as a ‘static’ discrete derivation is separated iff its connected components each have a unique root. A separated finite poset can be thought of as a family of rooted spaces - or ‘labeled’ lexical items (potentially with feature geometry).

**Claim 25.** For a separated derivation $\Delta$ and term $t$, we write $B_t = \{ x \in \Delta \mid t \preceq x \}$ for the block associated to $t$.

If $\Delta$ and $\Gamma$ are separated:

1. $\Delta \times \Gamma$ is separated, and its terms $(t, s)$ - which correspond exactly to pairs of terms $t \in \Delta$ and $s \in \Gamma$ - have blocks $B_{(t, s)} = B_t \times B_s$

2. $\Delta + \Gamma$ is separated, and its terms $t$ - exactly terms of either $\Delta$ or $\Gamma$ - are associated to blocks $B_t$ identical to those in the summand.

**Proof.** These follow immediately from the results in Claim 15.

---

**2.5. Constituency**

Recall that we call a map $\phi : \Delta \rightarrow \Gamma$ open if for each open subset $U \subset \Delta$, the image $\phi(U)$ is open in $\Gamma$. We call $\phi$ constituent-preserving if $\phi$ is open, term-preserving, and coherent. Recall also that each inverse image map $\phi^{-1} : \mathcal{O}(\Gamma) \rightarrow \mathcal{O}(\Delta)$ has a left adjoint $\phi_! : \mathcal{O}(\Delta) \rightarrow \mathcal{O}(\Gamma)$ taking each $D \in \mathcal{O}(\Delta)$ to the smallest $G \in \mathcal{O}(\Gamma)$ such that $D \subset \phi^{-1} G$.

If $\Delta$ is a derivation and $U$ is an open subset, we denote the subderivation on $U$ by $\Delta / U$.

We first see what openness has to do with constituency for finite partial orders.

**Claim 26.** If $X$ and $Y$ are finite partial order trees, an order preserving function $f : X \rightarrow Y$ is an open map iff it takes constituents to constituents, i.e. if $K \subset X$ is a constituent, then $f(K) = \{ y \in Y \mid \exists x \in K \text{ such that } f(x) = y \}$ is a constituent of $Y$.

**Proof.** Let $f : X \rightarrow Y$ be an open map of trees. $\Rightarrow$) A constituent $K$ will have open image $f(K)$, but since there is a unique minimal element $k \in K$, and this function is order-preserving, then all elements of $f(K)$ are ordered above $f(k)$; in particular, $f(K)$ has a minimal element, and since $Y$ is a tree, this is a constituent. $\Leftarrow$) Suppose $f$ is constituent-preserving. If $U$ is any open set of $X$, then $U$ factors uniquely into constituents $U = \bigsqcup K_i$. 

---

pt($\Delta$) $\mapright{pt(\phi)}$ pt($\Gamma$)

$q_{\Delta}$ $\downarrow$

$T(\Delta)$ $\mapright{q_{\phi}}$ $T(\Gamma)$


45
But each \( f(K_i) \) is a constituent (in particular, open) by hypothesis, and, for any set function, the image of a union is the union of the images, and hence \( f(U) \) is open.

Claim 27. Let \( \phi : \Delta \to \Gamma \) be an open morphism between arbitrary derivations. Then the induced map \( \top_D \to \top(\phi_D) \) is surjective for each \( D \in \mathcal{O}(\Delta) \).

Proof. Since \( \phi \) is open, the map \( D \to \phi_D \) is surjective. This is equivalent to the statement that \( \phi^{-1}_D : \mathcal{O}(\phi_D) \to \mathcal{O}(D) \) mapping \( G \) to \( \phi^{-1}G \cap D \) is injective. Now \( \tau(\phi_D) \subset \mathcal{O}(\phi_D) \) is a sublattice inclusion, and so is \( \omega(D) \subset \mathcal{O}(D) \). Furthermore, the induced map takes \( G \in \tau(\phi_D) \) to an element of \( \omega(D) \), taking \( G \) to \( \phi^{-1}_D(G) \). Since \( \phi^{-1}_D \) is injective, its restriction to these subsets is injective, so the associated \( \top_D \to \top(\phi_D) \) is surjective.

Claim 28. Let \( \phi : \Delta \to \Gamma \) be a constituent-preserving map between separated derivations. Then the induced \( q_\phi : T(\Delta) \to T(\Gamma) \) is open.

Proof. In appendix.

As a corollary, if \( \Delta \) and \( \Gamma \) are trees of terms, a constituent-preserving map \( \phi : \Delta \to \Gamma \) in the derivational sense is associated to a constituent-preserving map \( q_\phi : T(\Delta) \to T(\Gamma) \) in the order-theoretic sense.

Recall that in a finite partial order tree \( T \), any open subset \( U \subset T \) factors uniquely into a coproduct \( U = \bigsqcup_{i \in I} K_i \) of constituents \( K_i \subset T \), where the \( K_i \) are the connected components of \( U \).

For trees of terms, we have a similar result.

Claim 29. Let \( \Delta \) be separated, and let \( U \subset \Delta \) be an open subset such that \( \Delta/U \hookrightarrow \Delta \) is a constituent-preserving embedding. Then \( \Delta/U \) is separated. Furthermore, if \( \Delta \) is a forest of terms, then so is \( \Delta/U \).

Proof. Let \( B_t = \{ x \in \Delta \mid t \preceq x \} \) for any term \( t \in \Delta \). For any \( x \in B_t \), if \( x \in U \), then \( B_t \subset U \). This can be seen from the fact that since \( \Delta/U \) must be transitive, \( x \) must be close to some term \( s \preceq x \) of \( \Delta/U \). But since the inclusion is term-preserving, \( s \preceq x \) in \( \Delta \), and by separation of \( \Delta \) we have \( t = s \). Since \( \Delta/U \) is closed under derivational dominance, \( B_t \subset U \). Then, two elements are clearly \( R \)-similar in \( \Delta/U \) if in \( \Delta \), and there is a blockwise ordering between elements of \( \Delta/U \) if there is a blockwise ordering between them in \( \Delta \). Since each block overlapping with \( U \) is totally contained in \( U \), they inherit the ordering from \( \Delta \) and \( \Delta/U \) is separated.

If \( \Delta \) is a forest of terms, it is immediately true that \( \Delta/U \) is, since if \( t, s \) are terms of \( \Delta/U \), \( U_t \cap U_s \) in \( \Delta/U \) must be the same as in \( \Delta \).

It is an immediate corollary that when \( \Delta \) is a tree of terms, each constituent-preserving embedding \( \Delta/U \hookrightarrow \Delta \) factors into trees of terms \( \Delta/K_i \). Since each inclusion \( \Delta/K_i \hookrightarrow \Delta/U \)
and $\Delta/U \hookrightarrow \Delta$ are constituent-preserving embeddings, their composites are, and we get constituent-preserving embeddings $\Delta/K_i \hookrightarrow \Delta$ from trees of terms, viewed as a ‘derivational constituents’ of $\Delta$. In this way, each constituent-preserving embedding corresponds uniquely to a family of derivational constituents of $\Delta$. Clearly, a constituent-preserving embedding from a tree of terms into a tree of terms corresponds to all the points up to some term (‘completed step’) of $\Delta$.

**C-command.** Constituent-preserving maps reveal c-command as a kind of ‘invariant’.

We first establish a ‘spatial’ characterization of c-command.

**Claim 30.** Let $X$ be any finite partial order with open set lattice $\mathcal{O}(X)$. This lattice is a Heyting algebra in a unique way, where $U \Rightarrow V$ is the largest $A \in \mathcal{O}(X)$ such that $U \cap A \subset V$.

**Proof.** See JOHNSTONE or MAC LANE or BORCEUX.

As usual in Heyting algebras, we define $\neg a$ in an algebra $H$ to be $a \Rightarrow 0$, where 0 is the bottom element of $H$. Then, in a lattice $\mathcal{O}(X)$, $\neg U$ is the largest open subset disjoint from $U$.

**Claim 31.** Let $X$ be a finite partially ordered tree with open set lattice $\mathcal{O}(X)$. Let $K \subset X$ be any constituent of $X$. Then the connected components of $\neg K$ are exactly the constituents c-commanding $K$.

**Proof.** In appendix.

We link the above result to open (constituent-preserving) maps.

**Claim 32.** An order-preserving function $f : X \rightarrow Y$ between finite partial orders is open if and only if $f^{-1} : \mathcal{O}(Y) \rightarrow \mathcal{O}(X)$ is a morphism of Heyting algebras, i.e. $f^{-1}(U \Rightarrow V) = f^{-1}U \Rightarrow f^{-1}V$.

**Proof.** See BORCEUX.

As an immediately corollary, if $f : X \rightarrow Y$ is an open map, then $f^{-1}(\neg K) = \neg(f^{-1}K)$. We then might expect that constituent-preserving maps between trees preserve c-command in the reverse direction in some sense.

**Claim 33.** Let $f : X \rightarrow Y$ be a constituent-preserving map between finite partially ordered trees. Let $K \subset Y$ be any constituent of $Y$ and let $V \subset Y$ be any constituent c-commanding $Y$. Then each component of $f^{-1}V$ c-commands some component of $f^{-1}K$.

**Proof.** In appendix.

As a corollary, if $f^{-1}K$ is itself a constituent, then $f^{-1}V$ consists of constituents mapping into $V$ which c-command $f^{-1}K$. 

47
2.6. Summary

In §2.1, we defined a category Der of derivations, which are generalizations of finite partial orders. Namely, a derivation $\Delta$ is made up of a space $X$, thought of as a ‘state space’ of a derivation, together with an ‘atlas’ $\omega$ which shows how the derived objects look at each open $U \subset X$ via a map $U \to \top_U$. Morphisms of derivations $\phi : \Delta \to \Gamma$ are order-preserving functions on the underlying spaces which locally compare derived SOs $\top_U \to \top_V$ whenever $\phi(U) \subset V$ for $U \subset \Delta$ and $V \subset \Gamma$.

In §2.2, we described qualitative properties of derivations. One major aspect of derivations is that they capture projection intrinsically. Given a point $x \in \Delta$, for each open $U \subset \Delta$ such that $x \in U$, we will have a point $x_U \in \top_U$ corresponding to the ‘projection’ of $x$ as it is represented in the derived SOs at $U$. $a$ is a projection of $b$ essentially if $b$ maps to $a$ under some series of operations. We also described how the underlying topological properties of $\Delta$ relate to the derived objects via the sheaf condition. Namely, inclusions $U \hookrightarrow V$ correspond to ‘state updates’ $\top_U \to \top_V$, and unions $\bigcup U_i = U$ correspond to ‘collation’ of derived objects $\top_U \to \top_V$, showing in what sense our ‘atlas’ various continuously over the state space. In fact, there is a pair of adjoint functors relating derivations to derived objects. Namely, every finite partial order can be thought of as a ‘static’ derivation containing just those SOs $(P \mapsto iP)$, while every derivation is associated to a finite partial order of SOs it derives $(\Delta \mapsto \top_\Delta)$. These functors determine each other. We also can characterize which points are in the ‘same step’ via a closeness relation.

In §2.3, we showed concrete properties of Der. Namely, we showed that traditional categorical definitions of monomorphism, epimorphism, isomorphism, regular monomorphism, image factorization, product, and coproduct correspond to injections, surjections, ‘bi-morphic’ bijections, subspace embeddings, surjection-embedding morphism factorization, cartesian product, and disjoint union, respectively. These lead to many good straightforward constructions on derivations, and show how Der is intuitively ‘spatial’.

In §2.4, we generalized topological descriptions of trees and forests to derivations. Namely, we gave a characterization of derivations whose ‘roots with respect to closeness’ separate the derivation cleanly into ‘steps’. The ordering of these steps as blocks is determined by the derivational dominance relation between the elements within them. We characterized maps which preserve this block structure through term-preserving and coherent maps. We can associate the blocks of each such derivation with an actual partial order in a canonical way, such that coherent maps lead to order-preserving functions between the associated partial orders.

In §2.5, we constructed constituent-preserving maps between trees and showed how they preserve c-command for finite partial orders. We then generalized the result to trees of terms for derivations. We also showed how forests of derivations in this sense factor into component trees, just as with partial order forests. This gave us a characterization of ‘derivational constituents’ - subderivations consisting of some completed step and all preceding steps.
3. Formal theory of grammars and syntactic operations

Throughout this paper, we have been treating derivations from a ‘model’ or ‘representational’ perspective, where we study sets with structure which meet certain conditions. All of the properties studied so far (being separated, being a forest or tree of terms, being discrete or indiscrete) are all properties which can be studied by looking at the categorical properties of derivations, regardless of how they were built or assumptions about their language-specific properties.

We now study ways to construct a derivation recursively, starting with some basic objects which we can think of as lexical items together with a fixed set of rules.

This requires developing a theory of syntactic operations, both with respect to what they do and what kinds of objects they manipulate, as well as how they are reflected in the derivation. A theory of constraining these syntactic operations should also be developed. This in itself is a separate research program, which we cannot do justice here.

We should like to say a bit more about structure-building than algebraic operations do. While we would like to think of an operation as taking some family of syntactic objects \( \{S_i\}_{i \in I} \) and producing a new one \( Z \), we should like the structure of each of the inputs to have some relation to the structure of \( Z \). Specifically, we would like the operation to see the \( S_i \) and \( Z \) as structured things, and to relate those structures, something which an algebraic operation doesn’t do by itself.

Essentially, we would like homomorphisms \( f_i : S_i \to Z \) which relate the structure of each input to some substructure of the output. If the \( S_i \) and \( Z \) are ordered structures, we will minimally want the \( f_i \) to be order-preserving. Since \( FPos \) has finite coproducts, the family of \( f_i \)'s is equivalent to a single map \( +_{i \in I} f_i : \coprod_{i \in I} S_i \to Z \) from their coproduct. We can define a syntactic operation on some family of syntactic objects \( S_i \) simply to be a map \( f : \coprod_{i \in I} S_i \to Z \) from their coproduct which meets certain conditions, particular to our theory of syntactic operations.

The main goal will then be to build a mechanism which we can use to assign sets of operations to derivations in a systematic way. We defer the actual task of developing a substantive theory of which operations are used in natural language, though we give a general framework for theorizing rules and the objects they generate. All operations in this section are non-movement operations.

Sketch of implementation of syntactic operations for recursively constructing derivations

We take an \( n \)-ary operation of derivations to be a map \( \coprod \Delta_i \to Z \), equivalent to a collection of \( n \) maps \( \Delta_i \to Z \) and hence \( n \) maps \( \top_{\Delta_i} \to Z \). There is an intuitively natural way to extend a derivation to include ‘one new step’ \( Z \), built from a syntactic operation. If derivations \( \{\Delta_i\}_{i \in I} \) have final stages \( \top_{\Delta_i} \), then their coproduct \( \coprod_{i \in I} \Delta_i \) will have final stage \( \coprod_{i \in I} \top_{\Delta_i} \), and we can apply an operation \( f : \coprod_{i \in I} \top_{\Delta_i} \to Z \) to it. We extend the derivation
\( \prod \Delta_i \) along \( f \) by constructing a derivation which has \( \prod \Delta_i \) as an open subderivation, where the space \( Z \) is ‘on top’ of all of the preceding operations, with mappings \( f_i : T_{\Delta_i} \to Z \) from each of the final stages from the input.

\[ X \rightarrow \ldots \rightarrow Y \rightarrow A \rightarrow \ldots \rightarrow B \]

\( T_{\Delta_1} \rightarrow Z \rightarrow T_{\Delta_n} \)

Figure 17: Informal picture of the derivation constructed by extending a family of derivations \( \{\Delta_i\}_{i \in I} \) along a syntactic operation \( f : \prod_{i \in I} T_{\Delta_i} \to Z \).

We will construct such extensions in full generality. However, if \( Z \) is not rooted (“uniquely labeled”), then it will not be a term in the sense described in Def. 8. Throughout this section, all of the operations we will propose make the common assumption that the output \( Z \) is always rooted - i.e. the output of any syntactic operation has a unique labelling element, contra the alternatives laid out in, e.g., [Citko, 2008]. In a particular grammar, if the basic lexical items are all rooted, and all operations produce rooted structures, we can show inductively that the generable derivations are all trees of terms.

This framework successfully, and through simple means (order-preserving functions), captures how a syntactic operation reflects structure of the inputs in the output and represents that structure in a derivation. However, all ‘instances’ of syntactic operations are unrelated from this perspective, in that each operation will apply to a different domain (the particular inputs involved), and hence are different functions. We then would like to develop an approach to rules, which give instructions on how to systematically construct operations on each family \( \{\Delta_i\}_{i \in I} \) using a single template operation.

To a given tuple of objects \( (\Delta_i)_{i \in I} \), there may be multiple distinct ways an operation can apply (or none). So we should actually construct a rule as an assignment \( G : C \to \text{Set} \) which takes a given tuple of derivations \( (\Delta_i)_{i \in I} \) and assigns it a set of \( G \)-operations of the form \( f : \prod_{i \in I} \Delta_i \to Z \).

We will actually do this functorially. We also functorialize the ‘extension’ operation. From these two constructions, we will be able to take a rule \( G \) and applicable tuple \( (\Delta_i)_{i \in I} \) and operation \( h : \prod \Delta_i \to Z \in G(\Delta_i) \) to obtain a new derivation \( (\prod \Delta_i)^h \) in a functorial manner. This means that a homomorphism \( \phi_i : (\Delta_i) \to (\Gamma_i) \) between input objects

---

\( ^7 \)We could of course relax our notion of separation and constituent to non-rooted spaces to accommodate more general theories of syntactic operations, though we will not do that here.

\( ^8 \)We do not yet exclude the possibility that this label is projected by more than one head in the case of two heads totally ‘fusing’.

\( ^9 \)See example of agreement in §1.5.1.
leads to a homomorphism between output objects $Z \to Z'$ and hence between extended derivations $(\prod \Delta_i)^h \to (\prod \Gamma_i)^{h'}$.

3.1. Syntactic operations

First we consider the standard definition of an $n$-ary operation on set.

An $n$-ary operation is defined to be any function $f : X^n \to X$ where $X^n = X \times \ldots \times X$ $n$ times.


take a pair $(a, b) \in X \times X$ to its product $f(a, b) \in X$. $f(a, b)$ does not have to have any relationship to $a$ or $b$, other than $f$ itself. In other words, $f$ as an operation does not care whether $a$ or $b$ are structured ‘by itself’ (without describing more properties of $X$ and $f$).

Compare this with, say, Set-models of some first order theory $T$, and homomorphisms between models (e.g. a theory of trees or linear orders). A homomorphism $\phi : A \to B$ will preserve structure (interpretation of relation and function symbols) between the two structured objects.

Not only would we like to have an assignment of a pair of syntactic objects $(A, B)$ to new syntactic objects $Z$, but we should like to relate structure of $A$ and $B$ to $Z$ through a pair of maps $m_A : A \to Z$ and $m_B : B \to Z$, or equivalently a single map $m : A + B \to Z$.

3.1.1. Motivating example

Following [Boston et al., 2010], we construct an example of an operation on a pair of rooted spaces $(A, B)$. Write the minimal element of each partial order as $a$ and $b$ respectively. We are interested in the operation attaching the root of $B$ to the root of $A$. Formally, this is constructed as the coproduct of $A$ and $B$, introducing the order relation $a \leq b$, and closing under transitivity. See Fig. 18.

![Figure 18: An instance of an operation attaching the root of $B$ to the root of $A$.](image)

However, we are not just interested in this instance of attaching one root to the other, but rather all instances of similar root-attachment.
**Pushout lemma.** We construct the above attachment with a pushout. We first establish the following important basic claim about pushouts.

**Claim 34.** In any category, for any pair of arrows $f : A \rightarrow B$ and $g : A \rightarrow C$, an object $D$ together with maps $k : C \rightarrow D$ and $j : B \rightarrow D$ such that $jf = kg$ is a pushout if it is universal with respect to this property. This means that for any other object $E$ and pair of maps $m : C \rightarrow E$ and $n : B \rightarrow E$ such that $nf = mg$, there is a unique map $u : D \rightarrow E$ such that $uk = m$ and $uj = n$. See picture below.

Given a morphism $f : A \rightarrow B$ and morphism $g : A \rightarrow C$, we often call $k : C \rightarrow D$ the pushout of $f$ along $g$. This can be thought of as finding the ‘best’ translation of $f$ into $C$ in the context $g$. As with all universal constructions, $D$, $k$, and $j$ are all unique up to unique isomorphism. This means that if $D'$, $k'$, and $j'$ also give a pushout, then there is a unique isomorphism $i : D \cong D'$ such that $ik = k'$ and $ij = j'$.

The most fundamental property of pushouts is that they *paste* in any category.

Given a morphism $f : A \rightarrow B$ and morphism $g : A \rightarrow C$, we often call $k : C \rightarrow D$ the pushout of $f$ along $g$. This can be thought of as finding the ‘best’ translation of $f$ into $C$ in the context $g$. As with all universal constructions, $D$, $k$, and $j$ are all unique up to unique isomorphism. This means that if $D'$, $k'$, and $j'$ also give a pushout, then there is a unique isomorphism $i : D \cong D'$ such that $ik = k'$ and $ij = j'$.

The most fundamental property of pushouts is that they *paste* in any category.

The usual statement of the pushout lemma is that if the left square is a pushout square, the right square is a pushout square if and only if the outer rectangle is a pushout. That is, we could push $k$ out along $f$ to obtain $j$. We could then push $j$ out along $g$ to obtain $c$. On the other hand, we could start by composing $f$ and $g$, then push $k$ out along $gf$. The pushout lemma states that $c$ is also this pushout.

The proof is straightforward from the universal property of pushouts, and does not rely on any particular properties of the objects or morphisms themselves.

Pushouts are used in categories of directed graphs in graph grammars, first explored in [Ehrig et al., 1973] and later in [Ehrig et al., 1997] as a generalization of string replacement rules.

We first give a simple unrestricted example of a pushout for partial orders. Let $2 = \{a, b\}$ be the discrete partial order of two unordered points, and let $S$ be the Sierpinski space of two points $\{a, b\}$ such that $a < b$. We will be interested in pushouts of the obvious function $f : 2 \rightarrow S$ acting by identity on elements.

A context $k : 2 \rightarrow P$ to any partial order is the same thing as a selection of any pair of elements $k(a), k(b) \in P$. To push $f$ out along $k$, we construct the ‘legs’ of a pushout.
A cocone on this diagram is an fposet $Q$ and operation $f : P \to Q$ with map $z : S \to Q$ such that $f k = z f$. $f$ determines the function underlying $z$: $z$ must map $a \mapsto f(k(a))$ and $b \mapsto f(k(b))$. Since $z$ must be order-preserving. Together, these say that a cocone on this diagram is any operation $f : P \to Q$ such that $f(k(a)) \leq f(k(b))$.

The map $a : P \to \hat{P}$ which adds the relation $k(a) \leq k(b)$ in $P$ and closes under transitivity together with the induced map $m : S \to \hat{P}$. The universal property says that there is a unique factorization of $f$ into $ua$.

**Pushout lemma and functoriality.** The pushout lemma says we can push a morphism out along others functorially. If $k : 2 \to P$ is the selection of elements $(p, q)$ in $P$, any order-preserving function $g : P \to Q$ leads to a selection of points $(gp, gq)$ in $Q$. There is an induced map $\hat{P} \to \hat{Q}$, which has the same underlying function as $g$, but between posets with the relevant order relations added. This corresponds to the pushout diagram:

$$
\begin{array}{ccc}
2 & \xrightarrow{k} & P \\
\downarrow f & & \downarrow g \\
S & \to & \hat{P} \\
\end{array}
\quad
\begin{array}{ccc}
P & \xrightarrow{g} & Q \\
\downarrow & & \downarrow \\
\hat{P} & \to & \hat{Q} \\
\end{array}
$$

The vertical arrows represent the addition of the order relation between the selected pair (adding $a \leq b$, $ka \leq kb$ and $gka \leq gkb$, respectively) - i.e. the ‘structural changes’. The top horizontal arrows correspond to picking/translating contexts, while the bottom horizontal arrows correspond to the associated homomorphisms between the ‘output’ objects once the order relations have been added.

The pushout lemma says that this construction can be made functorial functorial. We define $G : \mathbf{FPos} \to \mathbf{Set}$ which takes a finite partial order $P$ and returns the set of (isomorphism classes)$^{10}$ of morphisms $h : P \to \hat{P}$, where $h$ is the universal map out of $P$ such that $hp \leq hq$ for some pair of elements $(p, q)$ of $P$. The construction is functorial because if $m : P \to Q$ is any order-preserving function, and $h : P \to \hat{P}$ is any morphism adding an order relation $p \leq q$ between elements of $P$, then we may push $h$ out along $m$ to obtain $g : Q \to \hat{Q}$, which adds the order relation $m(p) \leq m(q)$. $G$ is effectively the rule which takes any fposet $P$ and pair of points $p, q$ and adds the relation $p \leq q$.

---

$^{10}$We say two operations $h : P \to A$ and $g : P \to B$ are isomorphic if there is an isomorphism $i : A \cong B$ such that $ih = g$. This is just the usual notion of isomorphism in the category of $\mathbf{FPos}$ maps out of the fixed object $P$. 

---

53
Generation of a rule. This functor seems to be ‘generated’ by a single operation: \( f : 2 \to \mathcal{S} \). We can associate the object 2 with a functor \( \mathbf{FPos}(2, -) : \mathbf{FPos} \to \mathbf{Set} \). The Yoneda lemma states that for any set-valued functor \( G : \mathcal{C} \to \mathbf{Set} \), the elements of \( G(\mathcal{X}) \) are in bijective correspondence with natural transformations \( \mathcal{C}(\mathcal{X}, -) \equiv y\mathcal{X} \to G \).

Then, \( f \) is associated with a natural transformation \( y(f) : y2 \to G \). For each object \( P \), this natural transformation takes an element \( k : 2 \to P \in y2(P) \) to the element of \( G(P) \) which can be constructed as the pushout of \( f \) along \( k \). In this case, \( y2 \) is essentially the functor which takes a partial order and returns the set \( |P| \times |P| \) - the cartesian product of the underlying set. The natural transformation takes an element \((p, q)\) of this set and associates it to the order-preserving map \( h : P \to \tilde{P} \in G(P) \) adding the relation \( p \leq q \).

For this completely unrestricted form, we view a rule as a functor \( G : \mathbf{FPos} \to \mathbf{Set} \) which returns for each \( P \) a set of (isomorphism classes) of order-preserving functions \( h : P \to Z \). We say that \( G \) is generated by an operation \( f : M \to X \in G(M) \) if the associated natural transformation \( y(f) : yM \to G \) is epimorphic, meaning that each coordinate \( \mathbf{FPos}(M, P) \to G(P) \) is surjective. Essentially, \( f \) generates \( G \) if each operation on each object \( P \) arises as the pushout of \( f \) along some ‘context’ \( k : M \to P \).

Restricting rules. We are often going to be interested not in a single syntactic object, but tuples of them. For the case of binary rules, we will be interested in a functorial assignment which takes in a pair \((P, Q)\) of syntactic objects and returns a set of rules \( h : P + Q \to Z \) which is functorial by pushout. However, we often want to restrict ‘where’ an operation can apply, and must restrict the morphisms between inputs accordingly if it is to be functorial. The category of tuples of objects with some restricted class of morphisms between them can be thought of as the formalization of the ‘context’ of admissible rule applications.

In the motivating example, we are interested only in rooted objects \( A \) and \( B \) and the operation \( h : A + B \to Z \) which attaches the root of \( B \) to \( A \). However, even restricting to the subcategory \( \mathcal{C} \subset \mathbf{FPos} \times \mathbf{FPos} \) of pairs of rooted partial orders is not sufficient, since arbitrary order-preserving maps need not preserve roots. See Fig. 19 for an example.

We would then like to not only restrict attention to the rooted spaces - spaces which this operation can act on - but we would also like to restrict attention to morphisms between those spaces which preserves the properties of that object relevant for determining proper application of the operation. In this case, the inclusions represented by the left vertical maps in Fig. 19 did not preserve roots, which is what led to the failure of being able to compare the output of the top operation with the output of the bottom one.

However, if we restrict our operation to working on \( \mathcal{C} \subset \mathbf{FPos} \times \mathbf{FPos} \), a subcategory of pairs of rooted finite partial orders with pairs of root-preserving maps between them, the construction becomes functorial. See Fig. 20.
Figure 19: If we consider the coproduct of two trees properly included into another pair, and for each pair we attach the righthand summand to the root of the lefthand summand, then the vertical set-function between the ‘outputs’, marked with $\mathcal{G}$, which makes the diagram commute, is not order-preserving. For, $a \leq b$ in the top right tree, but $a \not\leq b$ in the bottom right one. There is no map completing this diagram.

Figure 20: The set function $\checkmark$ taking an element $x$ to $f(x)$ or $g(x)$ (depending on whether that element came from the $a$-phrase or $b$-phrase) is order-preserving and makes this diagram commute, viewing the lefthand maps as $f + g : A + B \to R + S$. This is a pushout diagram.
The generating operation \( f : 1 + 1 \cong 2 \to S \) can be seen as the ‘template operation’ which attaches one element to another. The category \( C \) can be seen as the formalization of the structural description of the context the rule applies in; in this case, pairs of rooted finite partial orders and selection of their roots.

A context category \( C \subset FPos^n \) together with an operation \( f : \prod_{1 \leq i \leq n} M_i \to Z \) with \( (M_i)_{1 \leq i \leq n} \in C \) together determine the functor \( G : C \to \text{Set} \). This functor takes any object of \( C \) and returns all operations which can be viewed as an instance of \( f \) applied in some \( C \)-admissible context.

### 3.1.2. Formal theory of operations

**Operations.** We first construct a category of operations on derivations formally.

**Definition 29.** ([Mac Lane, 1971], II.6) Given categories and functors

\[
E \xrightarrow{T} C \xleftarrow{S} D
\]

the **comma category** \( (T \downarrow S) \), also written \( (T, S) \), has as objects all triples \( \langle e, d, f \rangle \) with \( d \in \text{Obj } D \), \( e \in \text{Obj } E \) and \( f : Te \to Sd \), and as arrows \( \langle e, d, f \rangle \to \langle e', d', f' \rangle \) all pairs \( \langle k, h \rangle \) of arrows \( k : e \to e' \), \( h : d \to d' \) such that \( f' \circ Tk = Sh \circ f \). In pictures,

\[
\begin{array}{ccc}
\text{Objects} & E & \xrightarrow{T} & C & \xleftarrow{S} & D \\
\langle e, d, f \rangle & \xrightarrow{f} & \langle e', d', f' \rangle
\end{array}
\]

with the square commutative. The composite \( \langle k', h' \rangle \circ \langle k, h \rangle = \langle k' \circ k, h' \circ h \rangle \), when defined.

**Definition 30.** Define \( \text{Der} \downarrow \text{FPos} \) to be the comma category defined by the following equation.

\[
1_{\text{Der}} : \text{Der} \to \text{Der} \leftarrow \text{FPos} : i
\]

We call this the **category of operations**.

Concretely, an object of \( \text{Der} \downarrow \text{FPos} \) is a triple \( \langle \Delta, P, f : \Delta \to i(P) \rangle \), which we can unambiguously write as \( f : \Delta \to P \). A morphism in this category is a pair \( \langle \phi : \Delta \to \Gamma, h : P \to Q \rangle \) such that the following diagram commutes.

\[
\begin{array}{ccc}
\Delta & \xrightarrow{f} & P \\
\phi \downarrow & & \downarrow h \\
\Gamma & \xrightarrow{g} & Q
\end{array}
\]

**Definition 31.** Define \( \Delta \downarrow \text{FPos} \) to be the comma category defined by the following equation.
Here, the left map is the functor selecting the object $\Delta$ and its identity morphism, and $i$ is the usual inclusion. We call this the **category of operations on** $\Delta$.

The objects of $\Delta \setminus \text{FPos}$ can be thought of as operations $f : \Delta \to Z$. A morphism between operations $f : \Delta \to Z$ and $g : \Delta \to Y$ is a mapping $t : Z \to Y$ between outputs, such that $tf = g$.

**From partial orders to derivations.** By the adjunction $\top(\_ \to \_)$, a map $h : \Delta \to Z$ from a derivation to a partial order is determined totally by its behavior on the derived objects, $\top h : \top\Delta \to Z$. It is then straightforward to define a context $C \subset \text{Der}^n$ and generating operation $\bigsqcup M_i \to Z$, leaving the remainder of the construction the same. We just need to verify 2 facts.

**Claim 35.** Let $h : \Delta \to Z$ be any operation, and let $\phi : \Delta \to \Gamma$ be any morphism. Then there is a partial order $P$ together with maps $h' : \Gamma \to P$ and $f : Z \to P$ such that $fh = h'\phi$, which is universal with respect to partial orders with this property.

That is, if $Q$ is any partial order and $k : \Gamma \to Q$ and $g : Z \to Q$ any pair of derivation morphisms such that $gh = k\phi$, then there is a unique order-preserving $u : P \to Q$ such that $k = uh'$ and $g = uf$. As with all universal constructions, $(P, h', f)$ is determined up to unique isomorphism, in that if $(\overline{P}, \overline{h'}, \overline{f})$ is any other such partial order, then there is a unique isomorphism $i : P \to \overline{P}$ such that $h'i = \overline{h'}$ and $fi = \overline{f}$. We will say that $(P, h', f)$ is **fposet-universal** or just **universal** with respect to $\phi$ and $h$. We often abuse terminology and call $h'$ the pushout of $h$ along $\phi$.

**Proof.** We give a construction of $h'$. Apply $\top(\_ \to \_)$ to the diagram consisting of $\phi$ and $h$ to get a diagram of finite partial orders. Take any pushout of fposets.

Construct the associated map $h' : \Gamma \to P$ by precomposing $\hat{h'} : \top\Gamma \to P$ with the canonical map $\Gamma \to \top\Gamma$. \qed
Claim 36. The universal construction above has a ‘pushout lemma’. That is, given commuting squares as below with the left square universal, the right square is universal if and only if the whole square is universal.

\[
\begin{array}{ccc}
\Delta & \xrightarrow{\phi} & \Gamma \\
f \downarrow & & \downarrow \psi \\
X & \xrightarrow{a} & Y \\
\end{array}
\begin{array}{ccc}
& & \Sigma \\
& & \downarrow h \\
& & Z \\
\end{array}
\]

That is, we assume that \((Y, g, a)\) is \(f\)poset-universal with respect to \((\phi, f)\). The claim is that \((Z, h, b)\) is \(f\)poset-universal with respect to \((\psi, g)\) if and only if \((Z, h, ba)\) is \(f\)poset-universal with respect to \((\psi \phi, f)\). That is, \(h\) is the pushout of \(g\) along \(\psi\) iff it is the pushout of \(f\) along \(\psi \phi\).

Proof. This follows straightforwardly from the universal properties of each. The proof follows that of the usual pushout lemma identically. \qed

Rules. We define rules formally. We fix a subcategory \(\mathcal{C} \subset \text{Der}^n\) thought of as conditions on the arguments of the rule \(G\). Denote objects \(\Delta \equiv (\Delta_1, \ldots, \Delta_n) \in \mathcal{C}\) and morphisms \(\phi \equiv \langle \phi_1, \ldots, \phi_n \rangle : (\Delta_1, \ldots, \Delta_n) \to (\Gamma_1, \ldots, \Gamma_n)\). For each \(\mathcal{C}\)-object, denote \(\Xi \equiv \prod_{i \in I} \Delta_i\), and for each morphism \(\phi : \Delta \to \Gamma\), denote \(\overline{\phi} \equiv \phi_1 + \ldots + \phi_n : \Xi \to \Gamma\).

Definition 32. A \(\mathcal{C}\)-rule \(G : \mathcal{C} \to \text{Set}\) is a functor with the following properties:

1. For each \(\Delta \in \mathcal{C}\), \(G(\Delta)\) is a set of isomorphism classes of objects \(f : \Xi \to Z\) in \(\Delta \setminus \text{FPos}\).

2. Let \(\phi : \Delta \to \Gamma\) be a \(\mathcal{C}\)-morphism and let \(f : \Xi \to Z\) be a representative of an operation in \(G(\Delta)\). Then there is a (necessarily unique) element \(f' : \Gamma \to Z'\) in \(G(\Gamma)\) such that there is an \(f\)poset-universal diagram

\[
\begin{array}{ccc}
\Xi & \xrightarrow{\overline{\phi}} & \Gamma \\
\downarrow f & & \downarrow f' \\
Z & \xrightarrow{p} & Z' \\
\end{array}
\]

and \(G(\phi) : G(\Delta) \to G(\Gamma)\) maps \(f\) to \(f'\). Note that \(\overline{\phi}\) and \(f\) determine \(f'\) and \(p\).

Definition 33. Let \(\mathcal{C} \subset \text{Der}^n\) and let \(G : \mathcal{C} \to \text{Set}\) be a rule. Take any \(\mathcal{M} \in \mathcal{C}\) and operation \(f : \overline{\mathcal{M}} \to Z \in G(\mathcal{M})\). We say that \(f\) and \(\mathcal{C}\) generate \(G\) if the associated natural transformation \(y(f) : y\mathcal{M} \to G\) is epimorphic; that is, if each coordinate of the natural transformation is a surjective set function.

Claim 37. A rule \(G\) generated by \(f\) is completely determined by \(f\). That is, if \(G, H\) are any two \(\mathcal{C}\)-rules both generated by \(f\), then \(G = H\).
Proof. This is immediate by construction. Take any \( C \subset \text{Der}^n \), object \( M \in C \) and operation \( f : \overline{M} \to X \). If \( f \) generates \( G : C \to \text{Set} \), then to compute the values of \( G(\Delta) \), we simply take the pushout of \( \top_f \) along \( \top_\phi : \overline{\Delta} \to \overline{X} \) for each \( \phi : M \to \Delta \in C(\mathcal{M}, \Delta) \). □

We would like this construction to be functorial into \( \text{Der}\setminus\text{FPos} \). Importantly, for each \( G \)-operation on a tuple \( \Delta \), we should have an associated object \( f : \overline{\Delta} \to Z \) of \( \text{Der}\setminus\text{FPos} \).

**Definition 34.** ([Borceux, 1994a], 1.6.4) Consider a functor \( F : \mathcal{A} \to \text{Set} \) from a category \( \mathcal{A} \) to the category of sets. The category \( \int F \) of “elements of \( F \)” is defined in the following way.

1. The objects of \( \int F \) are the pairs \((A, a)\), where \( A \in |\mathcal{A}| \) is an object and \( a \in FA \).
2. A morphism \( f : (A, a) \to (B, b) \) of \( \int F \) is an arrow \( f : A \to B \) of \( \mathcal{A} \) such that \((Ff)(a) = b\).
3. The composition of \( \int F \) is that induced by the composition of \( \mathcal{A} \).

Using this standard construction, if \( G : C \to \text{Set} \) is a rule on \( C \subset \text{Der}^n \), \( \int G \) is a category whose objects pairs \((\Delta, h)\), where \( \Delta \in C \) and \( h : \overline{\Delta} \to Z \) is an isomorphism class of \( G \)-operations on \( \overline{\Delta} \). A morphism \( \phi : (\Delta, h) \to (\Gamma, g) \) is a \( \mathcal{C} \)-morphism \( \phi : \Delta \to \Gamma \) such that the pushout of \( \top_h \) along \( \top_\phi \) is \( \top_g \).

**Claim 38.** Let \( G : C \to \text{Set} \) be a rule on \( C \subset \text{Der}^n \). For each object \((\Delta, h : \overline{\Delta} \to Z)\) in \( \int G \), fix any representatives of \( Z \), \( h \), and the coproduct \( \overline{\Delta} \). Then \( A_G : \int G \to \text{Der}\setminus\text{FPos} \) taking \((\Delta, h)\) to \( h : \overline{\Delta} \to Z \) is a functor, and it is determined up to natural isomorphism.

**Proof.** It is functorial because each morphism in \( \int G \) leads to a commutative diagram of the appropriate sort. It is determined up to natural isomorphism because each object is determined up to isomorphism. □

We can think of an order-preserving map \( f : Y \to Z \) between SOs as *inclusive* if the underlying function is surjective. This just says that every point of \( Z \) comes from some point of \( Y \).

**Claim 39.** Let \( \mathcal{C} \subset \text{Der}^n \) be a condition category and let \( G : \mathcal{C} \to \text{Set} \) be a rule generated by \( f : \overline{M} \to Z \) such that \( \top_f \) has underlying set-function which is surjective. Then \( \top_h \) for each \( h \in G(\Delta) \) is surjective.

**Proof.** The epimorphisms in \( \text{FPos} \) are exactly the surjections, and epimorphisms are preserved by pushout. Since \( G \) is generated by an epimorphism, all of its sections are epimorphic, and hence surjective. □

We can then describe a whole rule as inclusive iff its generating operation is.
Definition 35. Let $\mathcal{C} \subset \text{Der}^n$ be a subcategory, and let $G : \mathcal{C} \to \text{Set}$ be a rule. We say that a $G$-operation $h : \Delta_1 + \ldots + \Delta_n \to Z$ is weakly extensive if each for each coordinate $h_i : \Delta_i \to Z$, the associated map of partial orders $\top_{h_i} : \top_{\Delta_i} \to Z$ is a subspace inclusion, i.e. a regular monomorphism. We say that $G$ is weakly extensive if all of its operations are weakly extensive.

Extensions of derivations. In this section, we characterize a functor $\text{ext} : \text{Der} \setminus \mathbb{FPos} \to \text{Der}$ taking in an operation $h : \Delta \to Z$ and returning a derivation $\Delta h$. The resulting derivation has many properties which allow us to view it as ‘$\Delta$ extended along $h$’. We should like to construct a derivation $\Delta f$, such that $\Delta f$ just consists of the original derivation $\Delta$ and space $Z$, such that the space $Z$ ‘follows’ $\Delta$.

Figure 21: Informal picture of an extension. The left hand side shows derived objects associated to open subsets of a derivation $\Delta$ mapping to a poset $Z$. The right side shows select stages of the extension $\Delta^f$, where $\Delta^f/\top_{\Delta} \cong \Delta$, with new stages $U \hookrightarrow Z$ for the open subsets of $Z$. There should be an arrow $D \to W$ in $\mathcal{O}(\Delta^f)$ from an object $D$ from $\mathcal{O}(\Delta)$ to an object $W$ from $\mathcal{O}(Z)$ iff the direct image of $f \circ u : D \to \top_{\Delta} \to Z$ lies in $W \subset Z$.

Essentially, from an operation $h : \Delta \to Z$, we would like to construct a new derivation $\text{ext}(h)$ which contains $\Delta$ and $Z$ which has the property that if $x \in \Delta$ goes to $h(x) = y \in Z$, then the corresponding element $x'$ in $\text{ext}(h)$ projects to the corresponding element $y'$ in $\text{ext}(h)$. We can characterize an extension of $\Delta$ by $h$ as the universal such derivation.

Definition 36. Let $h : \Delta \to Z$ be any operation. Consider any derivation $E$ and pair of morphisms $i : \Delta \to E$ and $j : Z \to E$, or equivalently, a single morphism $k : \Delta + Z \to E$. We say that $k$ takes $h$-images to projection if for every $x \in \Delta$, $k(h(x)) \sqsubset k(x)$, that is, $kx$ projects to $k(hx)$ in $E$. 

60
Claim 40. For any operation \( h : \Delta \rightarrow Z \), there is a universal derivation \( \text{ext}(h) \) together with a map \( k : \Delta + Z \rightarrow \text{ext}(h) \) which takes \( h \)-images to projection, in the sense that if \( k' : \Delta + Z \rightarrow E \) is any morphism taking \( h \)-images to projection, then there is a unique derivation morphism \( u : \text{ext}(h) \rightarrow E \) make the following diagram commute.

\[
\begin{array}{ccc}
\Delta + Z & \xrightarrow{k} & \text{ext}(h) \\
\downarrow{k'} & & \downarrow{u} \\
\Gamma + Y & \xrightarrow{j} & \text{ext}(g)
\end{array}
\]

We prove the claim by giving an explicit construction in Claim 43. It is an immediately corollary that the construction can be made functorial, by choosing any representative of \( \text{ext}(h) \) (all isomorphic).

Claim 41. Consider the mapping on objects \( \text{ext} : \text{Der}\backslash\text{FPos} \rightarrow \text{Der} \) taking \( h \) to \( \text{ext}(h) \). We claim this extends to a functor. Let \( h : \Delta \rightarrow Z \) and \( g : \Gamma \rightarrow Y \) be any operations with extensions \( k : \Delta + Z \rightarrow \text{ext}(h) \) and \( j : \Gamma + Y \rightarrow \text{ext}(g) \) and \((\phi, f)\) any morphism between them. There is a unique morphism \( u : \text{ext}(h) \rightarrow \text{ext}(g) \) making the following diagram commute.

\[
\begin{array}{ccc}
\Delta + Z & \xrightarrow{k} & \text{ext}(h) \\
| \phi + f \downarrow & & \downarrow{u} \\
\Gamma + Y & \xrightarrow{j} & \text{ext}(g)
\end{array}
\]

This mapping from \( \text{Der}\backslash\text{FPos} \)-morphisms to \( \text{Der} \)-morphisms gives the action of \( \text{ext} \) on morphisms, turning it into a functor. Since this functor arises from a universal construction, this defines the functor uniquely up to natural isomorphism.

The proof is straightforward: we have a morphism \( j \circ (\phi + f) : (\Delta + Z) \rightarrow (\Gamma + Y) \rightarrow \text{ext}(g) \). For any \( x \in \Delta \) and \( h(x) \in Z \), we have \( g(\phi(x)) = f(hx) \). Since \( j \) takes \( g \)-images to projection, we have \( j(f(hx)) \sqsupset j(\phi x) \), so \( j \circ (\phi + f) \) takes \( h \)-images to projection. Since \( \text{ext}(h) \) is universal with respect to this property, this induces the unique map \( u \).

We suggest a concrete construction of the functor. We first need the following lemma.

Claim 42. Let \( S \) be the Sierpinski derivation consisting of two points \( \{a, b\} \) such that \( a \sqsubset b \). A function \( f : S \rightarrow \Delta \) is a morphism if and only if \( fa \sqsubset fb \) is a projection relation.

It is straightforward that if \( f \) is a morphism, then \( fa \sqsubset fb \) must be a projection relation since morphisms must preserve projection. It is straightforward to check that for any pair \( x, y \in \Delta \) such that \( x \sqsubset y \) is a projection relation, the function sending \( a \) to \( x \) and \( b \) to \( y \) is in fact a morphism, as it follows directly from the definition of morphism. The following corollary is then immediate.

Claim 43. Let \( h : \Delta \rightarrow Z \) be any operation. For every \( x \in \Delta \), construct the disjoint union of \( \{x\} \) and \( \{hx\} \), and then take the disjoint union of all of these two-element sets. That is,
$X$ is the set consisting of $\Delta$, and for each $x \in \Delta$ a (distinct) element $hx$ representing its image. Construct the derivation $\Sigma$ as the sum of $|\Delta|$-many copies of $S$. That is, for each $x \in \Delta$, $S_x$ is a direct summand of $\Sigma$ isomorphic to $S$. We denote the minimal element of $S$ as $a$ and the other element as $b$, so that $a \subseteq b$. We construct a function $\alpha : X \to \Sigma$. For each $x \in X$ coming from $\Delta$, $\alpha$ sends $x$ to $b_x$, the largest element of $S_x$, and the corresponding element $hx \in X$ to $a_x$. Construct the function $\beta : X \to \Delta + Z$ sending $x \in X$ from $\Delta$ to $x \in \Delta$, and $hx$ to $hx \in Z$. $X$ can be turned into a derivation by taking the discrete topology, so that $\alpha$ and $\beta$ are automatically morphisms.

Consider the diagram below.

\[
X = \{x_1, hx_1, x_2, hx_2, \ldots\} \xrightarrow{\beta} \Delta + Z
\]

\[
\alpha \downarrow
\]

\[
\Sigma = S_{x_1} + S_{x_2} + \ldots
\]

It is useful to think of $S_{x_i}$ as the derivation drawn below.

\[
\begin{array}{c}
    hx_i \\
    \downarrow \\
    x_i
\end{array}
\]

Now consider any morphism $\gamma : \Delta + Z \to E$. If there is a morphism $\kappa : \Sigma \to E$ such that $\gamma \beta = \kappa \alpha$, it is uniquely determined, and it must be the map sending $hx_i$ and $x_i$ in $S_{x_i}$ to $\gamma(hx_i)$ and $\gamma(x_i)$, respectively. So the condition on $\gamma$ that there exists a $\kappa$ making the diagram commute is the same as the condition that $\gamma$ take $h$-images to projection. Consequently, the pushout of the diagram above is $\text{ext}(h)$. This pushout always exists since $\text{Der}$ is finitely cocomplete.

The functor $\text{pt}$, taking each derivation to its underlying partial order, preserves pushouts. We can then concretely construct the underlying partial order of $\text{ext}(h)$ as follows: take the partial order $\text{pt}(\Delta) + Z$ and add the relations $hx \leq x$ for each $x \in \Delta$ and close under transitivity. Since we only add relations from elements of $\Delta$ to elements of $Z$, no elements get identified, and the underlying set of $\text{ext}(h)$ is in correspondence with the elements of $|\Delta| + |P|$.

For an operation $h : \Delta \to Z$, we will often denote the extension as $\Delta^h$. We now show many ways in which this functor acts intuitively like an ‘extension of $\Delta$ along $h$’ on operations $h : \Delta \to Z$.

**Claim 44.** For any operation $f : \Delta \to Z$, consider the map to the extension $k : \Delta + Z \to \Delta^h$, or equivalently, two morphisms $i : \Delta \to \Delta^h$ and $j : Z \to \Delta^h$. We have the following properties:

(a) $i : \Delta \to \Delta^f$ is an open subderivation inclusion
(b) $j : Z \to \Delta^h$ is a subderivation inclusion, and $Z \to \Delta^h \to \mathbb{T}_{\Delta^h}$ is an isomorphism (call it $m$), where the second map is the canonical map from a derivation to its final derived object

(c) The composite $\Delta \to \Delta^h \to \mathbb{T}_{\Delta^h} \to Z$ is $h$, where the last map is the inverse $m^{-1}$.

A morphism $(\phi, h)$ between $f : \Delta \to Z$ and $g : \Delta \to Z'$ in $\text{Der}\setminus\text{FPos}$ is taken to a derivation morphism $\phi^h : \Delta^f \to \Gamma^g$ with the following properties:

(a) $\phi^h$ restricted to the subderivation $\Delta$ factors through $\Gamma \hookrightarrow \Gamma^g$ as $\phi$:

\[
\begin{array}{ccc}
\Delta^f & \xrightarrow{\phi^h} & \Gamma^g \\
\uparrow & & \uparrow \\
\Delta & \xrightarrow{\phi} & \Gamma
\end{array}
\]

(b) $\phi^h$ restricted to the subderivation $Z$ factors through $Z' \hookrightarrow \Gamma$ as $h$:

\[
\begin{array}{ccc}
\Delta^f & \xrightarrow{\phi^h} & \Gamma^g \\
\uparrow & & \uparrow \\
Z & \xrightarrow{h} & Z'
\end{array}
\]

(c) $\mathbb{T}_{\phi^h} : \mathbb{T}_{\Delta^f} \to \mathbb{T}_{\Gamma^g} \approx h : Z \to Z'$ are isomorphic morphisms of partial orders under the isomorphisms $\mathbb{T}_{\Delta^f} \approx Z$ and $\mathbb{T}_{\Gamma^f} \approx Z'$

The idea is that for each $f : \Delta \to Z$, we associate the extension of $\Delta$ by $f$, $\Delta^f$, which can be thought of as ‘$\Delta$ followed by $Z$ along $f$’, and for a morphism $(\phi, h) : (f : \Delta \to Z) \to (g : \Gamma \to Z')$, we have a morphism of ‘$\phi$ extended along $h$’ which acts like $\phi$ on the subderivations $\Delta$ and $\Gamma$ of the extensions, and acts like $h$ on the ‘new’ stage $h : Z \to Z'$.

Claim 45. Let $h : \Delta \to Z$ be an operation, and suppose that $Z$ is rooted.

1. If $\Delta$ is separated, then $\Delta^h$ is separated
2. If $\Delta$ is the sum of trees of terms (i.e. a forest of terms), then $\Delta^h$ is a tree of terms

Proof. SECTION APPENDIX?

We sketch one more concrete construction of $\Delta^h$, this time directly in terms of the structure of $\Delta$ and $Z$. We first need the concept of the basis of a space.

Definition 37. ([Munkres, 2000] §13) If $X$ is a set, a basis for a topology on $X$ is a collection $\mathcal{B}$ of subsets of $X$ (called basis elements) such that
(1) For each \( x \in X \), there is at least one basis element \( B \) containing \( x \).

(2) If \( x \) belongs to the intersection of two basis elements \( B_1 \) and \( B_2 \), then there is a basis element \( B_3 \) containing \( x \) such that \( B_3 \subset B_1 \cap B_2 \).

If \( B \) satisfies these two conditions, then we define the topology \( \mathcal{T} \) generated by \( B \) as follows: a subset \( U \) of \( X \) is said to be open in \( X \) (that is, to be an element of \( \mathcal{T} \)) if for each \( x \in U \), there is a basis element \( B \in B \) such that \( x \in B \) and \( B \subset U \). Each basis element is itself an element of \( \mathcal{T} \). By the definition of generate, every \( U \in \mathcal{T} \) is clearly a union of basis elements.

By the computations above, we know that the set underlying \( \Delta^h \) is essentially \(|\Delta| + |Z|\), and that the partial ordering on it is essentially the coproduct ordering, where we have added relations \( hx \leq x \) and closed under transitivity. We claim that there is a basis of \( \text{pt}(\Delta^h) \) of opens coming from \( \Delta \) and \( Z \).

**Claim 46.** Take any operation \( h : \Delta \to Z \) and its extension \( \Delta^h \). To every open subset \( U \subset \Delta \), we can associate an open subset \( U \subset \Delta^h \), simply because \( \Delta \subset \Delta^h \). To every open subset \( P \subset Z \), we can associate an open subset \( \{x \in \Delta^h \mid \exists p \in P, p \leq x \text{ in } \Delta^h\} = (P) \subset \Delta^h \), i.e. its upward closure in \( \text{pt}(\Delta^h) \). We have the following:

1. The collection
   \[
   B = \begin{cases} 
   U, & \text{U an open subset of } \Delta, \text{ or} \\
   (P), & \text{P is an open subset of } Z
   \end{cases}
   \]  (1)
   is a basis for the topology on \( \text{pt}(\Delta^h) \).

2. For an open subset of the form \( (P) \subset \Delta^h \), we have \( (P) \cap \Delta = \Delta^{h^{-1}}P \) as subsets of \( \Delta^h \).

3. For open subsets of the form \( (P) \) and \( (Q) \), we have \( (P \cap Q) = (P) \cap (Q) \) and \( (P \cup Q) = (P) \cup (Q) \). The same clearly holds for open sets from \( \Delta \), since \( \Delta \) is itself an open subset.

These follow directly from the construction of the ordering on \( \text{pt}(\Delta^h) \).

We then only have to equip \( \text{pt}(\Delta^h) \) with a local topology structure to obtain \( \Delta^h \). This is quite straightforward in terms of this basis. We define \( \omega^h \) on \( B \) as follows:

1. For \( U \) from \( \Delta \), we define \( \omega^h(U) = \omega U \)

2. For \( (P) \) from \( Z \), we define \( \omega^h((P)) = \{(Q) \mid Q \subset P \text{ is an open subset in } Z\} \)

Every such sheaf of finite distributive lattices on a basis of a finite space extends uniquely to a sheaf on the whole space by taking matching families. That is, for any open subset \( V \subset \Delta^h \), we can obtain \( V \) as a union of basis elements, and \( \omega^h(V) \) is just the union of matching families on that cover.\(^{11}\)

\(^{11}\)CITE typical sheaf resources
Claim 47. \( \text{pt}(\Delta^h) \) with the local topology defined above is isomorphic to the extension \( \Delta^h \).

Proof. SECTION APPENDIX

Grammars. With the results of the previous sections, it becomes straightforward to define a grammar.

We define a grammar as a set of lexical items \( \text{LEX} \) together with a set of rules \( \text{RULES} \). The elements of \( \text{LEX} \) are finite partial orders, while the elements of \( \text{RULES} \) are rules \( G : C \to \text{Set} \) where \( C \subset \text{Der}^n \).

We will usually be interested in grammars of a much more restricted form. For example, we might require that the elements of \( \text{LEX} \) are rooted partial orders, or even trees (possible even of restricted depth). We will usually require that each rule is binary (that is, \( C \subset \text{Der}^2 \) for each rule) and also generated by a single operation. Furthermore, we will usually only be interested in inclusive and weakly extensive rules where each operation produces a rooted partial order.

We can associate to each grammar \( G \) a language \( L_G \).

Definition 38. If \( G \) is a grammar, then we define a language \( L_G \) recursively as follows.

1. If \( X \in \text{LEX} \), then \( X \in L_G \)

2. Let \( G : C \to \text{Set} \) be a rule on \( C \subset \text{Der}^n \) in \( \text{RULES} \). If \( (\Delta_i) \in C \), \( h : \bigsqcup \Delta_i \to Z \in G(\Delta_i) \), and \( \Delta_i \in L_G \), then \( (\bigsqcup \Delta_i)^h \in L_G \)

We consider \( L_G \) a category by taking it to be a full subcategory of \( \text{Der} \). That is, the \( L_G \)-maps between a pair of objects \( \Delta \) and \( \Gamma \) is just the set of \( \text{Der} \)-maps between them.

Note that by Claim 45, if every item in \( \text{LEX} \) is rooted, and every operation produces a rooted object, then every object in \( L_G \) will be a tree of terms.

3.2. Equivalence of grammars

One advantage of having a description of derivations as structured objects is that we can define ‘extensional’ equivalence of grammars based on that structure.

Definition 39. If \( C, D \) are full subcategories of \( E \) with inclusions \( i, j \), we say that they are equivalent if there is a pair of functors \( F : C \to D \), \( G : D \to C \) such that \( jF \approx i \) and \( iG \approx j \) are naturally isomorphic.

If \( C \) and \( D \) are equivalent subcategories of \( E \), then for any pair of objects of \( C \), the \( C \)-morphisms between them are in bijection with the \( D \)-morphisms between them under \( F \), and conversely for objects of \( D \). Additionally, for each object of \( C \), there is an object of \( D \)

65
which is $E$-isomorphic to it, and conversely. Furthermore, $Fc \approx c$ for any $c \in C$, where the
isomorphism is taken in $E$, and conversely for $G$ and $D$.

We say that two grammars $G$, $H$ are extensionally equivalent if $L_G$ and $L_H$ are equivalent
as full subcategories of $\text{Der}$. Then, for every derivation which can be produced by $G$, an
isomorphic derivation can be produced by $H$ and vice-versa.

For example, choice of representation $\int G \to \text{Der}\setminus\text{FPos}$ of each rule does not change
a grammar up to equivalence. Similarly, if we add items to $\text{LEX}$ which are isomorphic to
existing ones, the category of derivations generated by the augmented $\text{LEX}$ is equivalent to
the smaller category of derivations.

It is straightforward to strengthen equivalence to isomorphism: we simply require that $FG = 1_D$ and $1_C = GF$. This will force the correspondence to be bijective.

Clearly, neither equivalence of grammars nor isomorphism makes reference to how the
derivations were constructed. This definition may still be used to the same effect even if we
restrict the recursive generation of derivations in more nuanced ways. In fact, it does not
even rely on the fact that the categories of interest are derivations; if we modify the base
category to include data like feature type or chains, we may still use the same definition.

### 3.3. Grammatical relations

The previous section views grammatical operations as a kind of ‘gluing together’ of spaces
$Y_i$ into a new space $Z$ along an order-preserving map $f : \bigsqcup Y_i \to Z$. We will view a
grammatical relation introduced by the operation $f$ as a measurement of how much the $Y_i$
come to overlap in $Z$ along $f$.

While the following definitions will make sense for any derivation, they have particularly
salient interpretations for trees of terms.

**Lexical items; max and min projections.** We first identify lexical items relativistically,
as done in C&T, [Muysken, 1982].

**Definition 40.** A *head* or *lexical item* in a derivation is a maximal term $t$, in that for
any term $s$, if $t \leq s$, then $t = s$.

**Definition 41.** A *lexical feature* of a derivation $\Delta$ is a point $x$ which is close to a head
$t \preceq x$.

**Definition 42.** A derivation homomorphism $\phi : \Delta \to \Gamma$ is *lexical-item preserving* if
it takes heads to heads. We are usually interested in lexical-item preserving morphisms
which are also *coherent* and *term-preserving*, and will assume all lexical-item preserving
morphisms have these properties unless stated otherwise.

Fix a derivation $\Delta$. For any lexical item $t$, we have an associated constituent $U_t$, whose
space $T_{U_t}$ represents the lexical item $t$ and its features at the moment of insertion into
the derivation. For any open $U$ such that $t \in U$, we have an inclusion $U_t \subset U$ and hence
an associated morphism $\top_{U_t} \to \top_{U}$, mapping the head $t$ and its features into the derived object at $U$.

**Definition 43.** Let $\Delta$ be any derivation, $t$ be a lexical item in $\Delta$, and let $U \in \mathcal{O}(\Delta)$ be any open such that $t \in U$. Recall that for any point $x \in \Delta$, we write the corresponding prime filter $P_x = \{ U \in \mathcal{O}(\Delta) \mid x \in U \}$.

We refer to $U_t$ as the **minimal projection** of $t$. We write $\top_{U_t}$ as $\min(t)$. We refer to the minimal element of $\omega U \cap P_t$ as the **maximal projection of $t$ at $U$**, and we write its associated derived SO as $\max(t)$, when $U$ is clear from context. For each such item $t \in U$, there is an associated composite of maps $\min(t) \to \max(t) \to \top_{U}$.

**Derivational overlaps.** Like in X-bar theory, we will be interested in ways in which the phrasal and head projections of two lexical items interact, which will give information about their grammatical dependencies, such as being a complement, specifier, or adjunct. Recall that in the represented space $\top_{U}$, we think of its points as representing heads and features (possibly fusions of them) coming from the lexical items, such that the labels are ordered by m-command, or, loosely, ‘domain’. An immediate dependent in this space can be thought of as being ‘governed’ or in the ‘minimal domain’ of another point.

**Definition 44.** For points $x, y \in X$ a finite partial order, we say that $y$ **immediately dominates/contains** $x$ if there is no $z \neq x, y$ such that $x > z > y$.

**Definition 45.** A **pullback** in any category $\mathcal{C}$ of two arrows $f : A \to X$ and $g : B \to X$, if it exists, is an object $P$ together with arrows $a : P \to A$ and $b : P \to B$ such that $fa = gb$ which is universal with respect to this property. That is, if $r : Q \to A$ and $s : Q \to B$ are any two morphisms such that $fr = gs$, then there is a unique morphism $u : Q \to P$ such that $au = r$ and $bu = s$. As with any universal construction, if the pullback exists, it is unique up to unique isomorphism in $\mathcal{C}$.

Pullbacks generalize intersection and preimages. For example, in $\bf{Set}$, if $f : A \to X$ and $g : B \to X$ are set-inclusions, then the intersection $A \cap B$ together with the two inclusions $a : A \cap B \to A$ and $b : A \cap B \to B$ is a pullback. If $f : Y \to X$ is any set-function and $g : S \to X$ is a subset inclusion, then $f^{-1}S = \{ x \in Y \mid f(x) \in S \}$ is a pullback, together with the subset inclusion $f^{-1}S \to Y$ and function $f_{f^{-1}S} : f^{-1}S \to S$ which is just $f$ restricted to $f^{-1}S$.

**Claim 48.** Let $f : A \to X$ and $g : B \to X$ be two order-preserving functions in $\bf{FPos}$. These have a pullback and it can be computed as follows. We define $A \times_X B$ to be the set $\{ (a, b) \in A \times B \mid f(a) = g(b) \}$. We give $A \times B$ the product ordering where $(a, b) \leq (a', b')$ iff $a \leq a'$ and $b \leq b'$, and we give $A \times_X B$ the subspace ordering. The functions $p_A : A \times_X B \to A$ and $p_B : A \times_X B \to B$ taking $(a, b)$ to $a$ and $b$ respectively are order-preserving, and these turn $A \times_X B$ into a pullback.
The pullback of two functions \( f \) and \( g \) gives us information about how much they overlap in their codomain.

Consider the ‘classical’ case of a tree \( T \) with constituents \( K \) and \( C \). Intersecting \( K \) and \( C \) gives one of three results: \( K, C, \) or \( \emptyset \). \( K \) results exactly when \( K \subset C \) is a subconstituent, and conversely \( C \) when \( C \subset K \). \( \emptyset \) results when there is no dependency between the two phrases. We generalize this analysis to pullbacks.

Let \( \Delta \) be any derivation, \( U \) any open subset, and \( x, y \in U \) any two lexical items. For each lexical item, we will have a series of maps \( \min(x) \to \max(x) \to \top_U \). The first map gives the data which shows how \( x \) projects from the lexical item to the ‘phrasal’ projection, where the second map takes its phrasal projection into the object \( \top_U \).

We then take the following square of pullbacks in \textbf{FPpos}.

\[
\begin{array}{ccc}
\min(x) \times \top_U & \min(y) & \longrightarrow & \max(x) \times \top_U & \min(y) & \longrightarrow & \min(y) \\
\downarrow & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\min(x) \times \top_U & \max(y) & \longrightarrow & \max(x) \times \top_U & \max(y) & \longrightarrow & \max(y) \\
\downarrow & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\min(x) & \longrightarrow & \max(x) & \longrightarrow & \top_U
\end{array}
\]

Figure 22: The 2-by-2 pullback comparison of the head-level and phrasal-level constituents associated to points \( x \) and \( y \) at an open \( U \in \mathcal{O}(\Delta) \) containing them, considered as pullback diagrams in \textbf{FPpos}.

The universal property of pullbacks will induce unique maps at all dotted arrows making the diagram in Fig. 22 commute.

In ‘classical’ cases, \( \max(x) \) and \( \max(y) \) are simply open subsets of \( U \), so their pullback is just their intersection. In fact, let us take the case where \( x \) and \( y \) are lexical items and \( x_U < y_U \) is an immediate domination relation. Consider the pullback of \( \max(x) \to \top_U \) and \( \max(y) \to \top_U \). Since \( x_U \leq y_U \), we have a map \( \max(y) \to \max(x) \) which composed with \( \max(x) \to \top_U \) is just \( \max(y) \to \top_U \). Then, the pullback must ‘contain’ all of \( \max(y) \) since the image of \( \max(y) \) is contained in the image of \( \max(x) \), akin to the classical result saying that the \( y \)-phrase is contained in the \( x \)-phrase.

Similarly, when \( x_U < y_U \), we know that there is a map \( \min(y) \to \max(y) \to \max(x) \) which composed with \( \max(x) \to \top_U \) is just \( \min(y) \to \top_U \). The pullback of \( \min(y) \to \top_U \) and \( \max(x) \to \top_U \) will again ‘contain’ \( \min(y) \) since the image of \( \min(y) \) is contained in the image of \( \max(x) \). This is again akin to the classical result, simply saying that the head \( y \) maps into the \( x \)-phrase at \( U \).

\footnote{In that for every element of \( \max(y) \), there is some element of the pullback projecting to it. When \( \max(x) \to \top_U \) is injective, so is its pullback.}
The pullback of \( \min(x) \) and \( \min(y) \) contains one point \((a, b)\) for each pair of lexical features \( x \preceq a \) and \( y \preceq b \) which become equal in \( U \). So this pullback is nonempty only if two features of the heads have identified, which we view as syntactic selection. In our example when \( x_U < y_U \) is immediate domination, this will imply that features of the heads \( x \) and \( y \) have identified and now reside somewhere in \( y_U \), indicating the selection of \( y \). We view this as saying that the \( y \)-phrase was somehow selected by \( x \), such as with complements, and perhaps some or all ‘selected’ specifiers.

Using this method, we will have to distinguish complements and ‘selected specifiers’ in some other way, as they will fall into the same class of ‘selected argument’ derivationally.\(^{13}\)

The pullback of \( \min(x) \) and \( \max(y) \) will be nonempty only if there is some element \( f \in \min(x) \) in the head \( x \) which maps into \( \max(y) \) (or more precisely, into the image of \( \max(y) \) in \( \top_U \)). This is akin to saying that the \( y \)-phrase at \( U \) is dependent on a feature from the head \( x \). In our case when \( x_U < y_U \) is an immediate domination relation, when the

\(^{13}\)This is often assumed in other formalizations of minimalist syntax, e.g. [Stabler, 1996]. The main issue arises if a head takes a specifier but no complement, since this specifier will look like a ‘first-merged selected argument’. In [Stabler, 1996], the distinction is made by linearization. [Kobele, 2006] initially says, ‘We call the expression first merged with a lexical item its complement, and later merged expressions are specifiers,’ but then this never allows specifiers of heads with no complements. [Kobele, 2006] later separates PF representations into three coordinates which simply assert the positions - (SPEC,HEAD,COMP), which might be able to be modified to distinguish them.
y-phrase is unselected, we think of this as a case of the y-phrase somehow being licensed by a feature of the head x or depending on it in some other way like for agreement features. We think of these as ‘unselected specifiers’. For example, we may have a C-head with a +wh feature and a n-phrase with a –wh feature. We may move the n-phrase to adjoin to the C-phrase, adding a relation C < n, while the -wh feature must be licensed by the +wh feature, depending on it.

Figure 24: A noun phrase adjoining to a C-phrase, where a feature of the C-head ends up in the domain of the n-phrase. min(C) → ⊤U and max(n) → ⊤U have nonempty pullback: it consists of the feature +wh along with the pair of maps sending it into the head C and the n-phrase at U.

It will also, however, be useful to represent certain agreement relationships this way. One such case might be when an adjective agrees with the φ-features of the head noun it modifies. This will indicate that the value of the φ-feature of the adjective is dependent on the φ-feature of the n-head.
This will classify certain adjuncts - those which undergo agreement with the head of the phrase they modify - as specifiers. We might choose to distinguish the last two cases based on the feature responsible for the overlap (a wh feature versus a $\phi$ feature), but for now we characterize these as structurally similar. This is not totally surprising given the typical characterization of constituents undergoing agreement in a specifier position. Such treatment of agreement is compatible with a feature-sharing view of agreement, and the overlap can be viewed as indicating the feature-sharing dependency. Discussion of both of configurations is discussed in [Danon, 2011], and the second in [Pesetsky and Torrego, 2007].

Any function with nonempty domain must have nonempty codomain, so nonemptiness of one set implies nonemptiness of all sets which it has maps into. For example, if $\min(x) \times_U \min(y)$ is nonempty, then all of the pullbacks will be nonempty. Take the fragment of a derivation given in Fig. 23. The pullback of $\min(V) \rightarrow \top_U$ and $\max(n) \rightarrow \top_U$ is nonempty since the selectional feature -$N$ enters the domain of the $n$-phrase as $N$. We are then interested in the whole collection of pullbacks which are nonempty, not simply whether a particular one is nonempty.

There are then 6 degrees of ‘connectedness’ two lexical items may have in $U$: all nonempty, all but $\min(x) \times_U \min(y)$ nonempty, $\min(x) \times_U \min(y)$ and $\min(x) \times_U \max(y)$ empty but other two not, $\min(x) \times_U \min(y)$ and $\max(x) \times_U \min(y)$ empty but other two not, only $\max(x) \times_U \max(y)$ nonempty, and all empty. In other words, we view the top
left square of Fig. 22 as a partial order of implications, and the degrees of connectedness correspond to the open sets of this partial order. We give these open sets suggestive names.

![Diagram](image)

Figure 26: The 6 upsets of the ‘lattice of impliciation of nonemptiness of the 2-by-2 pullback diagram of the head and phrasal projections of two points’. We give half of them names corresponding to its meaning in the case of one being in the minimal domain of the other, and label the other three as their formal opposite.

**Classical relations.** We specify to the case when the phrasal projection of $x$ at $U$ immediately dominates the phrasal projection at $y$; i.e., $x_U < y_U$ is an immediate domination relation in $\mathcal{T}_U$. We have already showed that this implies that the pullback $\max(x) \times_U \min(y)$ must be nonempty, and hence $y$ must be at least adjunct connected to $x$. Specifically, $y$ will be Adjunct-connected to $x$ if there is no lexical feature of $x$ which is dominated by $y$ at $U$; that is, if $\min(x) \times_U \max(y)$ is empty (and hence so is $\min(x) \times_U \min(y)$).

$y$ will be US-connected to $x$ if there is a lexical feature of $x$ which is dominated by $y_U$, the projection of $y$ at $U$, but no lexical feature of $x$ and $y$ have identified. This more appropriately might describe an ‘unselected specifier’, such as [Spec, CP], [Spec, TopP], or [Spec,TP] as opposed to, say, [Spec, vP]. We view any features in the overlap as the ‘licensing’ features of the $x$P, e.g. an EPP or wh feature, or possibly an agreement feature.

$y$ will be SA-connected to $x$ if there are lexical features of $x$ and $y$ which have identified at $U$. That is, there are features $x \leq a$ and $y \leq b$ such that $a_U = b_U$. We view this as indicating selection.

We give these formal definitions.

**Definition 46.** Let $\Delta$ be any derivation and $x$ and $y$ two lexical items such that $x$ is in the minimal domain of $y$ at $U \in \mathcal{O}(\Delta)$.

We say that $x$ is an **argument** of $y$ at $U$ if $x$ is SA-connected to $y$ at $U$.

We say that $x$ is an **unselected specifier** of $y$ at $U$ if $x$ is US-connected to $y$ at $U$. 

72
We say that \( x \) is an **adjunct** of \( y \) at \( U \) if \( x \) is Adjunct-connected to \( y \) at \( U \).

**Remaining formal overlap relations**

We have three remaining possible relationships between two lexical items and their projections at \( U \). These relations do not hold between \( x \) and \( y \) when \( y \) is in the minimal domain of \( x \) at \( U \), but only in more general situations.

Disconnected has the obvious meaning, where two lexical items are disconnected at \( U \) if neither is a subconstituent of the other, and they are in no agreement or selection relationship (and neither are their subconstituents). That is, they are totally disjoint. - Adjunct is simply the Adjunct relationship viewed from the perspective of the adjoined element.

The only ‘new’ relation is -Spec. This relation may appear when there is a feature common to both projections, but which doesn’t originate in the head of either. Such might be the case of multiple specification, as in the case of multiple wh-movement, where two phrases are not dependent on each other, in that the neither of the heads projecting the phrases have any relationship, but the phrases are licensed by a ‘common’ feature, originating outside of the heads of either phrase. They are not totally disconnected, as they share this licensing feature, but neither phrase depends on a feature from the head of the other.

![Diagram](C
\[\begin{array}{c}
C \\
| \\
n \\
| \\
-\text{wh} \\
| \\
+\text{wh}
\end{array}\n\begin{array}{c}
| \\
n \\
| \\
-\text{wh}
\]
)

Figure 27: Suppose that there are two \( n < -\text{wh} \) items which both attach to \( C \) whose -\text{wh} features are both dependent on \( C \)'s +\text{wh} feature. The two are -Spec-connected, since their phrases overlap, but there is no feature of either \( n \)-head ending up in the phrasal projection of the other.

This has not yet dealt with how to obtain these configurations. We will now look at how flavors of **merge** and **agree** arise as universally optimal solutions to obtain connectivity of these various types.
3.4. Toy grammatical rules

We first summarize some basic operations on finite partial orders.

**Adjuncts.** We have seen this case already. Adjunction can be seen as the operation which attaches the root of one tree to the root of another.

![Diagram of Adjunction](image)

Figure 28: Adjoining $Y$ to $X$

**Selected Arguments.** Now suppose further that we would like to have features of $X$ or $Y$ check with features of the other. For no mathematical reason other than symmetry and simplicity, let’s assume that the features originally in the minimal domain of $x_r$ are $X$’s features and similarly for $Y$, and further assume that all checking/agreement is done between features, and never between a ‘feature’ element and these ‘root’ (label) elements.

Suppose that $x_n$ is a selection feature, and it selects the feature $y_1$, and that selection is satisfied only through identification of elements.

Construct the monotone function $\beta$ on the following subspace of $X + Y$:

![Diagram of Monotone Function](image)
The colimit of this diagram is the following tree, where we have identified the features $x_n = y_1$ in the simplest way possible.

Figure 29: A universal construction representing feature-checking

Calling this new stage $Z'$, we see that the maps $X \rightarrow Z' \leftarrow Y$ describe $Y$ as the complement of $X$ at $Z'$. That is, the point $x_r$ is associated with minimal projection $X$, and similarly $y_r$ with $Y$, and the image of $y_r$ in $Z'$ is in the minimal domain of the image of $x_r$. Furthermore, the pullback of $X \rightarrow Z' \leftarrow Y$ is nonempty, consisting of a single element with projections picking out the feature $x_n$ of $X$ and $y_1$ of $Y$ (which can be thought of as the singleton set containing the ordered pair $\langle x_n, y_1 \rangle$), indicating that this is the feature pair they are head-to-head connected over.

**Unselected Specifiers.** Identifying features for checking is not the only option. We construct a feature-analogue of adjunction, where a feature $x_n$ goes into the domain of a feature $y_1$ to license it. For example, $x_n$ might be an EPP feature attaching to a nominal category feature; or a case feature which a nominal case feature depends on; or a $\phi$ feature which the $\phi$ feature of a modifier depends on. We could take our ‘adjoin $Y$ to $X$’ operation and additionally find the best possible map which orders the features as such.
This colimit is the tree below, which again we call $Z'$.

![Diagram](image)

If we check the grammatical relation between $X$ and $Y$ over $X \rightarrow Z' \leftarrow Y$, we find that $Y$ is an unselected specifier of $X$. More precisely in $(X + Y)'$, the lexical feature $x_r$ and the lexical feature $y_r$ are such that at $Z'$, the point $y_r$ is in the minimal domain of (immediately ordered above) $x_r$, and while the pullback of $X$ and $Y$ under $Z'$ in $\mathbf{FPos}$ is empty, the pullback of $\max(y_r)$ with $X$ in $Z'$ is nonempty (they overlap on the agreement feature $x_n$), and the pullback of $\max(x_r)$ with $Y$ in $Z'$ is also nonempty (they overlap on all of $Y$). So, $X$ is head-to-phrase and phrase-to-head related to $Y$, but not head-to-head related to it, while being immediately dominated by it (at $Z'$). Hence, $Y$ is an unselected specifier of $X$, corresponding to the fact that it has adjoined and been licensed.

From operations to rules. We turn the above local, non-movement operations into rules by describing their contexts. In each of the cases above, we relied on being able to specify roots, projections of lexical items, and projections of their features.

Definition 47. We say that $\Delta$ is rooted if $\top_\Delta$ is rooted. We say that $\phi : \Delta \rightarrow \Gamma$ between rooted derivations is root-preserving if $\top_\phi$ is root-preserving.
Definition 48. Let \( C \) be the category of pairs of rooted derivations with pairs of injective lexical item and root preserving morphisms:

\[
\text{ARGUMENT-SELECT} : C \to \text{Set} \quad \text{is the rule generated by the object } (S, S) \in C \quad \text{and the operation:}
\]

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
x_r \\
+ \\
x_n \\
y_1
\end{array}
\end{array}
\end{array}
\end{array} 
\to 
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
x_r \\
y_r \\
x_n = y_1
\end{array}
\end{array}
\end{array}
\]

\[
\text{SPECIFIER-MERGE} : C \to \text{Set} \quad \text{is the rule generated by the object } (S, S) \in C \quad \text{and the operation:}
\]

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
x_r \\
+ \\
x_n \\
y_1
\end{array}
\end{array}
\end{array}
\end{array} 
\to 
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
x_r \\
y_r \\
y_1 \\
x_n
\end{array}
\end{array}
\end{array}
\]

\[
\text{ADJOIN} : C \to \text{Set} \quad \text{is the rule generated by the object } (1, 1) \in C \quad \text{and the operation:}
\]

\[
\begin{array}{c}
\begin{array}{c}
x_r + y_r
\end{array}
\end{array} 
\to 
\begin{array}{c}
\begin{array}{c}
x_r \\
y_r
\end{array}
\end{array}
\]

These rules serve to create the correct kinds of dependencies between phrases. However, in practice, they must be restricted further. All of our features are ‘untyped’ (i.e. are not associated with syntactic category or feature type). We may want to further endow derivations with designations of type for each feature (e.g. nominal, verbal, \( \phi \), ...) and restrict rules to matching features of appropriate type. Similarly, we haven’t marked features as ‘checked’ or ‘requiring checking/active’. This is straightforward to do, but we leave it to \textsc{section appendix}, noting that this abstract level it is the dependencies which are important to syntactic structure. Similar assumptions are made by, e.g. \cite{Keenan and Stabler, 2003}.

Secondly, we have not imposed any restrictions on ‘order-of operations’. It is usually assumed that complement selection must precede all other operations \textit{if} it is to take place at all (thought needn’t in general). The other operations may take place in any order in
principle. Standard minimalist Bare Phrase Structure models simply impose this by stip-
ulation. Formal models like Stabler’s Minimalist Grammars do not have this as a general
principle, but rather hardcode it by serializing features in a particular way for each lexical
item of the language.

**Local and long-distance agreement.** The recursive method for building new derivations
by extending them along optimal - i.e. universal - ‘solutions’ to various feature configuration
‘problems’ given above captured the simplest aspects of Chomskyan grammar. However,
it has not yet touched on movement or long-distance agreement.

One way to capture long-distance agreement is fairly simple: we simply remove a locality
requirement on feature-agreement. As stated, feature agreement may only take place be-
tween features of the lexical items projecting the roots of the current objects. If we simply
loosen this constraint, such that when merging $\Delta$ and $\Gamma$ with roots $d \in T_\Delta$ and $g \in T_\Gamma$,
we may want to ‘adjoin’ $g$ such that $d < g$, but additionally, we may take a feature of the
head $d$, call it $j$, and have it agree with *any* feature $g < k$, where $k$ is a projection of a
non-root lexical feature. That is, when merging $\Delta$ and $\Gamma$, we find the universal map $f$
out of $T_\Delta + T_\Gamma$ such that $fd < fg$ and $fj < fk$. Long-distance agreement is then literally
just the same operation as agreement, with the locality constraint removed such that the
agreement can be done in situ. In pictures:
This feature $k$ may come from the lexical item projecting $g$, or it may come from a lexical item projecting to an element in the $g$-phrase, but not $g$ itself. Of course, like all syntactic operations, even if long-distance agreement is allowed, it should be subject to locality conditions. To describe these situations, we would have to develop a theory of contexts more general than the ones presented so far. That is, we may want to allow more than one feature to be manipulated in a single instance of \textsc{merge}, and various long-distance feature operations. Since we have developed the technology for building derivations very generally, this poses no problem, and behaves analogously to the simpler cases.

\textbf{Minimality.} One of the most common realizations of locality is minimality, with the ‘metric’ that if $j$ is a feature probing for another feature, and $k$ and $l$ are both c-commanded by $j$ and are potential targets of $j$, then $k$ is more local to $j$ than $l$ if $k$ asymmetrically c-commands $l$.

We now have many ways to interpret this. If $\Delta$ is a tree of terms, then the equivalence class of $j$ under the term-quotient will correspond to the term containing $j$, and similarly
for $k$ and $l$, so that we may compute c-command as usual. This will coincide with the typical notion of minimality in BPS-style trees. This notion of c-command is then going to align with a view of ‘derivational’ c-command, as it describes c-command across steps of the derivation. A weaker form of minimality is that a constituent $X$ asymmetrically c-commanded by an object $Y$ of the same ‘type’ (i.e., up to relativization) cannot move to a position which is not c-commanded by $Y$.

However, we can also look at ‘c-command’ within a single stage $X \in O(\Delta)$. Suppose that when a derivation $\Delta$ is merging with a derivation $\Gamma$, there is a probe in $T_\Delta$ looking for some feature to agree with (or, similarly, some feature in $T_\Delta$ looking within $T_\Delta$ itself for a feature to drive movement). ‘C-command’ within this tree will actually correspond to something roughly like c-command modulo projection - i.e. up to minimal domains.

We have not yet given a theory of copying/movement, but we can set this aside and just address the phrase-structural aspects. How stages inside of a derivation already capture ‘minimality up to minimal domains’ can be seen obviously from an example.

**Example.** Suppose that we have the following lexical items: a verb with space $\{o > V\}$, a light verb head with space $\{k, z, a > v\}$, and two nominals $\{N > O\}$ and $\{M > A\}$. We first merge $\{N > O\}$ with $\{o > V\}$, adjoining $\{N > O\}$ to $\{o, b > V\}$ and identifying $o$ and $N$, so that this object nominal is merged as a complement of the verb, forming the tree:

\[
\begin{array}{c}
V \\
\downarrow \\
O \\
\downarrow \\
\bar{o}
\end{array}
\]

Here $\bar{o}$ represents the image of $N$ and $o$. We then merge this verbal complex with the light verb, identifying the category feature $b$ with the selectional feature $z$ of the light verb, to form the tree:

\[
\begin{array}{c}
v \\
\downarrow \\
k \\
\downarrow \\
a \\
\downarrow \\
V \\
\downarrow \\
b \\
\downarrow \\
O \\
\downarrow \\
\bar{o}
\end{array}
\]

Finally, we merge the subject as a specifier, adjoining it to the light verb and letting the ‘EPP’ feature $a$ target the category feature $M$. 

80
Up to this point, the space of points, partitioned into terms, can be informally represented as the following tree of terms (or, similarly, the tree can be thought of as the stages which are terms, with arrows being arrows from $O(\Delta)$).
Figure 32: The space of points, clustered into terms, with arrows from $\mathcal{O}(\Delta)$

Figure 33: Quotient of the space of points into terms
Now suppose that $k$ is a ‘secondary’ specifier feature for accusative case marking, such as in C&T §4.10.1. In Chomsky’s analysis, we require that the object move to the (secondary) specifier position of $v$ to check its accusative case. Spatially, we must adjoin $O$ to $v$ to let the $k$ feature enter the domain of $O$’s $N$ feature (projected now to $\bar{o}$). But naively, this looks like a minimality violation. We would have to move the object $O$ - which is asymmetrically c-commanded by the subject $A$ - to a position not c-commanded by $A$. In terms of ‘global/derivational’ c-command, this will be true. However, no matter how copying/movement is realized, within the space $\top_\Delta$, adjoining $O$ to $v$ will move it up to a position c-commanding $A$, but which is still c-commanded by $A$. That is, we can copy $O$ (potentially in some derivational sense), and adjoin it to $v$, resulting in the following new stage.

\[
\begin{array}{ccc}
  & v & \\
  O_2 & A & V \\
  & \bar{o}_2 & M \quad b \quad \overline{O_1} \\
  k & \overline{o_1} & \\
\end{array}
\]

It is not resolved ‘where’ we copied the object from, but we can set this aside for now.\textsuperscript{14} No matter the form, the copy of $O$ is in a position mutually c-commanding $A$, but it does not move to a position outside the c-command domain of $A$. In this sense, the spatial structure of the derivation has already encoded the notion of ‘minimal domain’ and ‘equidistance’.

### 3.5. Compilations of rules

In this section, we explore an analytic benefit of having structured rules. We will study compiling a sequence of $n$-ary rules. For example, we may have an operation which merges a specifier but also agrees with it/checks some licensing configuration. We would like to see this single binary operation as a ‘compilation’ of two others - elementary MERGE additionally with an AGREE operation. Similarly, COMPLEMENT-MERGE might be a compilation of MERGE along with selection, while adjunction is just MERGE. This is similar, but not identical, to the theory developed in [Hunter, 2015].

The tools presented here are mainly presented for analytical purposes - to make sense of

\textsuperscript{14}If we copy after its merger as complement to $V$, it will carry over the fact that $o$ and $N$ have identified, i.e. it will carry over its status as complement. We could also just copy the ‘raw’ lexical item, prior to this identification. In the case of carrying over the ‘checked’ version, we could also copy the whole chunk of the derivation up to this projection of the element $O$. These issues are not often raised in Chomskyan syntax, but exist there as well. That is, if features change/become valued/etc throughout the derivation, when the copying is done will affect exactly the contents of the copy, of course bringing with it issues of identity between copies, etc.
how certain operations can be seen as composed of more primitive ones. However, we do not incorporate this technology into the grammar, we just analyze simple cases where we can see certain rules $G$ as arising from ‘compiled’ rules.

3.5.1. Categories of sequences

We first need to connect up the conditions of $m$ rules we wish to compile. Fix some finite sequence $S = (S, S', \ldots, S^{(m)})$ of subcategories of $\text{Der}^n$. These are to represent the coditions for $m$ many $n$-ary rules. We define a sequence of $S$ to be an $m$-tuple $(A, A', \ldots, A^{(m)})$ with $A^{(i)} \in S^{(i)}$, considered with $(n$-tuples of) derivation homomorphisms $l^{(i)} : A^{(i)} \to A^{(i-1)}$ for all $m \geq i \geq 1$. We can write a sequence as $A^{(m)} \xrightarrow{l^{(m)}} \cdots \xrightarrow{l'} A$.

It is to be thought of as a generalized parameter, in that we will use it to combine rules $r^{(i)} : A^{(i)} \to B^{(i)}$, taking the output of $r^{(i)}$, and using $A^{(i+1)}$ to select parameters in it with $r^{(i)}l^{(i+1)} : A^{(i+1)} \to A^{(i)} \to B^{(i)}$.

A morphism of sequences of $S$ is an $m$-tuple of $(n$-tuples of) derivation homomorphisms with $\phi^{(i)} : \Delta \to \Gamma$ in $S^{(i)}$, creating a commutative diagram:

$$
\begin{array}{cccc}
A^{(m)} & \xrightarrow{\phi^{(m)}} & X^{(m)} \\
\downarrow l^{(m)} & & \downarrow k^{(m)} \\
\cdots & & \cdots \\
\downarrow l' & & \downarrow k' \\
A' & \xrightarrow{\phi'} & X' \\
\downarrow & & \downarrow \\
A & \xrightarrow{\phi} & X
\end{array}
$$

We write the category of $S$-sequences as $S^*$. Each $S^{(i)}$ is by definition a subcategory of $\text{Der}^n$. We can take the pullback of all these subcategories, writing it $\bigcap S = C$. An object in this category is an $n$-tuple $(\Delta_1, \ldots, \Delta_n)$ which is in each $S^{(i)}$. A morphism in this category is an $n$-tuple of derivation homomorphisms $\langle \phi_j : \Delta_j \to \Gamma_j \rangle : (\Delta_1, \ldots, \Delta_n) \to (\Gamma_1, \ldots, \Gamma_n)$, i.e. a morphism in $\text{Der}^n$, such that this $\text{Der}^n$-morphism is in each $S^{(i)}$. $C \subset \text{Der}^n$ can be thought of as the subcategory of $n$-tuples which are in $S^{(i)}$ for all $i$ and preserve $S^{(i)}$-properties for all $i$.

There is a natural ‘diagonal’ inclusion functor $\delta : C \to S^*$, sending each $(\Delta_1, \ldots, \Delta_n)$ to the $S$-sequence consisting of only $(\Delta_1, \ldots, \Delta_n)$’s with the identity morphism between each object in the sequence. We intend to compile the $m$ rules into a single rule on the context $C$.

Take any object $A^* \equiv A^{(m)} \to \cdots \to A$ in $S^*$, and consider the representable functor $S^*(A^*, -)$. We compose $\delta$ with this functor, which gives for each $C$-object $(\Delta_1, \ldots, \Delta_n)$ the set $S^*(A^*, \delta(\Delta_1, \ldots, \Delta_n))$. We claim that each morphism in this set is totally determined.
by its first component, \( \phi : A \to (\Delta_1, \ldots, \Delta_n) \). We write \( \phi \) suggestively as \( l \), and notice that there is only one way to make the diagrams commute:

\[
\begin{array}{ccc}
A^{(m)} & \xrightarrow{l^{(m)}} & (\Delta_1, \ldots, \Delta_n) \\
\downarrow^{l(m)} & & \downarrow \\
\cdots & & \cdots \\
\downarrow^{l'} & & \downarrow \\
A' & \xrightarrow{l'} & (\Delta_1, \ldots, \Delta_n) \\
\downarrow^{l'} & & \downarrow \\
A & \xrightarrow{l} & (\Delta_1, \ldots, \Delta_n)
\end{array}
\]

Then, we can think of the sequence map \( A^* \to \delta(\Delta_1, \ldots, \Delta_n) \) simply as a morphism \( l : A \to (\Delta_1, \ldots, \Delta_n) \) in \( S \) such that \( l \cdot l^{(i)} \) is in \( S^{(i)} \) for each \( i \). In other words, this gives a 'spear' of selection of parameters in \( (\Delta_1, \ldots, \Delta_n) \) as depicted in the diagram:

\[
\begin{array}{ccc}
A^{(m)} & \xrightarrow{l^{(m)}} & (\Delta_1, \ldots, \Delta_n) \\
\downarrow^{l(m)} & & \downarrow \\
\cdots & & \cdots \\
\downarrow^{l'} & & \downarrow \\
A' & \xrightarrow{l'} & (\Delta_1, \ldots, \Delta_n) \\
\downarrow^{l'} & & \downarrow \\
A & \xrightarrow{l} & (\Delta_1, \ldots, \Delta_n)
\end{array}
\]

Claim 49. The inclusion \( \delta : C = \bigcap S \to S^* \) is always full. That is, for any pair of objects \( \Delta, \Gamma \in C \), the inclusion \( C(\Delta, \Gamma) \to S^*(\delta \Delta, \delta \Gamma) \) is actually bijective.

Proof. We show the map is surjective. Choose any \( l^* : \delta \Delta \to \delta \Gamma \). We know that this is determined by the first map \( l : \Delta \to \Gamma \), and hence all coordinates of \( l^* \) are \( l \). This is the image of \( l \) under \( \delta \).

\[\Box\]

3.5.2. Sequences of operations and compilations

Given a sequence \( A^* \) in \( S^* \), we define a sequence of operations on \( A^* \) to simply be a collection of \( \mathbf{FPpos} \) maps \( r^{(i)} : \prod_{1 \leq j \leq n} A_j^{(i)} \to B^{(i)} \) which we informally write \( r^{(i)} : A^{(i)} \to B^{(i)} \). That is, it is an operation in the same sense we have been using, specified for each object in the sequence. We write the whole sequence of operations informally as \( r^* : A^* \to B^* \).

Given a sequence of operations \( r^* : A^* \to B^* \) and a setting of parameters \( l : A^* \to (\Delta_1, \ldots, \Delta_n) \) from \( A^* \) to an object of \( C \), we construct the result of applying the sequence of operations \( r^* \) to \( (\Delta_1, \ldots, \Delta_n) \) along \( l \) as follows:

1. Construct the diagram below and take a fposet-universal diagram:

\[
\begin{array}{ccc}
A^{(m)} & \xrightarrow{l^{(m)}} & (\Delta_1, \ldots, \Delta_n) \\
\downarrow^{l(m)} & & \downarrow \\
\cdots & & \cdots \\
\downarrow^{l'} & & \downarrow \\
A' & \xrightarrow{l'} & (\Delta_1, \ldots, \Delta_n) \\
\downarrow^{l'} & & \downarrow \\
A & \xrightarrow{l} & (\Delta_1, \ldots, \Delta_n)
\end{array}
\]
\[
\prod_{i \in n} A_i \xrightarrow{r} B \\
\downarrow + i_l \\
\prod_{i \in n} \Delta_i \xrightarrow{r} P
\]

This is just the application of the first operation \( r \) on \( (\Delta_1, \ldots, \Delta_n) \) using the parameter selection \( l \).

We write the above diagram in shorthand to reduce clutter:\(^{15}\)

\[
\begin{array}{ccc}
A & \xrightarrow{r} & B \\
\downarrow & & \downarrow \\
\Delta & \xrightarrow{r} & P
\end{array}
\]

2. Compose \( l'_1 + \ldots + l'_n : A'_1 + \ldots + A'_n \to A_1 + \ldots + A_n \) with \( r \) to obtain \( r(l') : A'_1 + \ldots + A'_n \to B \),\(^{16}\) thought of as applying the operation \( r \) to the parameter-translation \( l' \). We further compose this with \( l \) to get a selection of parameters by \( A' \) in the output of the first operation acting on \( \Delta \):

\[
\begin{array}{ccc}
A' & \xrightarrow{r'} & B' \\
\downarrow r(l') & & \downarrow \\
A & \xrightarrow{r} & B \\
\downarrow & & \downarrow \\
\Delta & \xrightarrow{r} & P
\end{array}
\]

We then take the pushout of \( r' \) and \( lrl' \) to obtain a new output.

3. After \( m \) steps, the process completes, giving a ‘staircase’

\(^{15}\) The notation \( A \) is now ambiguous between \( A = (A_1, \ldots, A_n) \) and \( \prod A_i \). However, since \( B \) is in \( \text{FPos} \hookrightarrow \text{Der} \), not \( \text{Der}^n \), the notation \( r : A \to B \) disambiguates, indicating we must mean \( \prod A_i \).

\(^{16}\) We also ambiguously write as \( l' \) to mean \( l'_1, \ldots, l'_n \) or their sum \( + i_l' \). Again, the notation \( r(l') \) disambiguates: \( r \) has domain \( \prod A_i \) so we must mean the sum of the coordinates of \( l' \).
The process seems so ad hoc that it might not even be functorial. However, this is not the case. The pushout lemma guarantees that we can ‘paste’ pushout squares together to obtain a pushout square. Repeated applications of the pushout lemma show that we could also have started from the top of the staircase, pushing the vertical map out along the horizontal map sharing its domain.

In particular, we may compute $P(m)$ as the pushout of $l$ along a map $A \rightarrow \ldots \rightarrow X$, the composition of all maps lying even with the first ‘step’ of the staircase once all squares are filled in. We write $r^* : A^* \rightarrow B^*$ for this function, so that $P(m)$ can be computed directly from $l$ and $r^*$. We call $r^*$ the compilation of operations $r^* : A^* \rightarrow B^*$.

### 3.5.3. The rule generated by a sequence of rules

Let $S = (S, \ldots, S^{(m)})$ be a sequence of subcategories of $\text{Der}^n$. We want to say when a given sequence of operations $r^* : A^* \rightarrow B^*$ generates a $C$-rule $G : C \rightarrow \text{Set}$ when $C = \bigcap S$.

Let $G : C \rightarrow \text{Set}$ be a $C$-rule, and let $r^* : A^* \rightarrow B^*$ be an $S$-sequence of operations with compilation $\overline{r^*} : A \rightarrow Z$. We construct a functor $F_r : S^*(A^*, \delta-) \rightarrow G$ as follows. Each element of $S^*(A^*, \delta(\Delta))$ for a given $\Delta \in C$ is essentially an $S$-morphism $l : A \rightarrow \Delta$. We take $l$ to the isomorphism class of an operation $h : \Delta \rightarrow P$ if we can construct a pushout diagram in $\text{FPos}$:

\[
\begin{array}{c}
A \xrightarrow{l} \Delta \\
\downarrow \overline{r} & \downarrow h \\
Z & \rightarrow P
\end{array}
\]

We again say that the rule $G$ is generated by this sequence of operations $r^* : A^* \rightarrow B^*$ iff this natural transformation is epimorphic.
We give a proposition of how rule compilation simplifies when the rule is generated by a sequence of operations \( r^{(i)} : A^{(i)} \to B^{(i)} \) where the \( A^{(i)} \) are all equal and \( A^{(i)} \to A^{(i-1)} \) is the identity for all \( i \). In other words, all the rules to be compiled act on the same object lying in \( \bigcap S \).

Claim 50. Let \( S = (S_1, \ldots, S^{(m)}) \) be a sequence of subcategories of \( \text{Der}^n \), and let \( G : \bigcap S = C \to \text{Set} \) be a rule generated by the sequence \( r^* : A^* \to B^* \). If \( A^* = \delta A \) for some \( A \in C \), then \( G \) is generated by the compilation \( r^* : A \to Z \in G(A) \).

Proof. This follows from the fullness of the inclusion \( \bigcap S \subset S^* \). Fullness implies \( i : C(A, -) \to S^*(\delta A, \delta -) \) with coordinates \( i_\Delta : C(A, \Delta) \to S^*(\delta A, \delta \Delta) \) mapping \( h : A \to \Delta \) to \( \delta h : \delta A \to \delta \Delta \) is a natural isomorphism. Furthermore, the composition of \( i \) with any epimorphism \( S^*(\delta A, \delta -) \to G \) is an epimorphism, giving a single operation generating \( G \) by the Yoneda lemma.

3.5.4. A note on sequences

While we have constructed compilations of operations in terms of sequences, it may be that what we are really interested in is a family of operations \( f^{(i)} : M^{(i)} \to N^{(i)} \) on some other domain \( M^{(i)} \), where we apply each operation to the coordinate \( A^{(i)} \) of the parameter sequence. The reason we use the ‘intermediate’ step of introducing sequences of parameters \( A^* \) separate from these rules is that we may need more elements in toto than required by any one of the operations to accommodate them all simultaneously. We introduce the sequence \( A^* \) having at least the requisite structure to be able to relate all rules in the family, and then apply the family of operations to the sequence to get a sequence of rules from the family.

Formally, we say that an \( S \) family of operations is simply a collection of (epimorphic) operations \( f^{(i)} : M^{(i)} \to N^{(i)} \) where \( M^{(i)} \) is in \( S^{(i)} \). We apply this family to the sequence \( A^* \) along family of morphisms \( \phi^{(i)} : M^{(i)} \to A^{(i)} \), where each morphism is in \( S^{(i)} \) respectively by taking the requisite pushout at each coordinate to obtain operations \( r^{(i)} : A^{(i)} \to B^{(i)} \).

3.5.5. Examples

We look at three main cases of compilations of rules: argument selection as phrasal attachment + selection and specification as phrasal attachment + licensing. We actually view adjunction and licensing as the same operation, from opposite ‘points of view’.

Adjunction/phrasal attachment. Recall that we construct an adjunction rule using the condition category \( C \subset \text{Der}^2 \) of pairs of rooted derivations with injective lexical item and root-preserving maps between them and the operation \( f : 1 + 1 \to S \) on \((1, 1) \in C \) attaching the second singleton to the first, where \( S \) is the Sierpinski space.
Selection. We again restrict attention to $C \subset \text{Der}^2$. We use $S$ as the template diagram for a lexical item label with a single feature. We generate a rule $G$ from the operation $f : S + S \to T$, where $T$ is the (non-tree) partial order $a, b < c$, and $f$ maps each of the features of each copy of $S$ to the same element $c$, and leaves the roots alone (mapping them to $a$ and $b$). A $C$-map $k : S \to \Delta$ is essentially a selection of lexical item $l$ projecting to the root of $\top_{\Delta}$ (usually unique) and feature close to it. Selection is then just the identification of lexical features in two merging syntactic objects.

Licensing. We first construct ‘licensing’. We use the same $C$ as in the previous two rules. We again use $S$ as a template lexical item label and single feature. We generate licensing as $f : S + S \to X$ where $X$ is the (non-tree) space of 4 points with relations $a < c$ and $b < d < c$, where $a$ and $b$ are the roots of the copies of $S$, and $c$ and $d$ are the features close to them, respectively. This operation effectively takes a feature of the second argument and makes it depend on a feature of the first, by attaching a feature of the first to it.

We now give examples of compilations of rules. In each case, we will be compiling 2 rules, so $m = 2$, and each rule will be binary, so $n = 2$.

Argument selection. We view argument selection as adjunction + selection. We let $S = (C', C)$ both be $C$ as above. We let $A' \in C'$ and $A \in C$ both be $(S, S)$, where $S$ is the Sierpinski space, and we define the sequence $l' : (S, S) \to (S, S)$ to be the identity.

We apply adjunction to $(S, S)$ by targeting the roots to obtain $r : A \to B$. We apply selection exactly as defined to obtain $r' : A' \to B'$. The compilation is given below.

![Compilation of adjunction and selection](image)

Figure 34: Compilation of adjunction and selection. The $b$ phrase is an argument of the $a$ phrase, since it is in its minimal domain, and the selection feature $c$ and selectee $d$ have identified.

The pullback of the two categories is clearly $C$ since they are identical. Let $G : C \to \text{Set}$ be the rule generated by this sequence of rules. For any object $(\Delta_1, \Delta_2) \in C$, a morphism
l : A* → δ(Δ₁, Δ₂) is totally determined by l : A → (∆₁, ∆₂), and since l' is the identity and C' = C, any C-morphism will be a morphism of sequences. So G is just the ARGUMENT-SELECT rule from before.

**Specification.** We just draw the associated ‘staircase’ diagram, proceeding identically as before.

![Diagram](image)

Figure 35: Compilation of adjunction and licensing. The b phrase is a specifier of the a phrase, since b is in the minimal domain of a, and a licensing feature of a has attached to a feature of b (or more loosely, gone into the domain of b).

Like argument selection is phrasal attachment followed by feature identification, unselected specification can be seen as phrasal attachment followed by attachment of a feature from the adjoined phrase to the adjoining phrase.

### 3.6. Summary

In this section, we focused on methods for building derivations iteratively out of others.  

In §3.1, we focused on how to assign operations such as f : Δ + Γ → Z to tuples of derivations, which we called a *rule*. Namely, a rule G operates only in certain *contexts* specified by a subcategory C ⊂ Derⁿ in a functorial way. Furthermore, we required that this functorial behavior be fairly straightforward: if f : ∏ Δᵢ → Z is a G-operation, and ⟨φᵢ⟩ : (Δᵢ) → (Γᵢ) is a C-context preserving morphism, then we may push f out along +ᵢφᵢ to obtain an operation on ∏ Γᵢ. We constructed a category of all operations Der\FPpos, and we constructed a functor ext from this category to Der, which simply ‘extends’ the domain of an operation along that operation. We showed that these extensions have good properties: given a derivation Δ and operation h : Δ → Z, the input derivation is an open subderivation, the extended derivation internally represents the operation h, and when Z
is rooted and $\Delta$ is a forest, the extension is a tree. We then gave a simple inductive method for constructing derivations, which simply uses iterative application of some set of rules to some base set of lexical items.

In §3.2, we characterized a language associated to a grammar as a full subcategory of $\text{Der}$ of generable derivations. We gave a simple definition of equivalence of subcategories which we applied to these languages. Namely, two languages are equivalent as subcategories of $\text{Der}$ if for each derivation in one, there is an isomorphic derivation in the other, and vice-versa.

In §3.3, we analyzed grammatical relations. We described grammatical relations as a degree of overlap between heads which both map into a common step. When one head projects to a phrase in the minimal domain of another, we defined three increasingly close degrees of formal overlap: (unlicensed/non-agreeing) adjunct, (unselected) specifier, and (selected) argument. We characterized an adjunct as a phrase in the minimal domain of another, which undergoes no further licensing or selection; we characterized a specifier as an attached phrase which additionally undergoes licensing but not selection, while we characterized a selected argument as one which additionally undergoes selection. These come out as naturally ordered as the upsets of a partial order of implication of nonempty overlap.

In §3.4, we constructed toy rules which generate these relations. We gave a weak structural notion of feature ‘availability’ and how to specify context categories which select them. We then realized argument selection, specifier licensing, and adjuncts as arising from simple rules. We also suggested ways which these toy rules could be weakened to allow for more general long-distance agreement, and showed how derivational structure already captures certain properties of ‘minimality up to equidistance’ naturally, since derived SOs are already reduced ‘mod projection’.

In §3.5, we formalized the intuition that certain rules can be ‘decomposed’ into simpler primitives. For example, we realized argument selection as phrasal attachment + feature identification, while specification was realized as phrasal attachment followed by feature attachment, viewed as licensing. Adjunction is simply phrasal attachment. We gave a general method for compiling operations.

4. Copying

All constructions of the previous section describe derivations from a ‘phrase-structural’ perspective. They describe dependencies such as selection, licensing, and agreement, as well as the nesting of constituents. However, this phrase structure alone was not developed to model ‘transformations’.

It is straightforward enough to copy constituents in $\text{Der}$. We simply specify a constituent with an embedding and construct a duplicate via a coproduct. However, we would like to record this ‘copy-data’ in the object itself.
We do this simply through use of a morphism $e : \Delta \rightarrow \Delta$. We first look at a motivating example with trees. For a tree $T$, copy-data will be an order-preserving function $e : T \rightarrow T$ taking each element $x \in T$ to the ‘most recent’ element $y \in T$ it was copied from, and fixing it if it is a base-copy. Take the tree in Fig. 36. Note that the $'$ notation is not part of the structure of the tree. Rather, we encode the copy-data as a function $e : T \rightarrow T$ mapping $b' \mapsto b$, $d' \mapsto d$, and $f' \mapsto f$, and all other elements are fixed. This is an order-preserving map, and we can recover the fact that the constituent dominated by $b'$ is a copy by the fact that $b'$ is a minimal nonfixed element of $e$.

![Figure 36: A tree with copies](image)

Multiple copies and ‘remnant movement’ work the same.

![Figure 37: A tree with multiple copies and remnant movement](image)

The copy-data associated with the tree $T$ in Fig. 37 is the order-preserving map $e : T \rightarrow T$ mapping the $x''$ elements to $x'$ elements, and $x'$ elements to $x$ elements, and fixing all others. We can again see which constituents were copied as they are the minimal nonfixed elements. For these maps, $e$ eventually becomes ‘stable’ in that $e^n = e^{n+1}$ for sufficiently large $n$. This ‘eventual image’ maps each element to its base copy, so that we can tell which element it is a copy of.
Finally, we can view copy-data as a unary operation (‘reconstruct to previous copy’) on a SO $T$. Given two SOs-with-copy-data $(T, e)$ and $(S, f)$, the usual definition of a unary algebra homomorphism ‘preserves copy-data’. We say that $k : (T, e) \to (S, f)$ is a homomorphism if $k : T \to S$ is an order-preserving function such that $ke = fk$. That is, if $x \in T$ is a copy of $e(x) \in T$, then a homomorphism $k : (T, e) \to (S, f)$ requires that $k(x)$ be a copy of $k(e(x)) = f(k(x))$.

The main goal of this section is to generalize the above construction to derivations, and give a toy model for generating copies in derivations.

### 4.1. Copying with trees - toy example

We give an example for how to construct copy-data iteratively on trees, given a method for iteratively constructing trees.

Given binary trees $X$ and $Y$, we define (up to isomorphism) a tree $X \ast Y$ which consists of $X + Y + 1$ as points, with the coproduct ordering on $X + Y \subset X \ast Y$ such that $r \in 1$ is the least element of $X \ast Y$. In other words, we adjoin a new root $r$ which immediately branches into $X$ and $Y$.

We want to endow these trees with copy-data. We assume the base ‘lexical item’ trees $L$ come with the trivial endomorphism $1_L : L \to L$ fixing all elements, indicating that there are no nontrivial chains. If $(X, e)$ and $(Y, f)$ are two trees with copy-data, we can construct $(X \ast Y, e \ast f)$ such that $(e \ast f)(z) = z$ if $z$ is the new root, and otherwise acts like $e + f$. That is, if $z \in X$, then $(e \ast f)(z) = e(z)$, viewed as an element of $X \ast Y$, and symmetrically if $z \in Y$. $e \ast f$ is clearly order-preserving from $X \ast Y$ to itself.

This shows how to extend $\ast$ to trees with copy-data. Now, we want to construct a method of ‘adding copy-data’ if we copy a constituent $U \subset X$ and want to re-merge it with $X$. That is, we want to take a tree-with-copy-data $(X, e)$ and a constituent $U \subset X$ and form $X \ast U$, but where the copy-data on $X \ast U$ includes information that $U$ is a copy of a part of $X$.

We will write this function $e_U : X \ast U \to X \ast U$. Note we already have a pair of functions $e : X \to X$ and $u : U \hookrightarrow X$, which we can combine into a single function $g : X + U \to X$. We write the ‘copy’ of $U$ as $U'$ and compose this function with the inclusion $X \hookrightarrow X + U'$ to get a function $k : X + U' \to X + U'$. Since $X + U' \subset X \ast U$, we define $e_U$ to be $k$ on $X + U'$ and the identity on the root element. This gives an endomorphism $e_U$ which fixes the root, takes an element $x \in U'$ in the ‘copy’ to its position in $U \subset X$, and acts like $e$ on $X$. Importantly, $(X, e) \subset (X \ast U, e_U)$ is a subalgebra, and the only ‘new’ data added is the behavior on the copy $U'$, identifying it with the subset $U \subset X \subset X \ast U$ it was copied from. Interestingly, restricting $e_U$ to $U'$ does not give a subalgebra: to be a subalgebra the substructure must contain all prior copies down to the base copy.
4.2. Copies in Der

It is straightforward to generalize the 1-algebra construction to any category. Given a category \( \mathcal{C} \), the 1-algebras over \( \mathcal{C} \) has as objects \( \mathcal{C} \)-endomorphisms \( e : C \to C \) on an object \( C \in \mathcal{C} \), and has \( \mathcal{C} \)-morphisms \( k : (C, e) \to (D, f) \) such that \( ke = fk \) as morphisms.

We will denote the category of 1-algebras of derivations as \( \mathcal{1}\text{Der} \), and the category of 1-algebras of fposets as \( \mathcal{1}\text{FPos} \). We want to give various analogous constructions on these categories to talk about rules.

**Claim 51.** The following generalizations hold on the categories of 1-algebras:

1. There is a subcategory inclusion \( 1^i : 1\text{FPos} \to 1\text{Der} \) given by \( (P, e) \mapsto (iP, ie) \).

2. \( 1^i \) has a left adjoint \( 1\top(-) \) given by \( (\Delta, e) \mapsto (\top\Delta, \top e) \)

3. There is an extension functor \( 1\text{ext} : 1\text{Der}\setminus1\text{FPos} \to 1\text{Der} \) given by \( h : (\Delta, e) \to (P, r) \mapsto (\Delta^h, e^r) \), where \( e^r \) is \( \text{ext}(e, r) \) as in the diagram below.

   \[
   \begin{array}{ccc}
   \Delta & \xrightarrow{h} & P \\
   e \downarrow & & \downarrow r \\
   \Delta & \xrightarrow{h} & P
   \end{array}
   \]

   It acts functorially on \( (\phi, f) : (h : (\Delta, e) \to (P, r)) \to (k : (\Gamma, g) \to (Q, s)) \) as \( \phi^f : (\Delta^h, e^r) \to (\Gamma^k, g^s) \), where \( \phi^f \) is \( \text{ext}(\phi, f) \).

4. Every diagram below has a 1-fposet \( (P, f) \) with 1-morphisms \( k : (\Gamma, g) \to (P, f) \) and \( j : (Z, s) \to (P, f) \) which is *universal* among such 1-fposets.

   \[
   \begin{array}{ccc}
   (\Delta, e) & \xrightarrow{\phi} & (\Gamma, g) \\
   e \downarrow & & \downarrow k \\
   (Z, s) & \xrightarrow{j} & (P, f) \\
   \end{array}
   \]

5. A ‘pushout lemma’ holds for these diagrams. That is, if the left square is universal, the right square is universal if and only if the whole square is.

   \[
   \begin{array}{ccc}
   (\Delta, e) & \xrightarrow{\phi} & (\Gamma, g) & \xrightarrow{\psi} & (\Sigma, s) \\
   f \downarrow & \quad & g \downarrow & \quad & \downarrow h \\
   (X, l) & \xrightarrow{a} & (Y, m) & \xrightarrow{b} & (Z, n)
   \end{array}
   \]
Proof. The first claim is immediate from the functoriality of $i$. That $^1\top(-)$ is a functor is again immediate from the functoriality of $\top(-)$.

We give a unit and counit of an adjunction between them. Since $\top(-)$ is functorial, the unit $\eta_\Delta : \Delta \to i\top\Delta$ of $\top(-) \dashv i$ extends to a homomorphism $(\Delta, e) \to (i\top\Delta, i\top e)$ for any $(\Delta, e) \in 1\text{Der}$. We define $\eta : 1\text{Der} \to 1^1\top(-)$ as the natural transformation with coordinates $\eta_{(\Delta, e)} : (\Delta, e) \to (\top\Delta, \top e)$ which is $\eta_\Delta$ on the underlying derivation. The counit $\epsilon$ of $\top(-) \dashv i$ is an isomorphism $\epsilon_P : \top_i P \to P$ which extends to an isomorphism $\epsilon_{(P, r)} : (\top_i P, \top_i r) \to (P, r)$ for any $(P, r) \in 1\text{FPos}$. $\eta$ and $\epsilon$ give the unit and counit for an adjunction $^1\top(-) \dashv ^1i$ since $\eta$ and $\epsilon$ give one for $\top(-) \dashv i$.

We give proofs of the third and fourth claims by construction in SECTION APPENDIX. The fifth claim can be proven just using the universal properties of each square in a manner identical to the pushout lemma.

This will handle generalizations of both cases from finite trees.

Note 6. Suppose that the following-diagram of 1-algebras is universal for 1-fposets.

$$(\Delta, e) \xrightarrow{\phi} (\Gamma, g)$$

$$(Z, s) \xrightarrow{j} (P, f)$$

It does not follow that the following diagram is universal for fposets.

$$\Delta \xrightarrow{\phi} \Gamma$$

$$(Z, s) \xrightarrow{j} (P, f)$$

Summing w/o copying. It is straightforward to sum copy-data: if $(\Delta_i, e_i)$ are 1-derivations, their sum $\coprod (\Delta_i, e_i)$ is simply the sum of the $\Delta_i$ with function $+_i e_i$ acting on each summand and taking range within each summand. That is, $+_i e_i : \coprod \Delta_i \to \coprod \Delta_i$ takes an element $x \in \Delta_i$ to $e_i(x) \in \Delta_i$, viewed as a subderivation of the sum.

Given $n$ 1-derivations $(\Delta_i, e_i)$, we call a 1-morphism $h : \coprod (\Delta_i, e_i) \to (Z, s)$ from their sum to a 1-fposet an $n$-ary 1-operation on the $(\Delta_i, e_i)$. We take conditions for a non-copying $n$-ary rule of 1-derivations to be a subcategory $C \subset (1\text{Der})^n$. We define a non-copying $C$-rule to be a functor $G : C \to \text{Set}$ such that: (i) for each $(\Delta, e) \in C$, each element of the set $G(\Delta, e)$ is an isomorphism class of a 1-operation $h : (\Delta, e) \to (P, r)$, and (ii) for any $C$-morphism $\phi : (\Delta, e) \to (\Gamma, g)$ and $G$-operation $h : (\Delta, e) \to (P, r)$, the isomorphism class of the operation $h' : (\Gamma, g) \to (Q, s)$ is in $G(\Gamma)$ and $G(\phi)$ maps $h$ to $h'$, where $h'$ is any operation such that the following diagram is universal.
Copying. Take a 1-derivation \((\Delta, e)\) and open subset \(U \subset \Delta\). We have morphisms \(e : \Delta \to \Delta\) and \(u : \Delta/U \hookrightarrow \Delta\) which we can sum to obtain a morphism \(g : \Delta + \Delta/U \to \Delta\). This can be composed with the inclusion \(\Delta \hookrightarrow \Delta + \Delta/U\) to obtain a map \(e^U : \Delta + \Delta/U \to \Delta + \Delta/U\).

This ‘copying operation’ gives a new 1-derivation \((\Delta + \Delta/U, e^U)\). Much like ‘summing’ can be seen as a functor \(\sqcup : (1\text{Der})^n \to 1\text{Der}\), this ‘adding copy-data for one copied open’ can be made functorial. We denote by \(s\text{-Der}\) the category whose objects are 1-derivations \((\Delta, e)\) paired with an open subderivation \(U \subset \Delta\). The morphisms of this category \(\phi : ((\Delta, e), U) \to ((\Gamma, g), V)\) are 1-morphisms \(\phi : (\Delta, e) \to (\Gamma, g)\) such that \(\phi(U) \subset V\). We define a functor \(\text{copy} : s\text{-Der} \to 1\text{Der}\) which maps \(((\Delta, e), U)\) to \((\Delta + \Delta/U, e^U)\).

We take conditions for a copying rule to be a subcategory \(C \subset s\text{-Der}\). We define a copying \(C\)-rule to be a functor \(G : C \to \text{Set}\) such that: (i) every element of \(G((\Delta, e), U)\) is an isomorphism class of a 1-morphism \(h : (\Delta + \Delta/U, e^U) \to (P, r)\), and (ii) if \(\phi : ((\Delta, e), U) \to ((\Gamma, g), V)\) is a \(C\)-morphism and \(h : (\Delta + \Delta/U, e^U) \to (P, r)\) is a \(G\)-operation, the isomorphism class of the operation \(h' : (\Gamma + \Gamma/V, g^V) \to (Q, s)\) is in \(G((\Gamma, g), V)\) and \(G(\phi)\) maps \(h\) to \(h'\), where \(h'\) is any operation such that the following diagram is universal.

\[
\begin{array}{ccc}
(\Delta + \Delta/U, e^U) & \xrightarrow{\phi + \phi^V_U} & (\Gamma + \Gamma/V, g^V) \\
\downarrow h & & \downarrow h' \\
(P, r) & \xrightarrow{f} & (Q, s)
\end{array}
\]

4.3. Generation, compilation, and properties of rules for \(1\text{Der}\)

It was shown above how to straightforwardly generalize rules from derivations to 1-derivations, which keep track of copy-data. The main difficulty in generalizing from derivations to 1-derivations is how to produce operations \(h : (\Delta, e) \to (Z, s)\) which are 1-morphisms systematically from generators. The main issue rests in the fact that the rules \(r : M \to Z\) and contexts for them \(k : M \to \Delta\) which we are often interested in do not and should not care about copy-data. Furthermore, ignoring copy-data, a universal map of the kinds we have explored already may produce maps \(h : \Delta \to Z\) which are not underlying 1-morphisms. We give an example.

Let \((\tilde{S}, e)\) be the Sierpinski space \(a' < a\) together with the copy-data mapping \(a' \mapsto a\) and \(a \mapsto a\). Let \((1, !)\) be the space \(\{b\}\) with the identity function as copy-data. Let \(h : 1 + \tilde{S} \to Z\).
be the order-preserving map attaching $b$ to $a'$. There is no 1-algebra structure on $Z$ which will admit $f$ as a homomorphism. To see this, note that if $(X,e)$ is any 1-algebra and $h : X \to Y$ any surjective function, the 1-algebra structure on $Y$ is determined if $h$ is to be a homomorphism. The demand that $h$ puts on $Z$ in our example says that if $f$ were to be the requisite unary operation on $Z$, $f(b) = b$ but also $f(a') = a$. However, this is not order-preserving. See Fig. 38.

![Figure 38: An order-preserving function which cannot be a 1-algebra homomorphism.](image)

**Universal constructions for 1-algebras.** As mentioned, we typically don’t want the ‘template’ derivations we use to specify which elements to combine to be sensitive to copy-data. That is, we will still be interested in operations $r : M \to X$ in contexts $k : M \to \Delta$, where $M$, $r$, $X$, and $k$ do not see copy-data. However, $\Delta$ will have copy-data on it, and we want to *produce* operations $h : (\Delta,e) \to (Z,s)$ which do preserve copy-data. We handle both of these issues by generalizing the pushout construction. We will allow the ‘arms’ of the upper corner of the pushout diagram to ignore copy-data, while the ‘completion’ of the diagram must produce a morphism which is a 1-algebra homomorphism.

Given a 1-derivation $(\Delta,e)$, an operation $r : M \to X$ and a context $k : M \to \Delta$, we want the best 1-algebra homomorphism $h : (\Delta,e) \to (Z,s)$ completing the square with sides $r$ and $k$. More precisely, given $(\Delta,e)$, $r$ and $k$, we would like to find a 1-fposet $(Z,s)$, order-preserving function $g : X \to Z$ and 1-operation $h : (\Delta,e) \to (Z,s)$ such that $hk = gr$ as derivation morphisms such that $h$ is universal with respect to this property. That is, if $(Y,t)$ is any 1-algebra of finite partial orders, $m : (\Delta,e) \to (Y,t)$ any 1-algebra homomorphism, and $n : X \to Y$ any map of finite partial orders such that $mk = nr$ as morphisms of derivations, then there is a unique 1-algebra homomorphism $u : (Z,s) \to (Y,t)$ such that $uh = m$ and $ug = n$. See Fig. 39.

![Figure 39: Universal map with one 1-algebra homomorphism leg.](image)
We will say that a square diagram of the form above (without \( Y, u, m, \) and \( n \)) is \((\text{fposet},\text{hom})\)-universal if it is universal in the sense above. We distinguish this from the case before. Let \( \Delta, \Gamma \) and \( \Sigma \) be any derivations, and let \( \phi \) and \( \psi \) be any 1-morphisms. Let \( P \) be any fposet, and let \( k \) and \( j \) be any 1-morphisms.

\[
\begin{array}{c}
(\Delta, e) \\
\downarrow \phi \\
(\Sigma, s) \\
\downarrow j
\end{array}
\begin{array}{c}
\rightarrow
\begin{array}{c}
(\Gamma, g) \\
\downarrow k \\
(P, f)
\end{array}
\end{array}
\]

Figure 40: (hom,hom)-universal diagram

We say the above diagram is \((\text{hom},\text{hom})\)-universal if for every 1-fposet \((Q, s)\) and pair of 1-morphisms \( x : (\Gamma, g) \to (Q, s) \) and \( y : (\Sigma, s) \to (Q, s) \) such that \( x\phi = y\psi \), there is a unique 1-morphism \( u : (P, f) \to (Q, s) \) such that \( x = uk \) and \( y = uj \).

When it is clear from context, we will informally call the right vertical map the pushout of the left vertical map along the top horizontal map - e.g. in Fig. 39, \( h \) is the pushout of \( r \) along \( k \); in Fig. 40, \( k \) is the pushout of \( \psi \) along \( \phi \).

**Claim 52.** For any pair of derivation morphisms \( k : M \to \Delta \) and \( r : M \to X \) such that \( \top_r : \top_M \to \top_X \) is surjective (inclusive) and any 1-derivation structure \( e : \Delta \to \Delta \) on \( \Delta \), there is an \((\text{fposet},\text{hom})\)-universal square completing it.

\[
\begin{array}{c}
M \\
\downarrow k
\end{array}
\begin{array}{c}
(\Delta, e) \\
\downarrow h \text{ (hom)}
\end{array}
\begin{array}{c}
X \\
\downarrow g
\end{array}
\begin{array}{c}
(\Gamma, g)
\downarrow h' \text{ (hom)}
\end{array}
\begin{array}{c}
(Z, s) \\
\downarrow f \text{ (hom)}
\end{array}
\begin{array}{c}
(Q, s)
\end{array}
\]

Furthermore, suppose the left square is \((\text{fposet},\text{hom})\)-universal. Then the right square is \((\text{hom},\text{hom})\)-universal if and only if the whole square is \((\text{fposet},\text{hom})\)-universal.

\[
\begin{array}{c}
M \\
\downarrow k
\end{array}
\begin{array}{c}
(\Delta, e) \\
\downarrow h \text{ (hom)}
\end{array}
\begin{array}{c}
X \\
\downarrow g
\end{array}
\begin{array}{c}
(\Gamma, g) \\
\downarrow h' \text{ (hom)}
\end{array}
\begin{array}{c}
(Z, s) \\
\downarrow f \text{ (hom)}
\end{array}
\begin{array}{c}
(Q, s)
\end{array}
\]

**Proof.** We give a proof of the first claim by giving a construction of it in APPENDIX. The second claim can be proven just using the universal properties.

(right square ⇒ outer square) Let \((A, a)\) be any 1-fposet and let \( x : (\Gamma, g) \to (A, a) \) be any 1-morphism and \( y : X \to A \) any morphism such that \( x\phi k = yr \). By the universal property of the left square, there is a unique 1-morphism \( u : (Z, s) \to (A, a) \) such that \( y = ug \) and \( x\phi = uh \). We then apply the universal property of the right square to get a unique map \( v : (Q, s) \to (A, a) \) such that \( u = vf \) and \( x = vh' \). This produced a unique 1-morphism \( v : (Q, s) \to (A, a) \) such that \( x = vh' \) and \( y = ug = v(fg) \).
outer square ⇒ right square) Let \((A, a)\) be any 1-fposet and \(x : (\Gamma, g) \to (A, a)\) and \(y : (Z, s) \to (A, a)\) any 1-morphisms such that \(x\phi = yh\). Then \(x\phi k = yhk = ygr\), and we can apply the universal property of the outer square to obtain a unique 1-morphism \(u : (Q, s) \to (A, a)\) such that \(x = uh'\) and \(yg = u\bar{f}g\). Consider the 1-morphisms \(y, uf : (Z, s) \to (A, a)\), \(yg : X \to (A, a)\), and \(u\bar{f}h = uh'\phi : (\Delta, e) \to (A, a)\). Since \(yh = x\phi = uh'\phi\) by assumption, \(y\) gives a map splitting \(yg\) and \(u\bar{f}h\). But since \(u\bar{f}g = yg\), so does \(uf\). By the universal property of the left square, \(uf = y\) as morphisms of derivations, and hence as 1-morphisms. This produces a unique 1-morphism \(u : (Q, s) \to (A, a)\) such that \(uh' = x\) and \(uf = y\).

Generators for a non-copying rule. We can emulate the method of generating operations through universal constructions, even though the objects lie in different categories. Let \(i : D \to (\text{Der})^n\) be a context and \(G : D \to \text{Set}\) a rule. Let \(j : C \to \text{Der}^n\) be a subcategory and \(F : (\text{Der})^n \to \text{Der}^n\) be the functor forgetting 1-algebra structure in each coordinate, i.e. mapping \((\Delta, e_i)\in C \mapsto (\Delta_i)\in C\). Suppose we have a functor \(U : D \to C\) such that \(Fi = jU\). For a given \(M \in C\), we compose \(U\) with \(C(M, -)\) to get a functor \(C(M, U-) : D \to \text{Set}\).

Now suppose that \(r : \overline{M} \to X\) is an inclusive operation on \(\overline{M}\). For any \((\Delta, e) \in D\), we have \(U(\Delta, e) = \Delta \in C\), so that \(C(M, \Delta)\) is just the set of \(C\)-morphisms \(\phi : M \to \Delta\). Suppose that for some rule \(G : D \to \text{Set}\), for any \((\Delta, e) \in D\) and \(\phi \in C(M, \Delta)\), the pushout \(h : (\overline{\Delta}, \overline{e}) \to (Z, s)\) of \(r\) along \(\overline{\phi} : \overline{M} \to \overline{\Delta}\) is in \(G(\Delta, e)\). Then by Claim 52, the collection of functions \(\eta(\Delta, e) : C(M, U(\Delta, e)) \to G(\Delta, e)\) mapping each \(\phi \in C(M, U(\Delta, e))\) to the pushout of \(r\) along \(\overline{\phi}\) is a natural transformation. We say that \(G\) is generated by \((\mathcal{M}, r, C)\) if this natural transformation \(\eta : C(M, U-) \to G\) is epimorphic. In other words, every \(G\)-operation on \((\Delta, e) \in D\) arises from the pushout of \(r\) along \((\sum\text{ of})\) some \(C\)-map \(\phi\). Often, \(D\) will arise in a canonical way from \(C\) such as the pullback of \(C \to \text{Der}^n\) along \(F : (\text{Der})^n \to \text{Der}^n\).

Generators for a copy-rule. Let \(i : D \to \text{Der}^2\) be a context, and \(G : D \to \text{Set}\) a \(D\)-rule. Let \(j : C \to \text{Der}^2\) be a subcategory and \(F : \text{Der}^2 \to \text{Der}^2\) be the functor mapping \(\phi : ((\Delta, e), U) \to ((\Gamma, g), V)\) to \((\phi, \phi|[\overline{U}]) : ((\Delta, \Delta)/U) \to (\Gamma, \Gamma/V)\). Suppose we have a functor \(K : D \to C\) such that \(Fi = jK\). For a given \(M \in C\), we construct \(C(M, K-) : D \to \text{Set}\).

Now suppose that \(r : \overline{M} \to X\) is any operation on \(\overline{M}\) in \(C\). For any \((\Delta, e) \in D\), we have \(K((\Delta, e), U) = (\Delta, \Delta/U) \in C\), so that \(C(M, (\Delta, \Delta/U))\) is just the set of \(C\)-morphisms \(\phi : M \to (\Delta, \Delta/U)\). Suppose that for some rule \(G : D \to \text{Set}\), for any \((\Delta, e, U) \in D\) and \(\phi \in C(M, (\Delta, \Delta/U))\), the pushout \(h : (\overline{\Delta + \Delta/U}, e/U) \to (Z, s)\) of \(r\) along \(\overline{\phi} : \overline{M} \to \overline{\Delta + \Delta/U}\) is in \(G((\Delta, e), U)\). Then by Claim 52, the collection of functions \(\eta((\Delta, e), U) : C(M, K((\Delta, e), U)) \to G((\Delta, e), U)\) mapping each \(\phi \in C(M, K((\Delta, e), U))\) to the pushout of \(r\) along \(\overline{\phi}\) is a natural transformation. We say that \(G\) is generated by \((\mathcal{M}, r, C)\) if this natural transformation \(\eta : C(M, K-) \to G\) is epimorphic. In other words, every \(G\)-operation on \((\Delta, e, U) \in D\) arises from the pushout of \(r\) along \((\sum\text{ of})\) some
Compilations and properties of rules. We can generalize many of the properties of rules on derivations, such as inclusiveness and weak extension, to rules on 1-derivations. We can still characterize an operation \( h : (\Delta, e) \to (P, r) \) as inclusive if \( h : (\top_{\Delta}, \top_e) \to (P, r) \) is surjective, and say that \( G \) is inclusive if each \( G \)-operation is. Similarly, we can say that an \( n \)-ary operation is weakly extensive if for each input \( \Delta_i \), the map \((\top_{\Delta_i}, \top_{e_i}) \to \prod (\top_{\Delta_i}, \top_{e_i}) \to P\) is a subspace inclusion for each \( 1 \leq i \leq n \), and say that an \( n \)-ary rule \( G \) is weakly extensive if each \( G \)-operation is.

XXX (write more general version?) Similarly, we can generalize the compilation technique to 1-derivations in various ways. For example, let \( C^{(i)} \subset \text{Der}^n \) be conditions. \( \bigcap C^{(i)} \subset \text{Der}^n \) is a subcategory, and we take the pullback of this subcategory along \( F : (\text{Der})^n \to \text{Der}^n \), call it \( D \). An object of \( D \) is an \( n \)-tuple \((\Delta_1, e_1), \ldots, (\Delta_n, e_n)\) such that \((\Delta_1, \ldots, \Delta_n) \in \bigcap C^{(i)} \), and a \( D \)-morphism \( \phi : (\Delta_i, e_i)_{i \in \mathbb{N}} \to (\Gamma_i, g_i)_{i \in \mathbb{N}} \) is any \( n \)-tuple of 1-derivation morphisms such that \( \phi : (\Delta_i)_{i \in \mathbb{N}} \to (\Gamma_i)_{i \in \mathbb{N}} \) is a \( \bigcap C^{(i)} \)-morphism. There is a diagonal inclusion \( \delta : D \rightharpoonup C^* \) taking \((\Delta_i, e_i)_{i \in \mathbb{N}} \) to the sequence where every coordinate is \((\Delta_i)_{1 \leq i \leq n} \) with all maps the identity. For any \( A^* \in C^* \) and \((\Delta, e) \in D \), a \( C^* \)-morphism \( l^* : A^* \to \delta(\Delta, e) \) is essentially a map \( l : A \to \Delta \) in \( C \) such that \( l \circ l' \ldots \circ l^{(i)} : A^{(i)} \to \ldots A' \to A \to \Delta \) is a \( C^{(i)} \)-morphism for all \( 1 \leq i \leq m \). We write \( r^* : A^* \to B^* \) informally for a collection of \( r^{(i)} : \bigsqcup_{1 \leq j \leq n} A_j^{(i)} \to B^{(i)} \) for each \( 1 \leq i \leq m \). We can again construct a `staircase’ and complete each diagram with a universal square. The universal properties of these squares have `pushout lemmas’, and so rules can be compiled. Specifically, we only need `pushout lemmas’ for two types of diagrams of fposets and 1-fposets:

\[
\begin{array}{ccc}
A & \longrightarrow & B \\
\downarrow & & \downarrow \\
(D, d) & \overset{\text{hom}}{\longrightarrow} & (E, e) \\
& \overset{\text{hom}}{\longrightarrow} & (F, f)
\end{array}
\quad
\begin{array}{ccc}
A & \longrightarrow & (C, c) \\
\downarrow & & \downarrow \\
D & \longrightarrow & E \\
\overset{\text{hom}}{\longrightarrow} & & \overset{\text{hom}}{\longrightarrow}
\end{array}
\]

Figure 41: We say that a square consisting of a pair of maps of fposets \( A \to B \) and \( A \to C \), a 1-fposet map \((C, c) \to (Q, q)\) and fposet map \( B \to Q \) is universal if the lower right corner is a 1-fposet \((Q, q)\) such that the morphism from the 1-fposet is a 1-morphism, the square commutes as a map of fposets, and \((Q, q)\) is universal

\[100\]
It is straightforward to generalize to cases when \( D \subset \bigcap C^{(i)} \) is a subcategory. Similarly, we can generalize to \( D \)-rules when \( D \subset \text{Der}^\ast \) by pulling back over the functor \( F : \text{Der}^\ast \to \text{Der}^2 \) taking \(((\Delta, e), U)\) to \((\Delta, \Delta/U)\).

4.4. Copy-data and chains.

We also face a linguistic issue which we will only touch on here: what to copy? This is one of the main topics studied in [Kobele, 2006].

Although we have been speaking loosely of traces as being ‘structurally identical’ to their antecedents, a moment’s reflection should reveal that it is not clear exactly what this should be taken to mean. There are two aspects of structure that might be relevant to us here - the first is the derived structure, what Chomsky calls the structural description of an expression, and the second is what we might call the derivational structure, the derivational process itself.

The main problem with using copies, in any model, is that they will not represent any changes to them throughout the derivation. For example, imagine an XP needs to be licensed in some way, and achieves that licensing through movement. If this copied XP is literally the same object as the base XP, then it should still believe it needs to be licensed! Similarly, the head of the XP itself should not think it still needs a complement if one is selected. So upon copying, we desire to copy an item as it looks at the stage of the derivation where it is copied. However, this runs into problems of its own in principle. As Kobele remarks, “we need to make sure that the syntactic features that drive the derivation don’t get duplicated.” [Kobele, 2006]

We will hedge some of these issues here, though suggest how they can be addressed. For any endomorphism, \( e : \Delta \to \Delta \), we can get an equivalence relation out of \( e \): we take the coequalizer of \( e : \Delta \to \Delta \) and the identity \( 1_\Delta \). This will quotient \( \Delta \) by a relation where \( x \sim e(x) \sim e(e(x)) \sim \ldots \). For the endomorphisms of interest, those representing copy-data, we can consider two points to be ‘the same’ if they become equal under this ‘quotient’. In SECTION, we will construct a Minimalist Grammar tethered to derivations which ‘relocates’ the active features to the ‘new’ copy to deal with this issue.

While chains will become more abstract, we here will copy ‘derived objects’. We will always be taking an open \( U \in \omega \top_\Delta \) of the ‘current state’, and copying \( \Delta/U \) as described in the previous section. The new copy-data is constructed as \( e^U \) above. Recall that \( \Delta/U \) itself ‘recreates’ the parts of steps of \( \Delta \) contributing to \( \top_U \). This is part of what will allow us to see \( \top_U \) as a copied derived object, while still relating it to the base pieces it was assembled from.
5. Tethering objects of Der to Minimalist Grammars

FUNCTION TETHERING SEQUENCE OF LEXICAL ITEMS AND FEATURES TO DERIVATIONS
CORRESPONDING STABLE MULTIPLICATION WITH OPERATIONS

6. OTHER EXAMPLES?

FEATURE-SHARING WITH COPYING
MERCHANT HEAD ‘FUSION’
OTHER STUFF THAT PEOPLE WANT TO SEE WORKED OUT?

A. Proofs

Claim 5. For any point \( x \in \Delta \), for each open \( x \in U \subset \Delta \), there is an associated point \( x_U \in T_U \).

Proof. A prime filter can also be characterized as a subset \( P \subset D \) such that (i) \( d \in P \) and \( d \leq c \) implies \( c \in P \); (ii) if \( c, d \in P \), then \( c \land d \in P \); (iii) if \( \bigvee d_i \in P \), then there is some \( i \) such that \( d_i \in P \). If \( V \in P_x \subset \omega U \) and \( W \in \omega U \), \( V \subset W \), then \( W \in P_x \) since \( P_x \) is a filter, so \( P_x \cap \omega U \) is upward closed. If \( W, V \in P_x \cap \omega U \), then \( W \cap V \in P_x \) since \( P_x \) is a filter and \( W \cap V \in \omega U \) since \( \omega U \) is closed under meets in \( \mathcal{O}(\Delta) \). Finally, suppose \( V_i \in \omega U \) and \( \bigcup V_i \in P_x \cap \omega U \). Since \( P_x \) is prime, some \( V_i \in P_x \). So \( P_x \cap \omega U \) is prime in \( \omega U \).

Lemma 11.1. Der has terminal and initial objects \( 1 \) and \( 0 \) with underlying sets \( 1 \), a singleton, and \( \emptyset \), the empty set.

Proof. Let \( 0 \) have underlying set \( \emptyset \) with the trivial partial ordering of \( \emptyset \). \( \omega \emptyset = \{ \emptyset \} \) gives \( \emptyset \) the structure of a derivation. For any derivation \( \Delta \), there is exactly one set function \( 0 : 0 \hookrightarrow \Delta \) including the empty set into \( \Delta \). This will be a morphism, as \( 0 \) has no order relations to preserve, and for any \( V \in \omega U \) in \( \Delta \), \( 0^{-1} V = \emptyset \in \omega \emptyset = \omega 0^{-1} U \). This object is then initial.

Let \( 1 \) have any singleton as its underlying set with the ordering \( x \leq x \) for the single element of \( 1 \). We define \( \omega (1) = \{ 1, \emptyset \} \), and this gives the singleton the structure of a derivation. For any derivation \( \Delta \), there is exactly one set function \( ! : \Delta \rightarrow 1 \) taking every point of \( \Delta \) to the single point of \( 1 \). It is a morphism, since if \( a \leq b \) in \( \Delta \), then \( !(a) = x \leq x = !(b) \), and \( !^{-1} = \Delta \). By the definition of a derivation, both \( \emptyset = !^{-1} \emptyset \) and \( \Delta = !^{-1} \emptyset \) must be in \( \omega (\Delta) \), so \( ! \) is a morphism.

Lemma 11.2. The coproduct \( \Delta + \Gamma \) in Der can be computed as the sum of underlying spaces, with \( A + B \in (\omega \times \tau)(U + V) \) iff \( A \in \omega U \) and \( B \in \omega V \) for \( U \subset \Delta \) and \( V \subset \Gamma \).
Proof. The claim that the lattice $O(\Delta + \Gamma) \cong O(\Delta) \times O(\Gamma)$ follows straight from the duality between $\mathbf{FPos}$ and $\mathbf{FDL}$. We now check that $\omega \times \tau$ is actually a local topology on $\Delta + \Gamma$. Clearly, if $(A,B) \in (\omega \times \tau)(D,G)$, then $A + B \subset D + G$. If $(A,B), (J,K)$ are in $(\omega \times \tau)(D,G)$, then $A, J \in \omega D$ and $B, K \in \tau G$, so $(A \cap J, B \cap K) \in (\omega \times \tau)(D,G)$ and $(A \cup J, B \cup K) \in (\omega \times \tau)(D,G)$. So each $\omega \times \tau \subset O_{\Delta + \Gamma}$ is a subsheaf. To show it is a sheaf, let $\bigcup(D_i + G_i) = D + G$ be an open cover. and let $(A_i, B_i) \in (\omega \times \tau)(D_i, G_i)$ be a matching family. Then $A_i \in \omega D_i$ is a matching family on a cover of $D$ and $B_i \in \tau G_i$ is a matching family on a cover of $G$, producing elements $A \in \omega D$ and $B \in \tau G$. Then $(A,B)$ is the unique collation of the matching family $(A_i, B_i)$. This shows that $\Delta + \Gamma$ is a derivation.

We now have to show it has the universal property. Suppose that $a : \Delta \to \Xi$ and $b : \Gamma \to \Xi$ is any pair of morphisms. By duality, these correspond to homomorphisms of lattices in the other direction. We give a morphism of lattices $u^{-1} : O(\Xi) \to O(\Delta) \times O(\Gamma)$ taking $U \in O(\Xi)$ to $(a^{-1}U, b^{-1}U)$. Since $O(\Delta) \times O(\Gamma)$ is actually a product of lattices, $u^{-1}$ must be a lattice homomorphism. To show that it is a derivation morphism, take any $V \in \zeta U$ for $\zeta$ on $\Xi$. Since $a$ and $b$ are morphisms, $a^{-1}V \in \omega a^{-1}U$ and $b^{-1}V \in \tau b^{-1}U$. Then, $(a^{-1}V, b^{-1}V) \in (\omega \times \tau)(a^{-1}U, b^{-1}U).

Lemma 11.3. For a given pair of arrows $a, b : \Delta \to \Gamma$, the coequalizer $q : \Gamma \to Q$ can be constructed as follows. The open set lattice $O(Q)$ is isomorphic to the lattice \{U \in O(\Gamma) | a^{-1}U = b^{-1}U\}, and it determines the space $Q$ up to isomorphism. The local topology is given by $V \in \tau U$ for $V, U \in O(Q)$ if $V \in \tau U$, where $\tau$ is the local topology on $\Gamma$.

Proof. The set $O(Q)$ constructed as above is just the coequalizer of finite distributive lattices, and hence must be a lattice, and $O(Q) \hookrightarrow O(\Gamma)$ must be a lattice inclusion. We now show that $\tau$ does in fact turn $O(Q)$ into a derivation.

First we show that $\tau U \subset O_Q(U)$ is a sublattice. If $A, B \in \tau U$, then $a^{-1}A = b^{-1}A$ and $a^{-1}B = b^{-1}B$. Since $a^{-1}$ and $b^{-1}$ are lattice homomorphisms, $a^{-1}(A \cap B) = a^{-1}A \cap a^{-1}B = b^{-1}A \cap b^{-1}B = b^{-1}(A \cap B)$ so $A \cap B \in \tau U$. Since lattice homomorphisms preserve joins, this also implies that $A \cup B \in \tau U$.

Next we show that it is a subsheaf. That is, if $V \in \tau \hat{U}$ and $W \subset U$, then $V \cap W \in \tau \hat{W}$. If $V \in \tau \hat{U}$, then $V \in \tau U$; and if $W \subset U$ in $O(Q)$, then so in $O(\Gamma)$. Then $V \cap W \in \tau W$. We must then just show that $V \cap W \in O(Q)$, but this follows directly from the intersection preservation as above.

To see that it is a sheaf, we must check that for any cover $\bigcup_{i \in I} U_i = U$ for $U, U \in O(Q)$, if $V_i \in \tau U_i$ is a matching family, there is a unique collation $V \in \tau U$. If $\bigcup_{i \in I} V_i = U$ in $O(Q)$, then this holds in $O(\Gamma)$. A matching family on the $U_i$ is a matching family in $O(\Gamma)$, and we construct their collation as $\bigcup V_i \in \tau U$. The union of elements such that $a^{-1}V_i = b^{-1}V_i$ has this property since $a^{-1}$ and $b^{-1}$ preserve unions, so we can form the collation in $O(Q)$. 103
By construction, if \( V \in τU \), then \( V \in τU \), so the associated map \( q : Γ \to Q \) is in fact a morphism.

We must now only show that it is universal. Suppose that \( h : Γ \to Σ \) is any morphism such that \( ha = hb \). Then, \( a^{-1}h^{-1} = b^{-1}h^{-1} \). Choose any \( U \in O(Σ) \). Since \( a^{-1}h^{-1}U = b^{-1}h^{-1}U \), \( h^{-1}U \) must be in \( O(Q) \). We then construct a function \( u^{-1} : O(Σ) \to O(Q) \) sending \( U \) to \( h^{-1}U \). It follows immediately that this is a morphism of finite distributive lattices since \( O(Q) \hookrightarrow O(Γ) \) is a sublattice inclusion. Let \( ζ \) be the local topology on \( Σ \). If \( V \in ζU \), then \( h^{-1}V = u^{-1}V \in ζu^{-1}U = ζh^{-1}U \) since \( h^{-1} \) is the inverse image of a derivation morphism. So we have a unique morphism \( u : Q \to Σ \) such that \( uq = h \). □

**Lemma 11.4.** The product \( Δ \times Γ \) can be computed as the product of spaces, given the local topology \( ω \otimes τ \), determined by the following stalks. For each point \( (d, g) \in Δ \times Γ \), we define the stalk \( (ω \otimes τ)(U_{(d,g)}) \) to be \( \{ \bigcup_{i\in I} D_i \times G_i \mid D_i \in ωU_d \text{ and } G_i \in τU_g \} \).

**Proof.** The fact that the underlying set of the product is \( Δ \times Γ \) and the ordering/topology on it is as stated comes directly from the fact that \( pt \) (as a right adjoint) preserves products, if they exist. So if the product exists, it must exist on this partial order. The open set structure is the usual tensor product of lattices \( O(Δ) \otimes O(Γ) \) [Shmuely, 1979]. We then only have to show that \( ω \otimes τ \) as defined is actually a local topology to show that this is a derivation.

We first show that each coordinate of \( ω \otimes τ \) is a finite distributive lattice, and moreover a sublattice of \( O_{Δ×Γ} \) at that coordinate. For any pair of points \( d \in Δ \) and \( g \in Γ \), we show that \( (ω \otimes τ)(U_{(d,g)}) \) is a lattice. Clearly, \( U_{(d,g)} = U_d \times U_g \). The set is closed under unions in \( O(U_{(d,g)}) \), since a union of unions of ‘open boxes’ is itself a union of open boxes. If \( ∪_{j∈J} D_j \times G_j \) and \( ∪_{i∈I} D_i \times G_i \) are in \( (ω \otimes τ)(U_d \times U_g) \), we must show that \( (∪_{j∈J} D_j \times G_j) ∩ (∪_{i∈I} D_i \times G_i) \) is in \( (ω \otimes τ)(U_d \times U_g) \). But \( (∪_{j∈J} D_j \times G_j) ∩ (∪_{i∈I} D_i \times G_i) = (∪_{i∈I} D_i \times G_i) ∩ (∪_{j∈J} D_j \times G_j) \). But since \( D_i, D_j \in ωU_d \) by assumption, \( D_i \cap D_j \in ωU_d \), and similarly for \( U_g \). Then this is the union of open boxes, so it is in \( (ω \otimes τ)(U_d \times U_g) \). Then not only is \( (ω \otimes τ)(U_d \times U_g) \) a finite distributive lattice, it is a sublattice of \( O(U_d \times U_g) \).

Now we show that it is a subsheaf; i.e. we show that for each \( \bigcup D_i \times G_i \in (ω \otimes τ)(U_d \times U_g) \) and \( (d,g) \leq (a,b) \) that \( (∪ D_i \times G_i) ∩ U_a × U_b \in (ω \otimes τ)(U_a × U_b) \). Again by the distributive property, \( (∪ D_i \times G_i) ∩ U_a × U_b = (∪ D_i \times G_i \cap U_a × U_b) = (∪ D_i \cap U_a × G_i \cap U_b \). But \( U_a × U_b \in U_d × U_g \) implies \( U_a \subset U_d \) and \( U_b \subset U_g \), so each \( D_i \cap U_a \in ωU_a \) and \( G_i \cap U_b \in ωU_b \). Hence, on this basis, \( ω \otimes τ \subset O_{Δ×Γ} \), and its extension to the whole space remains a subsheaf of distributive lattices, and \( Δ \times Γ \) is a derivation.

To see that the projection functions are morphisms, we again only need to check local continuity, as the underlying functions are already order-preserving. If \( C \in ωD \), then we just have to check that \( C×Γ \in (ω \otimes τ)(D×Γ) \), and this is true since \( C \in ωD \) by assumption and \( Γ \in τΓ \) for any derivation. Symmetrically, the projection to \( Γ \) is a morphism.
To complete the proof that this is a product, we need to show for any \( t : \Pi \to \Delta \) and \( s : \Pi \to \Gamma \), there is a unique morphism \( u : \Pi \to \Delta \times \Gamma \) such that \( \pi_\Delta u = t \) and \( \pi_\Gamma u = s \). We define the function \( u : \Pi \to \Delta \times \Gamma \) taking \( p \in \Pi \) to \((tp, sp) \in \Delta \times \Gamma \). It is an order-preserving map since it is order-preserving in each coordinate; we only have to show that it is locally continuous.

We show that it is locally continuous on the stalks, which will imply local continuity on all opens. Take any \( \bigcup_{i \in I} D_i \times G_i \in (\omega \otimes \tau)(U_d \times U_g) \). We first simplify further to the case \( D \times G \in (\omega \otimes \tau)(U_d \times U_g) \). \( u^{-1}(U_d \times U_g) = \{ p \in \Pi \mid t(p) \in U_d \text{ and } s(p) \in U_g \} = t^{-1}U_d \cap s^{-1}U_g \). By assumption that \( t \) and \( s \) are morphisms, \( t^{-1}D \in \zeta^{-1}U_d \) and \( s^{-1}G \in \zeta^{-1}U_g \), where \( \zeta \) is the local topology on \( \Pi \). By restriction, \( t^{-1}D \cap s^{-1}G \in \zeta(t^{-1}U_d \cap s^{-1}U_g) \). For any union, since each of the inverse images of \( D_i \times G_i \) are in \( \zeta(U_d \times U_g) \), their union must be, since \( \zeta(U_d \times U_g) \) is closed under unions. This shows local continuity on the stalks of \( \Delta \times \Gamma \). This implies that local continuity extends to all opens by taking unions of matching families on the stalks.

**Lemma 11.5.** Let \( a, b : \Delta \to \Gamma \) be a pair of morphisms. Their equalizer can be computed as the underlying equalizer of sets \( S = \{ x \in \Delta \mid a(x) = b(x) \} \) given the subspace ordering together with the local topology \( \omega_S \) determined by stalks as follows. For any point \( x \in S \), we define \( \omega_S(S_x) = \{ A \cap S \mid A \in \omega_U \} \), where \( S_x = \{ y \in S \mid x \leq y \} \) and \( U_x = \{ y \in \Delta \mid x \leq y \} \).

**Proof.** \( S \) is clearly a partial order, and \( S \toplus \Delta \) an order-preserving function.

We first show that \( \omega_S(S_x) \subset O_S(S_x) \) is a sublattice. Let \( A \cap S \) and \( B \cap S \) be in \( \omega_S(S_x) \) such that \( A, B \in \omega_U \). Then \( (A \cap S) \cap (B \cap S) = (A \cap B) \cap S \) and \( (A \cap S) \cup (B \cap S) = (A \cup B) \cap S \) are in \( \omega_S(S_x) \) since \( A \cap B \) and \( A \cup B \) are in \( \omega_U \). To see that \( \omega_S \) is a presheaf on the basis of sets of the form \( S_x \), we must check that for \( A \cap S \in \omega_S(S_x) \) and \( x \leq y \) that \( A \cap S \cap S_y \in \omega_S(S_y) \). We have a restriction in \( \Delta \omega_U \to \omega_U \), taking \( A \) to \( A \cap U_y \in \omega_U \). Then \( A \cap U_y \cap S \in \omega_S(S_y) \). But \( A \cap S \cap S_y = A \cap S_y = A \cap U_y \cap S \). This is then a presheaf of distributive lattices on \( O(S) \), and its sheafification \( \omega_S : O(S)^{op} \to \text{F DL} \) is a subsheaf of \( O_S \). So \( S \) is a derivation.

To show that the inclusion \( m \) is a morphism, we must only check that \( V \in \omega_U \) implies \( (V \cap S) \in \omega_S(U \cap S) \). It is sufficient to find a matching family on a cover of \( U \cap S \) in \( S \) whose union in \( V \cap S \). We cover \( U \cap S \) with opens \( S_x \) such that \( x \in U \cap S \). For each \( x \in U \cap S \), we have a \( x \in U \), and we restrict \( V \) to \( U_x \) as \( V \cap U_x \in \omega_U \). We then obtain \( V \cap U_x \cap S \in \omega_S(S_x) \). Since the \( V \cap U_x \) are a matching family, so are \( V \cap U_x \cap S \). This matching family corresponds to an element \( \bigcup_{x \in U \cap S} V \cap U_x \cap S \). It is clear that this union is contained in \( V \cap S \). Conversely, for any \( y \in V \cap S \), we have \( y \in U \cap S \) since \( y \in V \cap U \). So \( y \in V \cap U_y \cap S \), and hence \( V \cap S = \bigcup_{x \in U \cap S} V \cap U_x \cap S \).

We must now show universality. Let \( h : \Xi \to \Delta \) be any morphism such that \( ah = bh \). \( h \) must factor through \( S \) as a function, since for all elements in the image \( a(h(x)) = b(h(x)) \) so \( h(x) \in S \). We call the function \( u : |\Xi| \to S \) and show it underlies a morphism. Since
$h$ is a morphism, it must be order-preserving, and hence $u$ must be order-preserving since $S$ is a subspace as a partial order. We must then just have to show that if $W \subseteq \omega_S(U \cap S)$, then $u^{-1}(W) \subseteq \zeta u^{-1}(U \cap S)$, where $\zeta$ is the local topology on $\Xi$. We again reduce to the case of stalks. Let $x \in S$ be any point. Since $h$ is a morphism, we know that for any $A \subseteq \omega_U$, we have $h^{-1}A \subseteq \zeta(h^{-1}U)$. $h^{-1}U = \{z \in \Xi \mid h(z) \in U\}$ in general, but since the image of $h$ is contained in $S$, then $h^{-1}(U \cap S) = \{z \in \Xi \mid h(z) \in U$ and $S\} = h^{-1}U$. Since every element of $\omega_S(U_x)$ comes from some $A \subseteq \omega_U$ restricted to $S$, we have that for any $A \cap S \in \omega_S(S_x)$ that $u^{-1}(A \cap S) \subseteq \zeta u^{-1}(S_x)$ since this is just the statement that $A \subseteq \omega U$ goes to $u^{-1}(A \cap S) = h^{-1}A \subseteq \zeta(h^{-1}U_x) = \zeta(u^{-1}(U_x \cap S))$. For the more general case of $W \subseteq \omega(U \cap S)$, we can take a cover of $U \cap S$ by $S_x$ such that $x \in U \cap S$ and reconstruct $u^{-1}W$ from the inverse images of $W \cap S_x$, since unions are preserved under inverse images of functions.

\[\text{Claim 14.} \quad \text{Let } \Delta \text{ be a derivation and } S \subseteq |\Delta| \text{ any subset. We turn } S \text{ into a partial order by considering it a subspace of } \text{pt}(\Delta). \text{ We then get a morphism of partial orders } S \hookrightarrow \text{pt}(\Delta) \text{ which extends to a morphism of derivations } S \hookrightarrow \text{pt}(\Delta) \to \Delta. \text{ Call this inclusion } i. \text{ The pushout of } i \text{ along itself exists, and the induced morphisms } s, t : \Delta \rightrightarrows \Delta + S \Delta \text{ have the property that } s(d) = t(d) \text{ if and only if } d \in S.\]

\[\text{Proof.} \quad \text{By Claim 13, if } \Delta + S \Delta \text{ is the derivational pushout of } i \text{ along itself, then } \text{pt}(\Delta + S \Delta) \text{ is the pushout of pt}(i) \text{ along itself in } \text{FPos}.\]

A pushout in $\text{FPos}$ can be computed as a pushout of preorders (partial orders without an antisymmetry requirement), followed by soberification, identifying any two elements $x, y$ such that $x \leq y$ and $y \leq x$. In turn, the pushout of preorders can be computed as the pushout of sets, and then endowing this set with the ‘final topology’.

Given morphisms $f : A \to B$ and $g : A \to C$, we first form the set-theoretic pushout $B +_A C$. This can be computed by taking the disjoint union $B + C$, then constructing the following relation on $B + C$. We write $xRy$ if there is an element $a \in A$ such that $f(a) = x$ and $g(a) = y$. We find the smallest equivalence relation containing $R$ and denote it $\sim$. The pushout of sets $B +_A C$ is $B + C$ modulo this equivalence relation, meaning we identify any two points which are equivalent. There is a canonical function $B + C \to B +_A C$ taking each element to its equivalence class. The functions associated to the pushout are $B \hookrightarrow B + C \to B +_A C$ and $C \hookrightarrow B + C \to B +_A C$.

We turn these functions into order-preserving functions by equipping $B +_A C$ with the ‘least preordering possible’. In other words, if $b \leq b'$ in $B$, then $[b] \leq [b']$ in $B +_A C$ and similarly if $c \leq c'$ in $C$ then $[c] \leq [c']$ in $B +_A C$. We then close this relation under transitivity. It is already closed under reflexivity, so this is a preordering on $B +_A C$, and the morphisms from $B$ and $C$ are order-preserving.

If $i : S \to X$ is a subspace inclusion, then the pushout of sets $X +_S X$ is effectively the set $X$ ‘doubled’ where we have identified points from $S$ in each copy. As a set pushout, the functions $s, t : X \rightrightarrows X +_S X$ have the property that $s(x) = t(x)$ iff $x \in S$. To complete
the theorem, we must simply show that the induced preordering on $X + S X$ is already antisymmetric.

If we look in each ‘copy’ of $X$ in $X + S X$, the ordering looks just as it does on $X$. We transitively add relationships $a \leq b$ between elements in different copies of $X - S$ if there is an element $s \in S$ such that $a \leq s \leq b$. If $a \leq b$ and $b \leq a$ in a single copy of $X$, then $a = b$ by virtue of antisymmetry there. If $a$ and $b$ are in different copies of $X - S$ such that $a \leq s \leq b$, then we must have elements $s, s' \in S$ such that $a \leq s \leq b$ and $b \leq s' \leq a$. Viewing all of these as elements of $X$, this means that $b$ and $a$ are equally ordered, so the only elements between them are representatives of $a = b$. But by assumption $a, b \not\in S$, a contradiction. So distinct copies of a point in $X - S$ cannot be ordered as such, and the ordering is antisymmetric.

Claim 22. Each forest of terms $\Delta$ factors into a coproduct of open subderivations $\Delta/U_{r_i}$, such that each $\Delta/U_{r_i}$ is a tree of terms and the inclusion $\Delta/U_{r_i} \hookrightarrow \Delta$ is a constituent-preserving embedding. Here, the $r_i$ are the minimal terms of $\Delta$.

Proof. First note that each $x \in \Delta$ is greater than some minimal term $r_i \leq x$. This follows directly from separation and the minimality of $r_i$. Now note that for any minimal terms $t$ and $s$ of $\Delta$, either $U_t \cap U_s = \emptyset$ or $t = s$. That is, the open sets associated to distinct minimal terms are disjoint. So each point is actually greater than exactly one minimal term. Together, this shows that $\coprod_{i \in I} \Delta/U_{r_i} = \Delta$, where $\{r_i\}_{i \in I}$ is the set of minimal terms of $\Delta$.

Clearly, $\Delta/U_{r_i}$ has unique minimal term $r_i$, so we only have to show that $\Delta/U_{r_i}$ is a forest of terms (which will imply that it is a tree of terms). Note that for any open $V \subset \Delta/U_{r_i}$, $\omega_{U_{r_i}} V = \omega V$. This actually implies that the inclusion $\Delta/U_{r_i} \hookrightarrow \Delta$ is a constituent-preserving embedding. The openness follows automatically. The coherence follows since if $a \leq b$ in $\Delta/U_{r_i}$, then $U_b \in \omega_{U_{r_i}} U_a$ implies $U_b \in \omega U_a$. Let $t \in U_{r_i}$ be a term in $\Delta/U_{r_i}$. By way of contradiction, assume that there is some $x \preceq t$ such that $x \neq t$ in $\Delta$. $x$ must belong to $U_{r_i}$, since it must be in some minimal term and the open sets associated to the terms are disjoint. By Claim 15, $x \preceq t$ in $\Delta/U_{r_i}$, contradicting that $t$ is a term.

By being a constituent-preserving subderivation of $\Delta$, each $\Delta/U_{r_i}$ must be transitive, and $R$ is an equivalence relation on it. To see that $R$ is regular, note that each block is totally contained inside a single $U_{r_i}$, so there is an ordering between blocks in each subderivation iff there is an ordering between them in $\Delta$. Similarly, there is a blockwise ordering between two elements in each subderivation iff there is a blockwise ordering between them in $\Delta$ for the same reason.

Claim 28. Let $\phi : \Delta \rightarrow \Gamma$ be a constituent-preserving map between separated derivations. Then the induced $q_\phi : T(\Delta) \rightarrow T(\Gamma)$ is open.

Proof. The following are equivalent:
(1) \( f : X \to Y \) is an open map of finite preorders

(2) For every \( x \in X \), if \( y \geq f(x) \in Y \), then there exists some \( z \in X \) such that \( f(z) = y \) and \( z \geq x \).

If \( f \) is an open map, and \( U \subset X \) an open subset, then \( f(U) \subset Y \) is open. By finiteness, \( U \) is generated by some finite set \( \{ x_1, \ldots, x_n \} \subset X \) such that for all \( x \in U \), \( x \geq x_i \) for some \( x_i \), a generator. Then for any \( f(x_i) \leq y \), by openness, some element of \( U \) must map to \( y \). Since this holds for all open sets, and for any \( x \in X \) the set \( \{ x' \in X : x' \geq x \} \) is open, (1) \( \Rightarrow \) (2). If (2) holds, then for any \( U \) we can use (2) on the generators to produce the necessary elements of \( U \) to hit all points greater than some element in the image \( f(U) \), so that this map is surjective, and then \( f \) is open, so that (2) \( \Rightarrow \) (1).

Now suppose that \( a \in pt(\Delta) \) has equivalence class \( [a] \) in \( \Delta \) and \( \phi : \Delta \to \Gamma \) is a term-preserving, term-coherent open map, which induces a map \( f \) between the quotients of points \( f : \Delta \to \Gamma \), taking \( f([a]) = [b] \). To show that \( f \) is open, for any \([c]\) such that \([b] \leq [c]\) in \( \Gamma \), we must find some \([z]\) such that \([z] \geq [a]\) in \( \Delta \) and \([z]\) maps to \([c]\).

If \( f([a]) = [b] \), that means there is some point \( x \) such that \( x \sim a \), i.e. \( U_x, U_a \leftrightarrow U_{t_i} \), such that \( pt(\phi)(x) = y \) and \( y \sim b \). In particular, by term-separation, there is a unique term \( t \) such that \( t \sim a \), so that \( t \) itself must map into the unique term containing \( b \), call it \( s \), since by term-coherence, \( t \) must be taken to a term, and the only term similar to \( s \) is \( s \).

Now, if we have \([c] \geq [b]\), by the regularity of the partition, we must have that \( c \) is blockwise greater than \( b \) in \( \Gamma \). [Codara, 2009] Choose some sequence satisfying the blockwise ordering requirements, so that \( c = x_0 \sim y_0 \geq x_1 \sim y_1 \geq \ldots \geq x_n \sim y_n = b \). For each step \( x_i \sim y_i \), by term-separation, each of \( x_i \) and \( y_i \) are uniquely similar to some term \( t_i \) of \( \Gamma \). We construct a blockwise path in \( pt(\Delta) \) mapping to this blockwise path by a back-and-forth method. Start with \( b = y_n \). Since we have established that there is some point \( x \) mapping to \( y \sim b \) under \( pt(\phi) \), there is also a term containing \( x \), namely \( t \), mapping to the term \( t_n = s \). In the blockwise sequence, we assume that there is some step \( x_n \leq y_{n-1} \), and we know that \( t_n \leq x_n \leq y_{n-1} \). But since \( t \) hits the point \( t_n \), and \( pt(\phi) \) is open, there must be some point \( t \leq y_{n-1} \) of \( pt(\Delta) \) which hits \( y_{n-1} \). Since \( t \sim a \), \( a \leq y_{n-1} \) are blockwise ordered in \( pt(\Delta) \). Now, \( y_{n-1} \) is contained in some unique term \( t_{n-1} \), and \( y_{n-1} \) is contained in some unique term \( t_{n-1} \) by term-separation. Since \( U_{t_{n-1}} \) contains \( U_{y_{n-1}} \) openly, and this map is term-coherent, \( t_{n-1} \) must be taken to the same partition as \( y_{n-1} \), and since \( pt(\phi) \) is term-preserving it must be taken to \( t_{n-1} \).

Now, for the inductive step, we have assumed that there is some \( x_{n-1} \sim y_{n-1} \) and \( x_{n-1} \leq y_{n-2} \). But by \( x_{n-1} \sim y_{n-1} \), we must have that \( t_{n-1} \leq x_{n-1} \), so \( t_{n-1} \leq x_{n-1} \leq y_{n-2} \). But \( t_{n-1} \) hits \( t_{n-1} \), and by openness, there is some point \( t_{n-1} \leq y_{n-2} \) of \( pt(\Delta) \) such that \( y_{n-2} \) hits \( y_{n-2} \). But since \( t_{n-1} \leq y_{n-2} \) and \( t_{n-1} \sim y_{n-1} \), we have a blockwise ordering \( y_{n-1} \leq y_{n-2} \), and hence a blockwise ordering \( a \leq y_{n-1} \sim y_{n-2} \).

Continuing, we obtain a blockwise ordering \( a \leq y_{n-1} \leq y_{n-2} \ldots \leq y_0 \) in \( pt(\Delta) \), where \( y_i \) hits \( y_i \) in the sequence. In particular, this final \( y_0 \) hits \( y_0 \sim c \). Since equivalence classes
in $\overline{A}$ are ordered iff elements in their class are blockwise ordered, we have that $[a] \leq [y_0]$. Since $y_0$ hits $y_0 \sim c$, $[y_0]$ must be mapped to $[c]$, so we can always produce the necessarily element, and the map is open.

Claim 31. Let $X$ be a finite partially ordered tree with open set lattice $O(X)$. Let $K \subset X$ be any constituent of $X$. Then the connected components of $\neg K$ are exactly the constituents c-commanding $K$.

Proof. First note that $\neg K = X - \text{cl}(K)$, where $\text{cl}(S)$ is the smallest closed set containing $S$. A closed set is any set which is the complement of an open set. But $\text{cl}(K)$ is just $K$ along with any node dominating any node in $K$. By the assumption that $K$ is a constituent, this is just $K$ along with the nodes dominating $k$, the root of the constituent $K$. Then, $\neg K$ is the set of nodes which are not in $K$ nor dominate $K$. Let $V$ be a connected component of $\neg K$. Then, take any node in $X$ properly dominating $V$, i.e. any node $y$ such that for all $v \in V$, $x < v$. This node $x$ must lie outside of $\neg K$; if not, then $x \in \neg K$, and hence by openness, the upset of $x$ contains $V$ and is connected, contradicting $V$'s maximality. Since $x < V$, we may assume also that $x \notin K$, since $K \cap V = \emptyset$. Then, $x$ must lie in $\text{cl}(K) - K$, the set of nodes irreflexively dominating $K$, also known as the boundary of $K$. That is to say, if $V$ is a connected component of $\neg K$, then for any node $x$ irreflexively dominating $V$, $x$ irreflexively dominates $K$.

Claim 33. Let $f : X \rightarrow Y$ be a constituent-preserving map between finite partially ordered trees. Let $K \subset Y$ be any constituent of $Y$ and let $V \subset Y$ be any constituent c-commanding $Y$. Then each component of $f^{-1}V$ c-commands some component of $f^{-1}K$.

Proof. Take a constituent $K \subset Y$ and $V \subset Y$ c-commanding $K \subset Y$. Choose a connected component $L \subset f^{-1}(V)$. Choose the greatest $x < L$, in the sense already established (i.e. the unique mother of the minimal element of $L$), which exists since $X$ is a tree. By monotonicity, $f(x) < f(l)$ for all $l \in L$, and $f(l) \in V$. In fact, $f(x) < V$, since $V$ has a root $r$ and since the points above $f(l)$ form a linear order (by assumption that $Y$ is a tree), $r$ and $f(x)$ must be linearly ordered. If $r \leq f(x)$, then $f(x) \in V$; but in this case, $x < l$ both map into $V$, and hence the upset of $x$ would be in the preimage of $V$, contradicting the maximality of $L$ as a connected component of $f^{-1}V$. So $f(x) < r$, hence $f(x) < V$. Since $V$ c-commands $K$ by hypothesis, $f(x) < K$. Let $Z$ be the upset of $x$ in $X$. By the assumption that $f$ is open, $f(Z)$ is the upset of $f(x)$, which contains $K$. Now, choose any point $m \in K$. By the assumption of openness, there must be some point $z \in Z$ such that $f(z) = m$. This $z$ belongs to some connected component of $f^{-1}K$, and suppose this component is dominated by $p$ (note that $p$ cannot be smaller than $x$ by monotonicity). Then $x < p \leq z$, so in particular $x < K_p$, where $K_p$ is the upset of $p$ in $X$. Then, for every node dominating $L$, this node will dominate $K_p$. In particular, for every connected component of $f^{-1}V$, we can produce a connected component of $f^{-1}K$ which
it c-commands. In other words, every connected component of $f^{-1}V$ c-commands some component of $f^{-1}K$.

References


