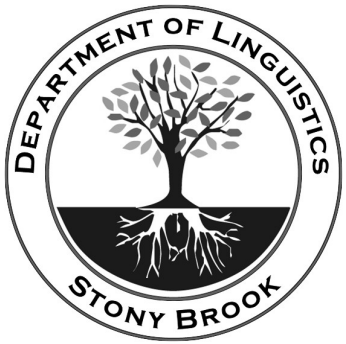
Learning Constraints over Representations of Your Own Choosing

Jeffrey Heinz

(joint work with Dakotah Lambert and Jon Rawski)



Stony Brook University

MIT

April 21-22, 2021

This Minicourse

1. Finite Model Theory provides a unified language for representing various linguistic structures.
2. Factors are "parts" of structures.
3. Grammars detail how a structure's well-formedness is based on its parts.
4. Too many distinctions hinders learning.
   $\Leftrightarrow$ Fewer distinction facilitates learning.
5. The factors are *partially ordered*. This *structure* helps a lot!
6. Applications to phonotactic learning
7. Inductive vs. Abductive Learning

Part I

Main Idea

Compositionality

Hypothesis: The well-formedness of a complex structure depends on its parts.

As a first approximation:

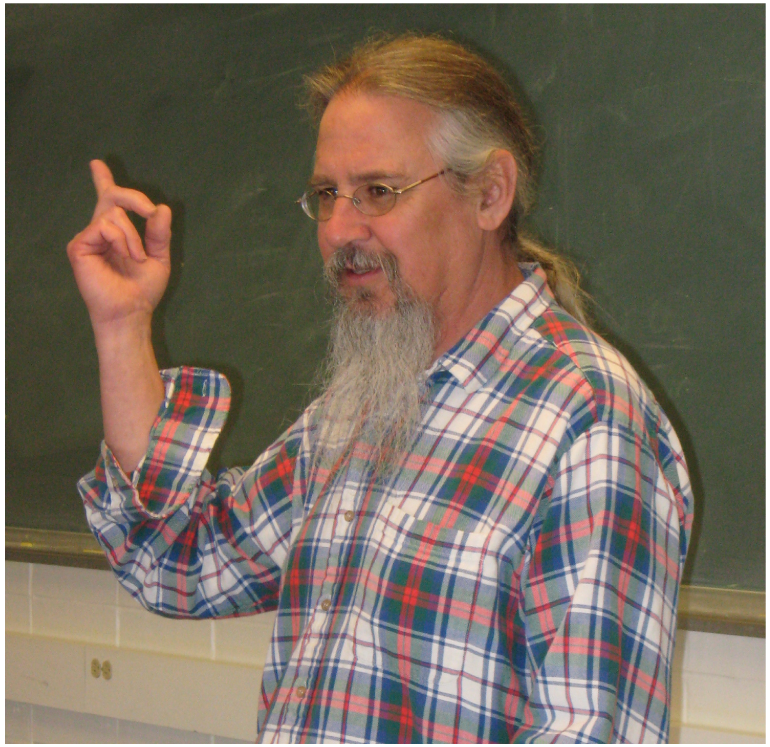
- We use sequences (strings) to model word structure.
- We use trees to model sentence structure.

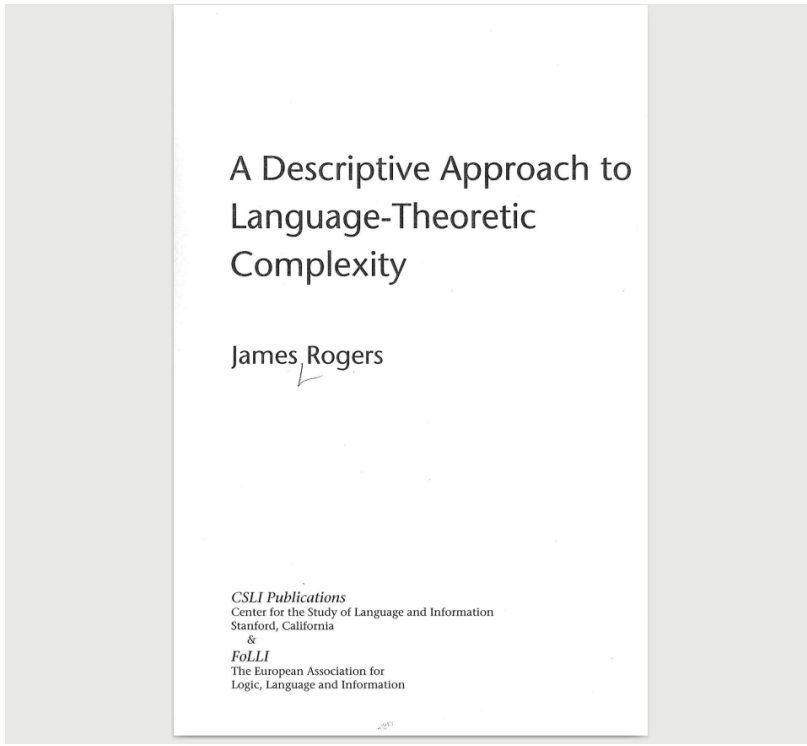
How can these notions be formalized? What are the consequences?

Managing Expectations

- The examples will mostly come from phonology, but the ideas are not specific to phonology.
- If the representations cannot be observed directly (like syntactic trees) they will first have to be inferred/parsed/obtained.

Model-theoretic Representations for Linguistic Structures





The 2017 SIGMOL S.-Y. Kuroda Prize is awarded to James Rogers (Earlham College). James Rogers's 1998 book, "A Descriptive Approach to Language-Theoretic Complexity," was the first comprehensive work to apply monadic second-order logic to the analysis of linguistic theories...`http://molweb.org/mol/award-2017.html`

Model Theory for Linguistics

Pullum 2007:7, "The Evolution of Model-Theoretic Frameworks in Linguistics"

> *I have tried to point out in the brief historical review above, however, that the flowering of this work that began in the middle 1990s was related to seeds planted some thirty years before. They were planted in stony ground, only inexpertly tended, and inadequately watered, but they were planted nonetheless. There is now an increasingly luxuriant garden to explore.*

Model Theoretic Representations for Strings

$$\mathcal{M}(w) := \left\langle \mathcal{D}_w; R_i, \sigma \right\rangle_{\sigma \in \Sigma}$$

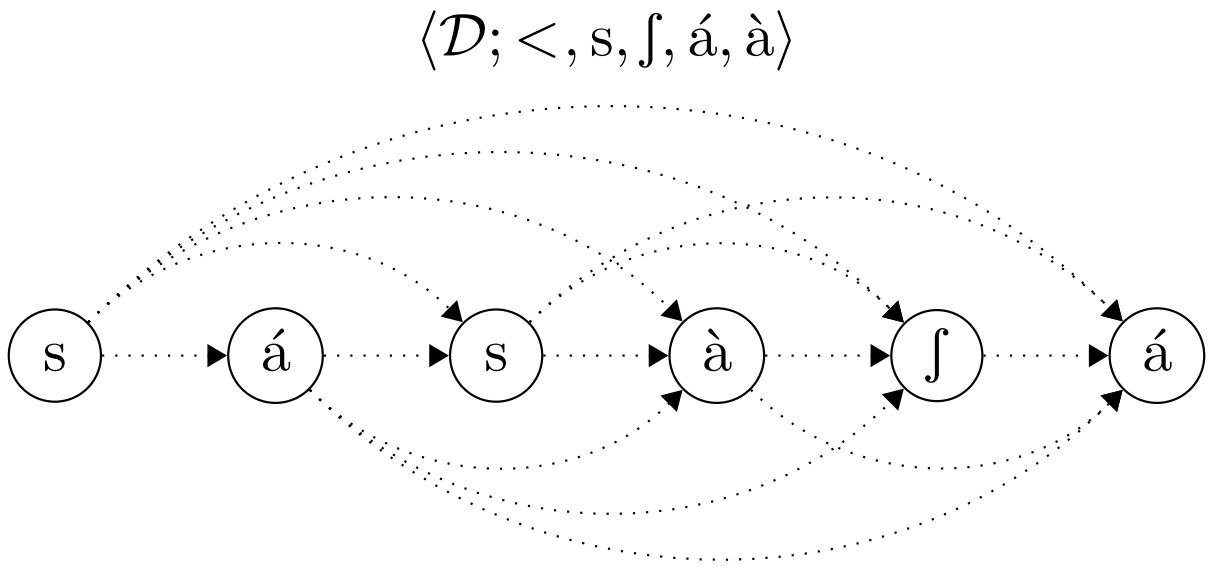
where

- $\mathcal{D}_w$ is isomorphic to an initial segment $\langle 1, \ldots, |w| \rangle$ of the nonzero natural numbers and represents the positions of w,
- each σ is a unary relation that holds for all and only those positions at which σ occurs, and
- the remaining R_i are the other salient relations, which are typically used to define how the positions are ordered

Word Models: Conventional Successor

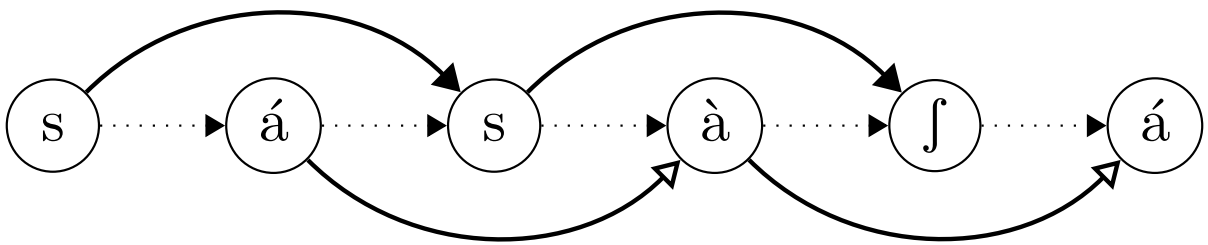
$$\langle \mathcal{D}; \lhd, \text{s}, \int, \acute{\text{a}}, \grave{\text{a}} \rangle$$

Word Models: Conventional Precedence



Word Models: Tier-Based Successor

$$\langle \mathcal{D}; \lhd, \lhd^{\{\text{s},\int\}}, \lhd^{\{\text{à},\text{á}\}}, \text{s}, \int, \text{à}, \text{á} \rangle$$



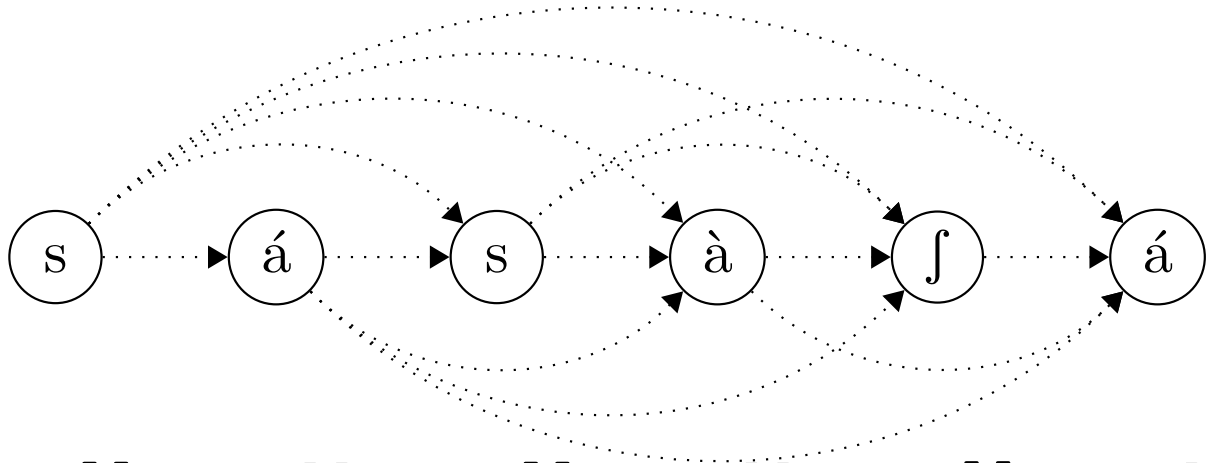
Word Models: Feature-Based Successor

$$\langle \mathcal{D}; \lhd, +\texttt{syll}, -\texttt{syll}, +\texttt{ant}, -\texttt{ant}, \texttt{H}, \texttt{L} \rangle$$

−syll	+syll	−syll	+syll	−syll	+syll
+ant	H	+ant	L	−ant	H

Word Models: Feature-Based Precedence

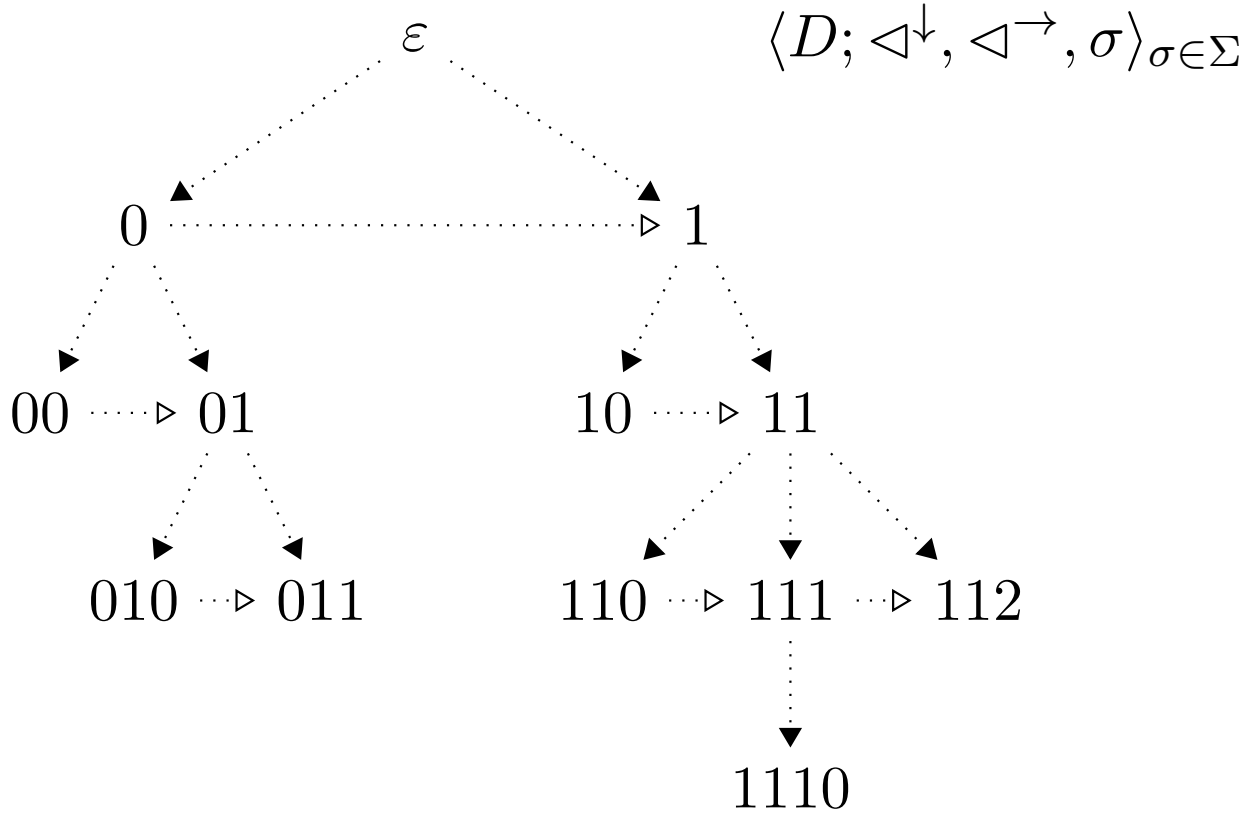
$\langle \mathcal{D}; <, +\texttt{syll}, -\texttt{syll}, +\texttt{ant}, -\texttt{ant}, \texttt{H}, \texttt{L} \rangle$



Other Word Models

1. Autosegmental structures (Jardine 2016, 2017)
2. Syllable structures (Strother-Garcia 2018, 2019)
3. Prosodic structures (Dolatian 2020)
4. Morphological structures (Dolatian 2020)
5. Articulatory Phonology structures (Chadwick 2020)

Tree Models: Conventional Dominance & Successor



For models of trees, the domain is isomorphic to the Gorn addresses (shown) of the nodes in the tree.

(Rogers 1998, Frank and Vijay-Shanker 2001)

Summary

Finite model theory provides a unified ontology and a vocabulary for representing many kinds of objects, by considering them as relational structures (see Libkin 2004 for a thorough introduction). This allows flexible but precise definitions of the structural information in an object, by explicitly defining its parts and the relations between them. This makes model-theoretic representations a powerful tool for analyzing the information characterizing a certain structure.

(Lambert, Heinz and Rawski, under review)

Part II

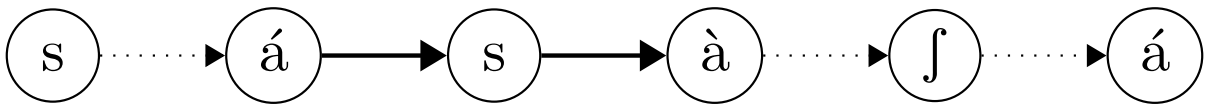
Factoring Structures

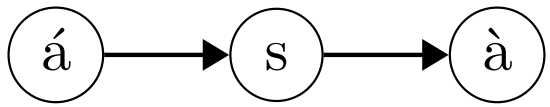
Factors

- We call the "parts" of a relational structure its "factors."
- Informally, factors are connected pieces of structure (so domain elements *connected* by the relations).
- The *size* of a factor is its number of elements.
- If A is a factor of B, we write $A \leq B$.

Some Example Factors

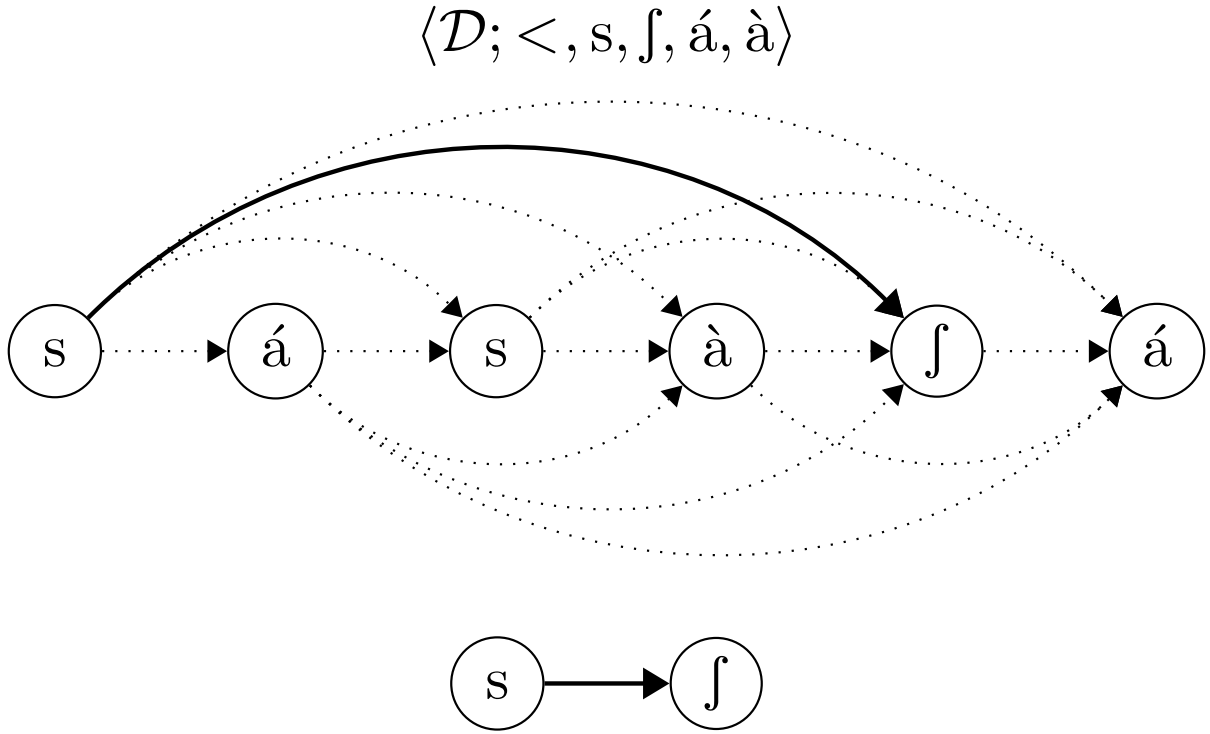
$$\langle \mathcal{D}; \lhd, \mathrm{s}, \int, \acute{\mathrm{a}}, \grave{\mathrm{a}} \rangle$$





Factors are substrings!

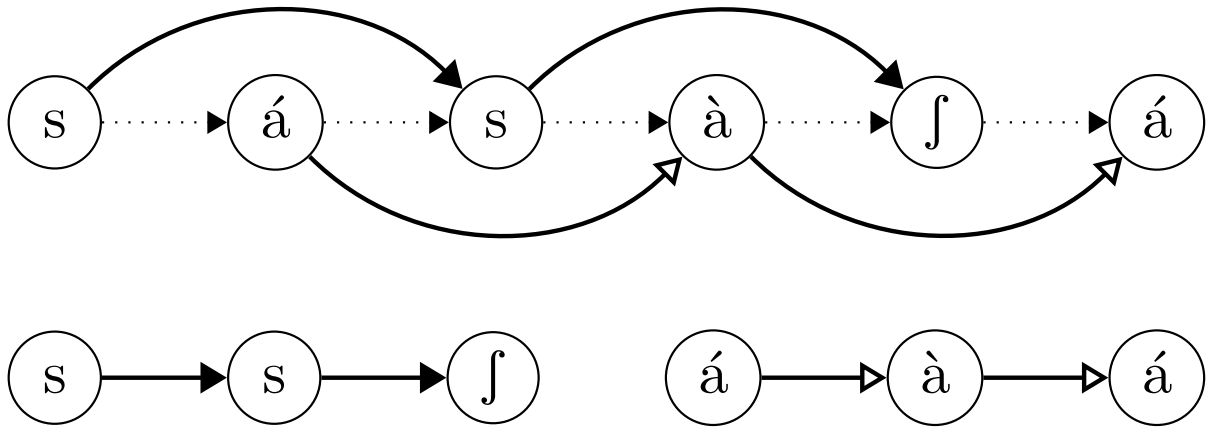
Some Example Factors



Factors are subsequences!

Some Example Factors

$$\langle \mathcal{D}; \lhd, \lhd^{\{\text{s},\int\}}, \lhd^{\{\text{à},\text{á}\}}, \text{s}, \int, \text{à}, \text{á} \rangle$$

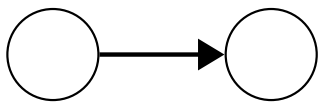


Factors are substrings on tiers!

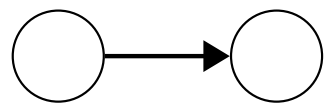
Some Example Factors

$\langle \mathcal{D}; \lhd, +\texttt{syll}, -\texttt{syll}, +\texttt{ant}, -\texttt{ant}, \texttt{H}, \texttt{L} \rangle$

−syll	+syll	−syll	+syll	−syll	+syll
+ant	H	+ant	L	−ant	H



−syll +syll

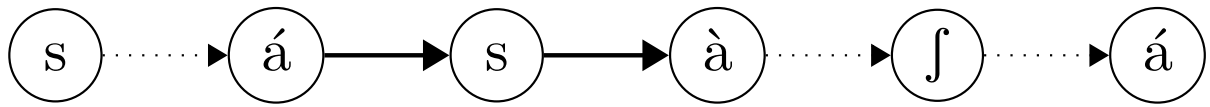


+syll −syll

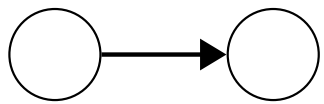
Factors are substrings of feature bundles!

Some Example Factors

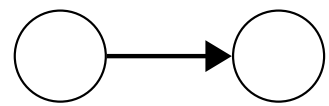
$\langle \mathcal{D}; \lhd, +\texttt{syll}, -\texttt{syll}, +\texttt{ant}, -\texttt{ant}, \texttt{H}, \texttt{L} \rangle$



−syll	+syll	−syll	+syll	−syll	+syll
+ant	H	+ant	L	−ant	H



H +ant

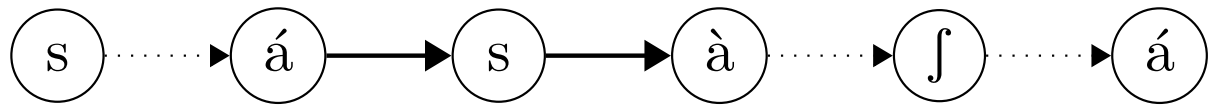


−syll L

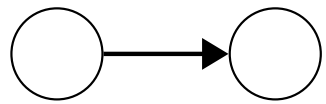
Factors are substrings of feature bundles!

Some Example Factors

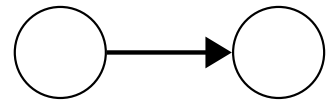
$\langle \mathcal{D}; \lhd, +\texttt{syll}, -\texttt{syll}, +\texttt{ant}, -\texttt{ant}, \texttt{H}, \texttt{L} \rangle$



−syll	+syll	−syll	+syll	−syll	+syll
+ant	H	+ant	L	−ant	H



H



L

Factors are substrings of feature bundles!

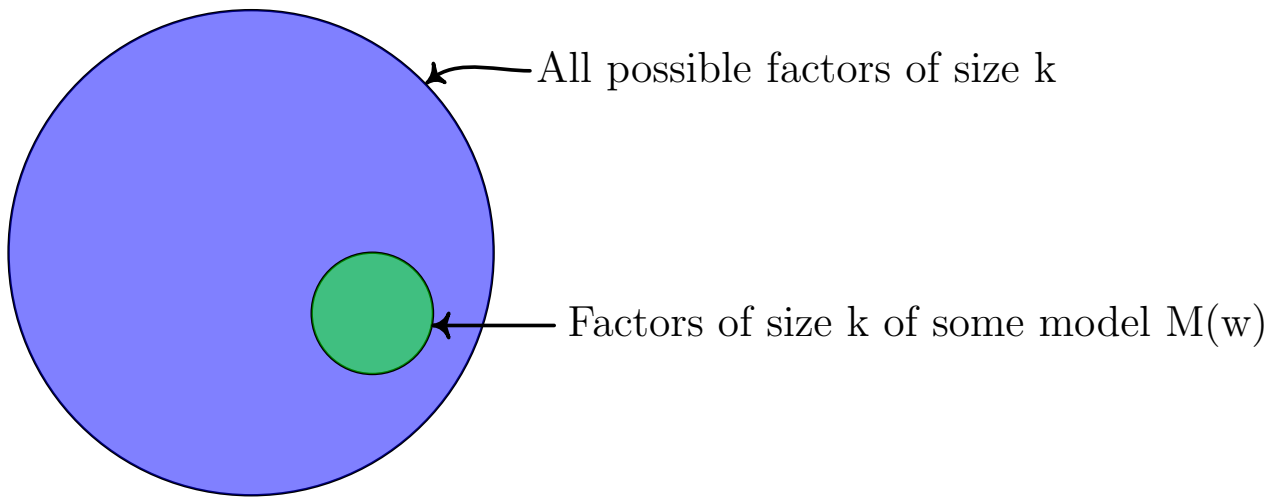
Gathering Factors

Given a model of a word $M(w)$, its factors of size k are finite.

$$F_k(M(w)) := \{m : m \leq M(w), |m| = k\}$$

Even the set of all possible factors of size k is finite.

$$\text{FAC}_k := \bigcup_{w \in \Sigma^*} F_k(M(w))$$



Part III

Some Factor-Based Grammars and Learning Them

Grammar (ideas)

If we have a well-formed structure, we can assume one of the following.

1. One or more of its factors licenses it.
2. None of its factors are ill-formed.
3. The particular combination of its factors licenses the structure.
4. The particular combination of its factors and their counts (up to t) licenses the structure.

- Note (4) subsumes (3) which subsumes both (1) and (2).
- If the factor size is set to k (and for (4), a threshold set to t), each of these ideas leads to "in principle" learnability.

Formal Grammars

If we have a well-formed structure, we can assume one of the following.

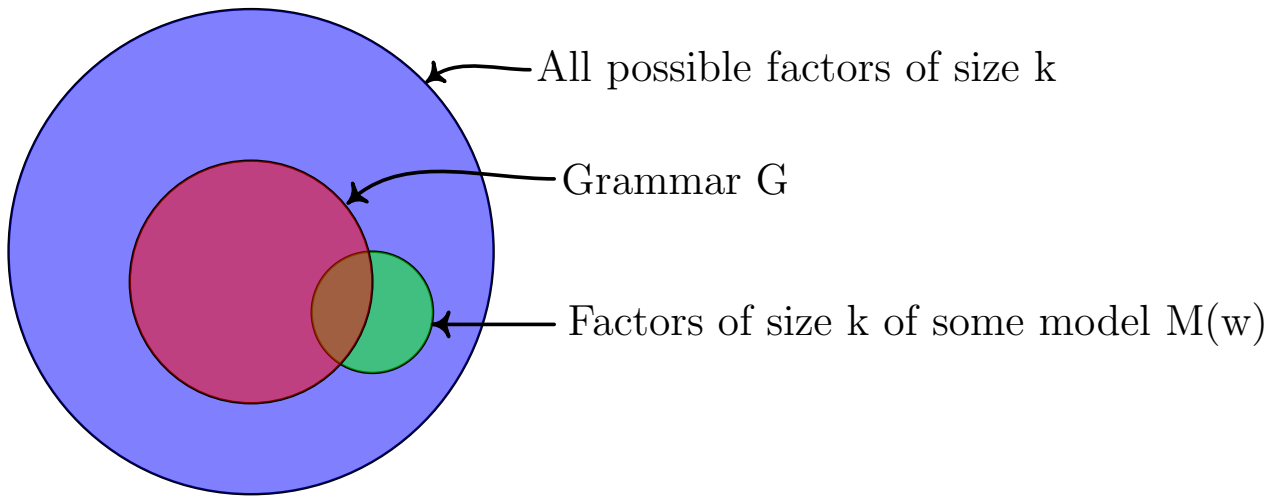
1. G is a set of licensing factors and G recognizes words whose models include at least some factor in G.
2. G is a set of forbidden (marked) factors and G recognizes words whose models' factors are disjoint with G.
3. G is a collection of sets of factors and G recognizes words whose models' set of factors is in G.
4. G is a collection of saturated multisets of factors and G recognizes words whose models' saturated multiset of factors is in G.

- Note (4) subsumes (3) which subsumes both (1) and (2).
- If factor size is set to k (and for (4), a threshold set to t), each of these ideas leads to "in principle" learnability.

Example for (1)

(1) G is a set of licensing factors and G recognizes words whose models include at least some factor in G.

- Disyllabic minimal word condition.
  G = { [+syll][+syll] }
  with feature-based precedence (or tier) model and $k = 2$

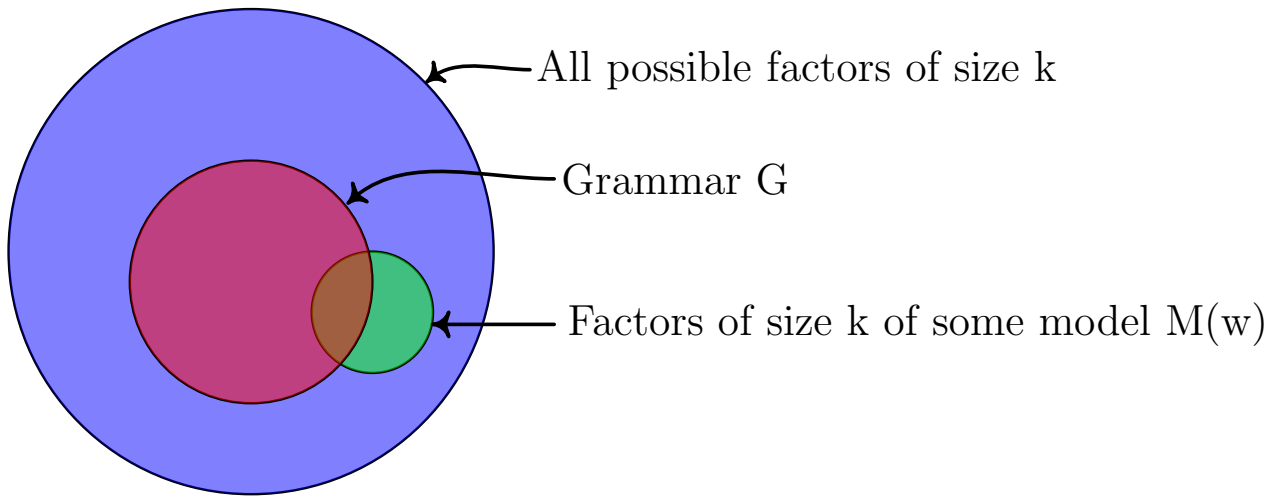


M(w) is accepted.

Example for (2)

(2) G is a set of **forbidden** factors and G recognizes words whose models' factors are disjoint with G.

- CV language
  G = { [+syll][+syll], [-syll][-syll] }
  with feature-based successor model and $k = 2$

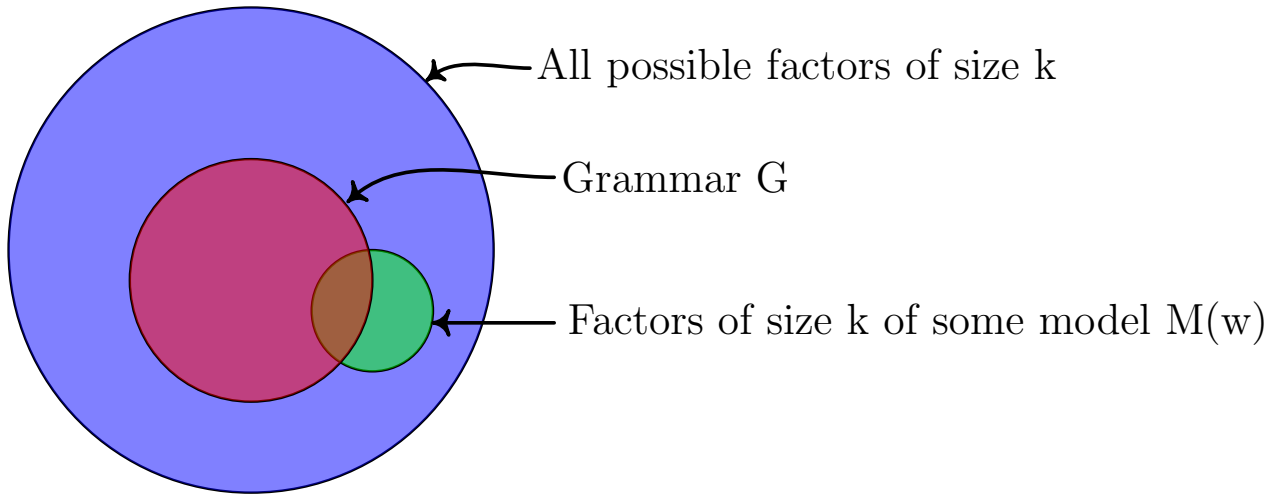


M(w) is rejected.

Example for (2)

(2) G is a set of **forbidden** factors and G recognizes words whose models' factors are disjoint with G.

- CV language
  G = { [+syll][+syll], [-syll][-syll] }
  with feature-based successor model and $k = 2$



M(w) is rejected.

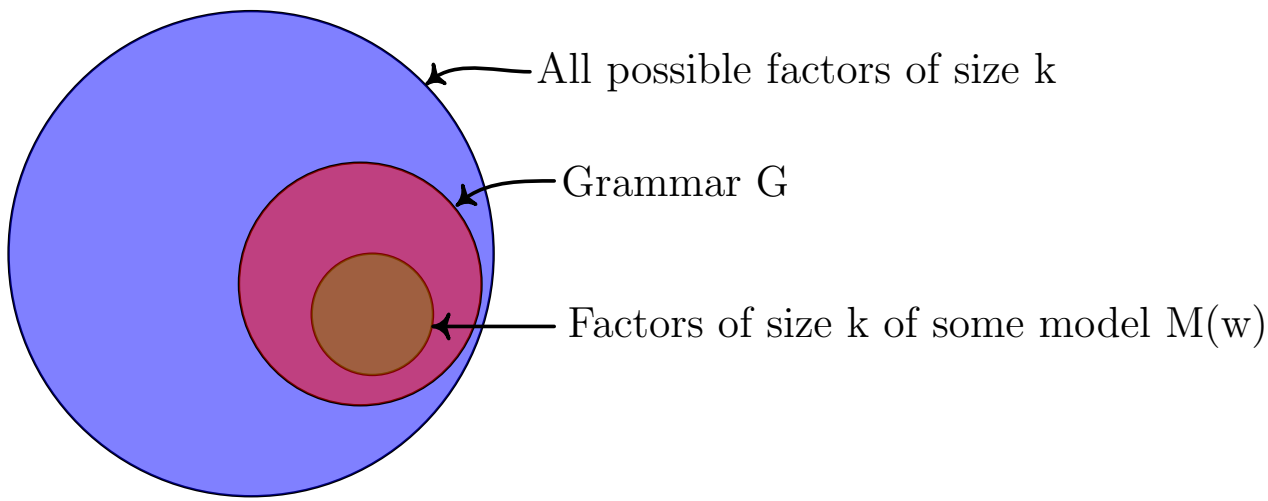
Most markedness constraints in phonology are here!!

Example for (2) continued

An equivalent way of thinking of these constraints is as follows.

(2) G is a set of **permissible** factors and G recognizes words whose models' factors **all belong** to G.

- CV language
  G = { [+syll][-syll], [-syll][+syll] }
  with feature-based successor model and $k = 2$

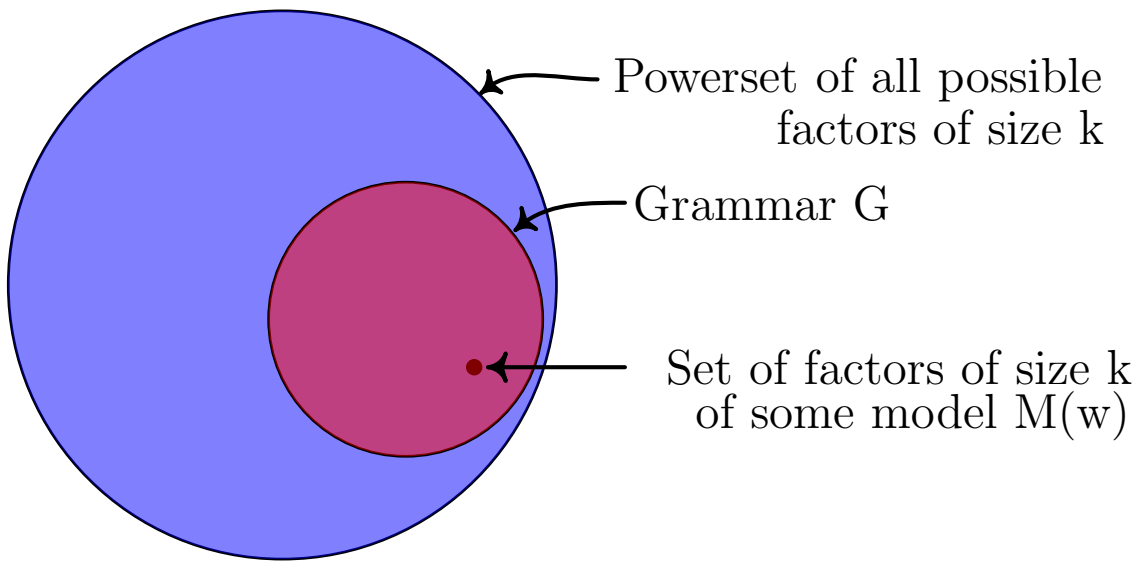


M(w) is accepted.

Example for (3)

(3) G is a collection of sets of factors and G recognizes words whose models' set of factors is in G.

- Words must contain both [s] and [ʃ].
  $\mathrm{G} = \{S : \{\mathrm{s}, \int\} \in S\}$
  with conventional successor model and $k = 1$

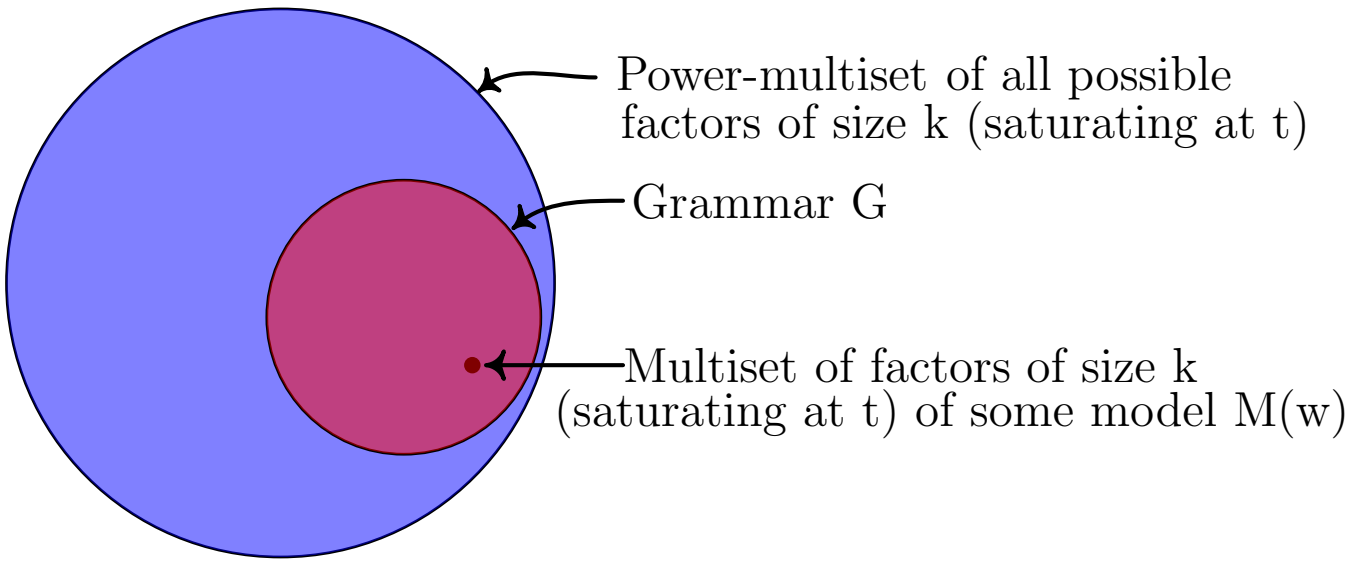


M(w) is accepted.

Example for (4)

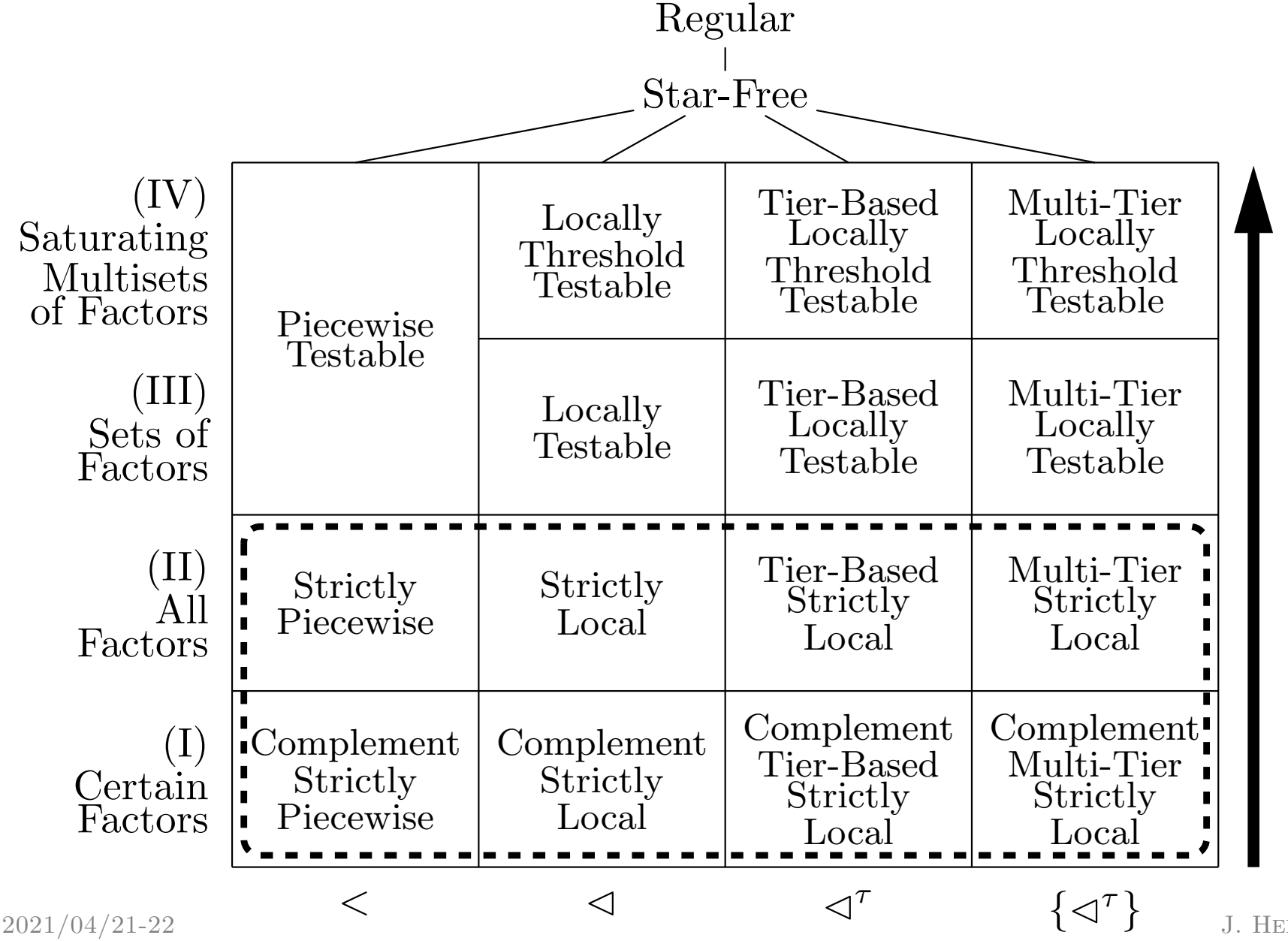
(4) G is a collection multisets of factors and G recognizes words models' multiset of factors is in G.

- Words must contain at least 3 CC substrings.
  G = { S : { ([-syll][-syll], 3) } $\in S$ }
  with feature-based successor model and $k = 2$ and $t = 3$



M(w) is accepted.

Classes of Formal Languages Obtained from Different Models with Different Grammars



Doing Linguistic Typology

Requires two books:

- "encyclopedia of categories"
- "encyclopedia of types"

Wilhelm Von Humboldt

Evidence from Stress Patterns
Rogers and Lambert 2019

1. They consider over 100 distinct stress patterns, expressed as regular grammars, from over 700 languages in the StressTyp2 database.
2. They develop methods to *factor* these grammars into primitive constraints. Virtually all constraints fall into these kinds:
    1. Class I with $\{\lhd\}$ (no LL)
    2. Class II with $\{\lhd\}$ (require $\acute{\text{L}}$)
    3. Class I with $\{<\}$ (no $\acute{\text{H}}\ldots\text{L}$)
    4. Class II with $\{<\}$ (require $\acute{\text{H}}\ldots\text{L}$)
    5. Class I with $\{<, \lhd\}$ (no $\text{H}\ldots\acute{\text{H}}\ltimes$)
3. See also their 2019 MoL paper.

Part IV

Learning these Grammars from Examples

Learning These Grammars from Examples

Learning Algorithms:

- $G_0 := \varnothing$
- $G_{i+1} := G_i \cup f(M(w))$

where

1. $f(M(w)) := M(w)$ iff $M(w) \leq k$ and $\varnothing$ otherwise
2. $f(M(w)) := F_k(M(w))$ *
3. $f(M(w)) := \{F_k(M(w))\}$
4. $f(M(w)) := \wr F_k(M(w)) \int_t$

The notation $\wr \ldots \int_t$ represents a multiset that saturates at a count of t.

*Learns the permissible version of the grammar.

Feasibility: Time and Space Complexity

Conventional model analysis (no features)

- Let n be the size of the input data.
- $|\Sigma|, k, t$ are fixed in advance are so are constants.

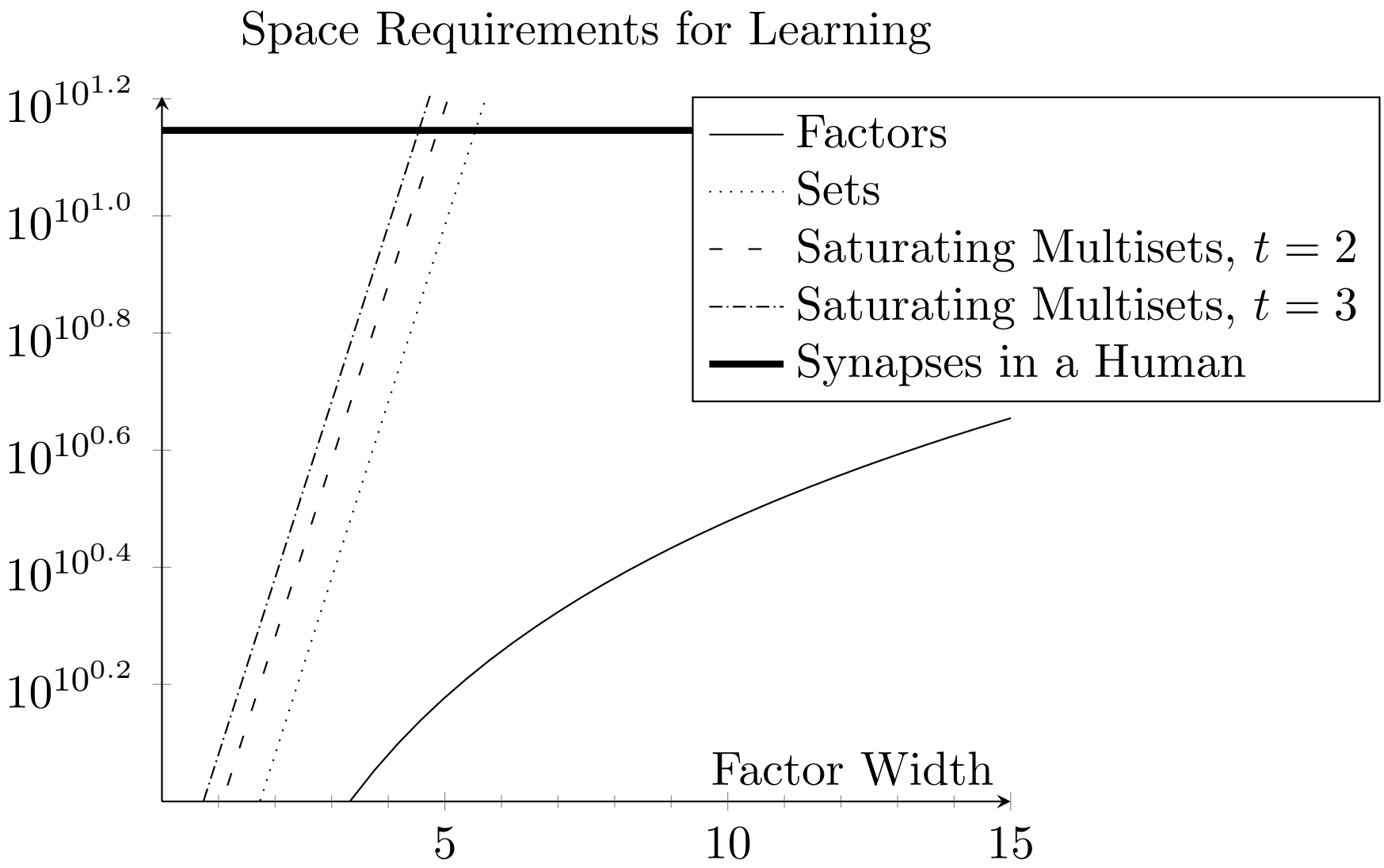
Algorithm	Time***	Space
I	polynomial in n	constant*
II	polynomial in n	constant*
III	polynomial in n	constant**
IV	polynomial in n	constant**

* This constant is singly exponential in k: $\mathcal{O}(|\Sigma|^k)$.

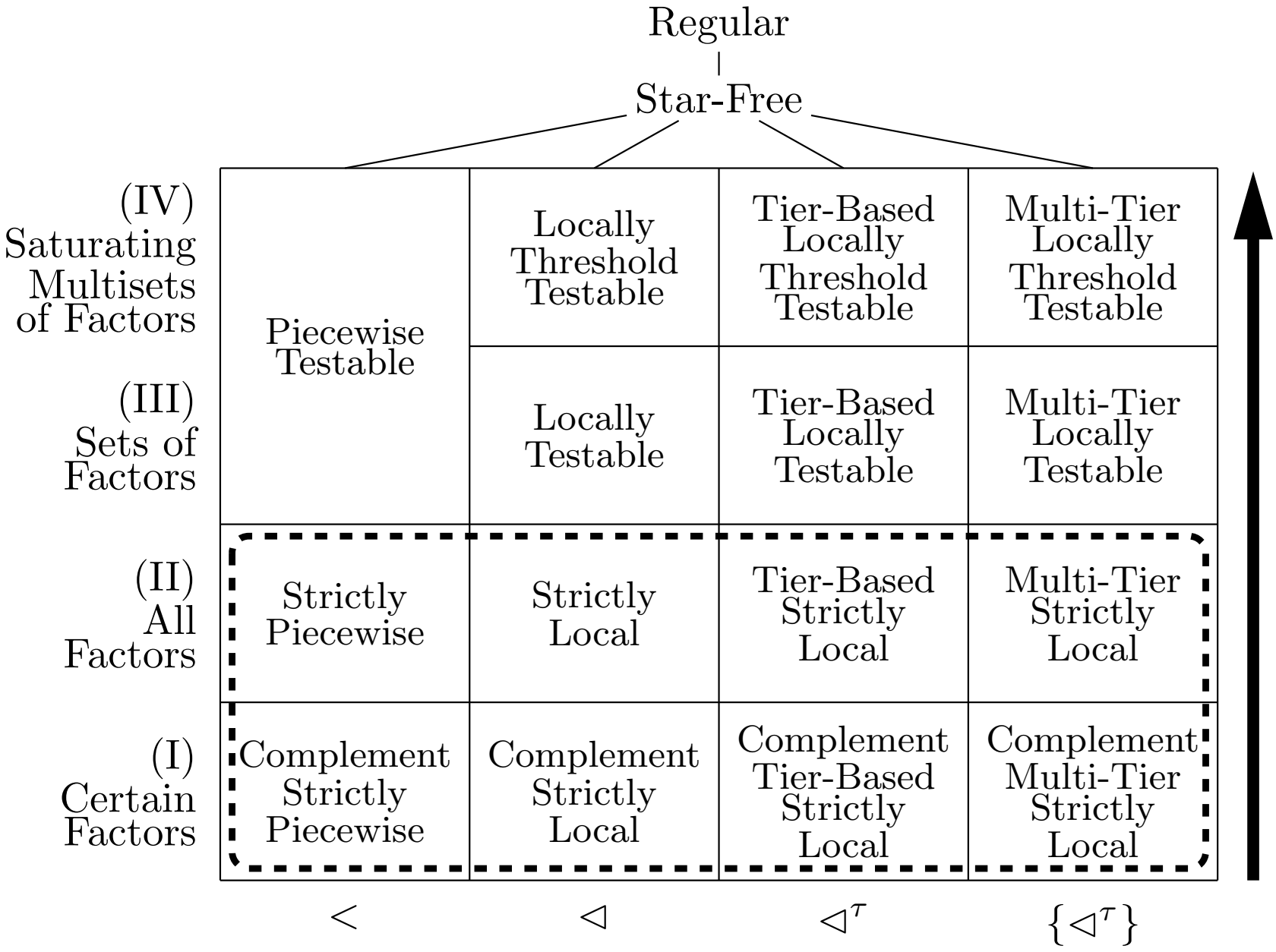
** This constant is doubly exponential in k: $\mathcal{O}((t+1)^{|\Sigma|^k})$.

*** Time complexity depends on the model and some additional choices. I and II are linear with certain optimizations but polynomial without them for some models. These optimizations are not available for III and IV.

Feasibility: Space Complexity



Feasibility: Space Complexity



Part V

What about Phonological Features?

What About Features?

- With n features, we can potentially distinguish 2^n symbols in a conventional model.
- So the space constants becomes even worse with feature-based models because $|\Sigma|$ is effectively larger.

The Selection Problem (Hayes and Wilson 2008:390)

> *. . . one still faces a formidable difficulty: the fact that an enormous number of distributional generalizations are consistent with any given set of surface forms.*

Hayes and Wilson 2008

To solve the selection problem, we assume that UG determines the feature inventory and the format of constraints, yielding a search space that is quite large and hence compatible with the inductive baseline approach. Nevertheless, in our experience it is effectively searchable, provided the right search heuristics are used.

Wilson and Gallagher 2018

What about . . . a nonstatistical model . . . that learns by memorizing feature sequences . . . ? The immediate problem confronting such a model is that any given segment sequence has multiple different featural representations.

Wilson and Gallagher 2018

For example, the attested dorsal-tier trigram [oqa] could be represented

- *with very general classes*
  (e.g., [+syll][−syll] [+syll] = VCV*),*
- *with maximally specific classes*
  (i.e., [+syll, −high, −low, +back][−cont, −son, +dorsal, −high, −cg][+syll, −high, +low] = [oqa]*),*
- *or at intermediate levels of granularity*
  (e.g., [+ syll, −high, −low][− cont, −son, +dorsal, −high][+ syll, −high, +low] = EQA*).*

Wilson and Gallagher 2018

If a hypothetical [model] judged a substring to be legal as long as it satisfied ***any*** *attested featural description, it would tolerate (among other structures) every VCV trigram and thus massively overgeneralize. If the model instead required* ***all*** *feature representations of a substring to be attested, it would be equivalent to [memorizing segmental trigrams] . . . Lacking a method for deciding which representations are relevant for assessing well-formedness—precisely the role played by statistics [here]—****learning . . . is doomed****.*

(emphasis added)

Part VI

Bottom Up Factor Learning

Bottom-Up Factor Inference Algorithm Chandlee et al. 2019

Theorem

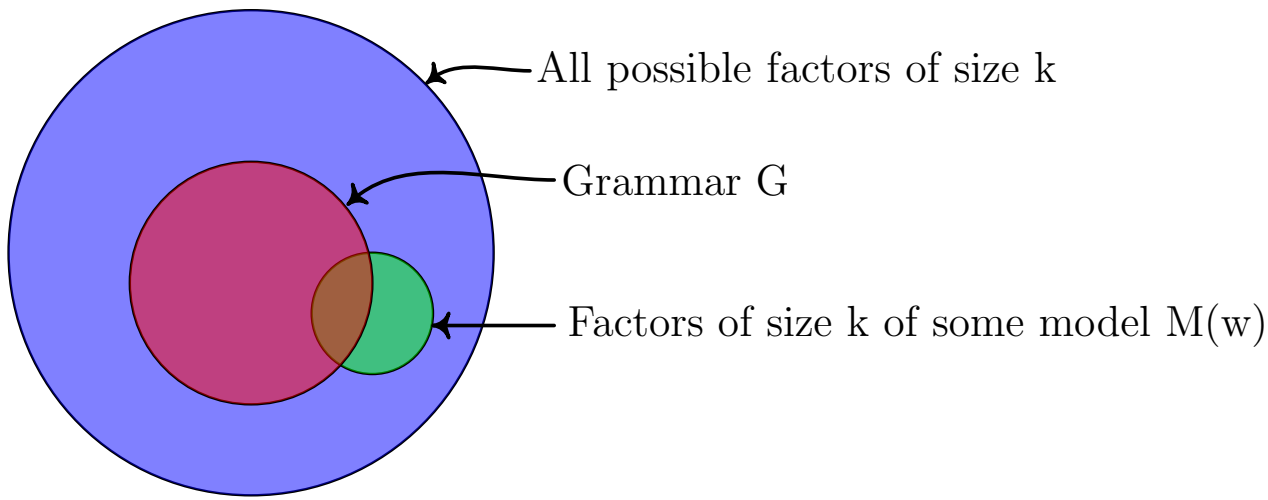
Given a finite positive data sample D, BUFIA finds a constraint grammar G such that:

1. *G is consistent, i.e. it covers the data:*
   - *$D \subseteq L(G)$*
2. *$L(G)$ is the smallest language in the class $\mathcal{L}$ which covers the data*
   - *for all $L \in \mathcal{L}$ where $D \subseteq L$, $L(G) \subseteq L$*
3. *the largest forbidden factor is of size k*
4. *G includes the most general factors $\mathbf{m}$ of any other grammars G' that also satisfy (1,2,3)*
   - *for all $\mathbf{m}' \in G'$, there exists $\mathbf{m} \in G$ such that $\mathbf{m} \leq \mathbf{m}'$.*

Forbidden Factor Grammars

(2) G is a set of **forbidden** factors and G recognizes words whose models' factors are disjoint with G.

- CV language
  G = { [+syll][+syll], [-syll][-syll] }
  with feature-based successor model and $k = 2$



M(w) is rejected.

BUFIA finds forbidden factors.

Foreshadowing

1. So Chandlee et al. 2019 provide a non-statistical method "for deciding which representations are relevant for assessing well-formedness."
2. However, the grammar obtained is redundant in a way I will make clear.
3. This is an under-appreciated fact faced by all phonotactic learner which use features.
4. This necessitates the need for clear *abductive* principles for determining *which* constraints are in the grammar.

What is Abduction?

- Abduction is inference to the best explanation.
- Many hypotheses are empirically equivalent.
- Abductive principles tell us which to select (cf. the Selection Problem).
- Abductive principles typically invoke concepts of simplicity and generality.

(Haig 2018)

Examples of Abductive Principles in Action

Hayes and Wilson 2008, 4.2.2:

- "First, shorter constraints (fewer matrices) are treated as more general than longer ones."
- "...we suggest that the value of a constraint is proportional to the number of segments contained in its classes, and our metric sorts constraints of a given length on this basis."

BUFIA also

- prefers more general constraints using the structure of factor space.
- orders features to sort same-length constraints but does so intrinsically as opposed to extensionally

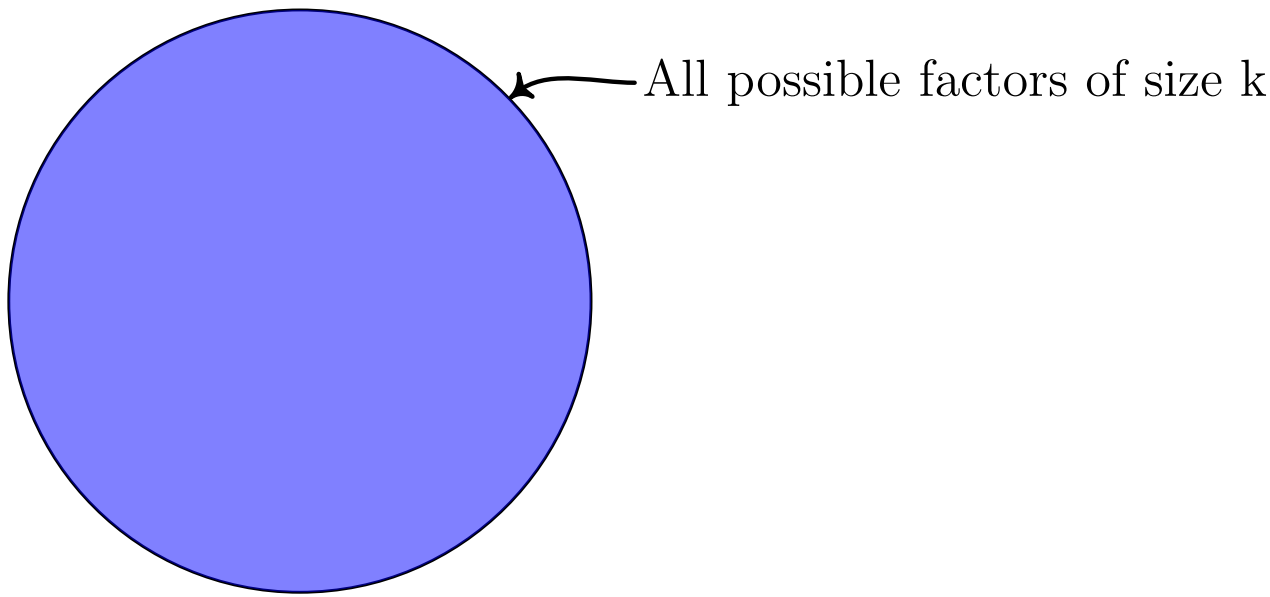
Part VII

Ordering Factors

Ordering Factors

The set of possible factors forms a partial order.

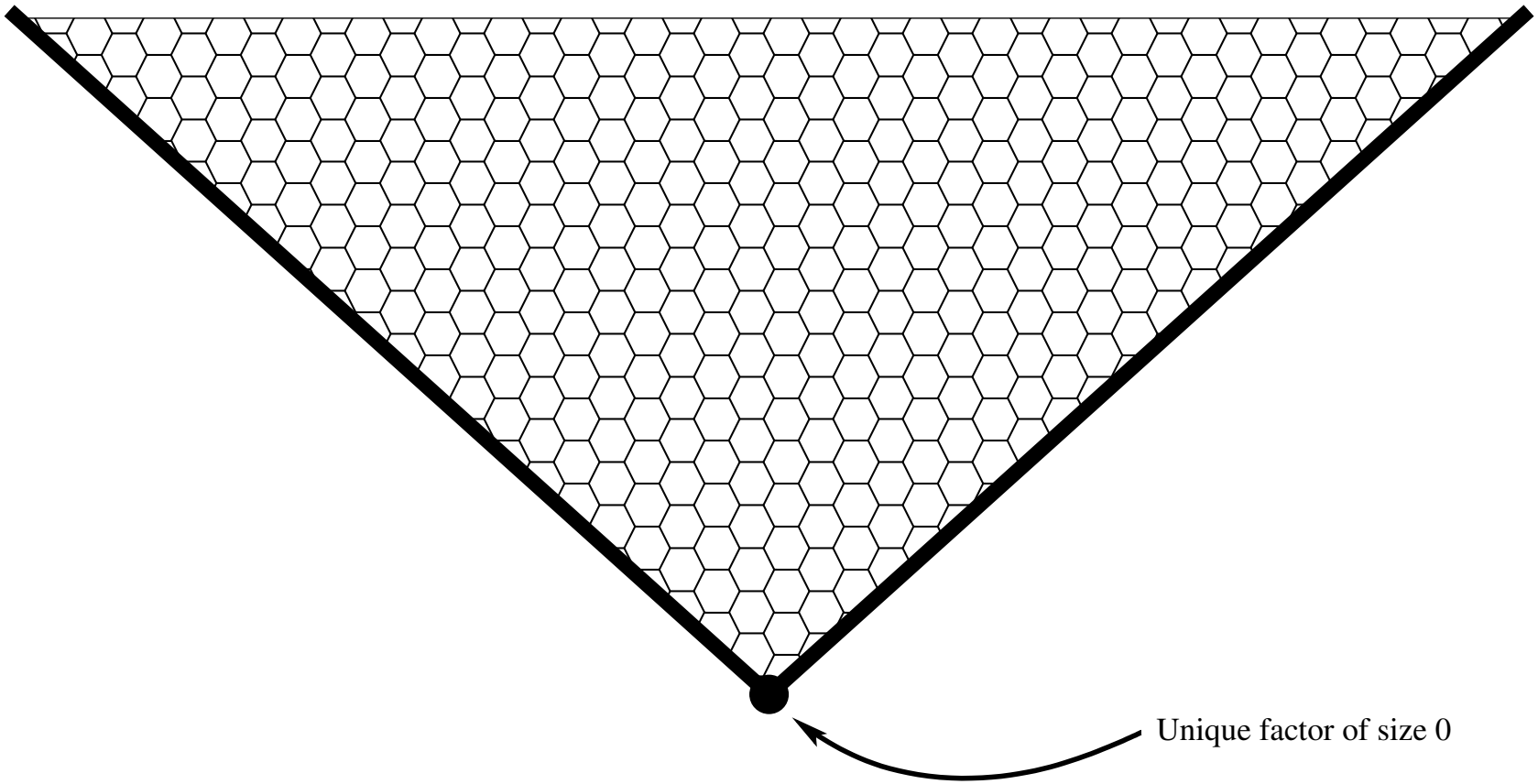
$A \leq B$ iff B contains A as a factor.



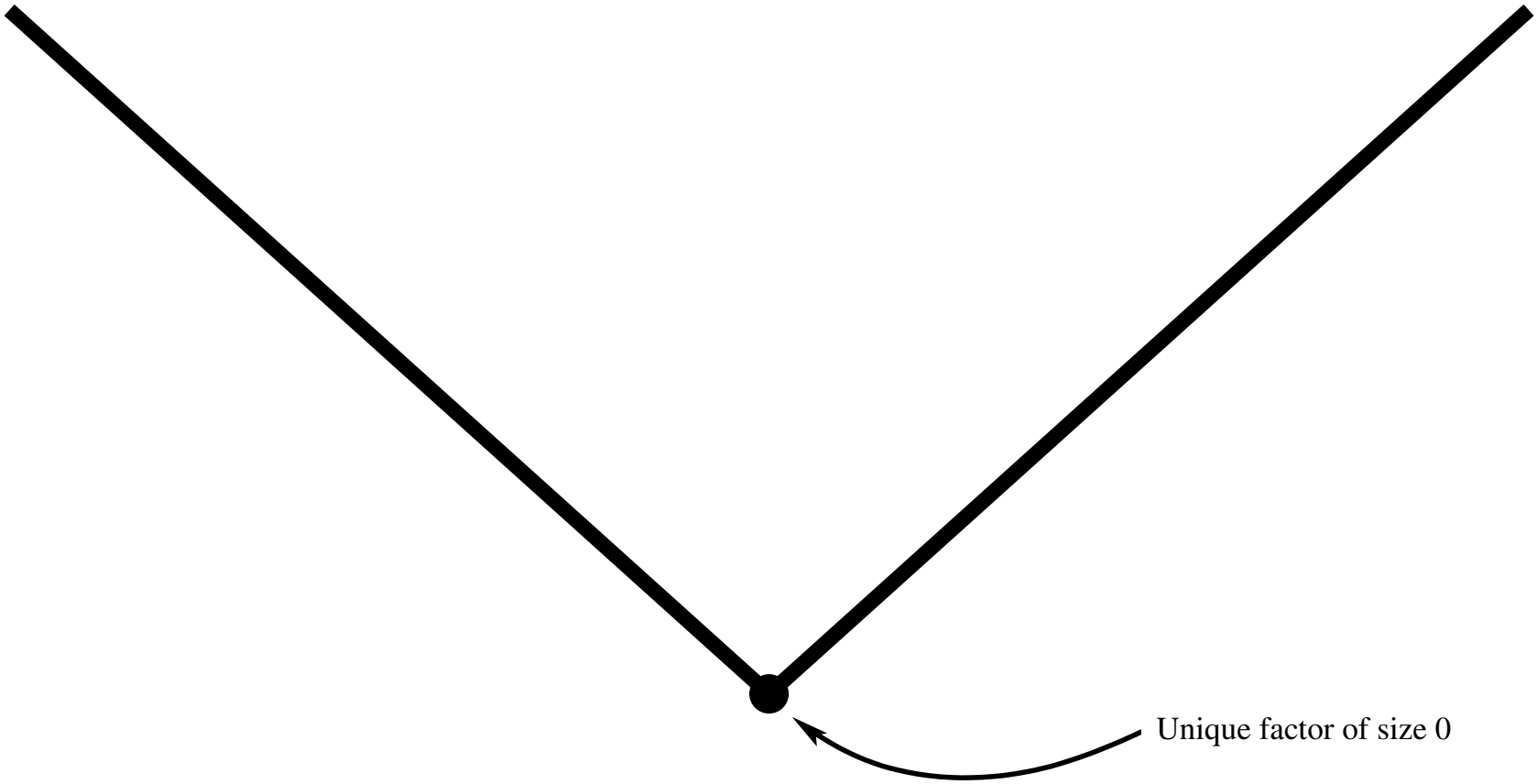
Ordering Factors

The set of possible factors forms a partial order.

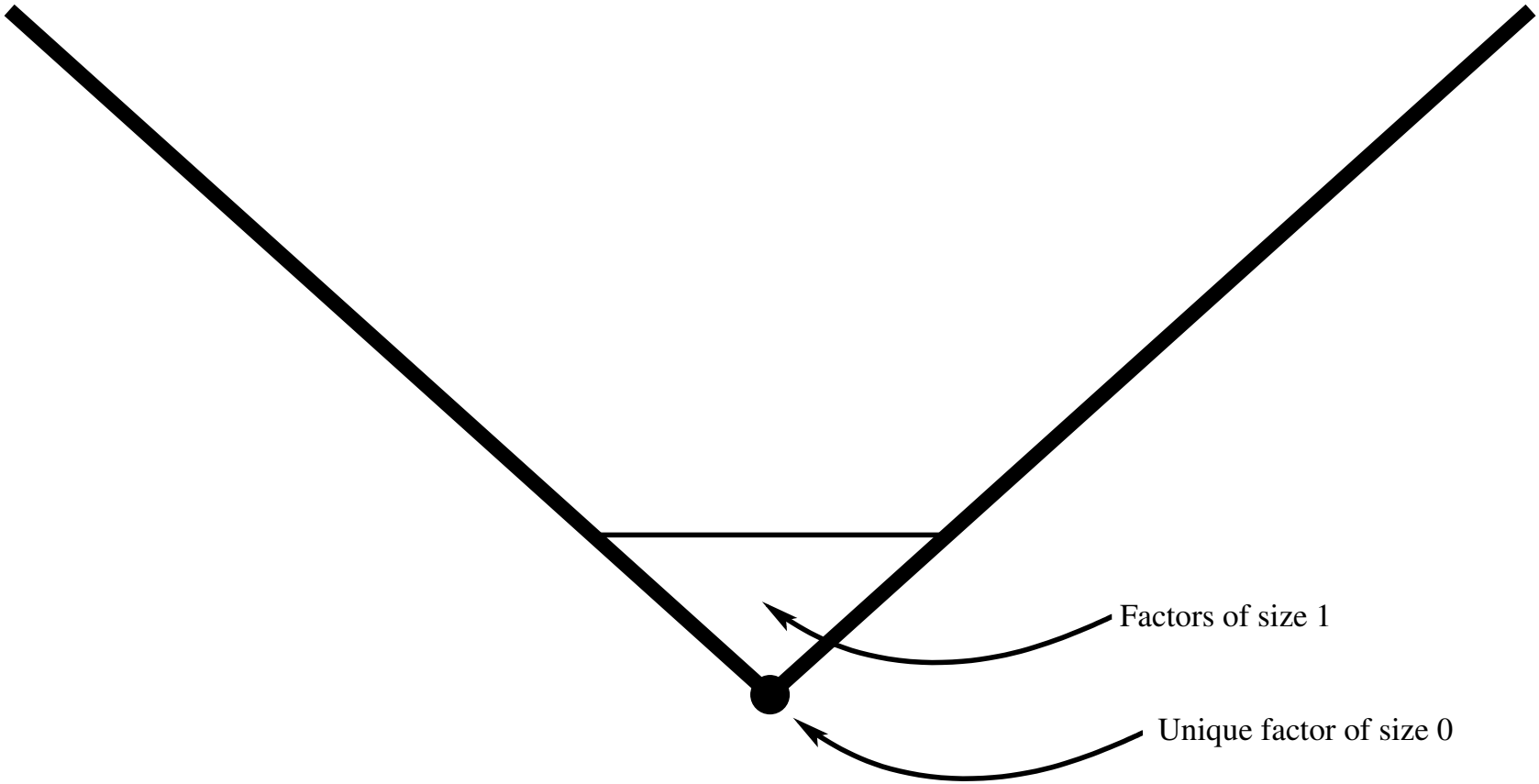
$A \leq B$ iff B contains A as a factor.



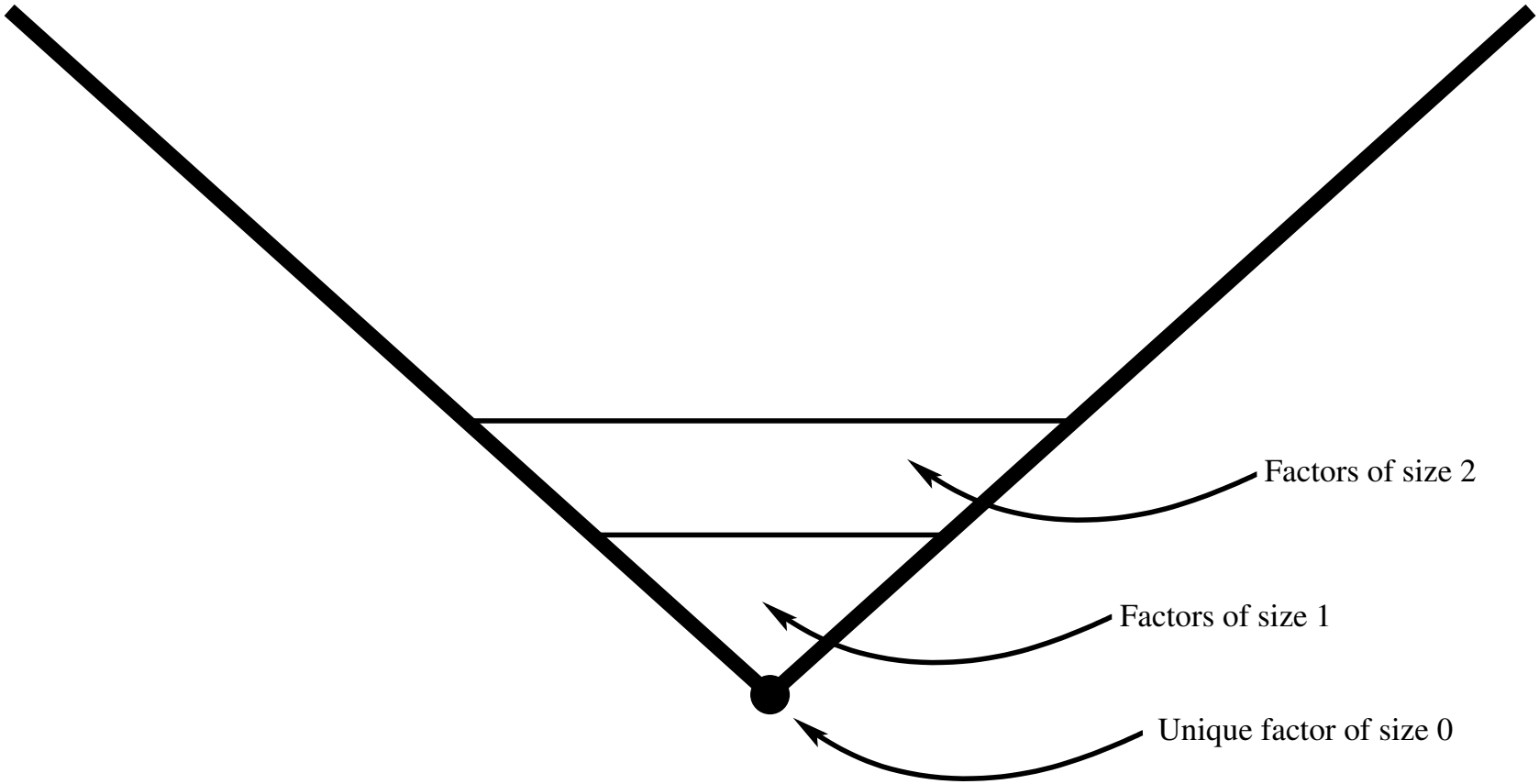
Factor Space



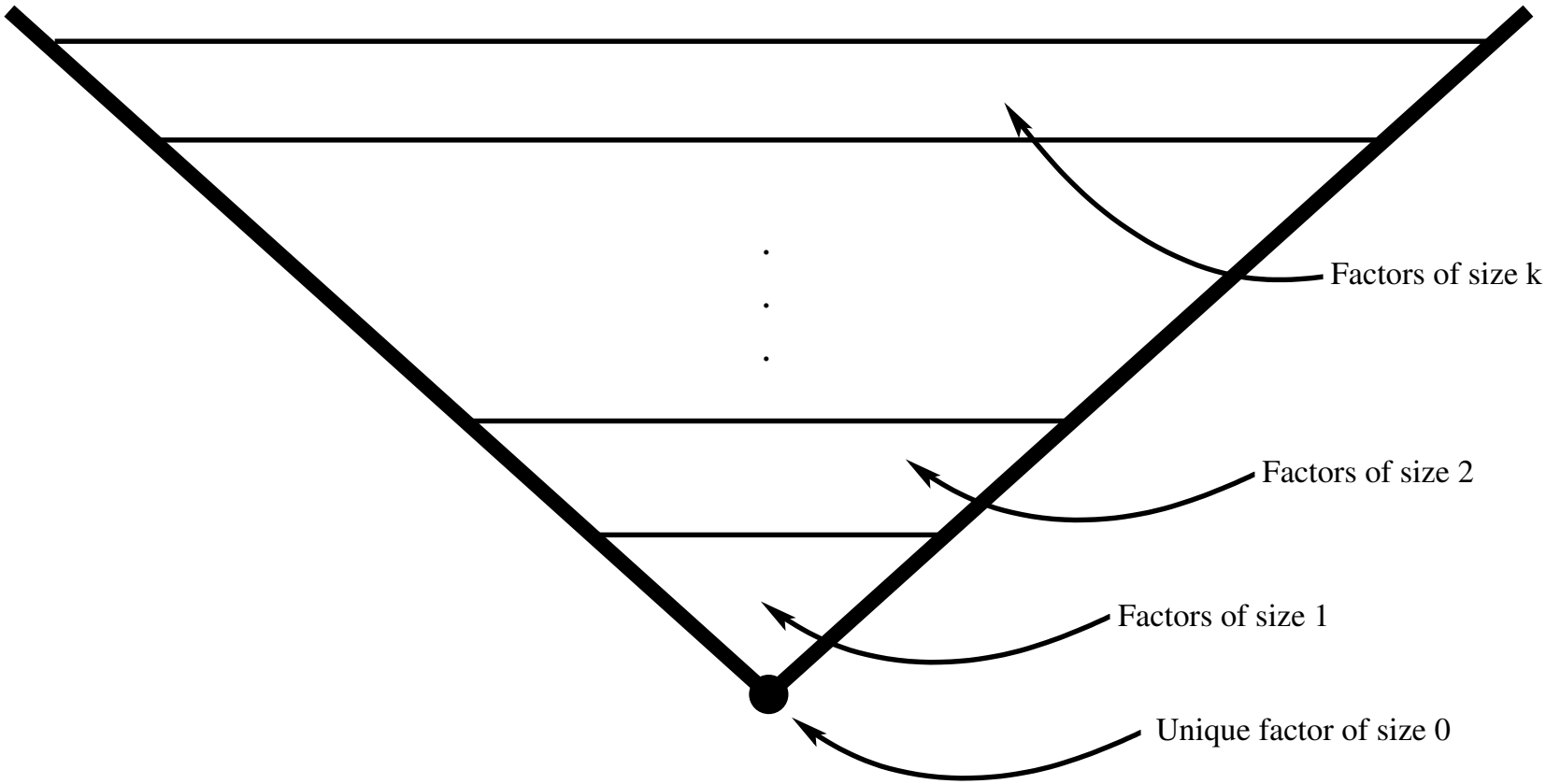
Factor Space



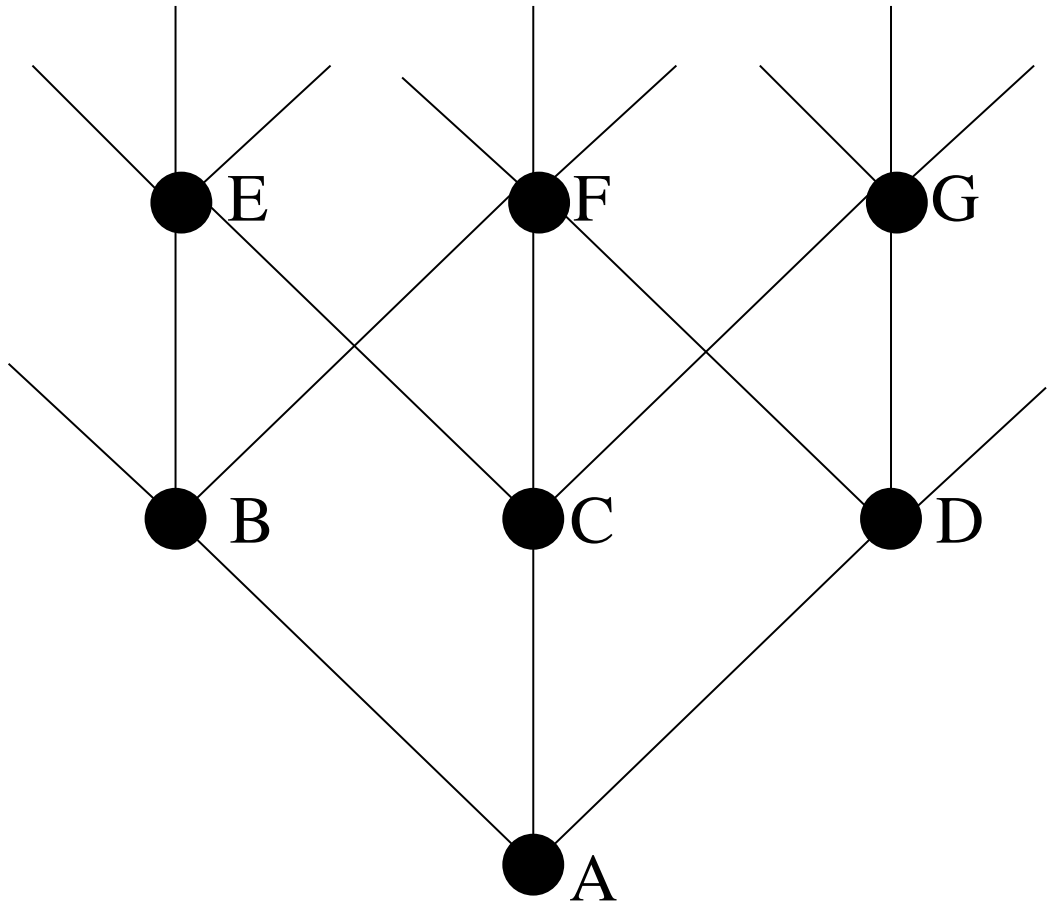
Factor Space



Factor Space

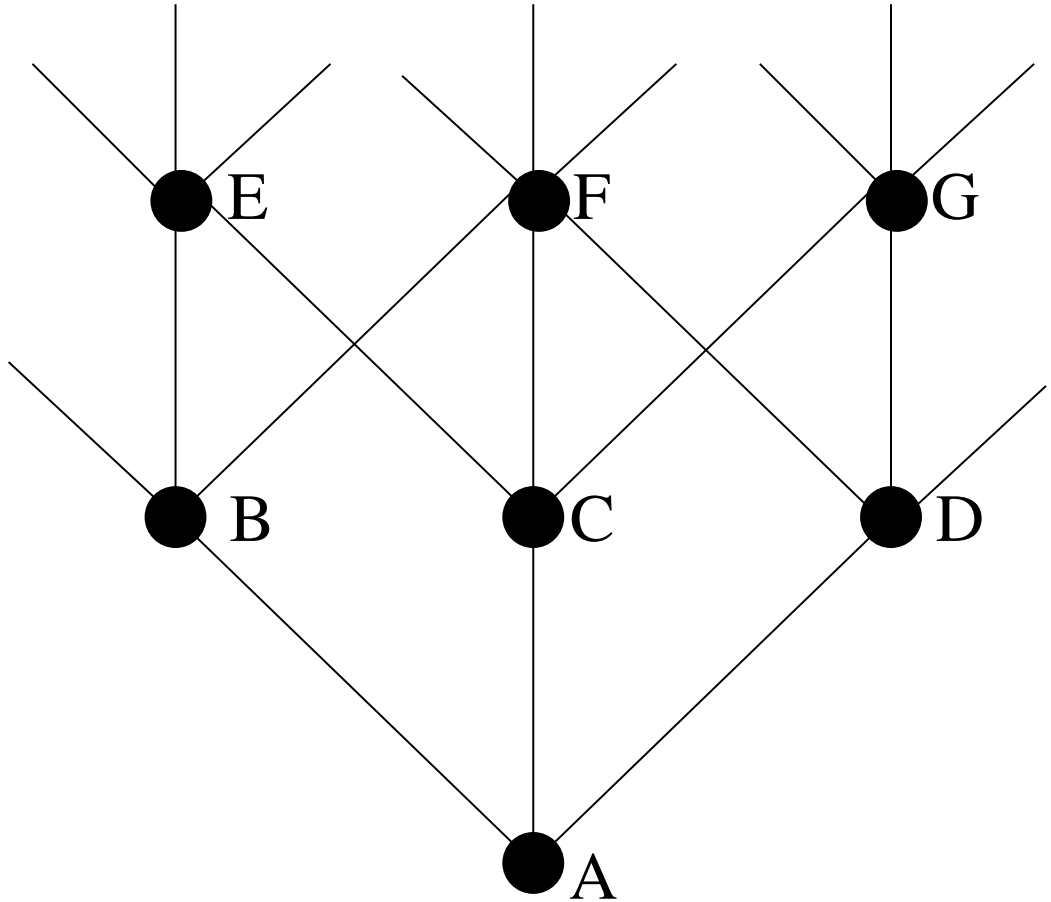


Partially Ordered Factors



More specific factors are higher up

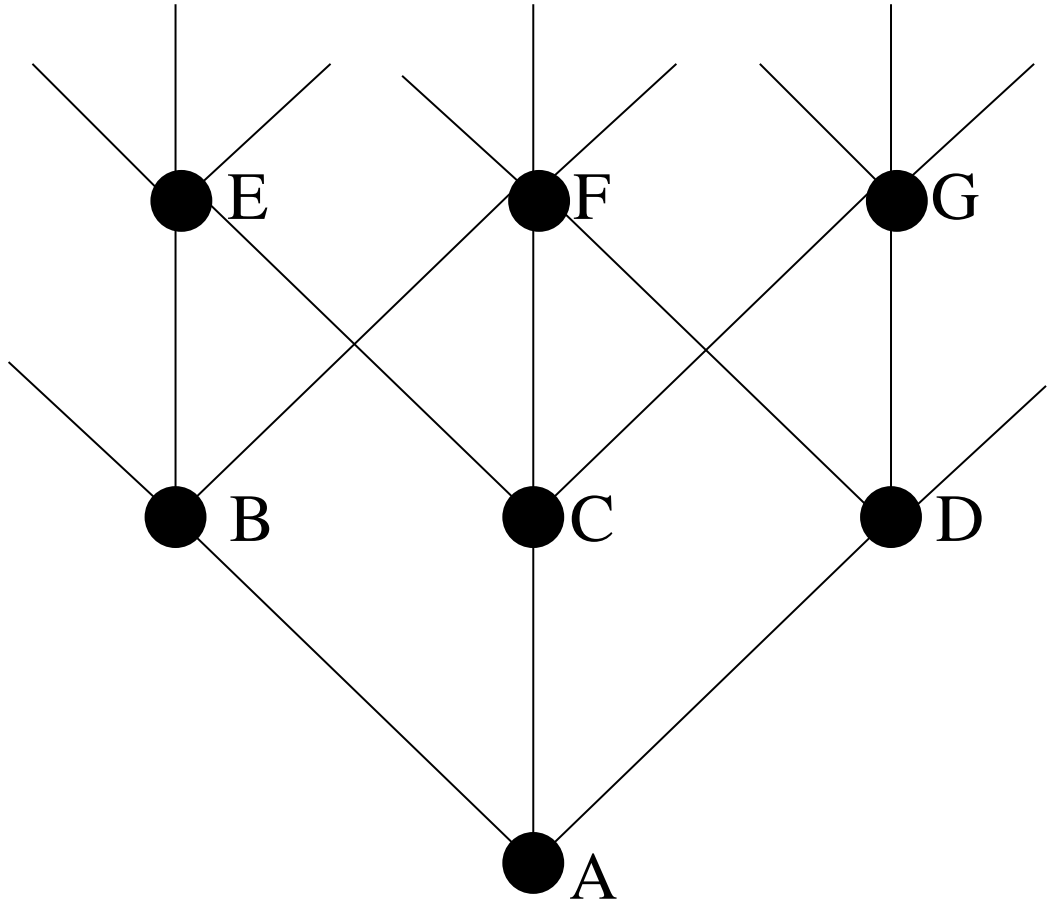
Partially Ordered Factors



"is a factor of"

A $\leq$ B, C, D, E, F, G

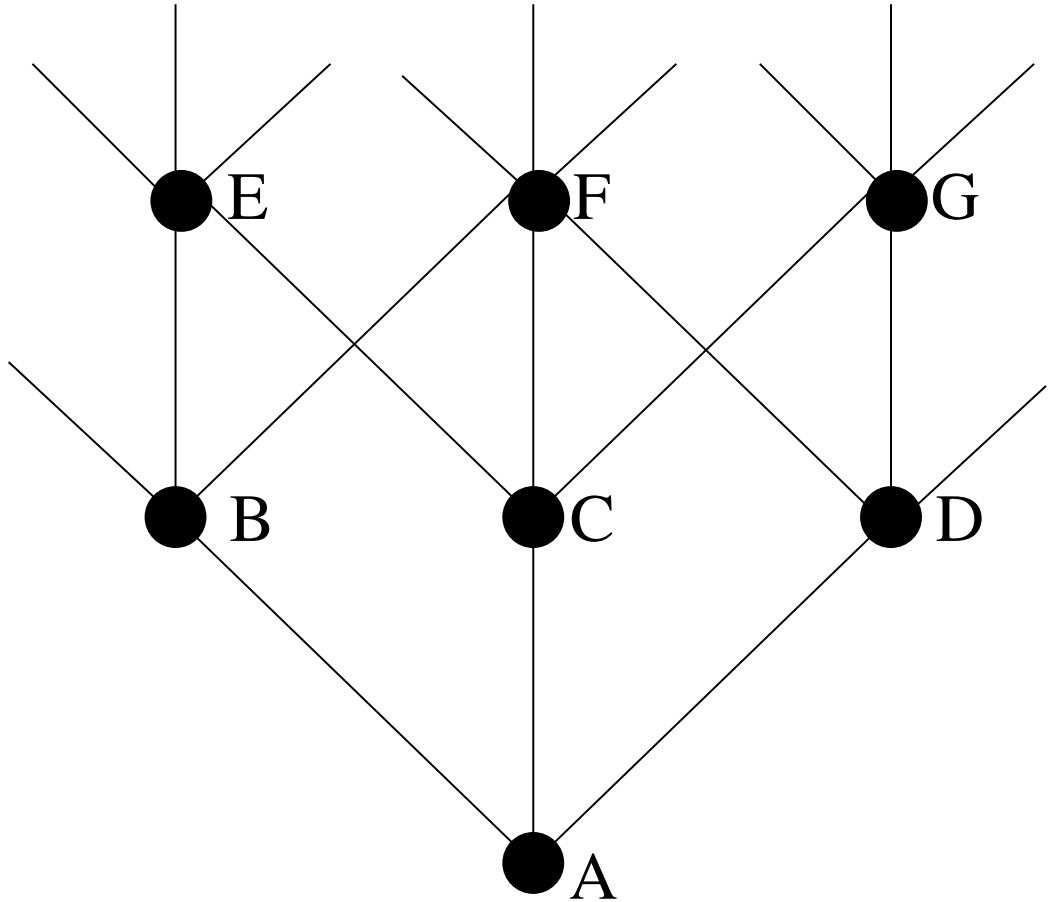
Partially Ordered Factors



"is a factor of"

$B \leq E, F$

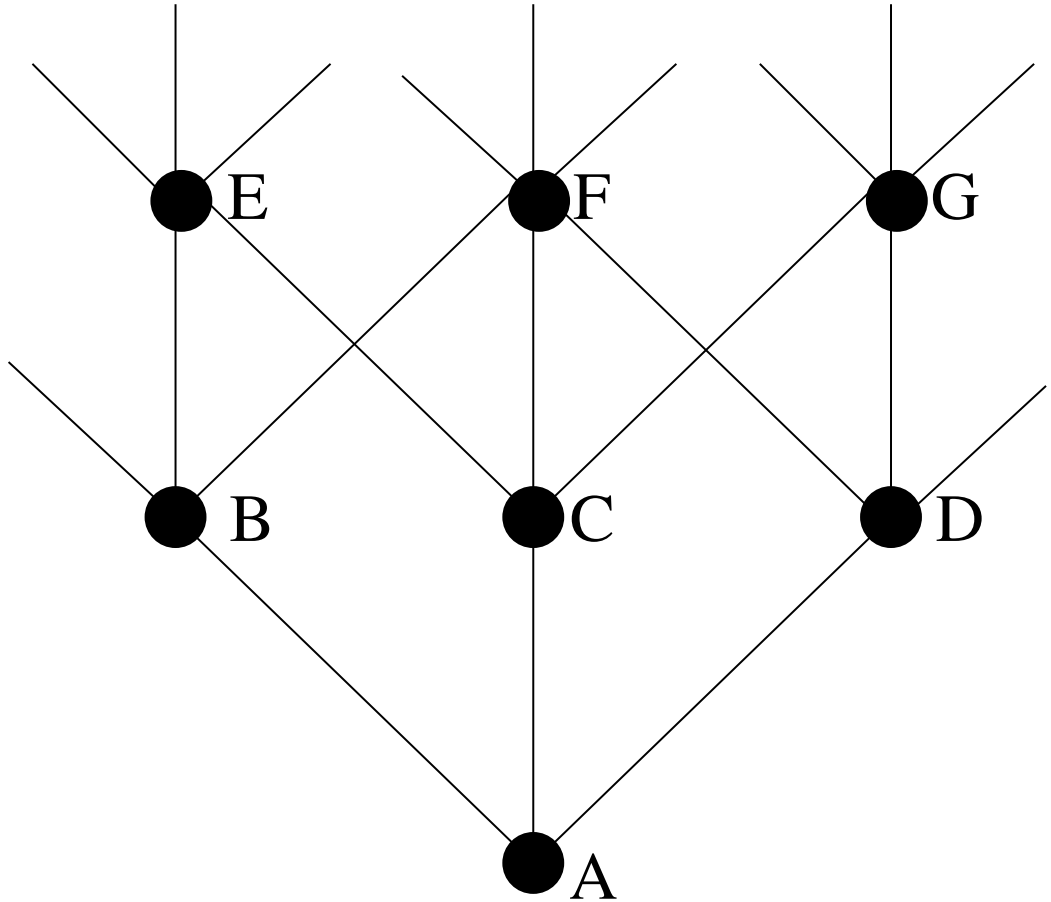
Partially Ordered Factors



"is a factor of"

$C \leq E, F, G$

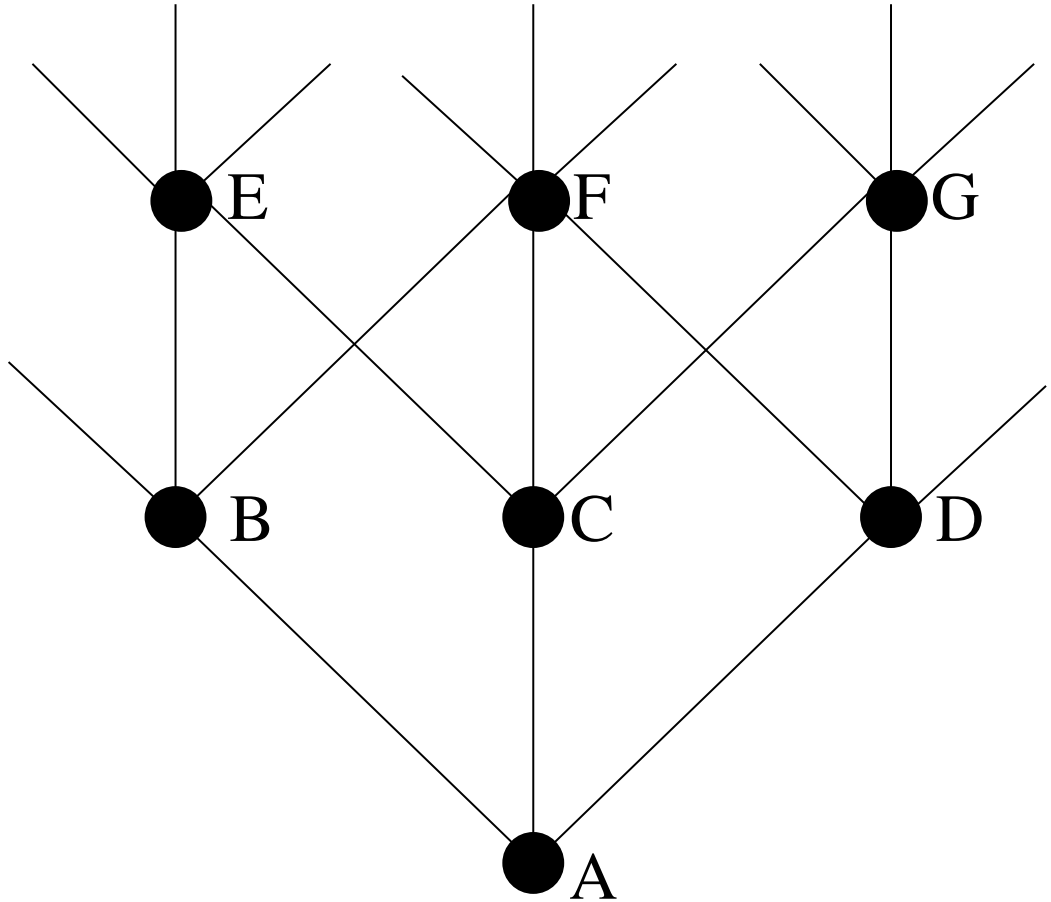
Partially Ordered Factors



"are incomparable"

B ~ C, D

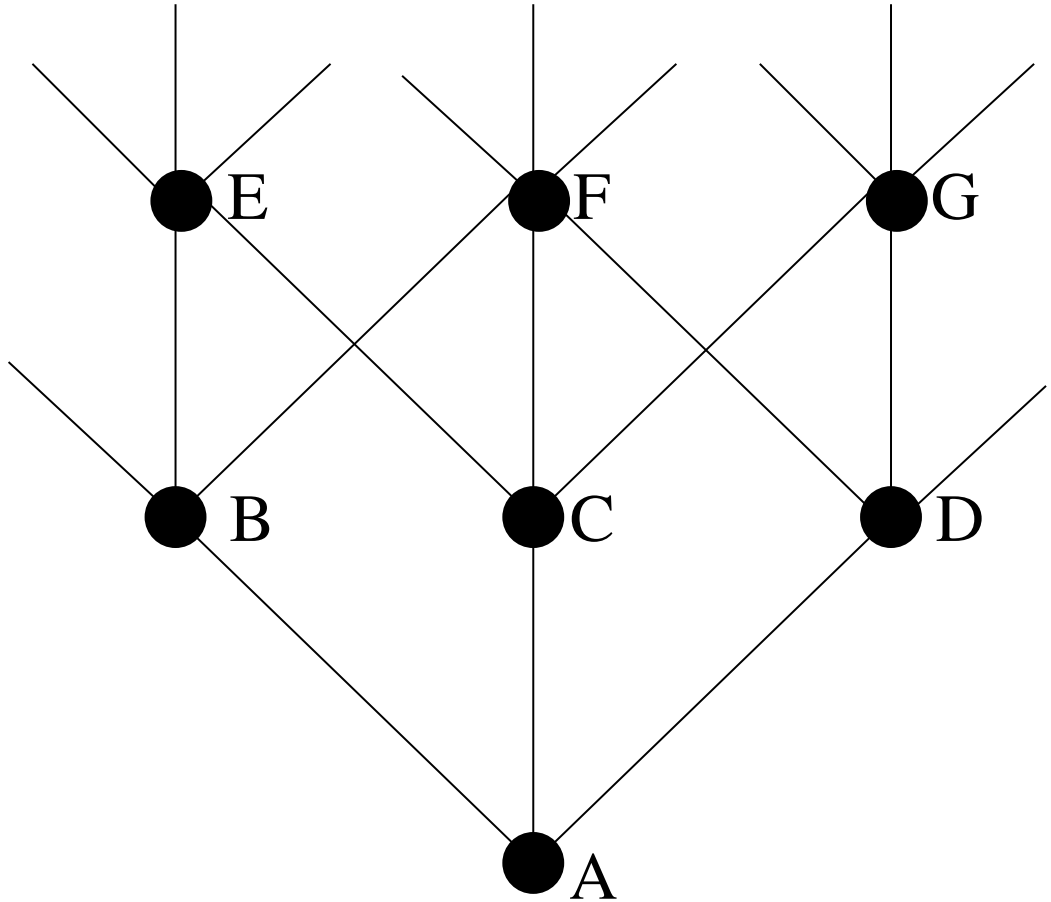
Partially Ordered Factors



"are incomparable"

F $\sim$ E, G

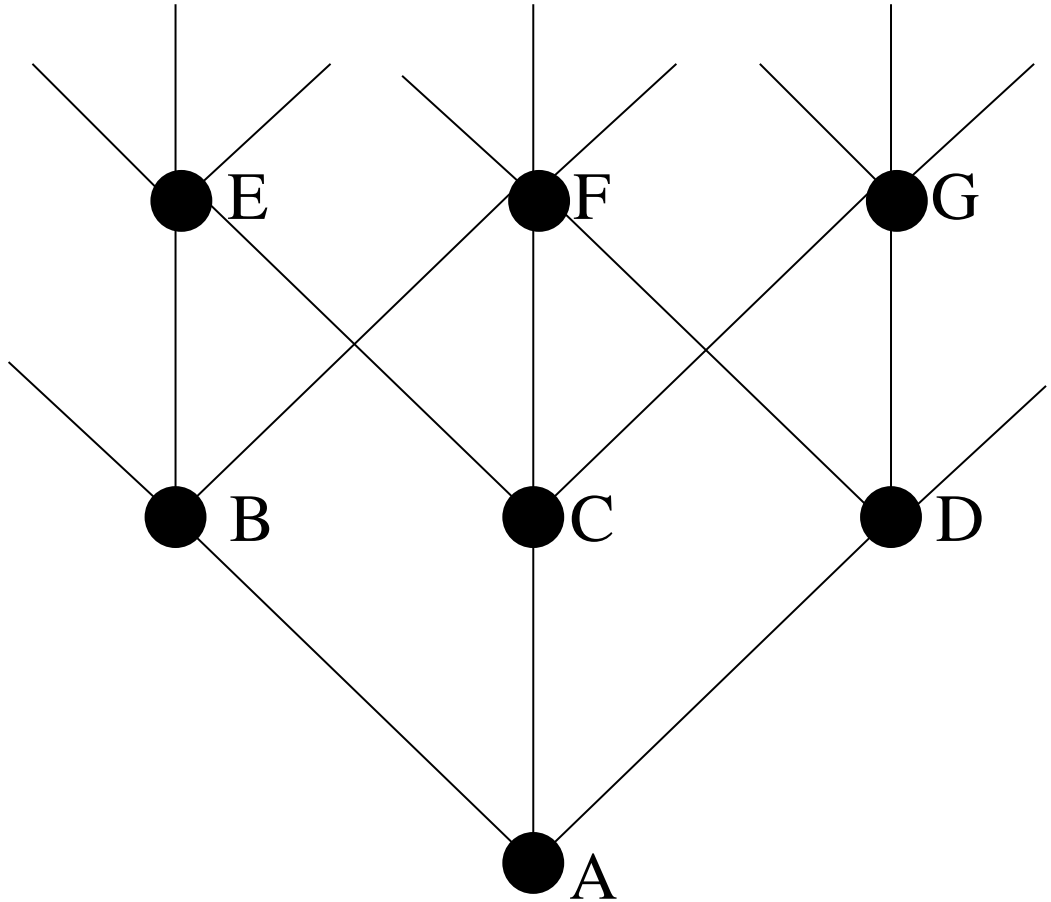
Partially Ordered Factors



"is a superfactor of"

B, C, D $\geq$ A

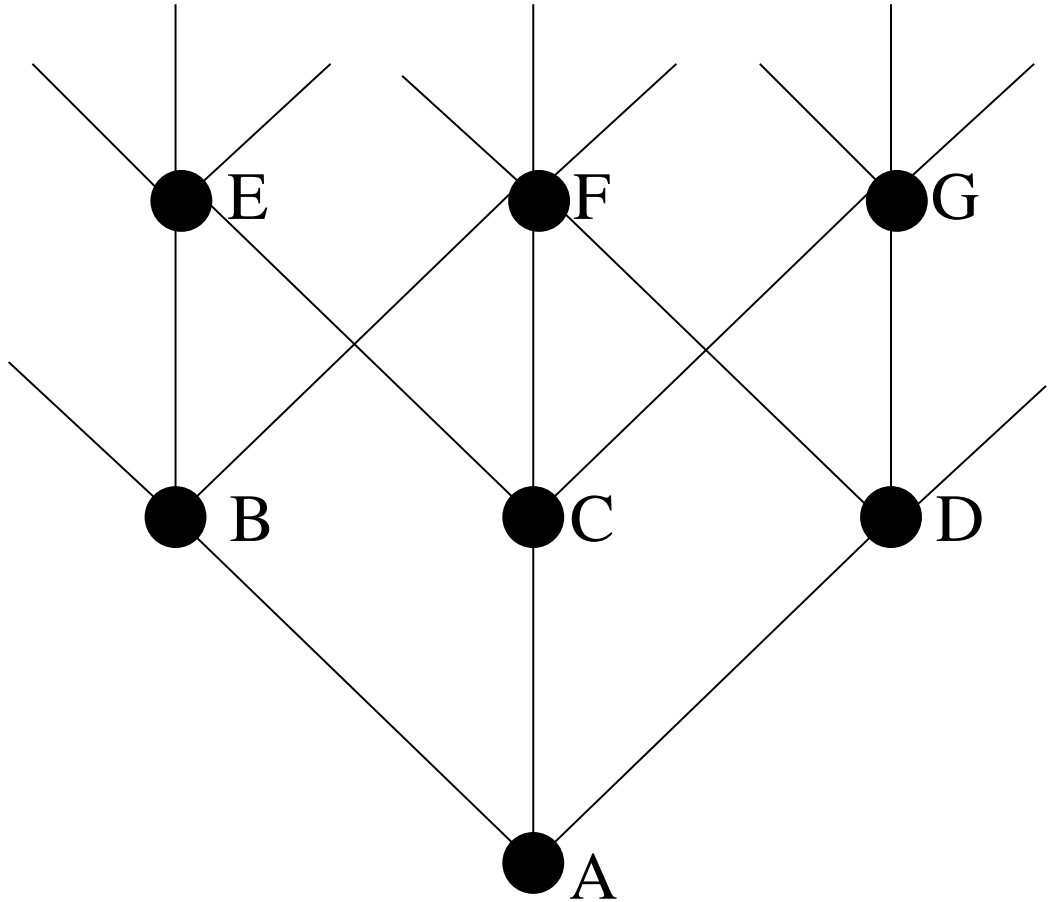
Partially Ordered Factors



"is a superfactor of"

E $\geq$ B, C, A

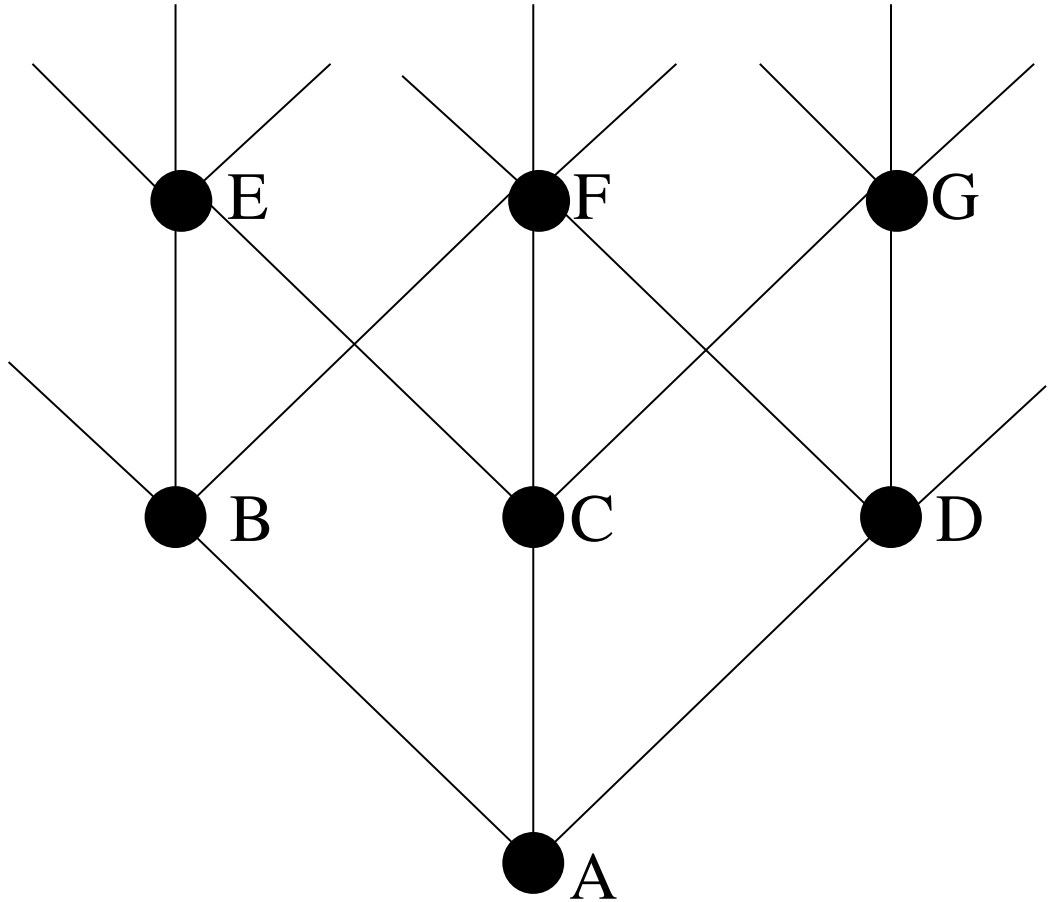
Partially Ordered Factors



"is a superfactor of"

G $\geq$ C, D, A

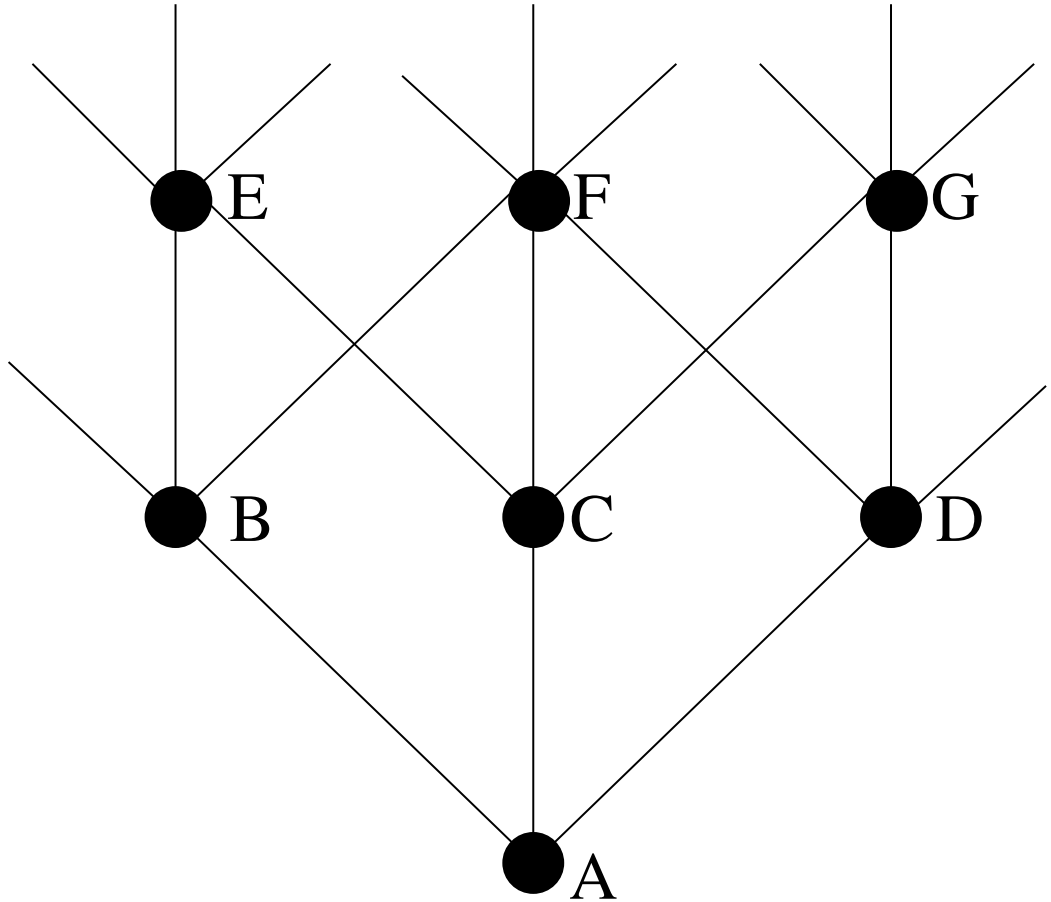
Partially Ordered Factors



"is a successive superfactor of"

B, C, D $\rhd$ A

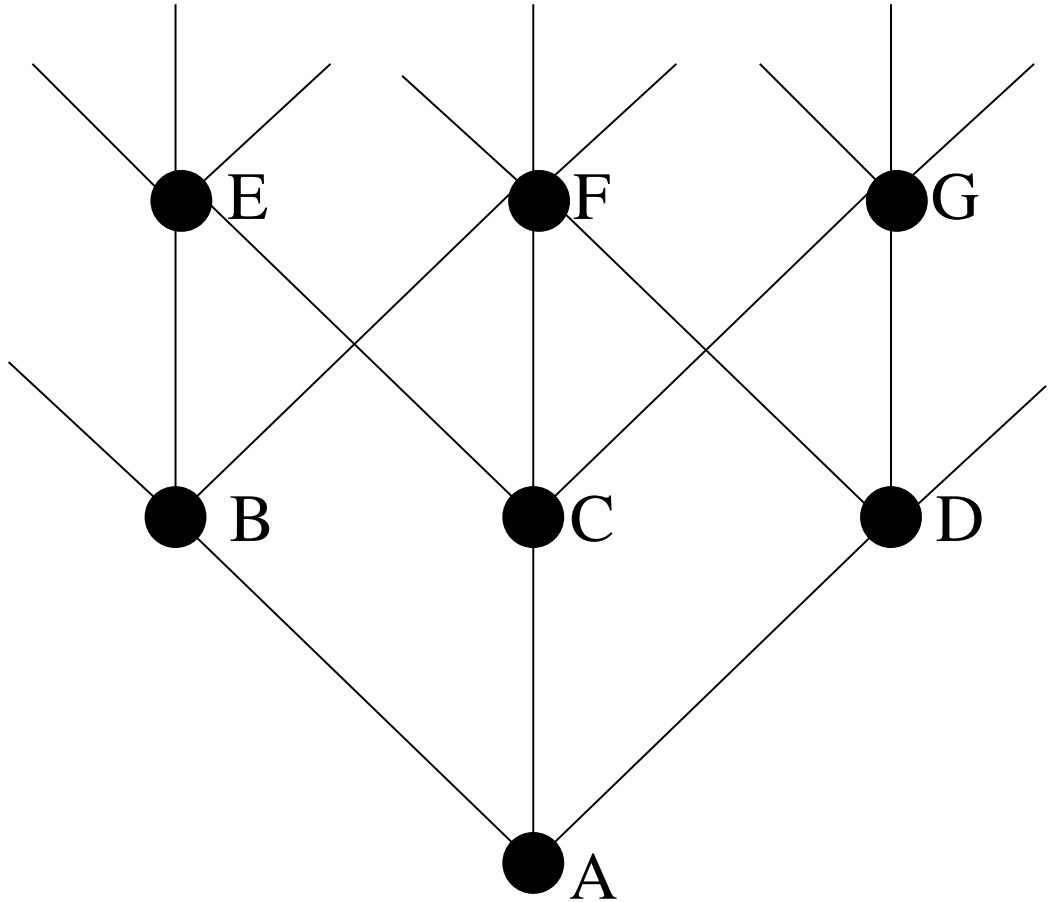
Partially Ordered Factors



"is a successive superfactor of"

E, F, G $\triangleright$ C

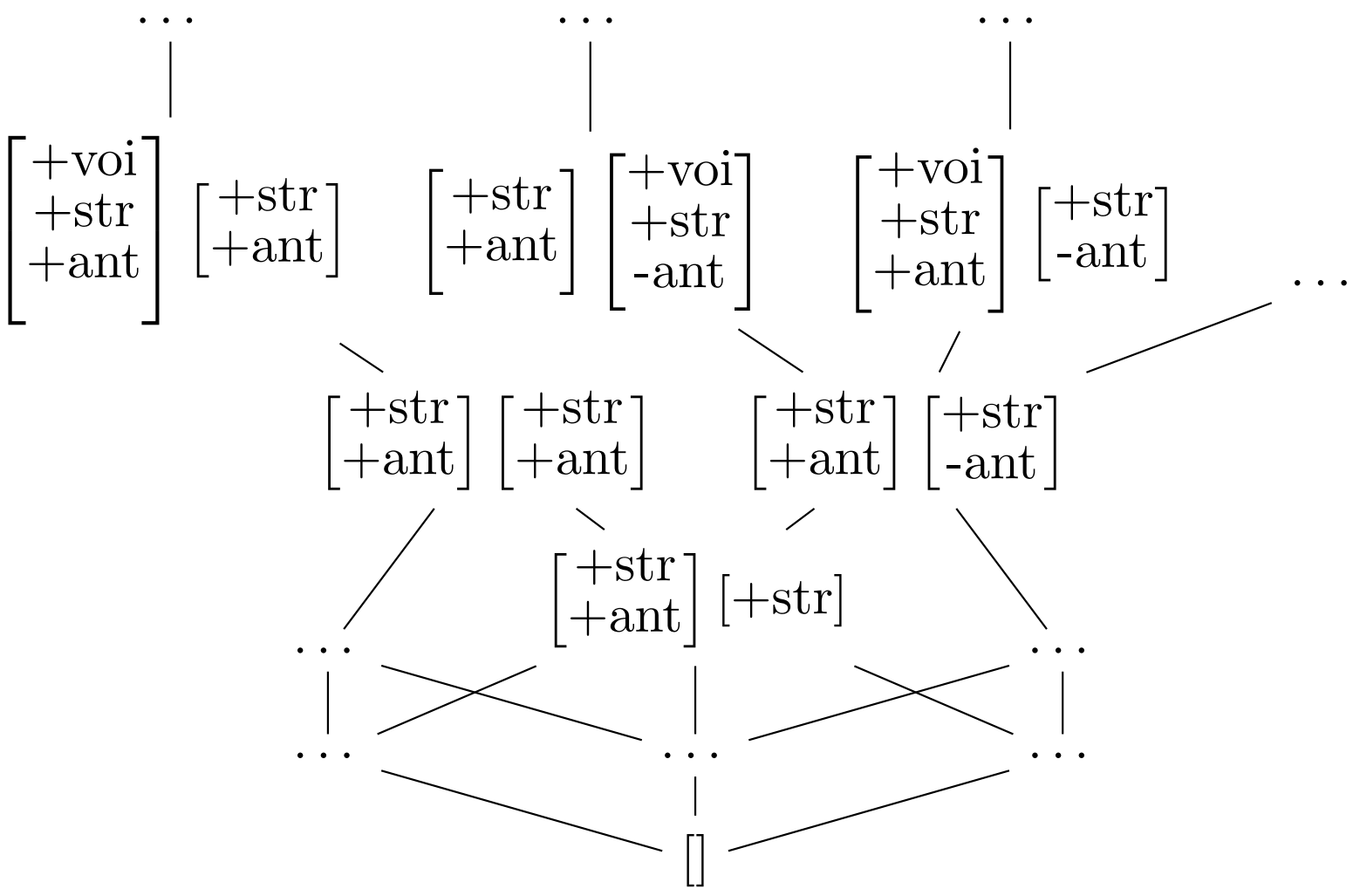
Partially Ordered Factors



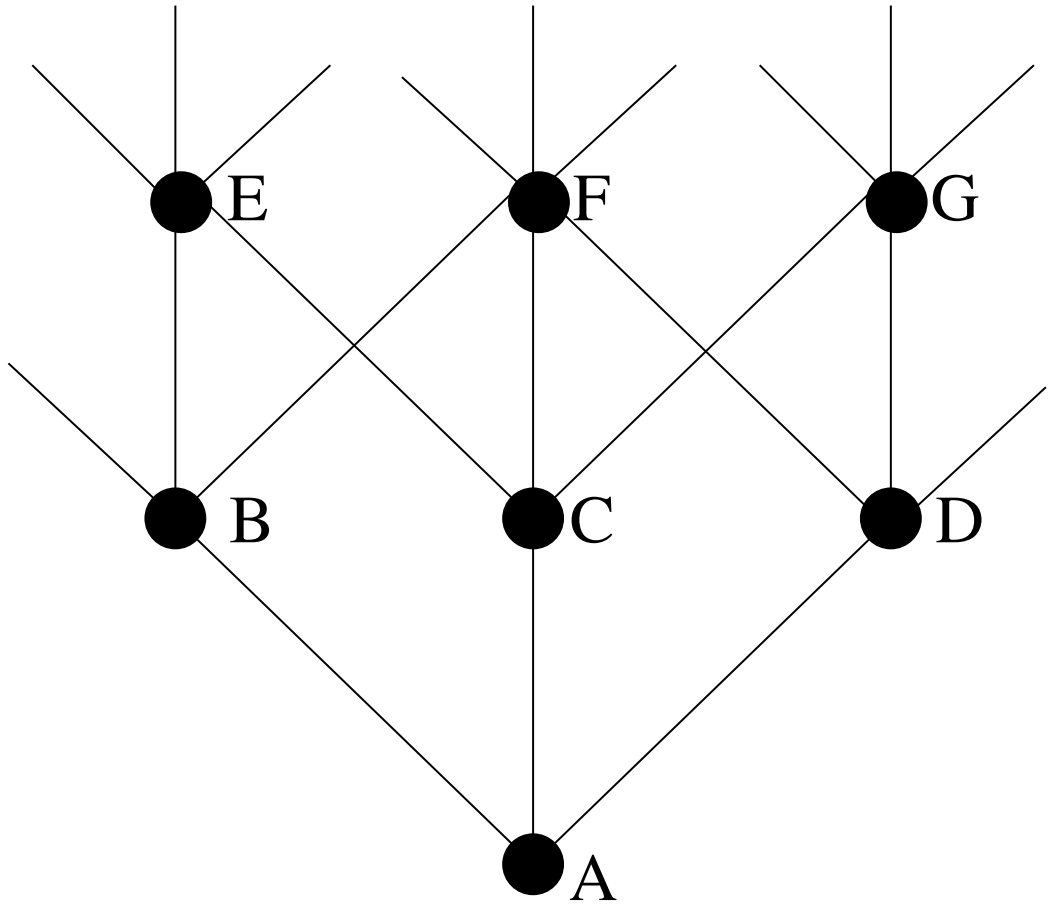
"is a successive superfactor of"

$G \rhd C, D$

Example: Sibilant Harmony with precedence (<)

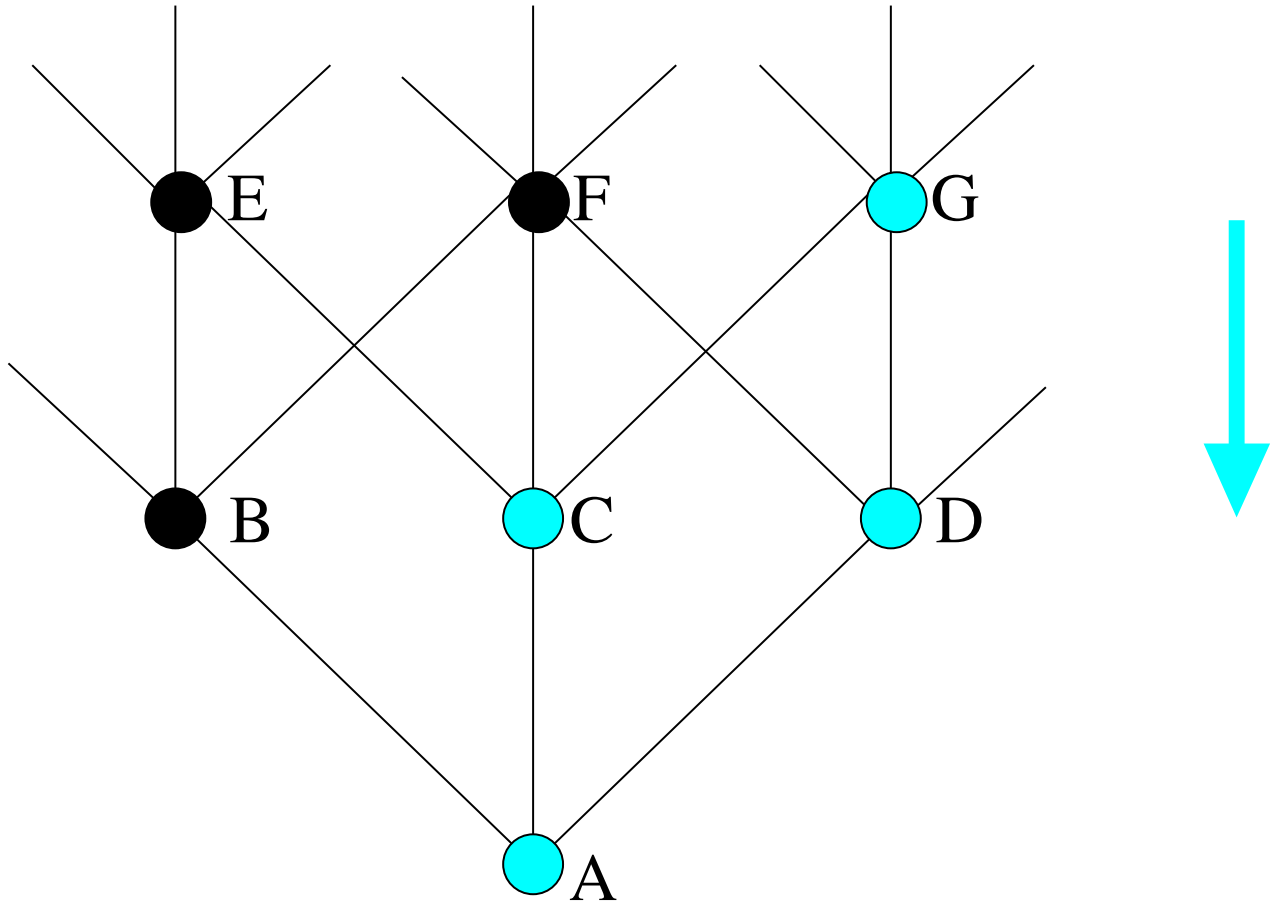


Inference via Factors



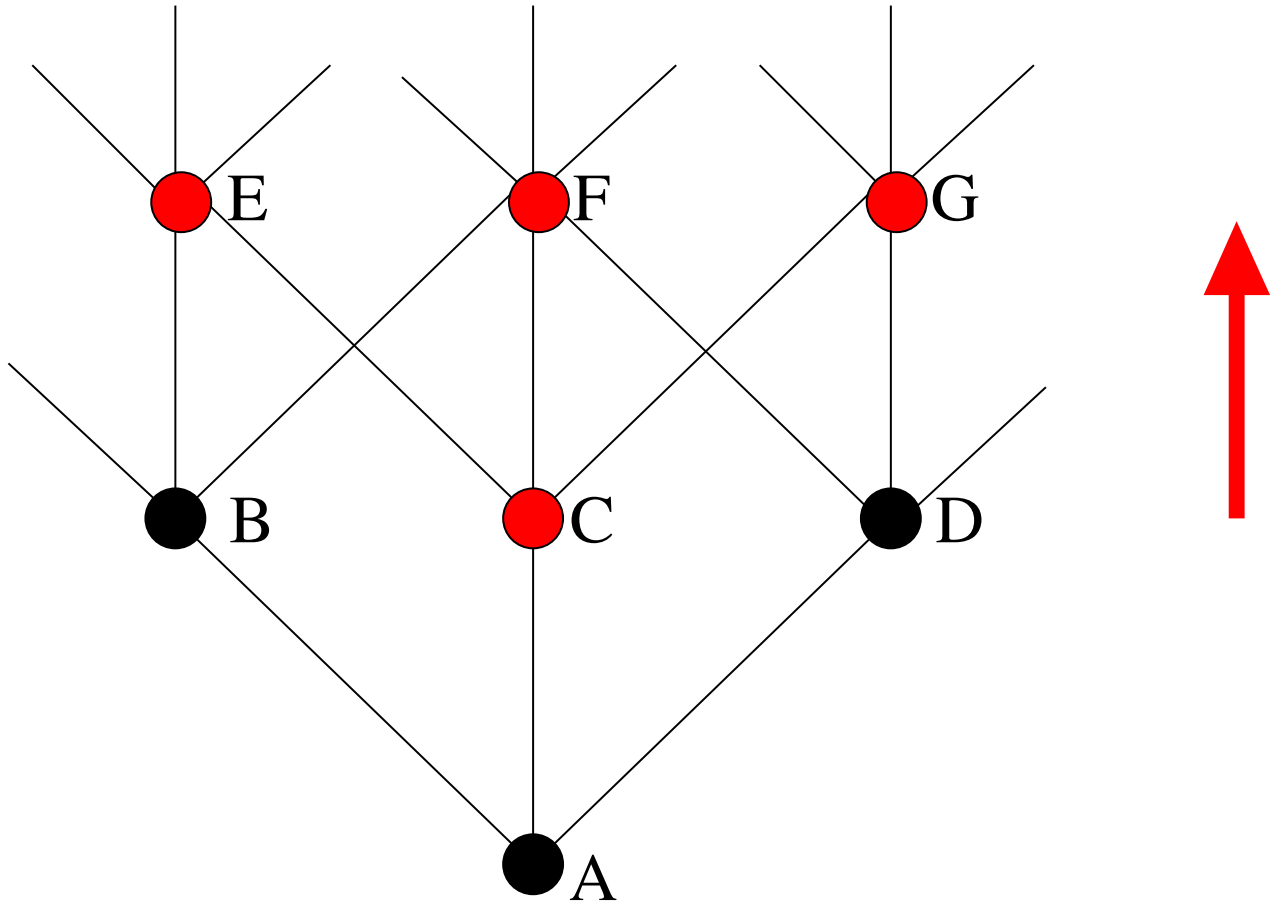
- If factor G is is permissible, we can conclude all of its factors are too.
- If factor C is forbidden, we can conclude all of its superfactors are too.

Inference via Factors



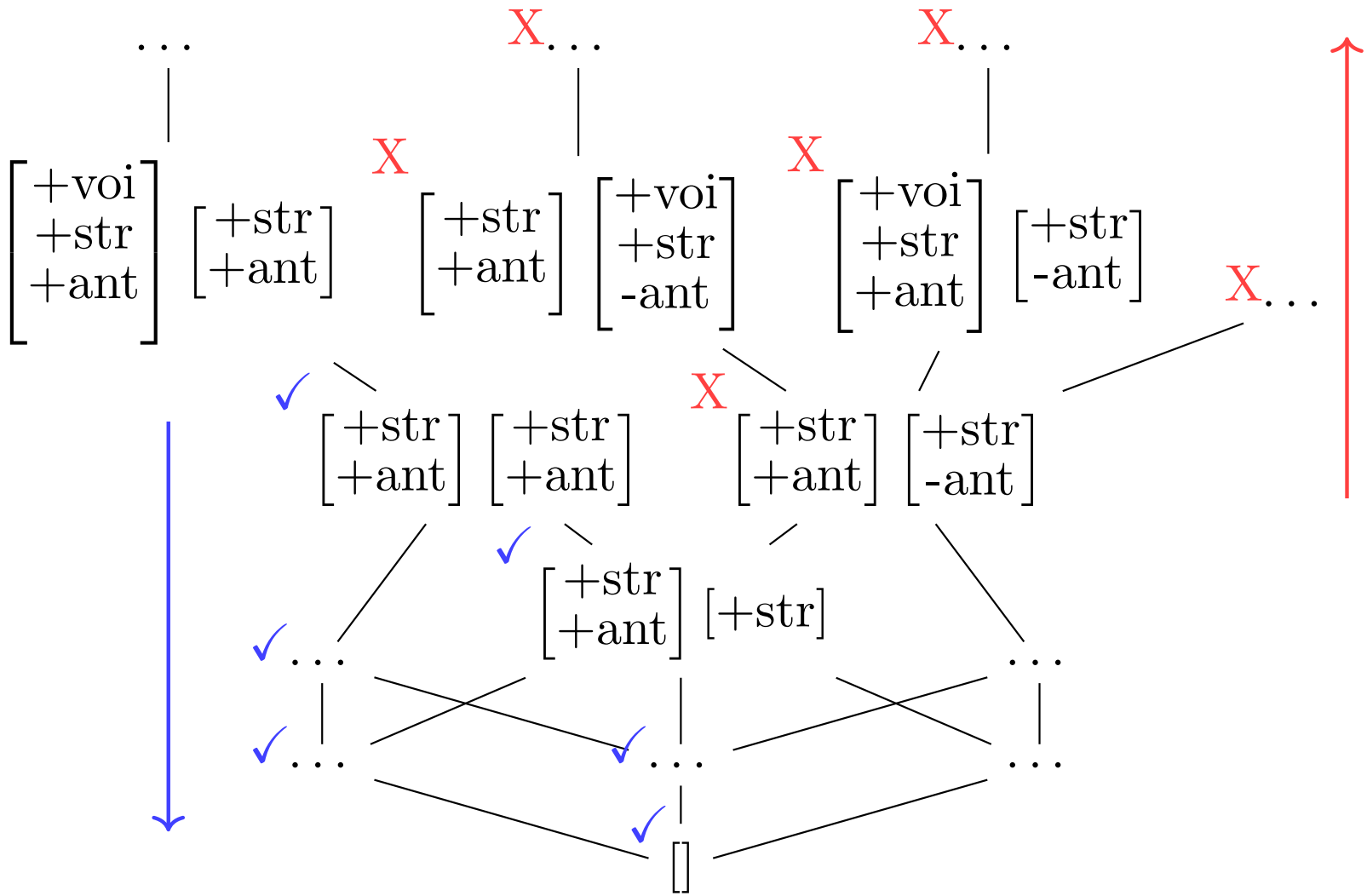
- If factor G is is permissible, we can conclude all of its factors are too.
- If factor C is forbidden, we can conclude all of its superfactors are too.

Inference via Factors



- If factor G is is permissible, we can conclude all of its factors are too.
- If factor C is forbidden, we can conclude all of its superfactors are too.

Example: Sibilant Harmony with precedence (<)



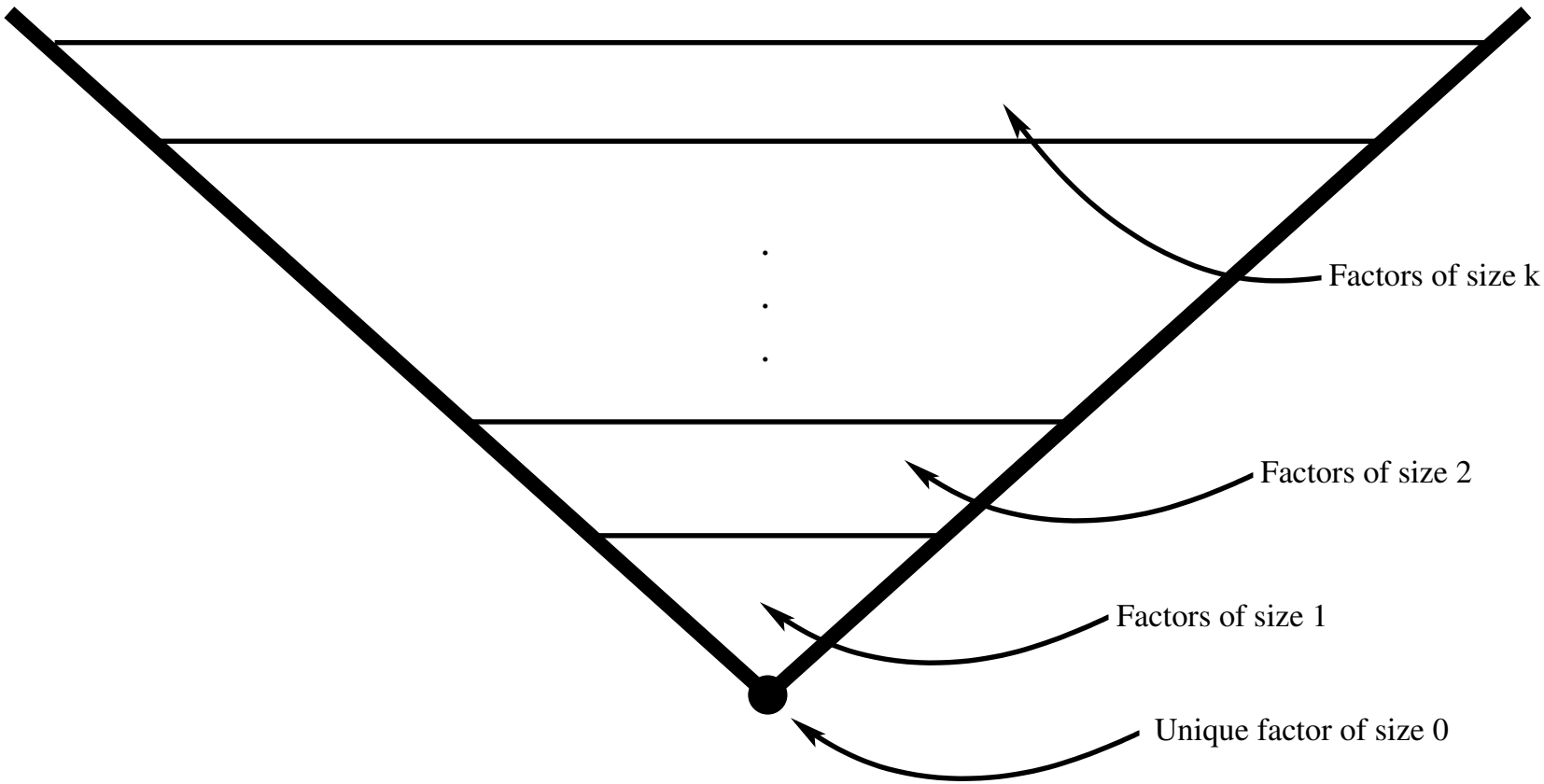
Part VIII

How BUFIA works and Discussion

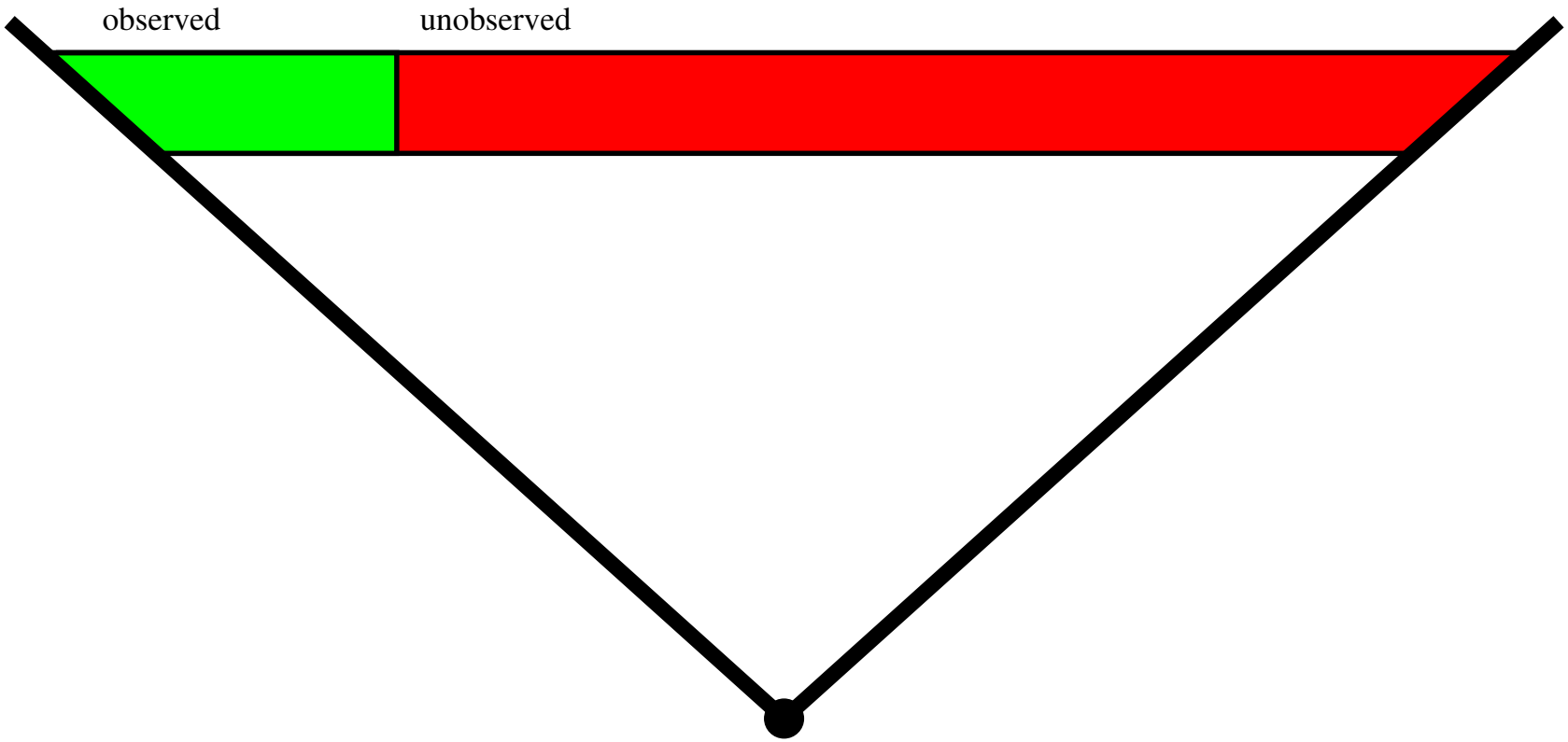
Bottom-Up Traversal of Factor Space

1. Put the most general factor (the one at the bottom) on the Queue.
2. Take a factor C from the Queue and check whether it is present in the observed data.
3. If so, discard it and add its successive superfactors which are not blocklisted to the Queue.
4. If not, add C to the grammar G as a constraint, remove all its superfactors from the Queue and blocklist them.
5. Repeat from step 2.

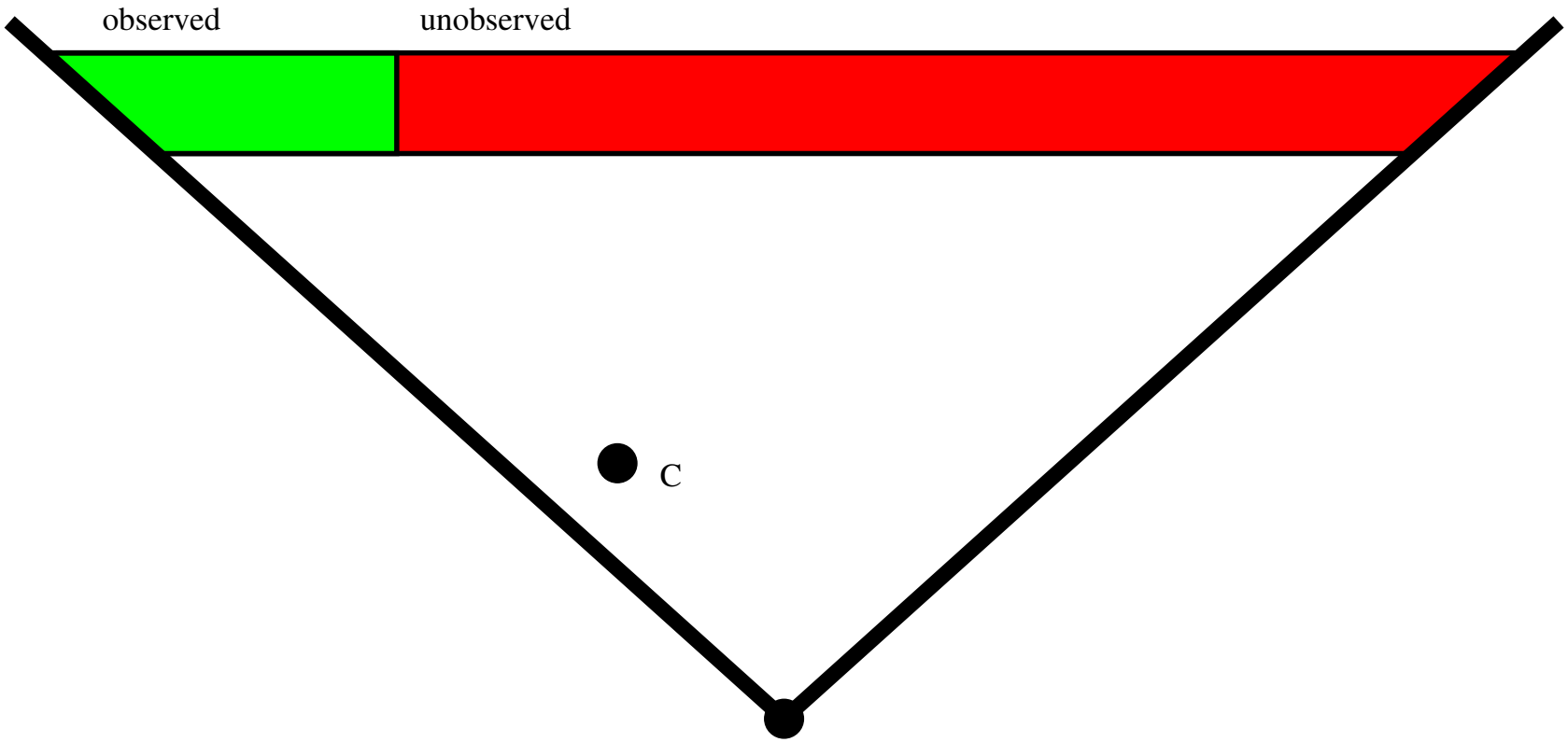
Step 2: Is a factor forbidden? It it added as a constraint to the grammar?)



Step 2: Is a factor forbidden? It it added as a constraint to the grammar?)

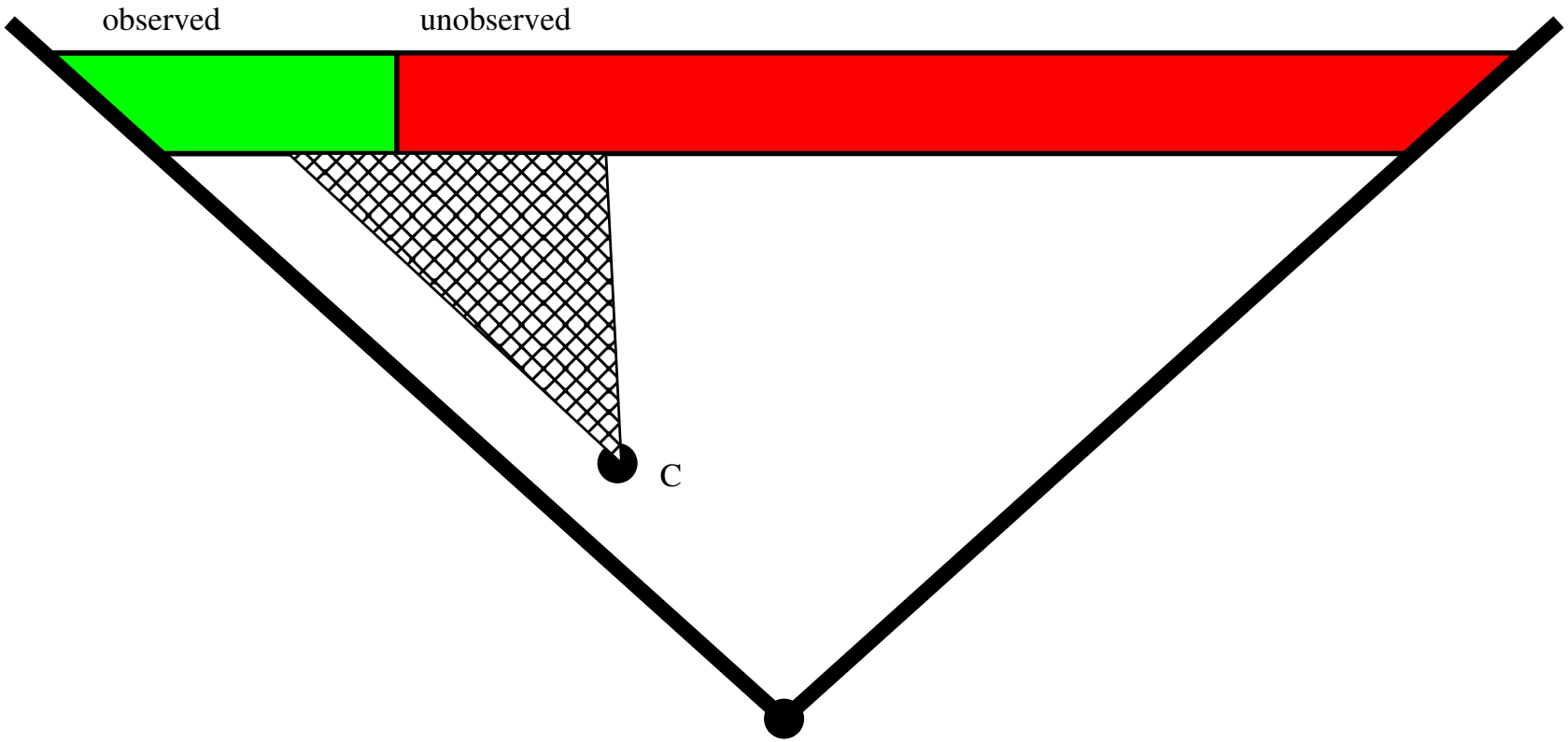


Step 2: Is a factor forbidden? It it added as a constraint to the grammar?)



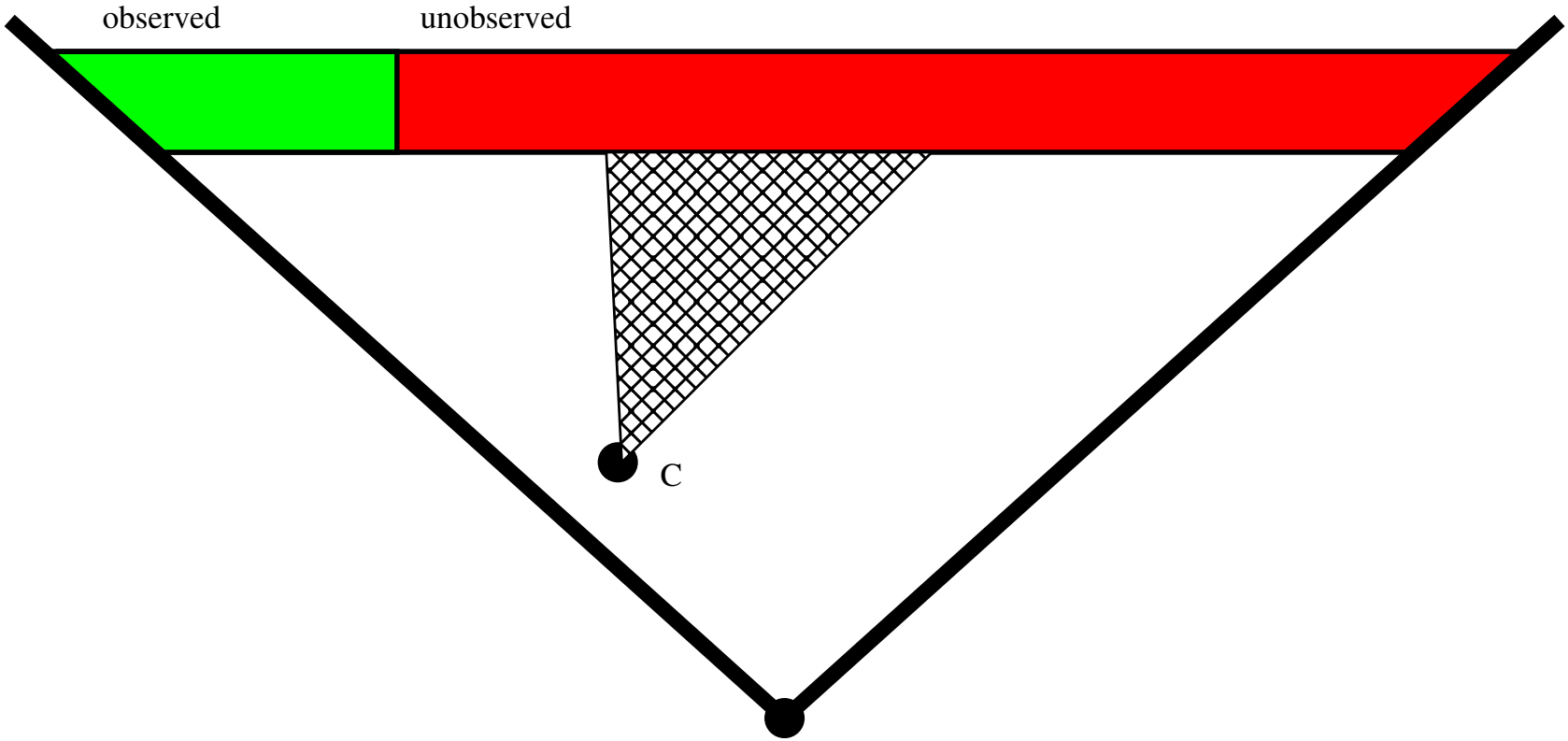
If C is a factor of the observed data, then conclude C is not forbidden.

Step 2: Is a factor forbidden? It it added as a constraint to the grammar?)



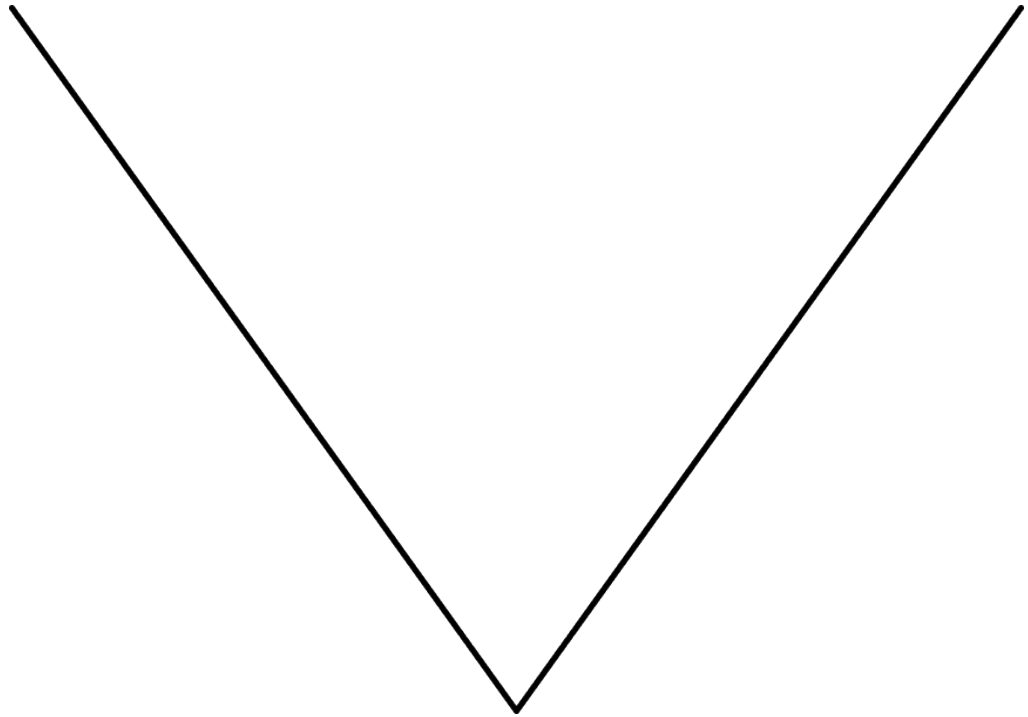
If C is a factor of the observed data, then conclude C is not forbidden.

Step 2: Is a factor forbidden? It it added as a constraint to the grammar?)

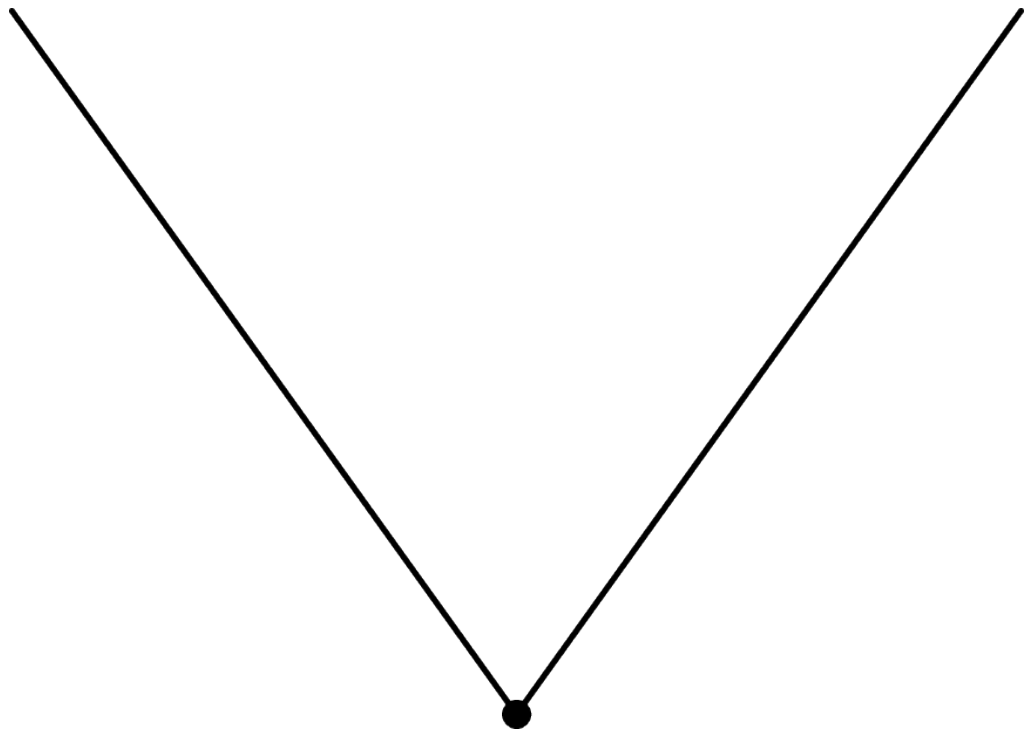


If the observed data does not include C as a factor, then conclude C is forbidden. All superfactors of C can henceforth be ignored!

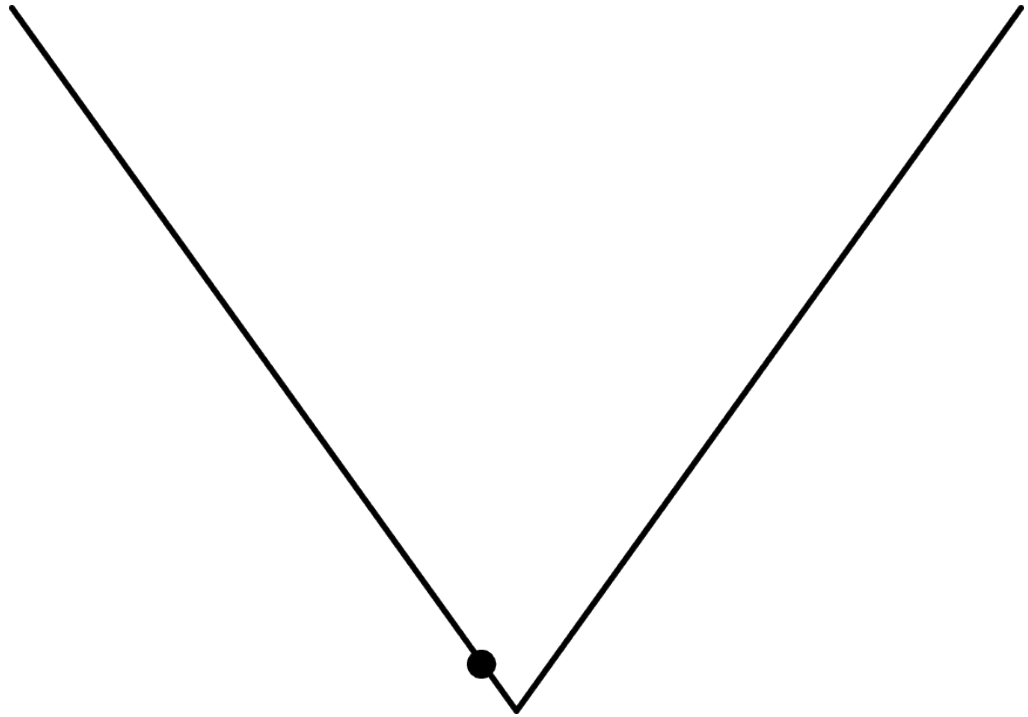
Proceed Bottom-Up in this way (Visualization)



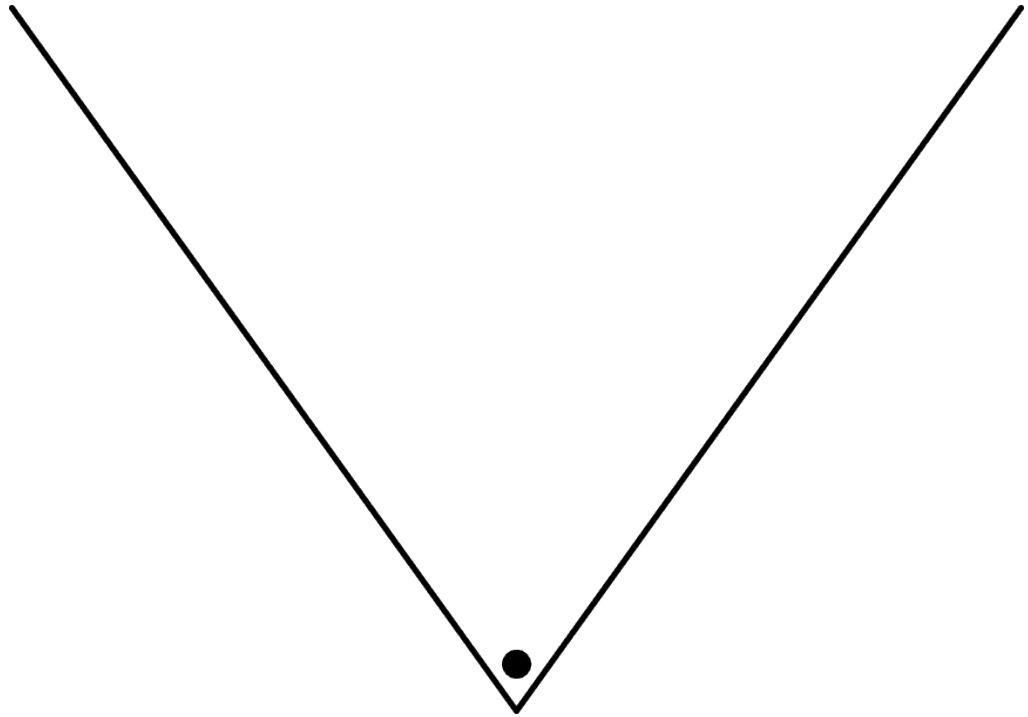
Proceed Bottom-Up in this way (Visualization)



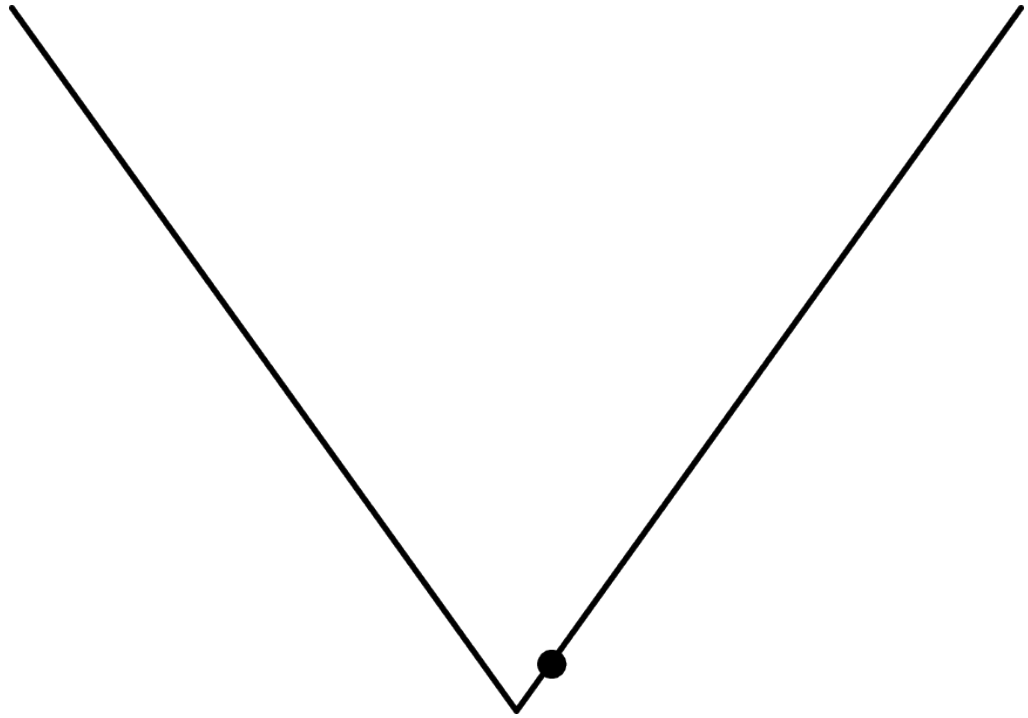
Proceed Bottom-Up in this way (Visualization)



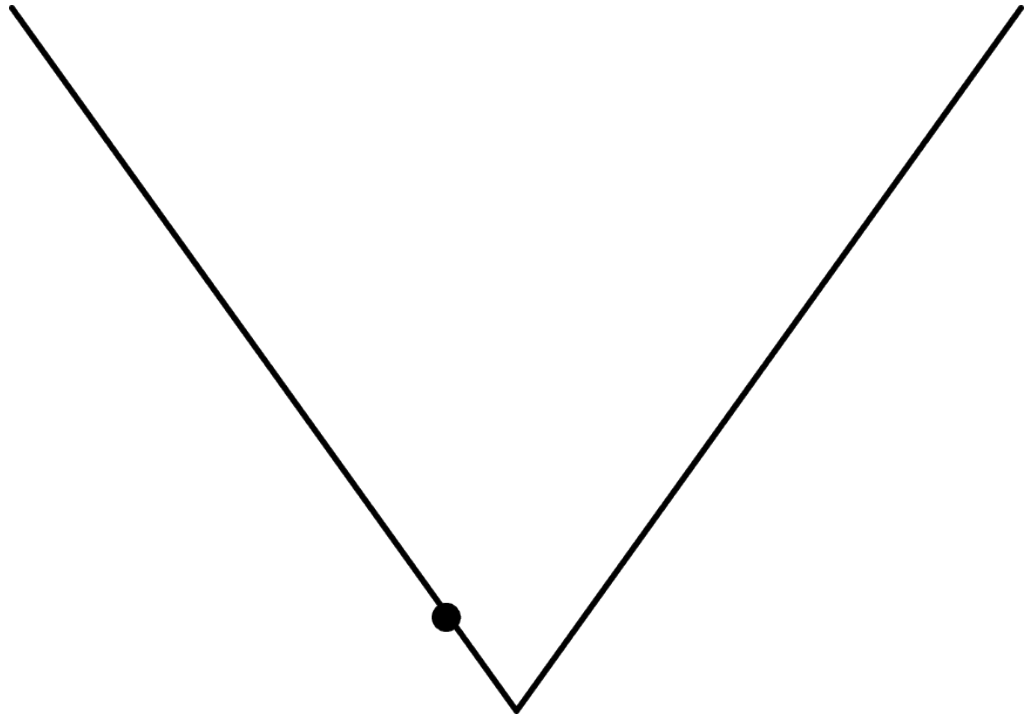
Proceed Bottom-Up in this way (Visualization)



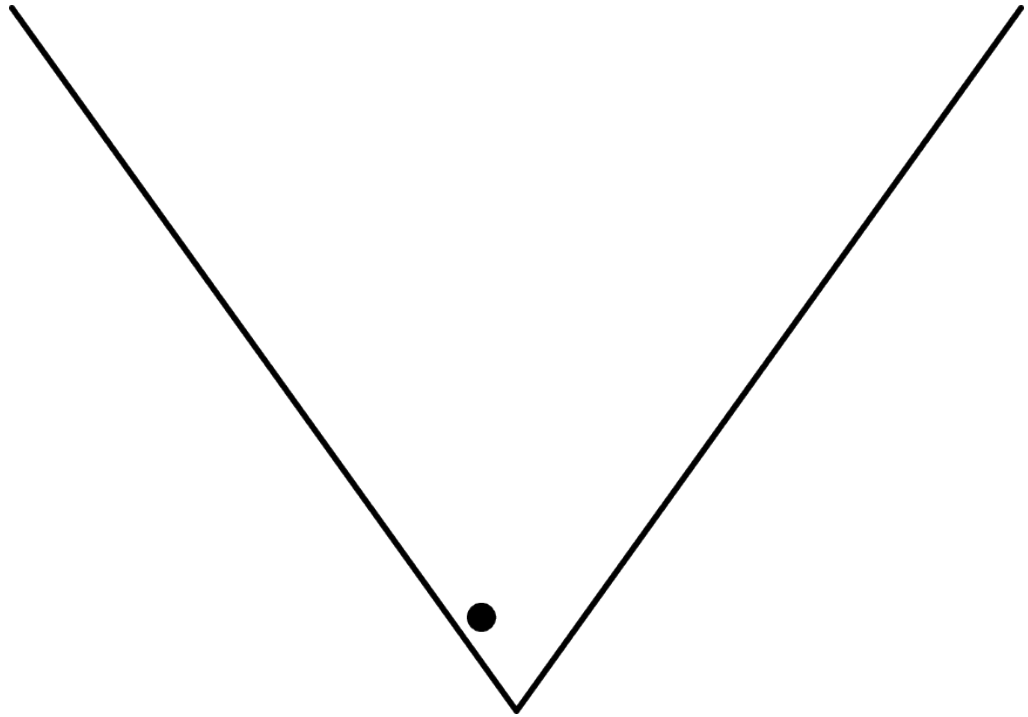
Proceed Bottom-Up in this way (Visualization)



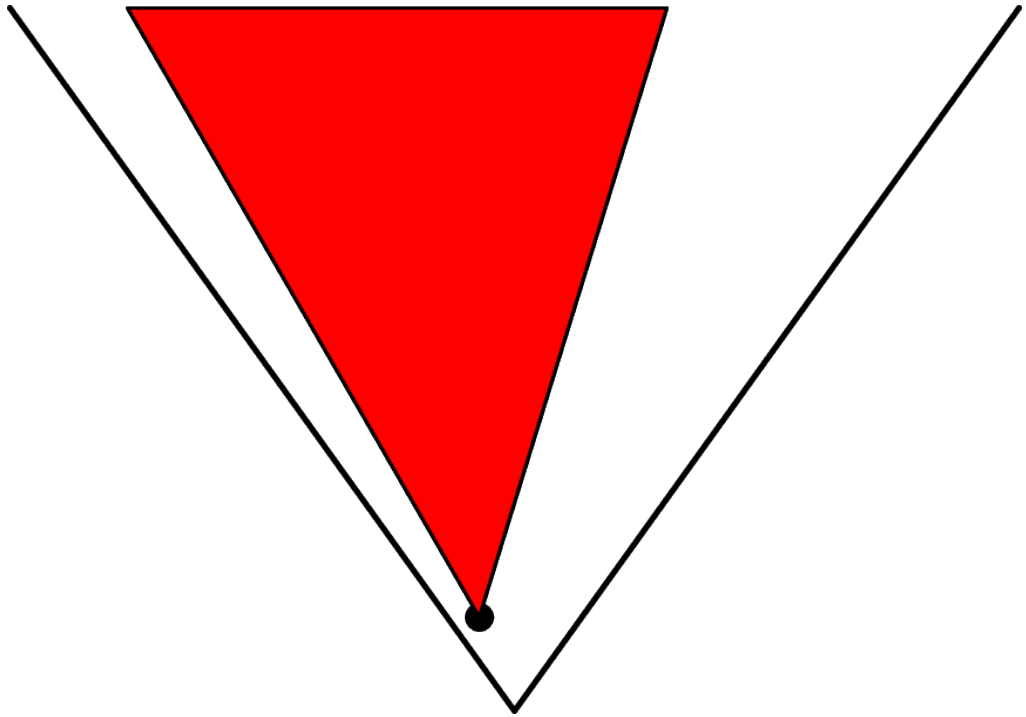
Proceed Bottom-Up in this way (Visualization)



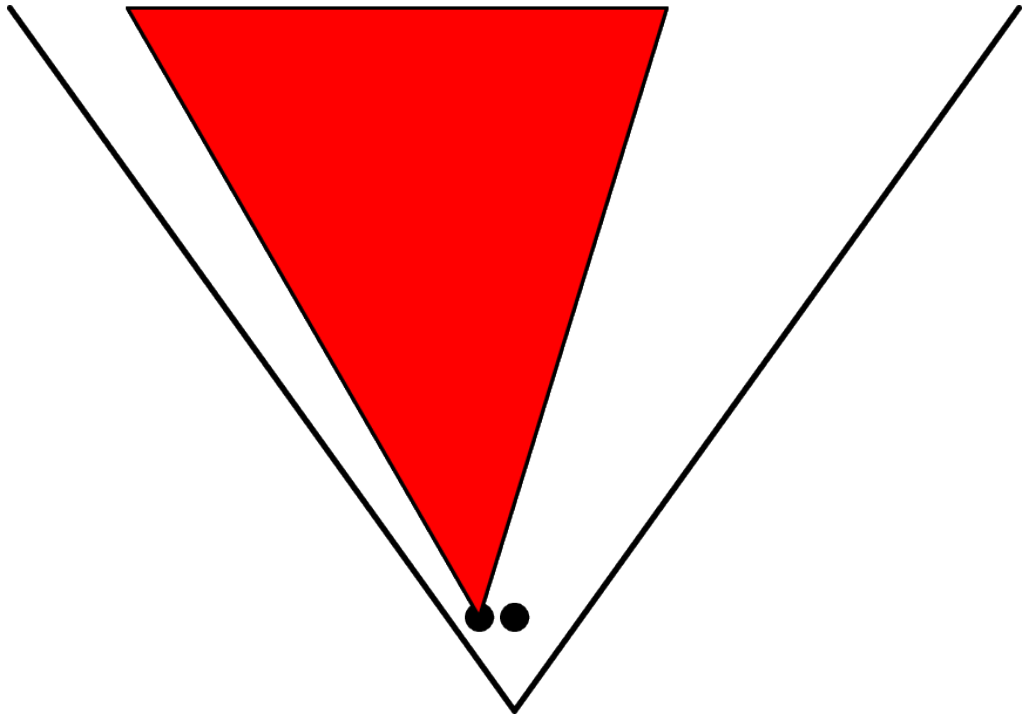
Proceed Bottom-Up in this way (Visualization)



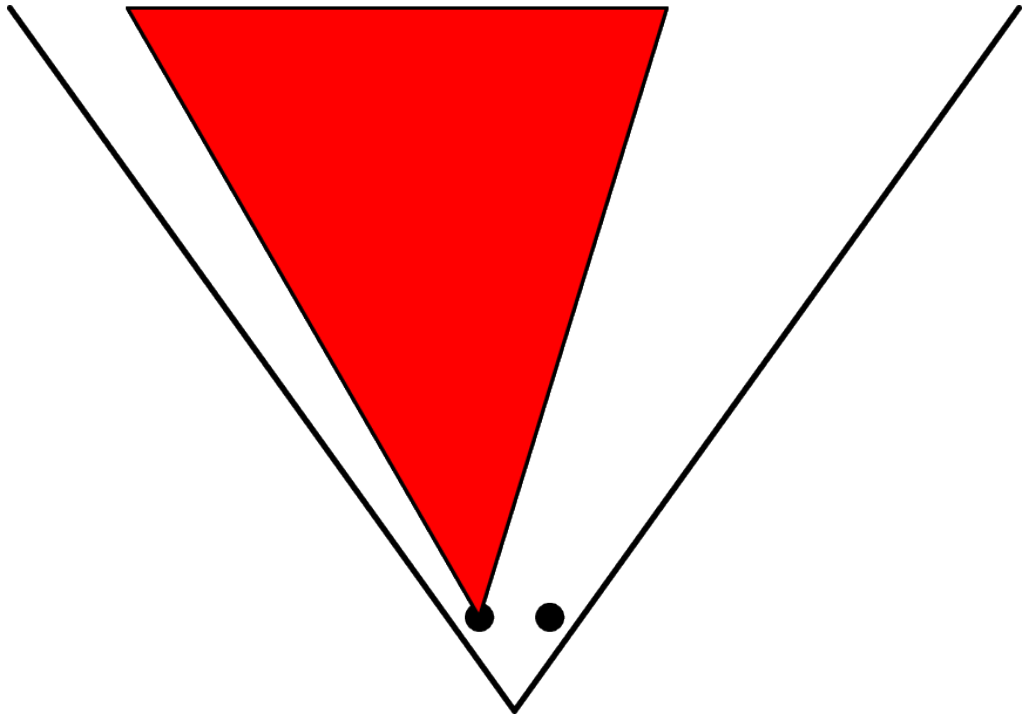
Proceed Bottom-Up in this way (Visualization)



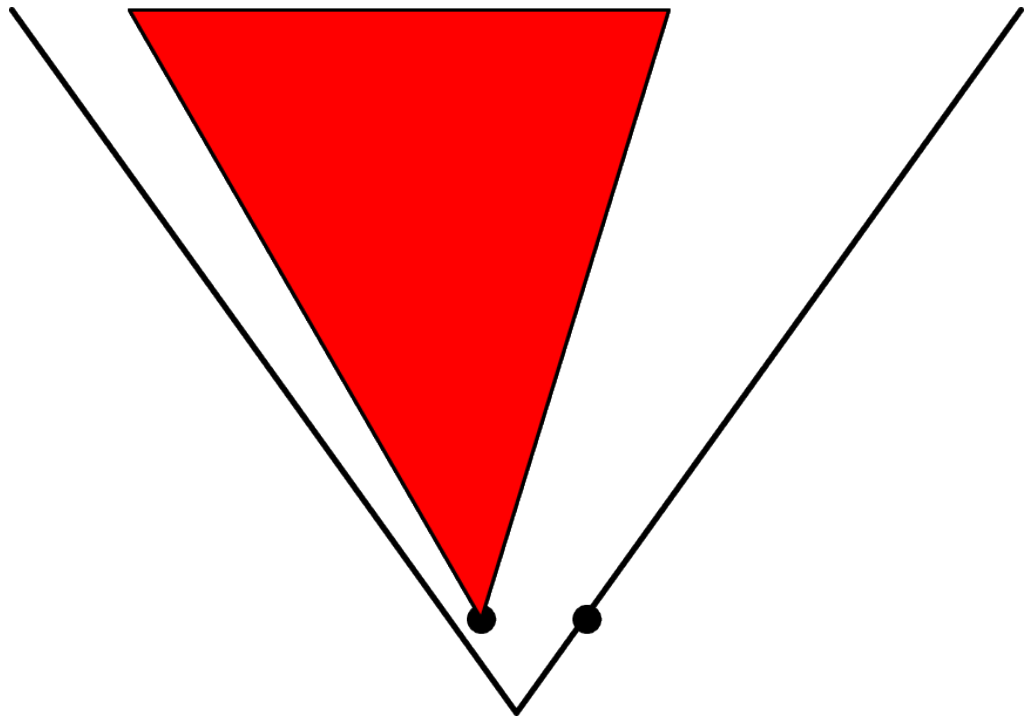
Proceed Bottom-Up in this way (Visualization)



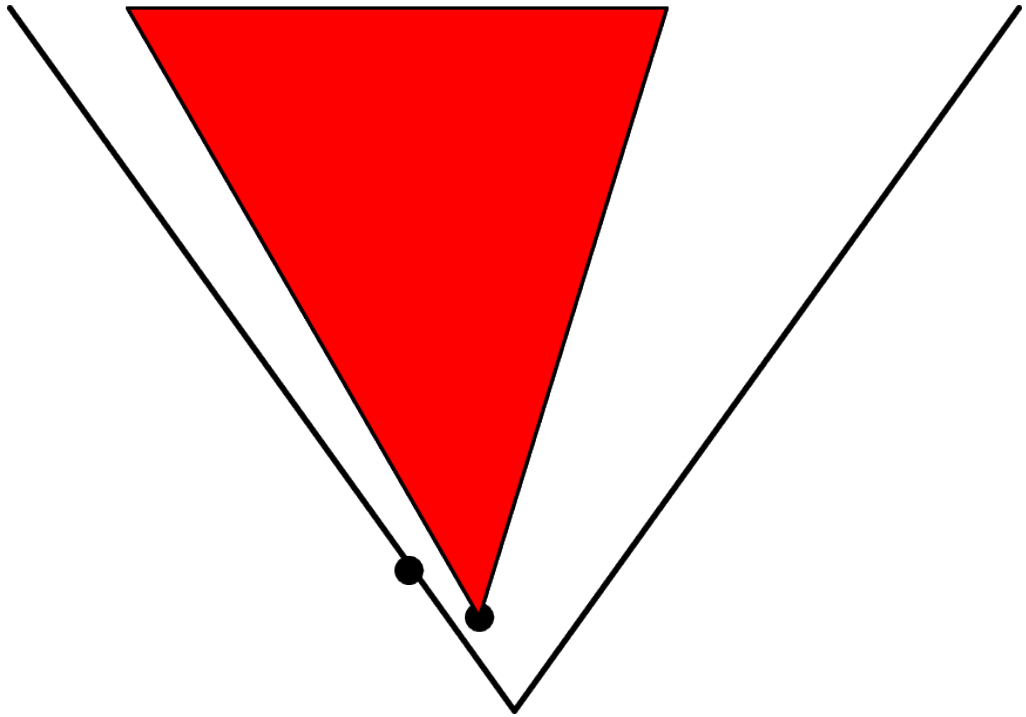
Proceed Bottom-Up in this way (Visualization)



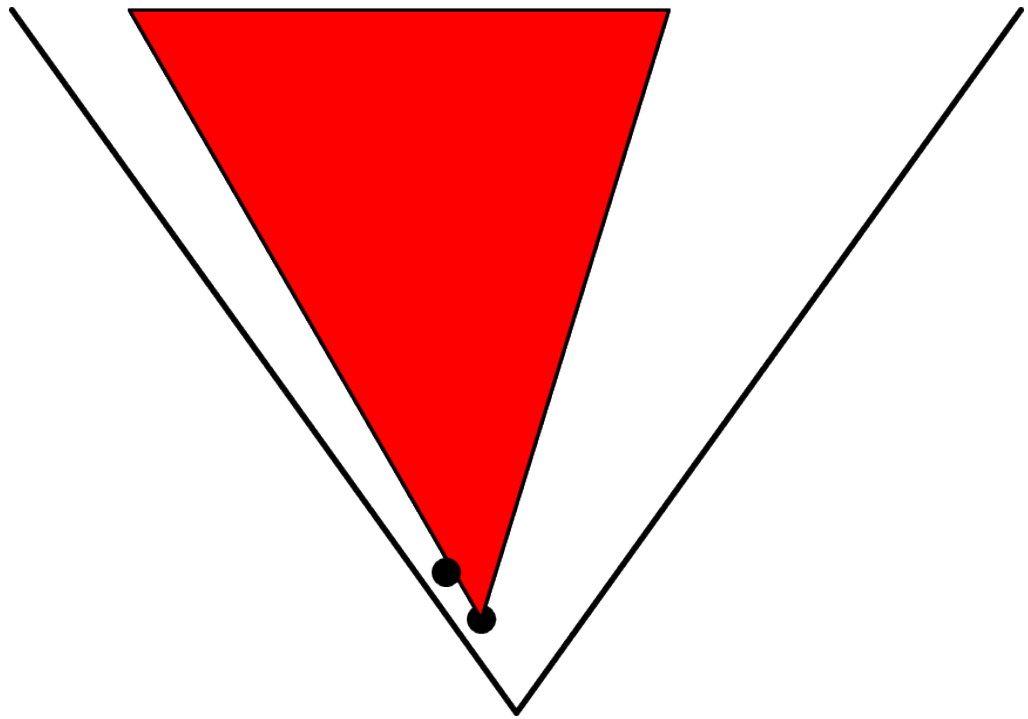
Proceed Bottom-Up in this way (Visualization)



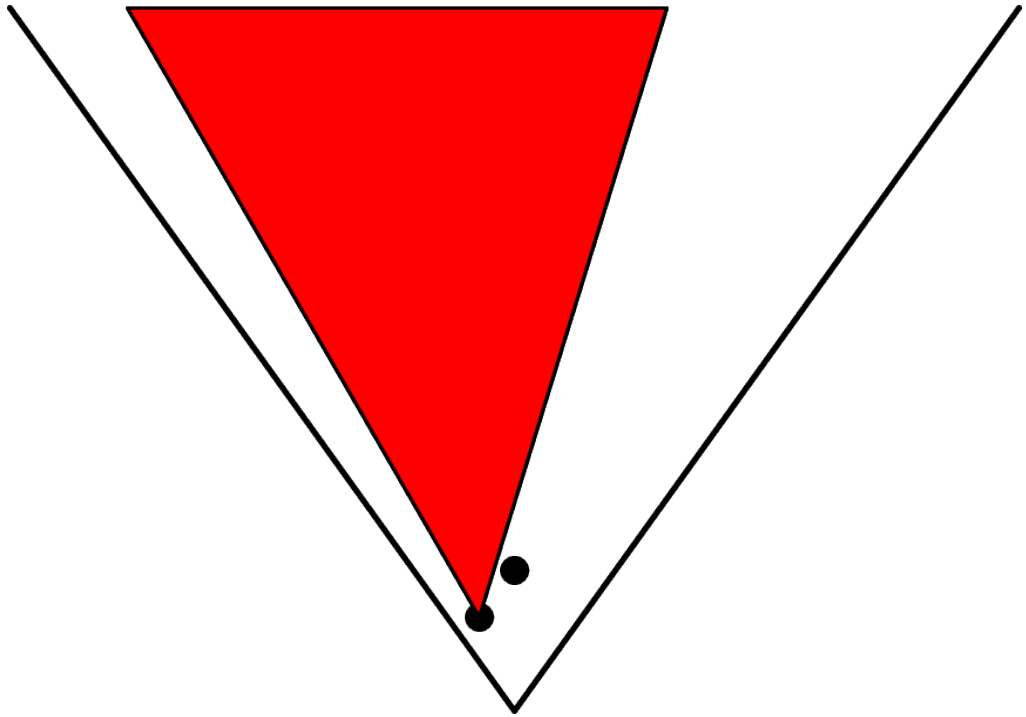
Proceed Bottom-Up in this way (Visualization)



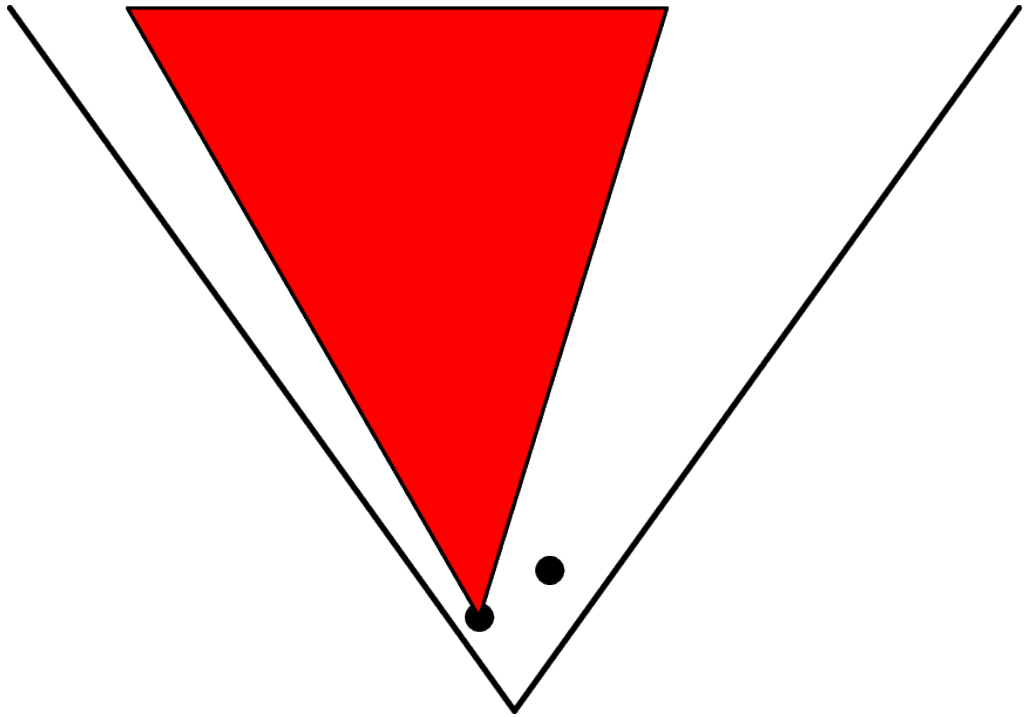
Proceed Bottom-Up in this way (Visualization)



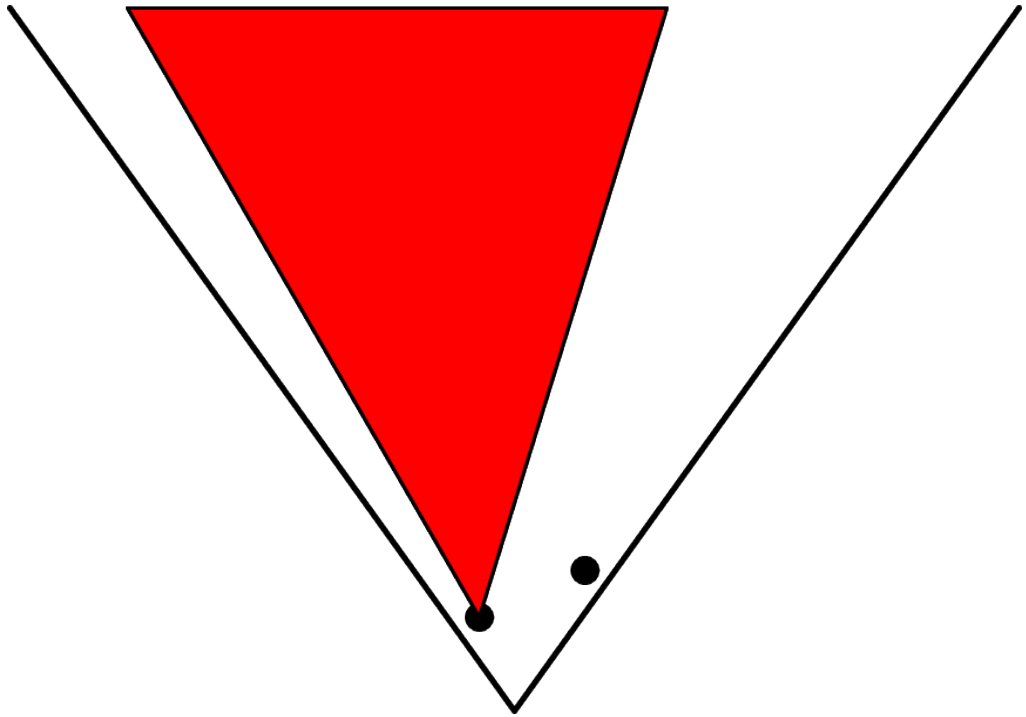
Proceed Bottom-Up in this way (Visualization)



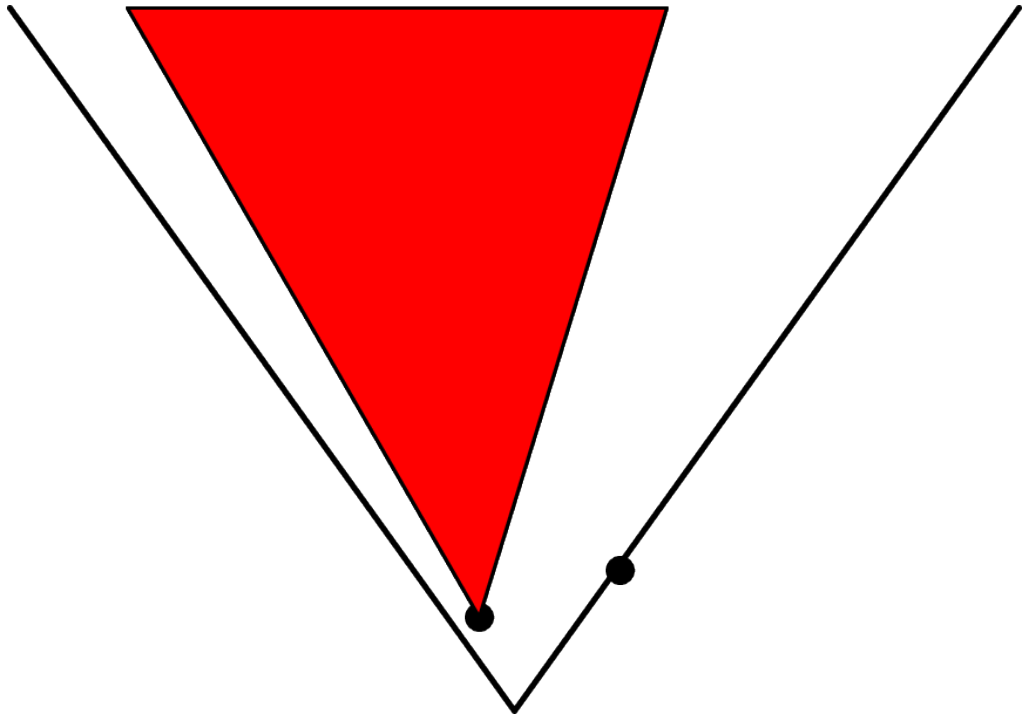
Proceed Bottom-Up in this way (Visualization)



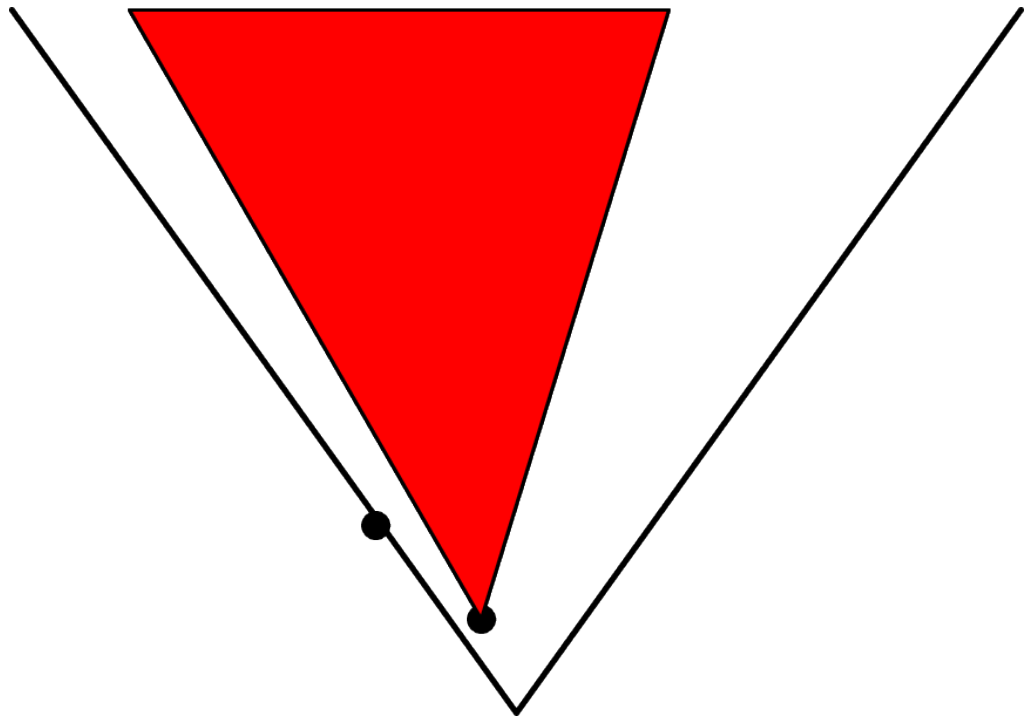
Proceed Bottom-Up in this way (Visualization)



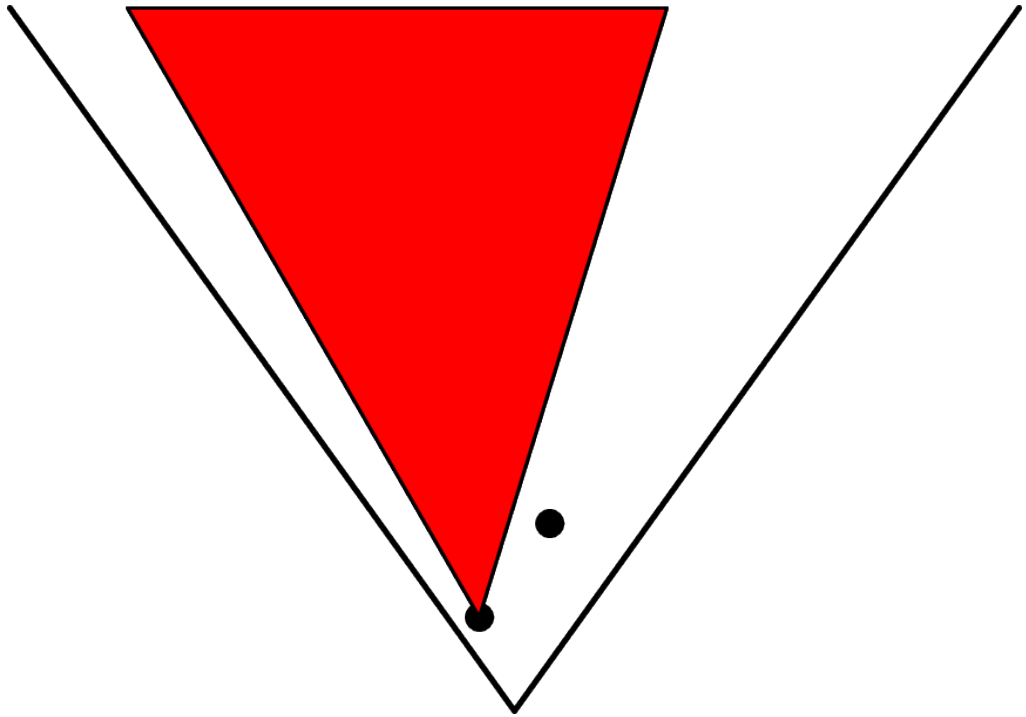
Proceed Bottom-Up in this way (Visualization)



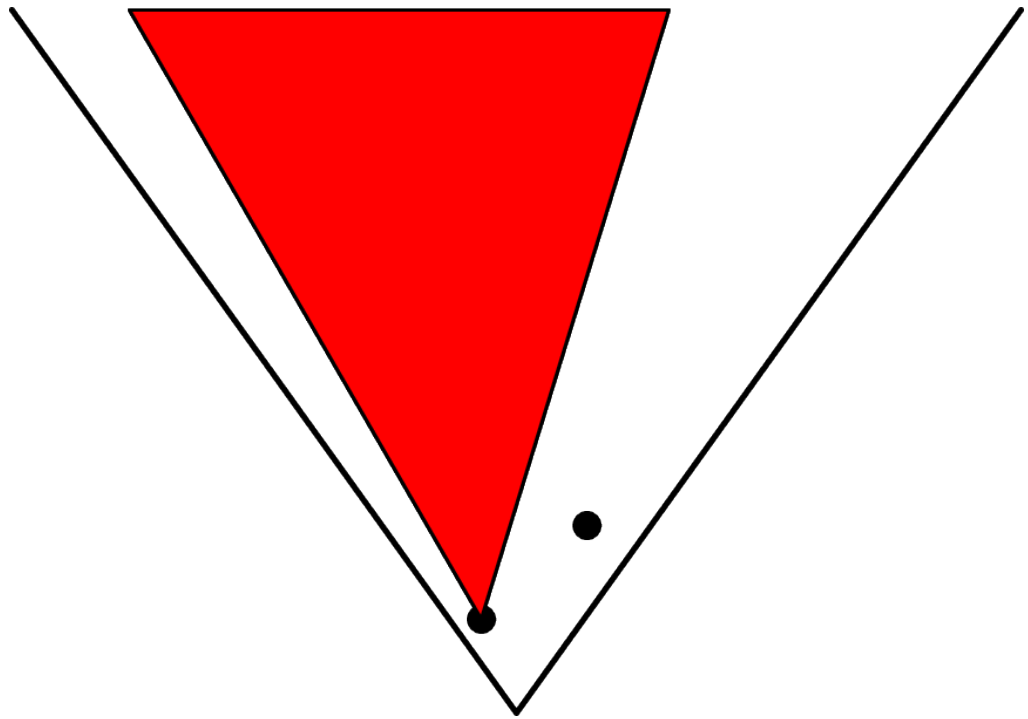
Proceed Bottom-Up in this way (Visualization)



Proceed Bottom-Up in this way (Visualization)



Proceed Bottom-Up in this way (Visualization)



Some Results

Hayes and Wilson's CMU initial cluster data

- 469 forbidden factors with size up to 2 (feature-based successor model)
  e.g. [+nasal][+dorsal]

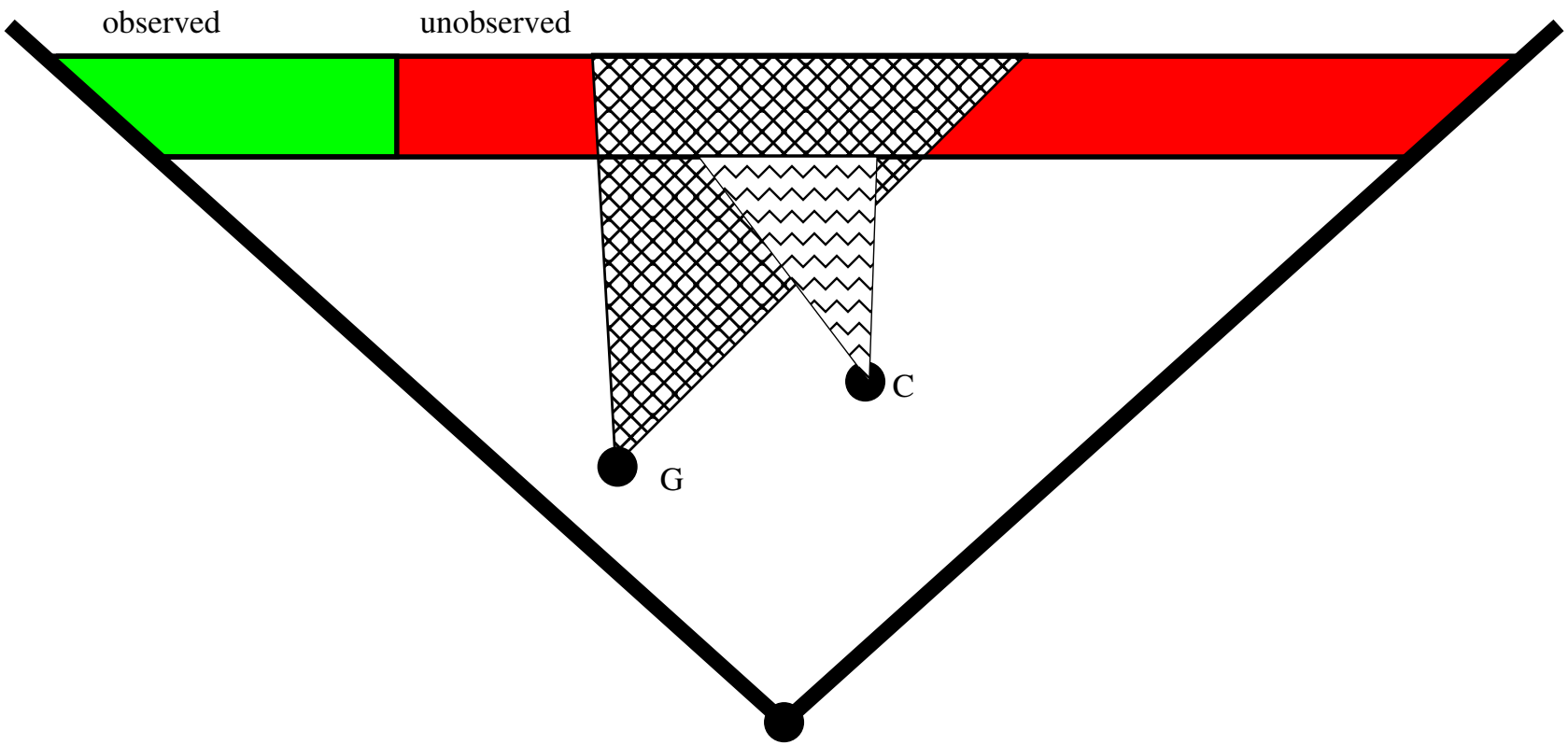
Gallagher's Quechua roots (p.c.)

- 1913 forbidden factors with size up to 2 (feature-based successor model)
  e.g. [+syllabic][+syllabic]
- 320 forbidden factors with size up to 2 (feature-based precedence model)
  e.g. [+cg][+cg]

The forbidden factors in these sets are pairwise incomparable!
Why so many?

Redundancy in the grammar

The fact that there are so many feature combinations means many incomparable constraints are "accomplishing the same thing."



The extension of C is already covered by G.

(Haig 2018)

What are we doing?

1. We assume observed factors are permissible.
2. We assume each unobserved factor warrants explanation.
3. The explanation we assume we seek is some constraint forbidding it.
4. Once we have one explanation, do we need another?

Here is as simple example illustrating what happens.

	i	u	e	o	a
high	+	+	-	-	-
back	-	+	-	+	-
low	-	-	-	-	+

Learning Data: { ii, aa }

k = 2

Hayes and Wilson (MaxEnt)

[+back]	6.186
[-low, -high]	2.162
[-low][-high]	5.766
[-high][-low]	5.766

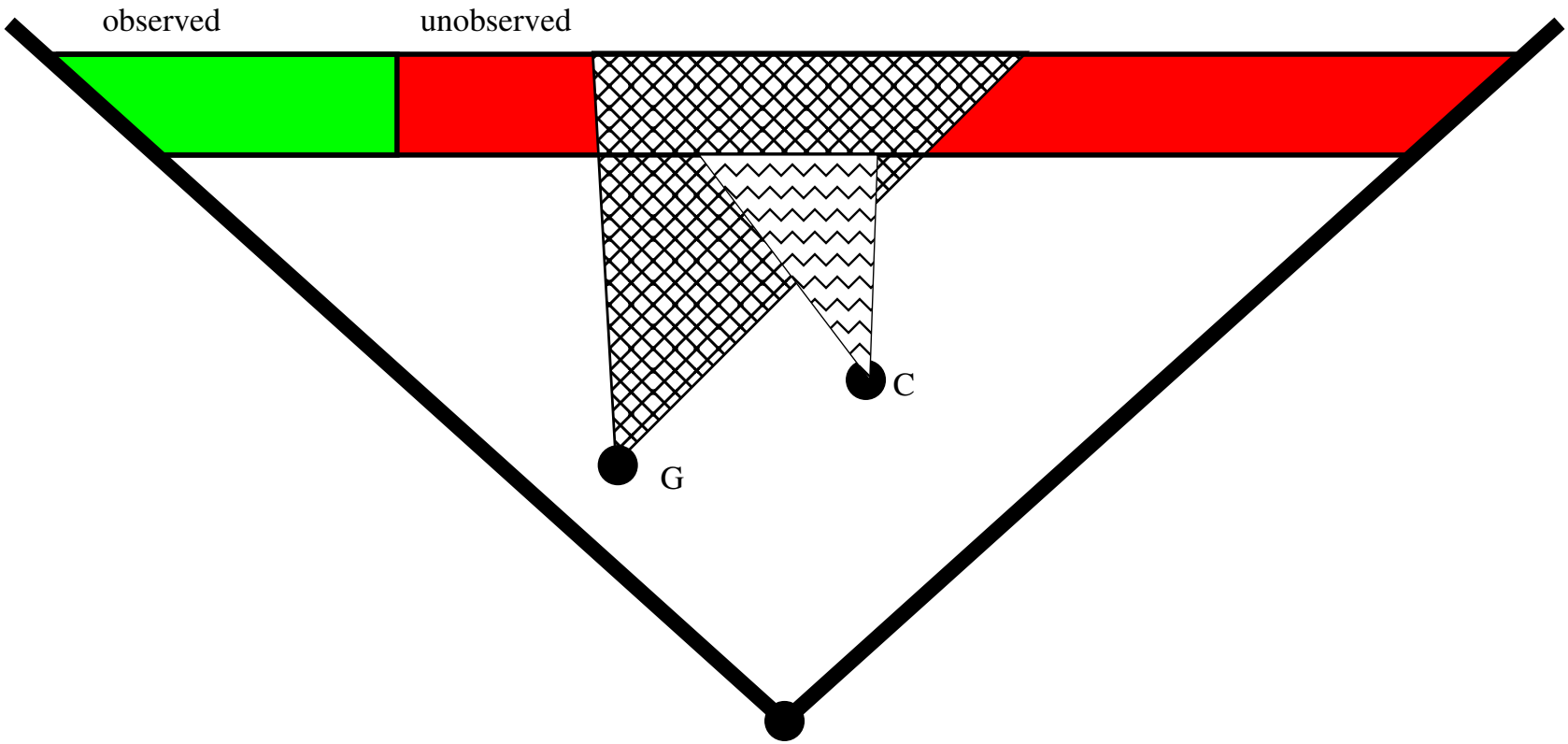
Here is as simple example illustrating what happens.

	i	u	e	o	a
high	+	+	-	-	-
back	-	+	-	+	-
low	-	-	-	-	+

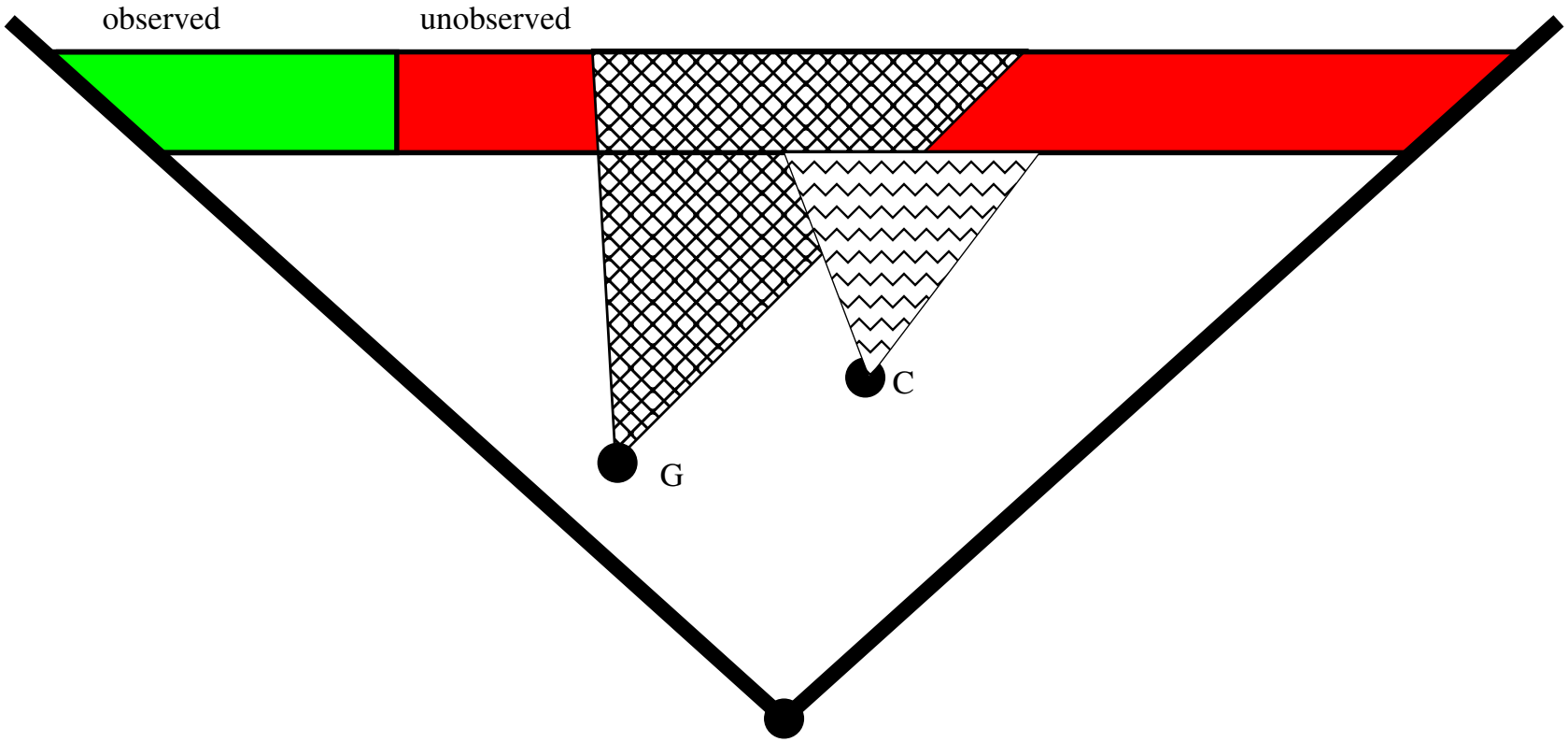
Learning Data: { ii, aa }
k = 2

Chandlee et al. (BUFIA)	
[+back]	[-high][-low]
[-high, -low]	[+low][+high]
[+high][-high]	[+low][-low]
[+high][+low]	[-low][-high]
[-high][+high]	[-low][+low]

Candidate Abductive Principles

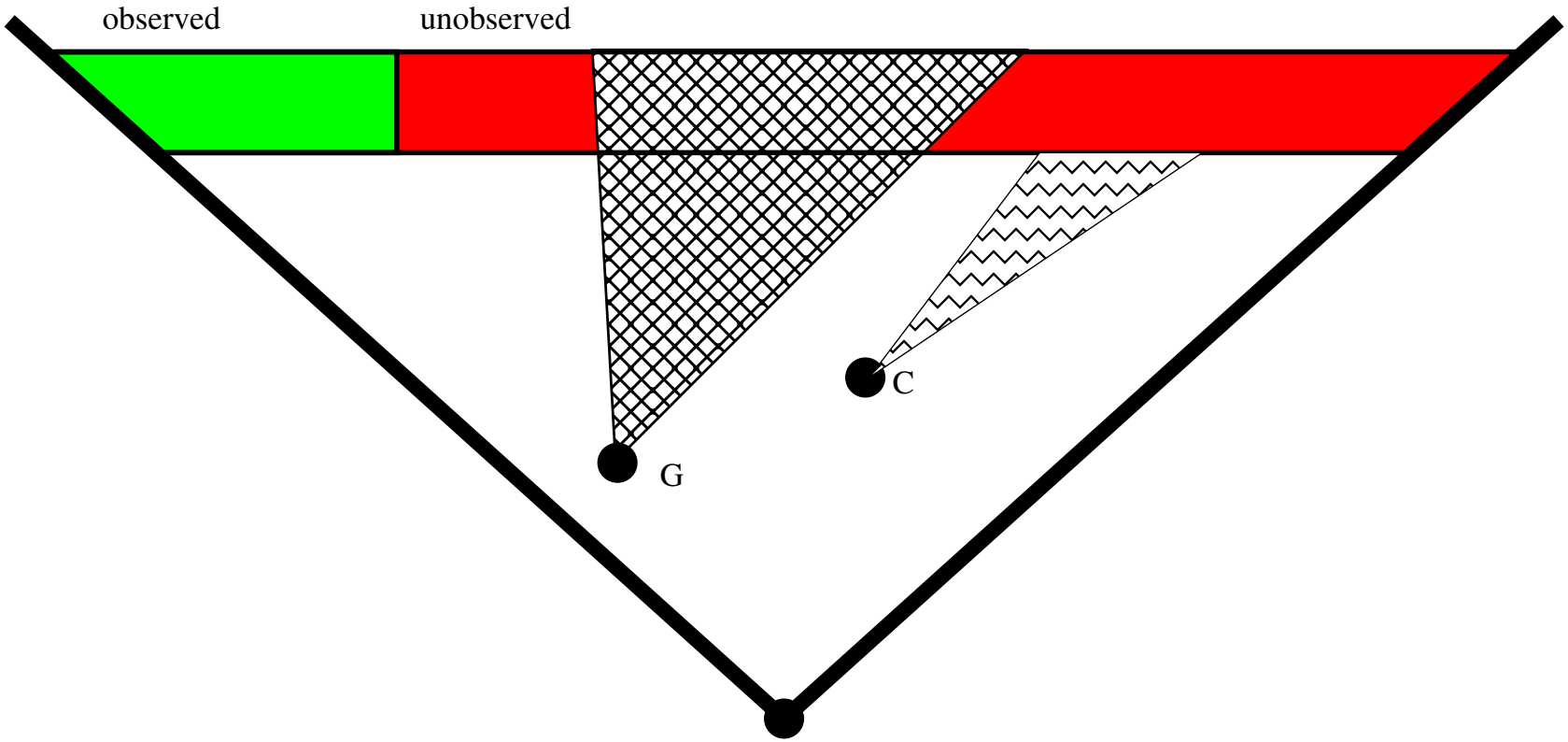


Candidate Abductive Principles



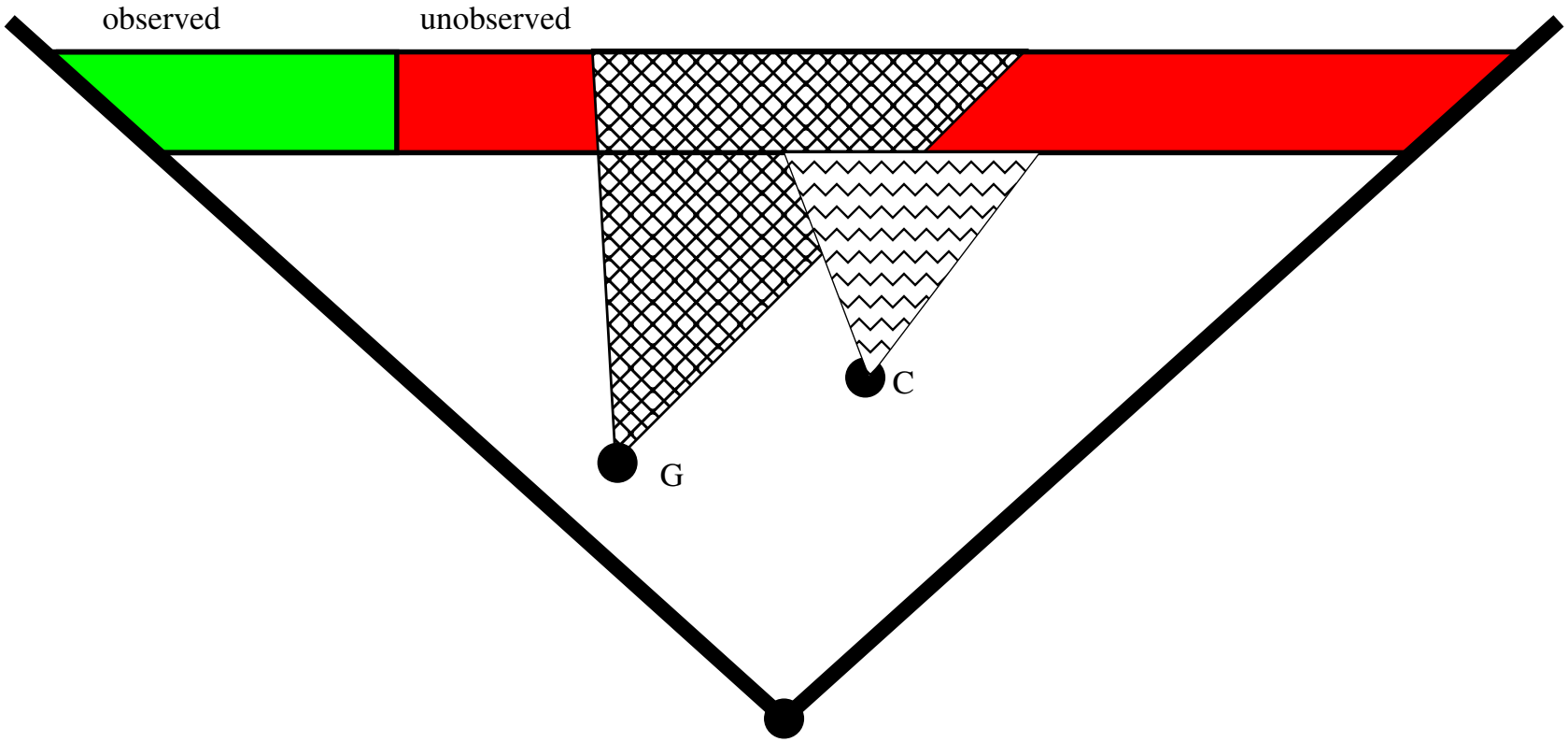
Principle 1: Add C to G only if C accounts for **any** unobserved factors that are not superfactors of G

Candidate Abductive Principles



Principle 2: Add C to G only if **all** the unobserved factors C accounts for are not superfactors of G.

Candidate Abductive Principles



The "gain" in Wilson and Gallagher 2018 is a statistical abductive principle that also navigates these issues.

Adding Abductive Principles to BUFIA

	i	u	e	o	a
high	+	+	-	-	-
back	-	+	-	+	-
low	-	-	-	-	+

Learning Data: { ii, aa }

Chandlee et al. (BUFIA + P1)

[+back]
[-high, -low]
[+high][-high]
[-high][+high]

Adding Abductive Principles to BUFIA

	i	u	e	o	a
high	+	+	-	-	-
back	-	+	-	+	-
low	-	-	-	-	+

Learning Data: { ii, aa }

Chandlee et al. (BUFIA + P2)
[+back]
[-back][-back, -high, -low]
[+low][-back, +high]
[-back, +high][+low]
[-back, -high, -low][+low]
[-back, -high, -low][-back, +high]

Inductive vs Abductive Principles

- The structure of the constraint space lets us induce general, incomparable constraints that are surface true (without statistics) up to some size.
- Abductive principles are used to determine which of these constraints ought to make up the grammar (what the best explanations are).
- In this way, we can understand exactly why the grammar we learn is what it is.
- This is not about statistics vs. structure: it is about understanding the different roles and contributions each can play with respect to the problems of induction and abduction.

English and Quechua again, BUFIA + P1

Hayes and Wilson's CMU initial cluster data

- 469 → 32
  forbidden factors with size up to 2
  (feature-based successor model)

Gallagher's Quechua roots (p.c.)

- 1913 → 89
  forbidden factors with size up to 2
  (feature-based successor model)
- 320 → 22
  forbidden factors with size up to 2
  (feature-based precedence model)

Part IX

Conclusion

Current/Future Work

- Run BUFIA with different abductive principles on different corpora.
- Whenever possible, examine learned grammars with speaker judgments and compare with other phonotactic learning models (Durvasula 2020, AMP)
- Explore constraint learning with BUFIA with other kinds of representations (autosegmental, syllable structure, . . .)
- Combine BUFIA with a syntactic parser for learning constraints on syntactic trees.

Minicourse Summary – THANKS!

1. Finite Model Theory provides a unified language for representing various linguistic structures.
2. Factors are "parts" of structures.
3. Grammars detail how a structure's well-formedness is based on its parts.
4. Grammars and representations together gives us an encyclopedia of categories of constraint types.
5. The feasibly learnable constraint types are the simpler ones which make fewer distinctions.
6. At least in phonology, that's where the markedness constraints appear to be.
7. The factors are *partially ordered*. This *structure* helps learning a lot!
8. It is important to be clear about both inductive and abductive principles when studying language learning.