How the Structure of the Constraint Space Facilitates Learning

Jonathan Rawski, Jeffrey Heinz, Adam Jardine, Jane Chandlee

North American Phonology Conference
May 5, 2018

This work was supported by NIH under grant #R01HD87133-01

The Talk in a Nutshell

Previously on this Topic

- Efficient Learning of Segmental Phonotactics and Mappings
- Question: How to extend these learners for feature-based constraints? (Cf. Hayes and Wilson 2008)

Today We

- Describe the lattice structure of the space of feature-based constraints
- Show how Learners can utilize this lattice to generalize constraints.

The Talk in a Nutshell

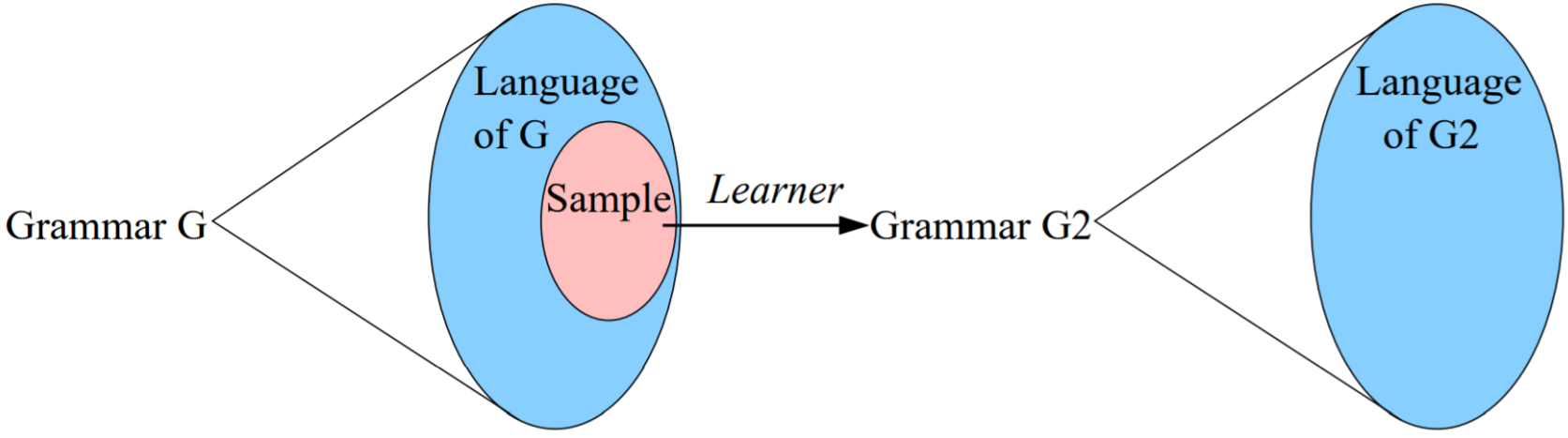
Previously on this Topic

- Efficient Learning of Segmental Phonotactics and Mappings
- Question: How to extend these learners for feature-based constraints? (Cf. Hayes and Wilson 2008)

Today We

- Describe the lattice structure of the space of feature-based constraints
- Show how Learners can utilize this lattice to generalize constraints.

The View from Learning Theory



- Learning cannot take place without a restricted hypothesis space.
- G2 is not drawn from an unrestricted set of possible grammars.
- Hypotheses available to the learner ultimately determine:
  - the kinds of generalizations made
  - the range of possible natural language patterns

(Nowak et al. 2002, Niyogi 2006, Jain et al. 1999, Osherson et al. 1986)

Learning with Locality

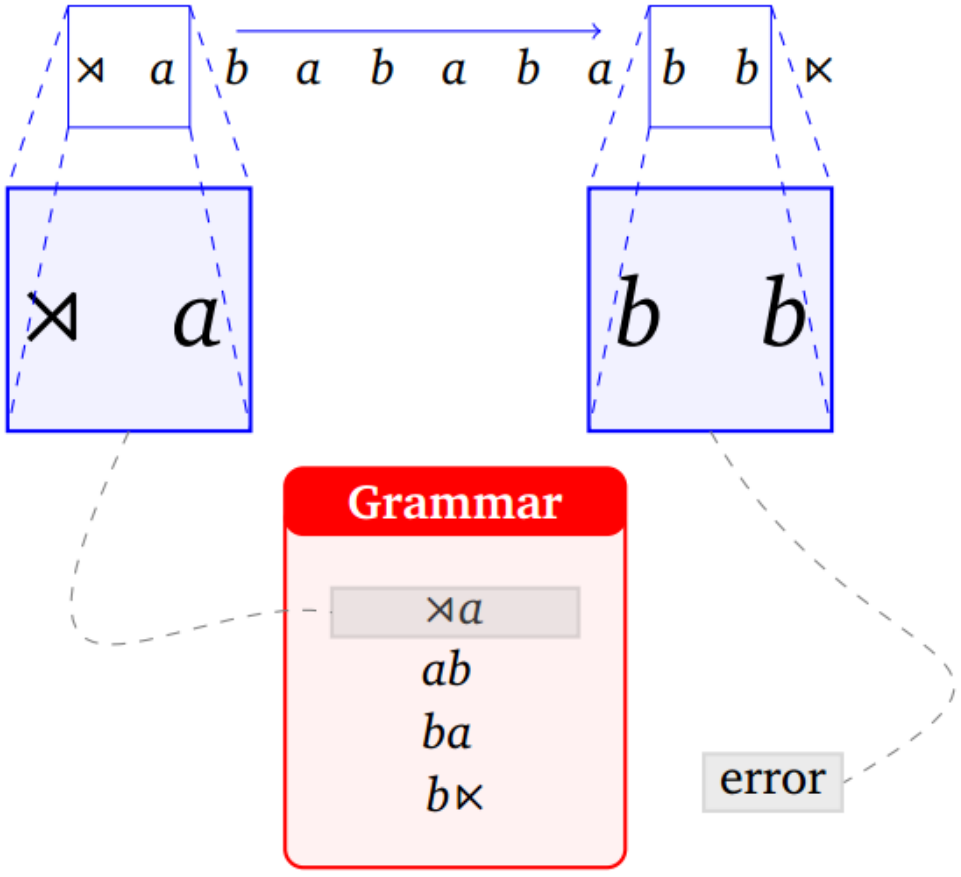


Figure: 1 window of size k

(Garcia et al. 1991, Heinz 2010)

Learning with Precedence

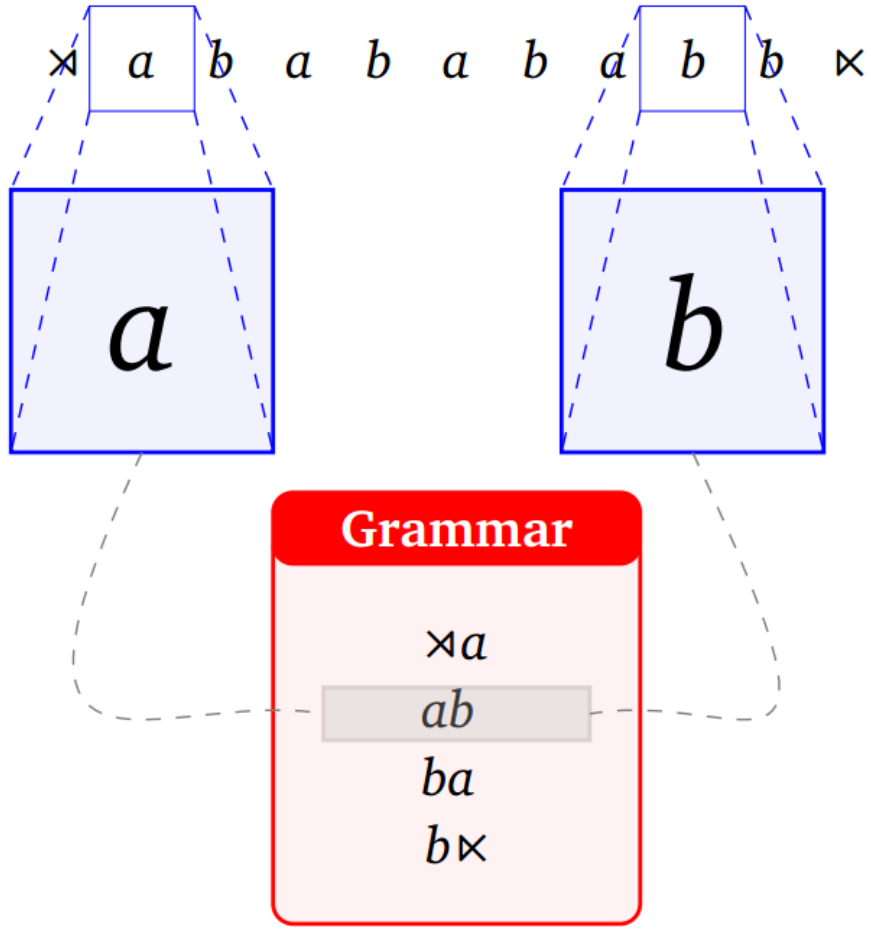


Figure: k windows of size 1

(Heinz 2010)

The Challenge of Features

Wilson & Gallagher in press

"Could there be a non-statistical model like [Heinz's] that learns by memorizing feature sequences? The problem confronting such a model is that any given segment sequence has may different featural representations. Without a method for deciding which representations are relevant for assessing wellformedness (the role that statistics plays in Maxent-Ftr) **learning is doomed**."

Example

Suppose the sequence *nt* is absent from a corpus. There are many possible constraints that could explain its absence:

*nt
*[+nasal][+coronal]
*[+consonant][+coronal,-continuant]
*[+sonorant][-sonorant]
....

How can a learner decide which of these constraints is responsible for the absence of *nt*?

Example

Suppose the sequence *nt* is absent from a corpus. There are many possible constraints that could explain its absence:

*nt

*[+nasal][+coronal]

*[+consonant][+coronal,-continuant]

*[+sonorant][-sonorant]

....

How can a learner decide which of these constraints is responsible for the absence of *nt*?

Example

Suppose the sequence *nt* is absent from a corpus. There are many possible constraints that could explain its absence:

*nt

*[+nasal][+coronal]

*[+consonant][+coronal,-continuant]

*[+sonorant][-sonorant]

....

How can a learner decide which of these constraints is responsible for the absence of *nt*?

Hayes & Wilson 2008

- Given an innate feature system, order a list of possible featural constraints by constraint length and generality.
- Input a batch of feature bundle strings as learning data
- Use MaxEnt and Observed/Expected ratios to discover the most general constraints.

Why no structure in the constraint space?

- The nature of features gives a particular order and structure to the space of possible constraints.
- Let the learner exploit this structure as much as possible when making inferences from data.
- Use its size to our advantage!

Feature Extension

Definition (Feature Extensions)

Let $\mathbf{s}$ and $\mathbf{t}$ be segments represented as bundles of n-ary features.

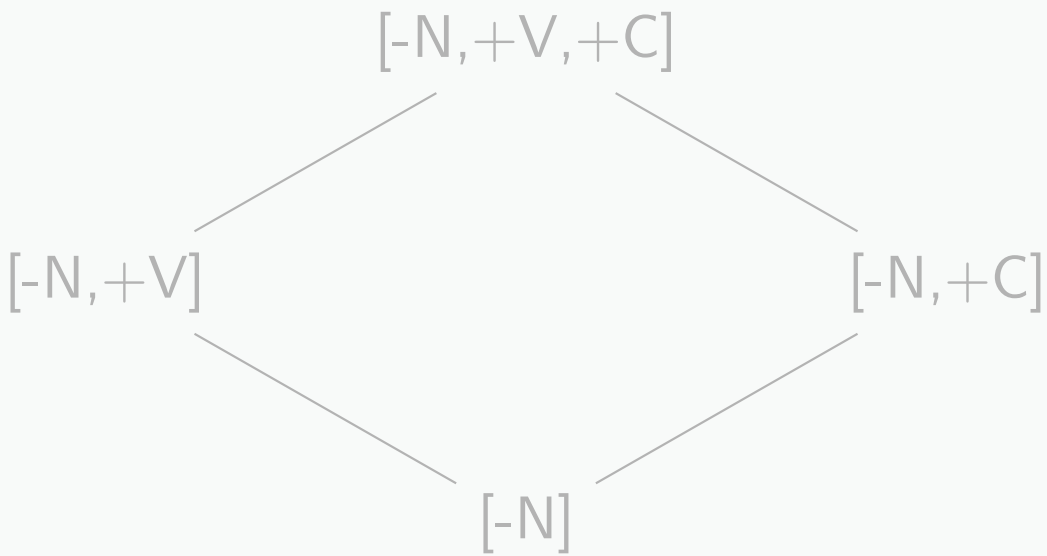
Then $\mathbf{t}$ is an **feature extension** of $\mathbf{s}$ for grammar G ($\mathbf{s} <_G \mathbf{t}$) iff $\mathbf{t}$ is the result of inserting one or more n-ary features of G in $\mathbf{s}$.

Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

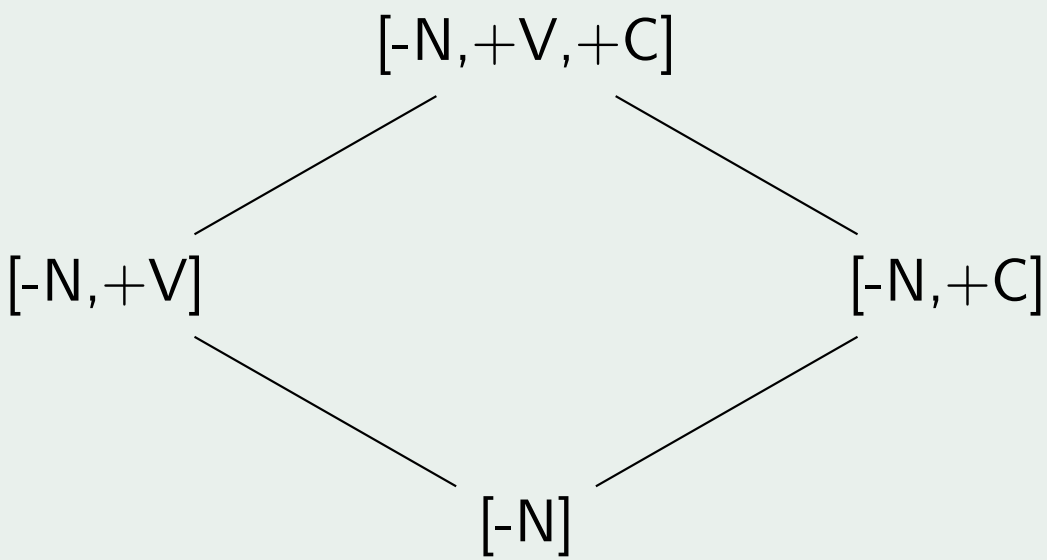


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

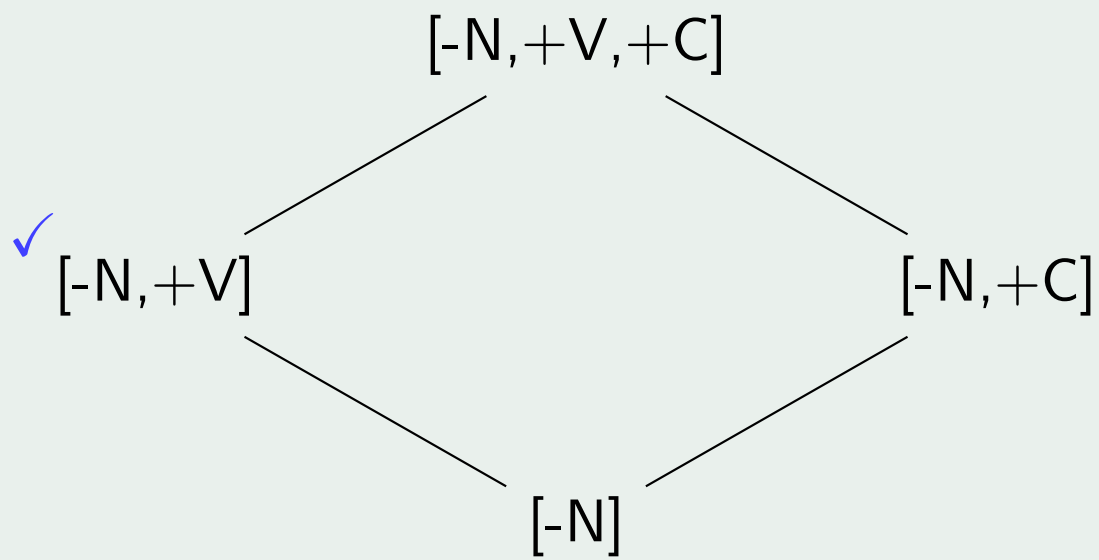


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

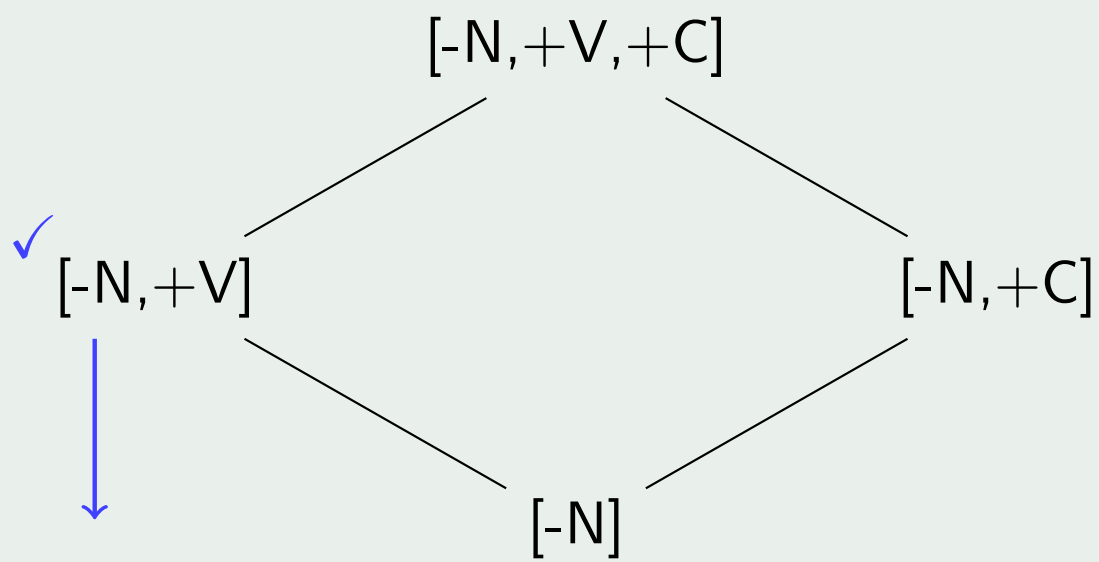


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

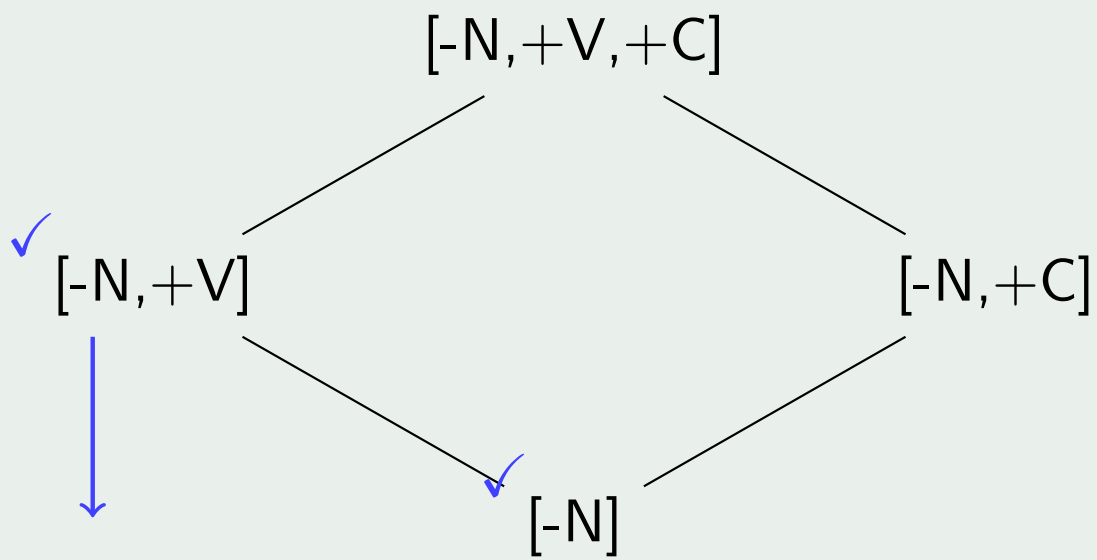


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

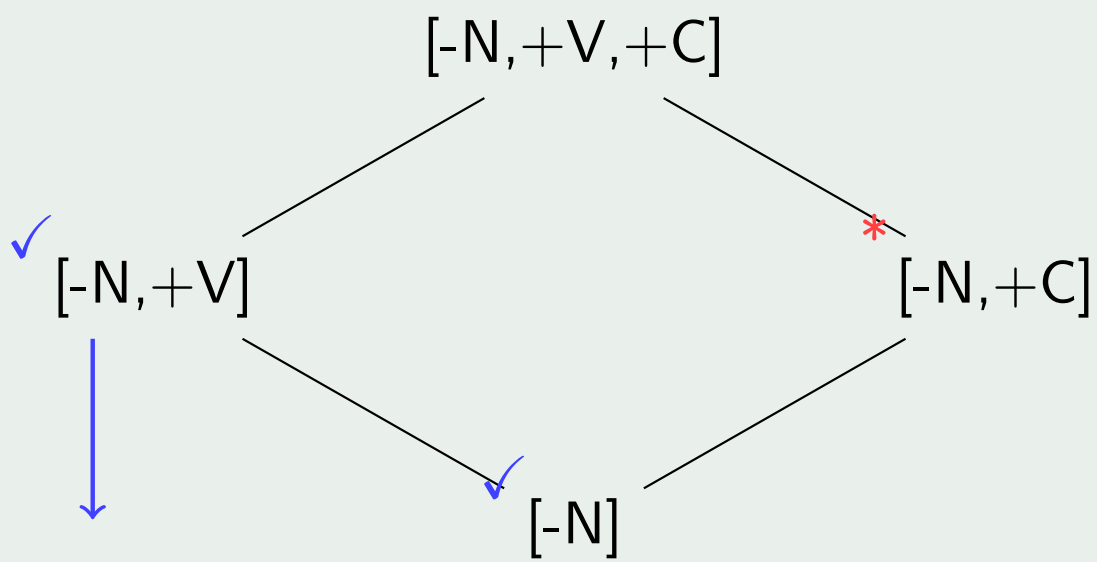


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

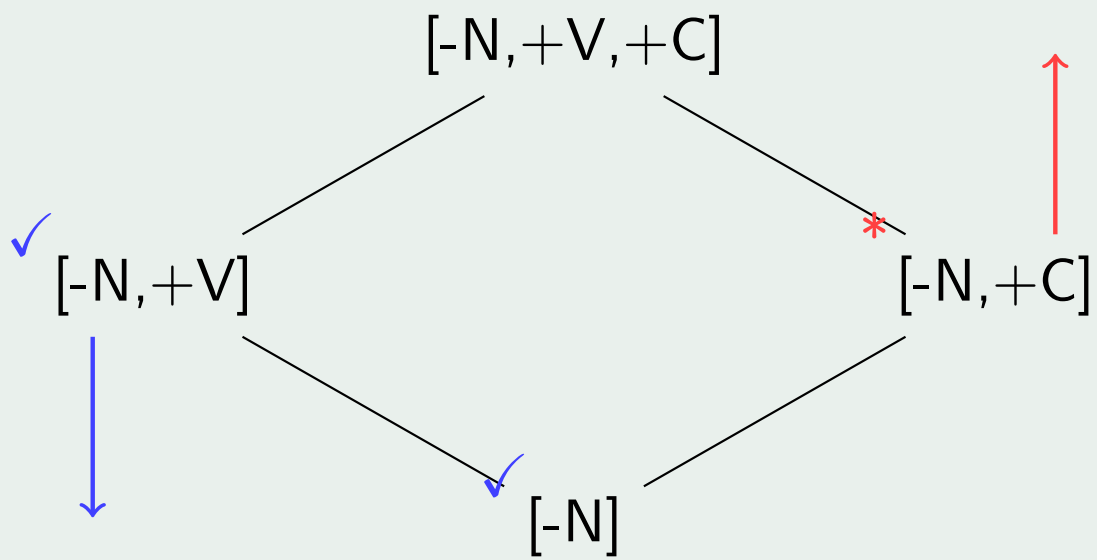


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

Example

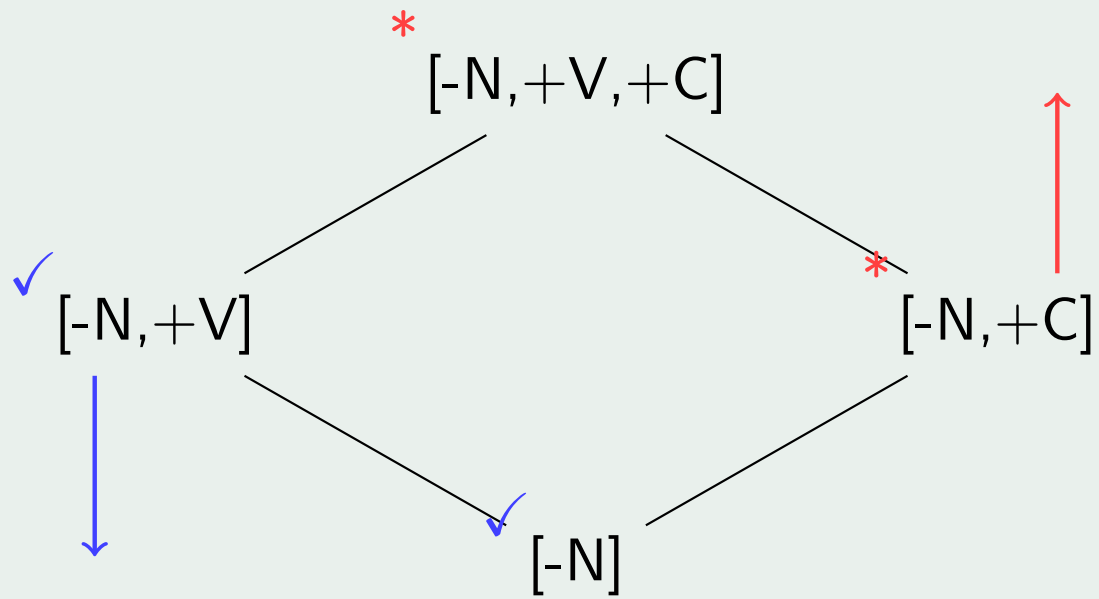


Containment Closure

Containment Closure

If **t** is a feature extension of **s** for G and G generates **t**, then G generates **s**.

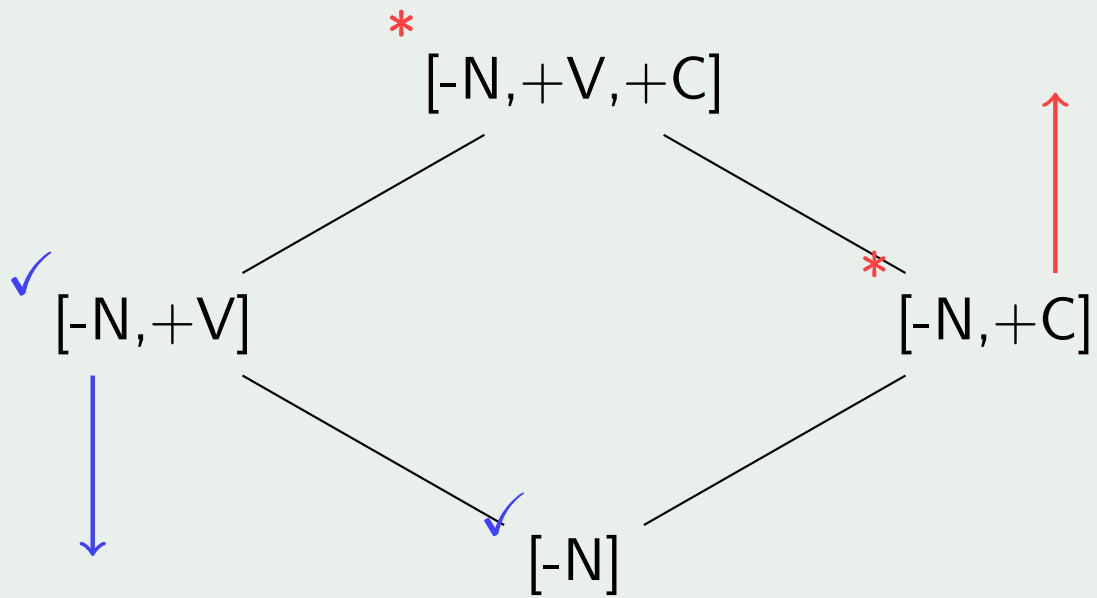
Example



Parallels to Logical 'And'

- ⇓ **Grammaticality is Downward Entailing w.r.t.** $<_G$
  $a \wedge b = 1$ implies $a = 1$
- ⇑ **ungrammaticality is upward entailing w.r.t.** $<_G$
  $a = 0$ implies $a \wedge b = 0$

Example



Example with Singular Segments

Input Data: Feature Strings of Length 1

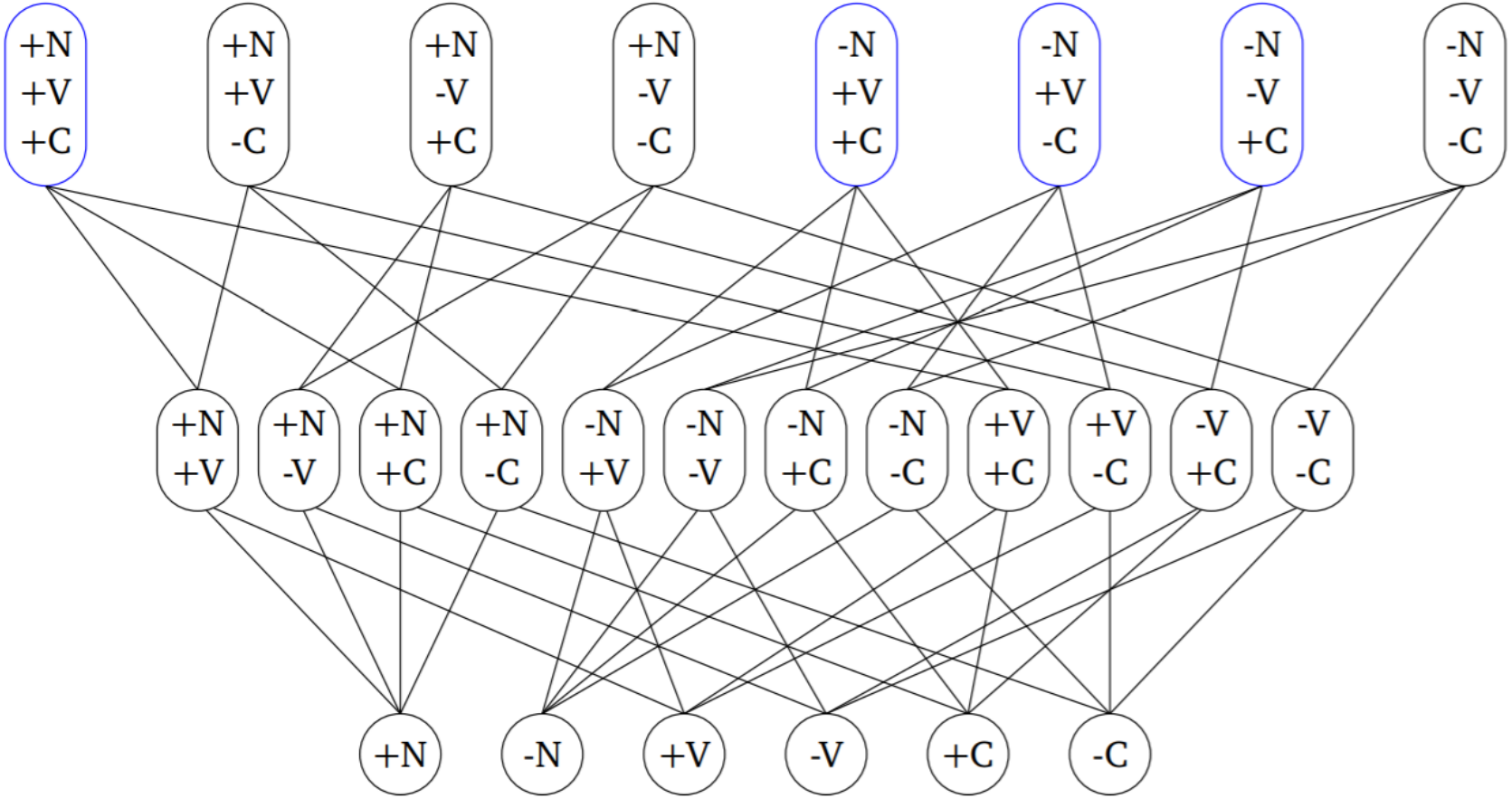
Suppose we observe in a language

- [+N,+V,+C] (voiced nasal consonants),
- [-N,+V,+C] (voiced nonnasal consonants),
- [-N,-V,+C] (voiceless nonnasal consonants),
- [-N,+V,-C] (voiced nonnasal vowels),

What constraints ought to be posited?

[-N,-V,+C] is a feature extension of [-N,-V], [-N,+C], [-V,+C].
These are feature extensions of [-N], [-V], [+C].
And the empty feature bundle [∅].
This partial ordering forms a **semi-lattice**.

Example


+N +V +C
+N +V -C
+N -V +C
+N -V -C
-N +V +C
-N +V -C
-N -V +C
-N -V -C
+N +V
+N -V
+N +C
+N -C
-N +V
-N -V
-N +C
-N -C
+V +C
+V -C
-V +C
-V -C
+N
-N
+V
-V
+C
-C

Example

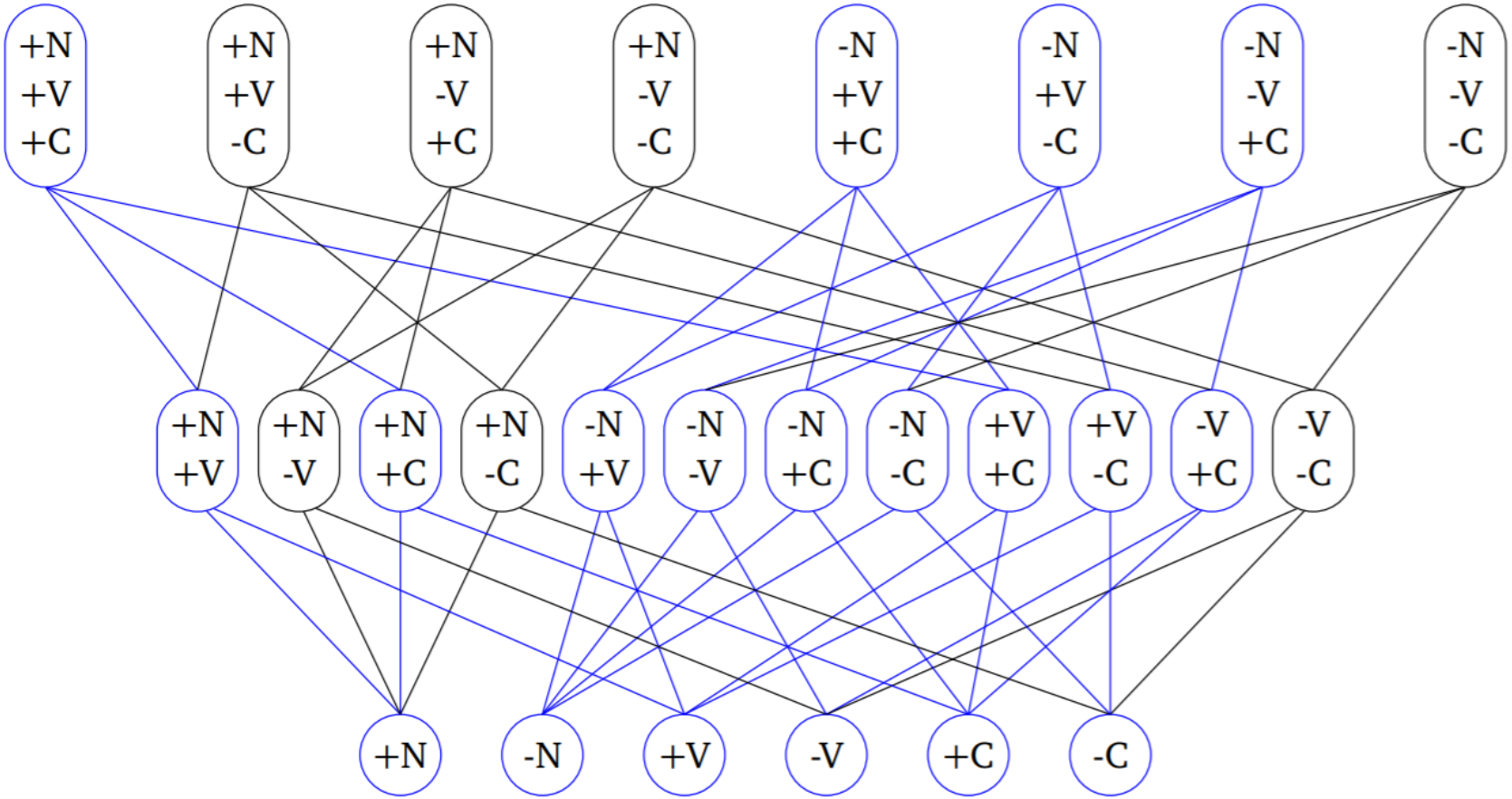


Figure: The learner eliminates ALL factors contained in observed examples

Example

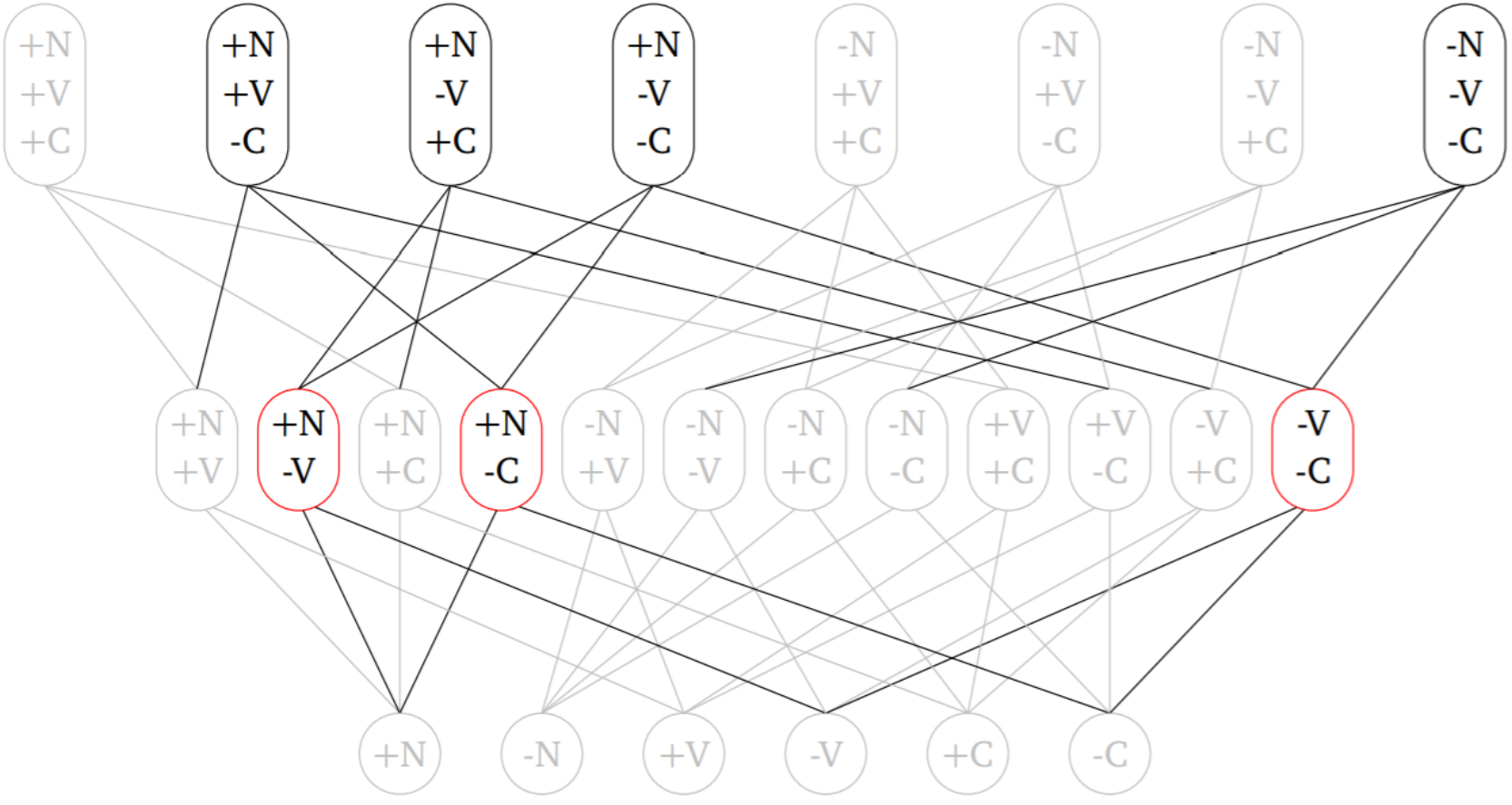
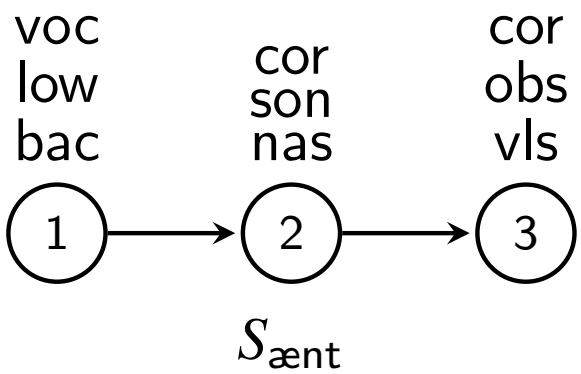
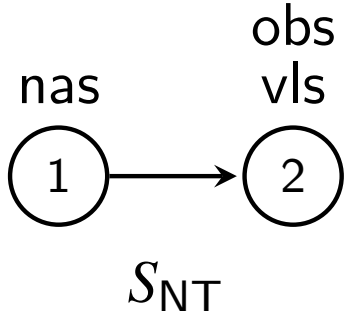


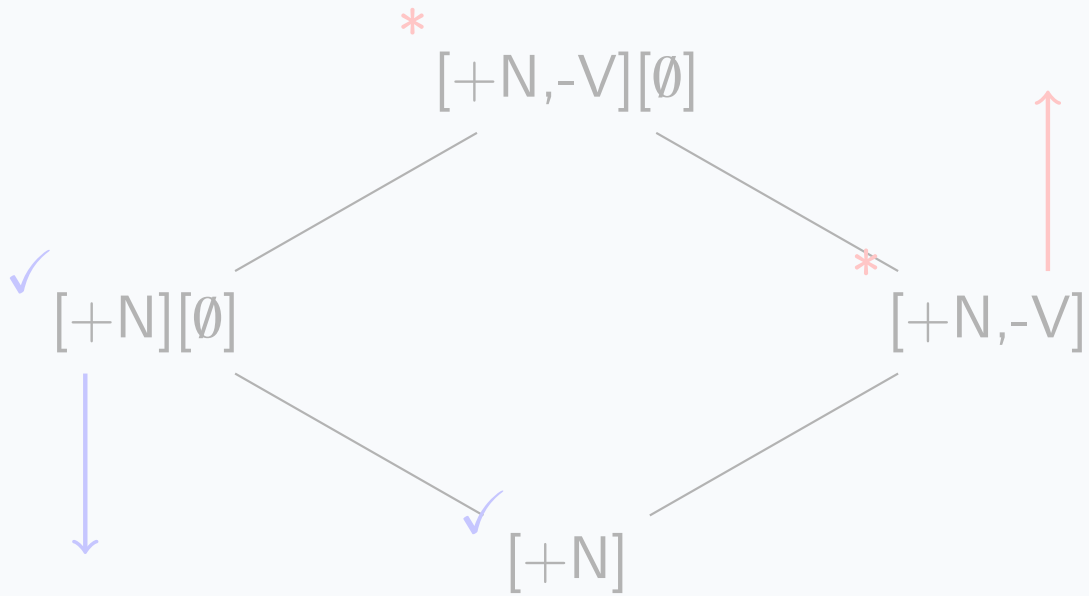
Figure: The set of most general ones which remain are the constraints!

Moving to Substructures

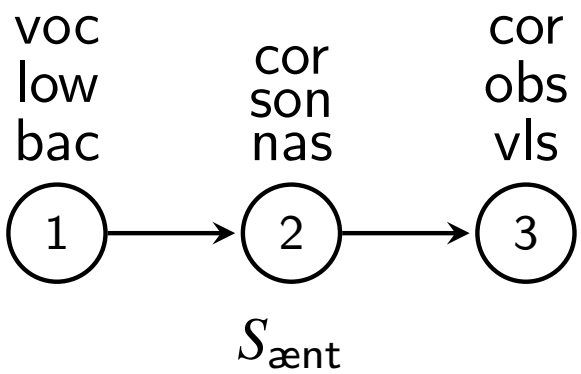


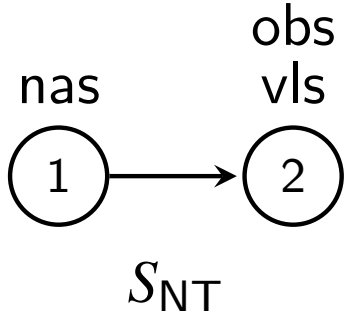


Containment Closure Still Holds

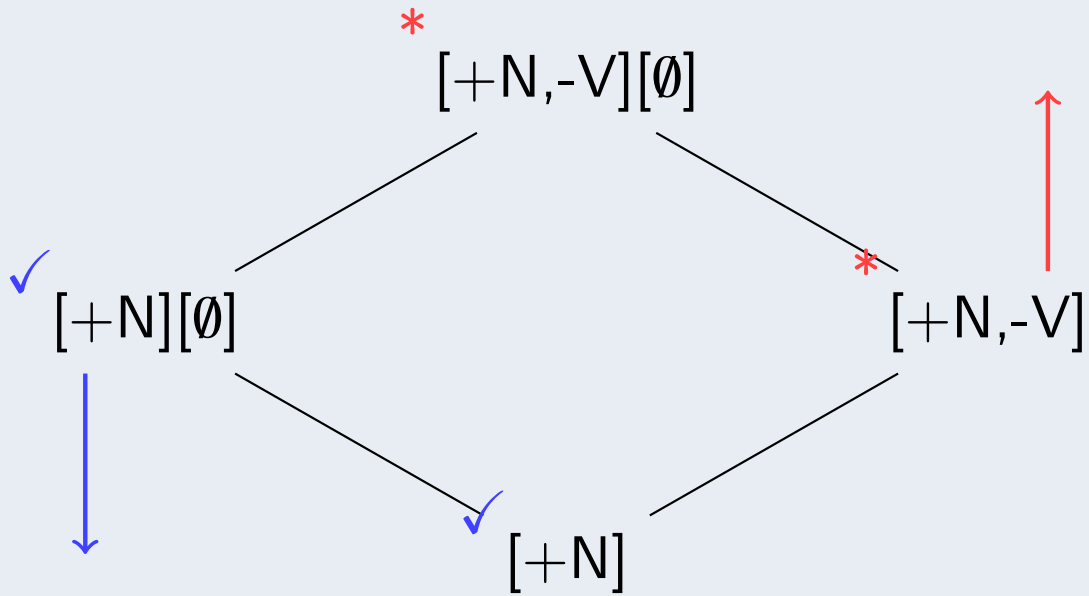


Moving to Substructures





Containment Closure Still Holds



Semilattice Explosion (Hayes and Wilson 2008)

Table 2
Number of possible constraints for various values of $|C|$ and n

		$\|C\|$			
		30	**100**	**200**	**400**
	1	30	100	200	400
	2	900	10,000	40,000	160,000
n	**3**	27,000	1,000,000	8 million	64 million
	4	810,000	100 million	1.6 billion	26 billion
	5	24 million	10 billion	320 billion	10 trillion

|C| is the number of natural classes and
n is the length of the constraint.

Is Learning doomed? No. But Hayes & Wilson are right that the direction of induction matters.

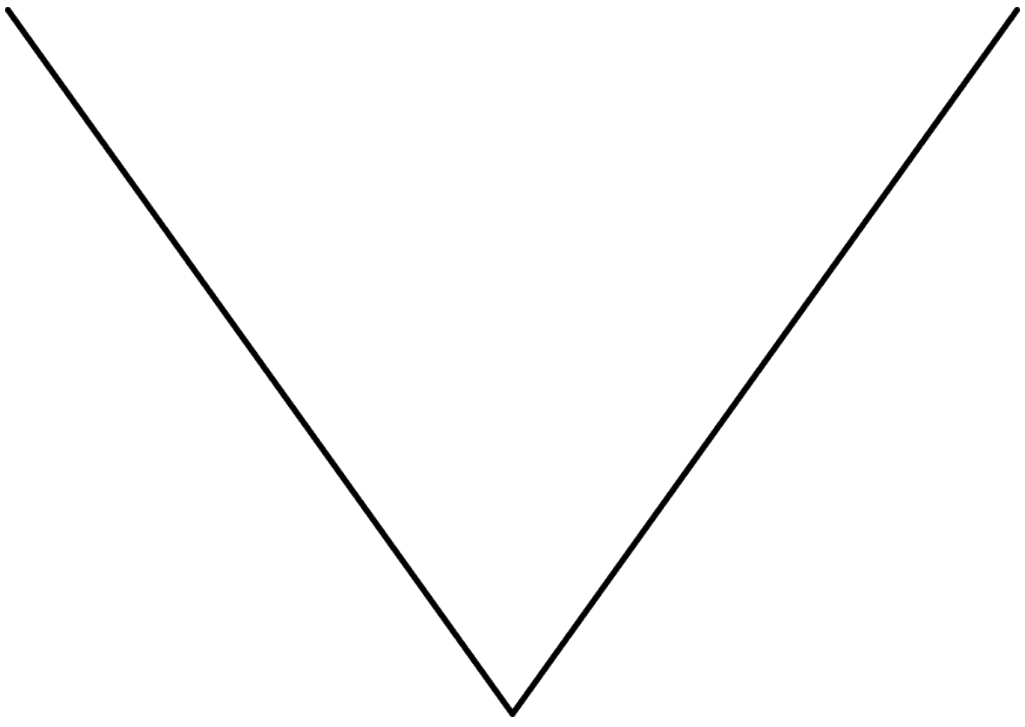
Top-Down Induction

- Start at the most specific points (highest) in the semilattice
- Remove all the substructures from the lattice that are present in the data.
- Collect the most general substructures remaining.

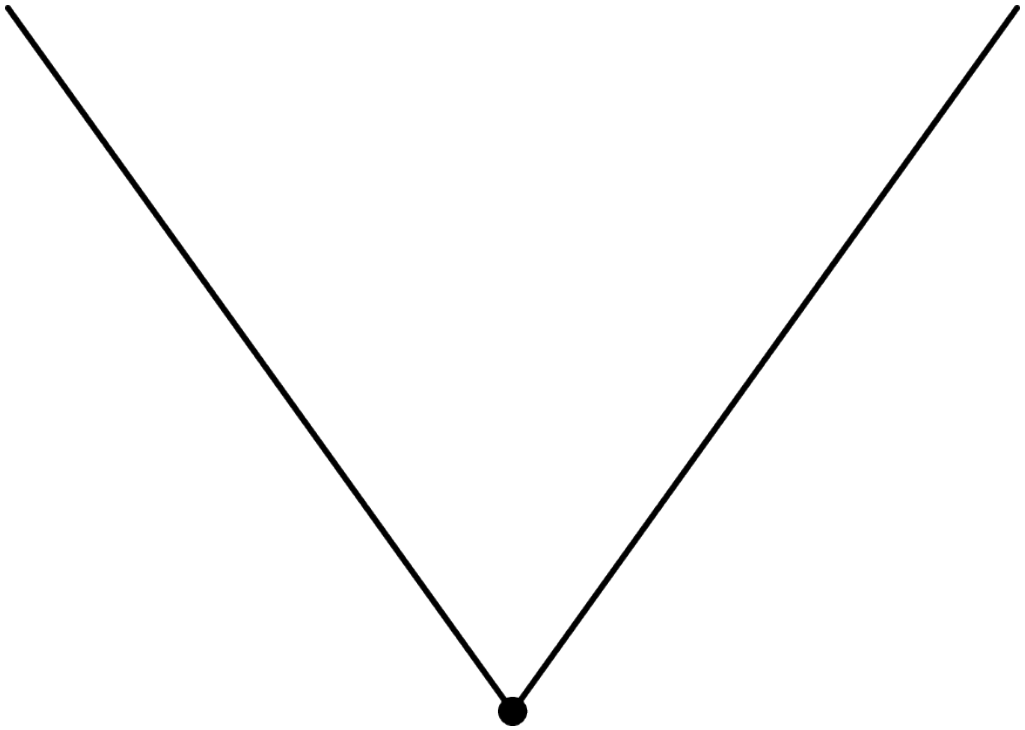
Bottom-Up Induction

- Beginning at the lowest element in the semilattice,
- Check whether this structure is present in the input data.
- If so, move up the lattice, either to a point with an adjacent underspecified segment, or a feature extension of a current segment, and repeat.

2D Schema of a Semilattice of Constraints

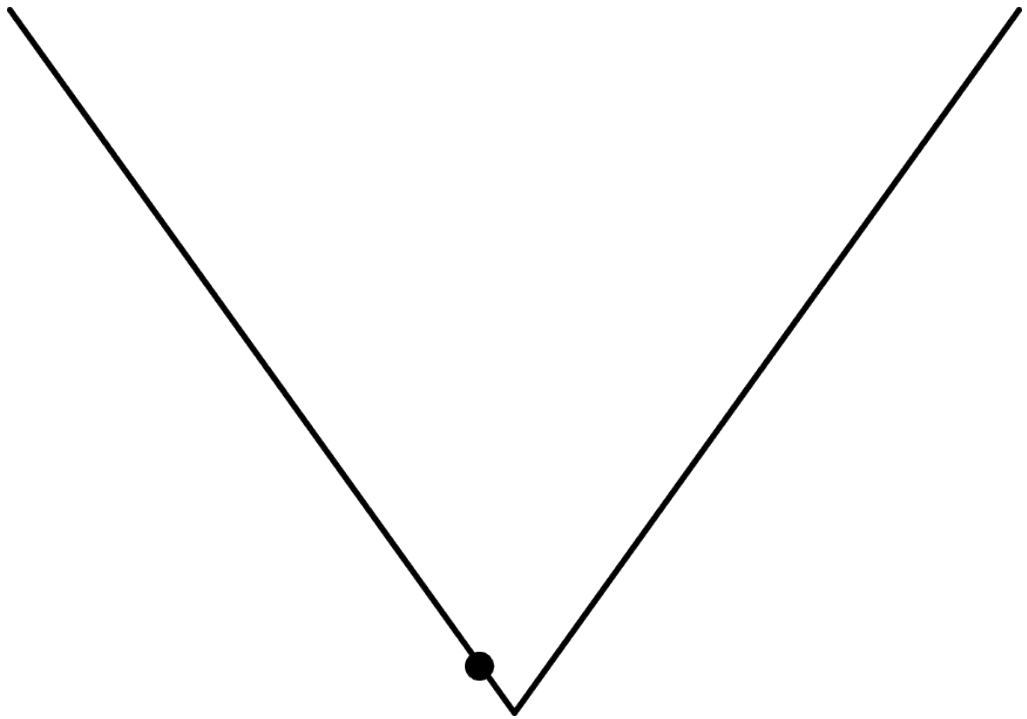


2D Schema of a Semilattice of Constraints



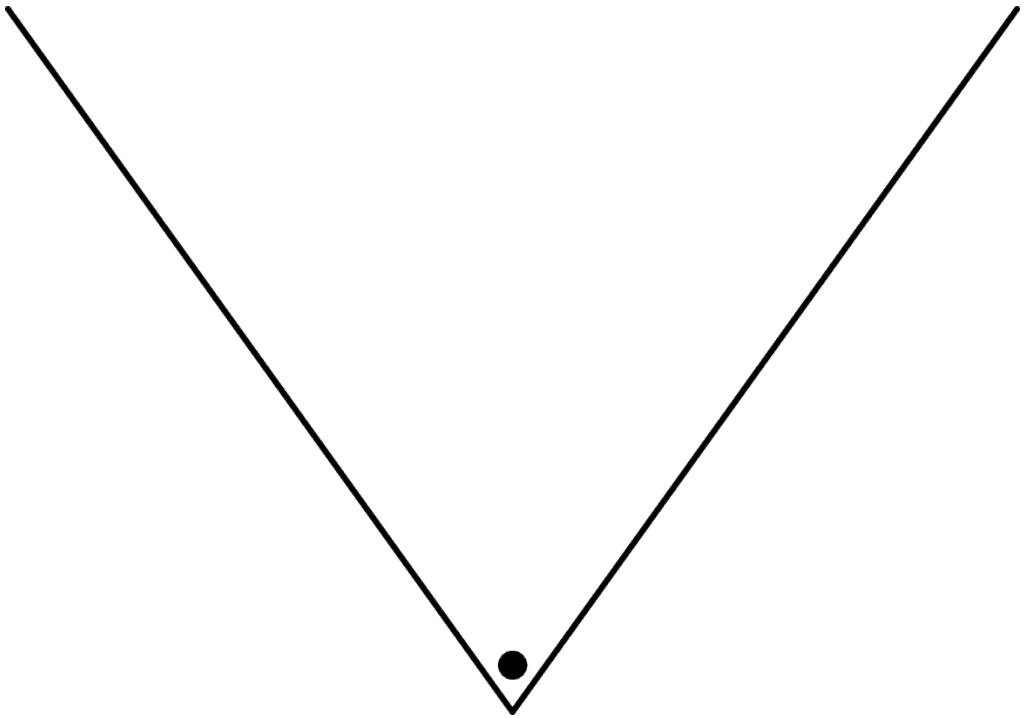
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



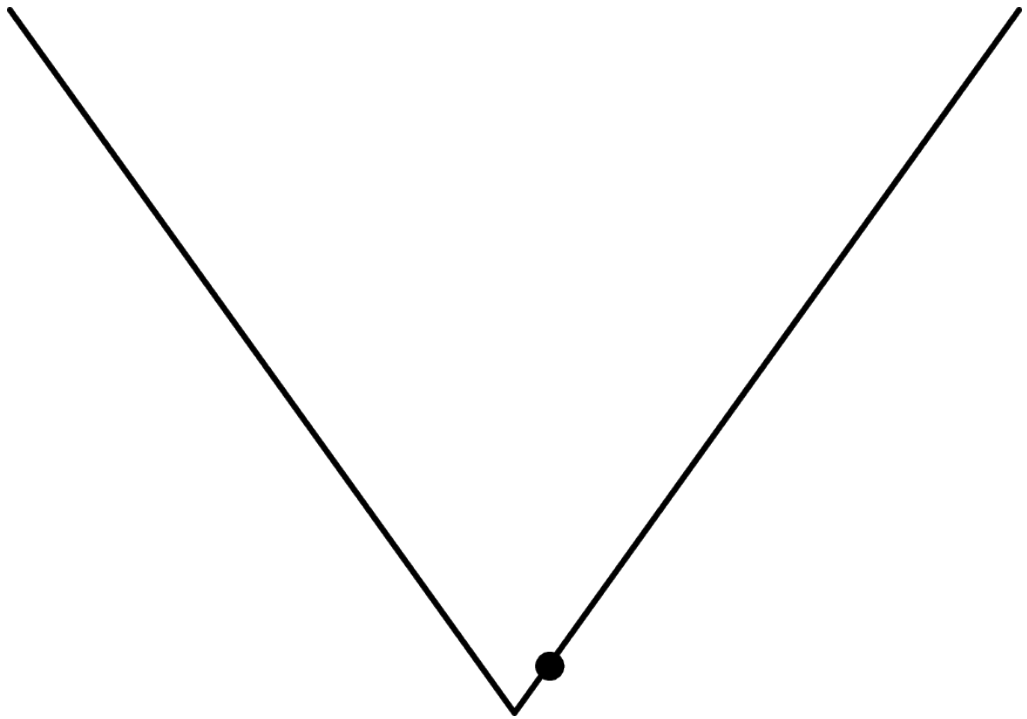
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



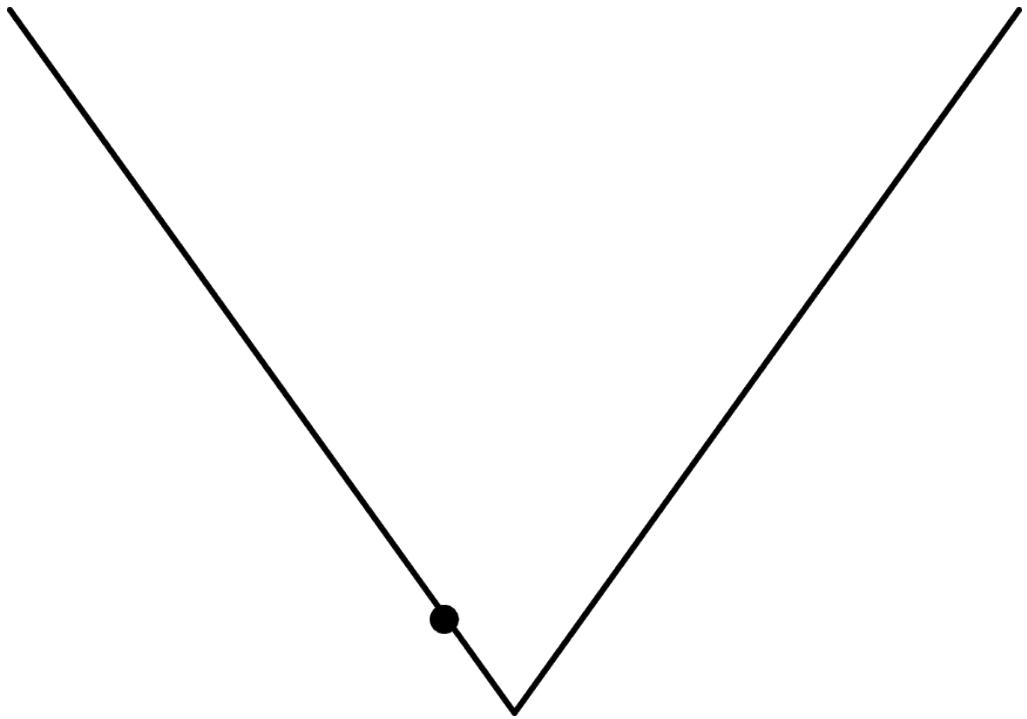
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



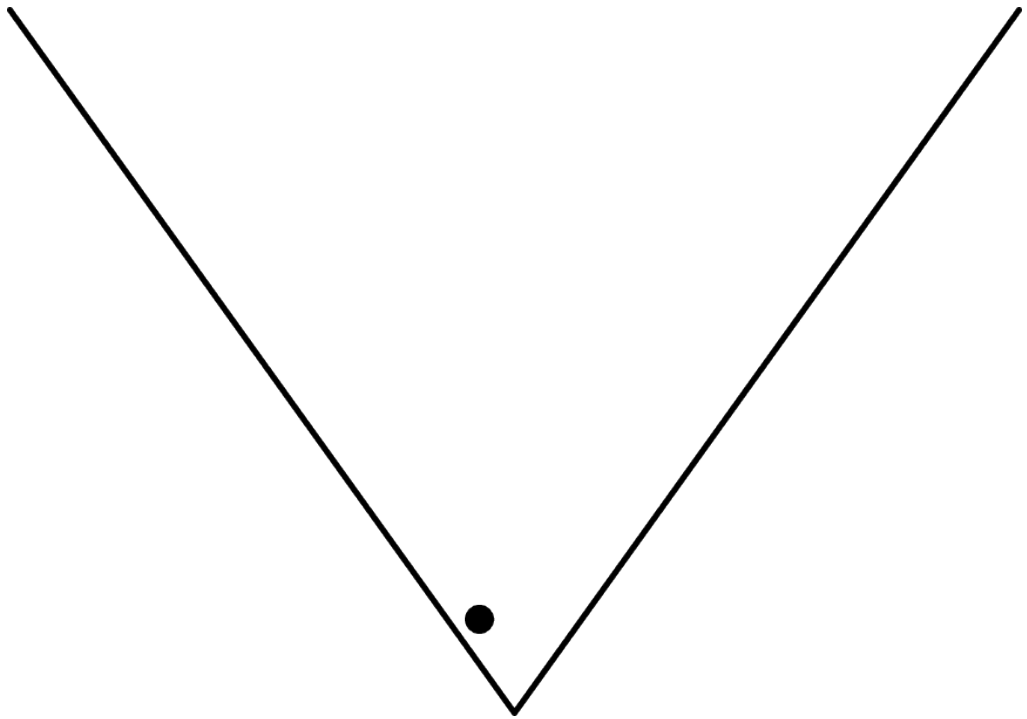
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



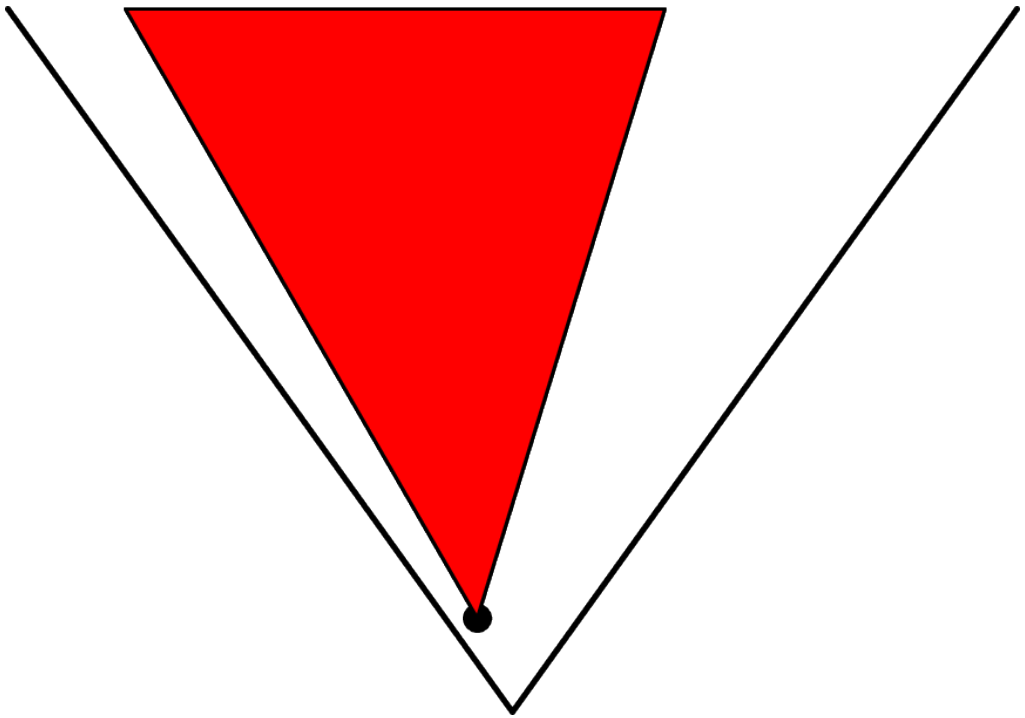
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



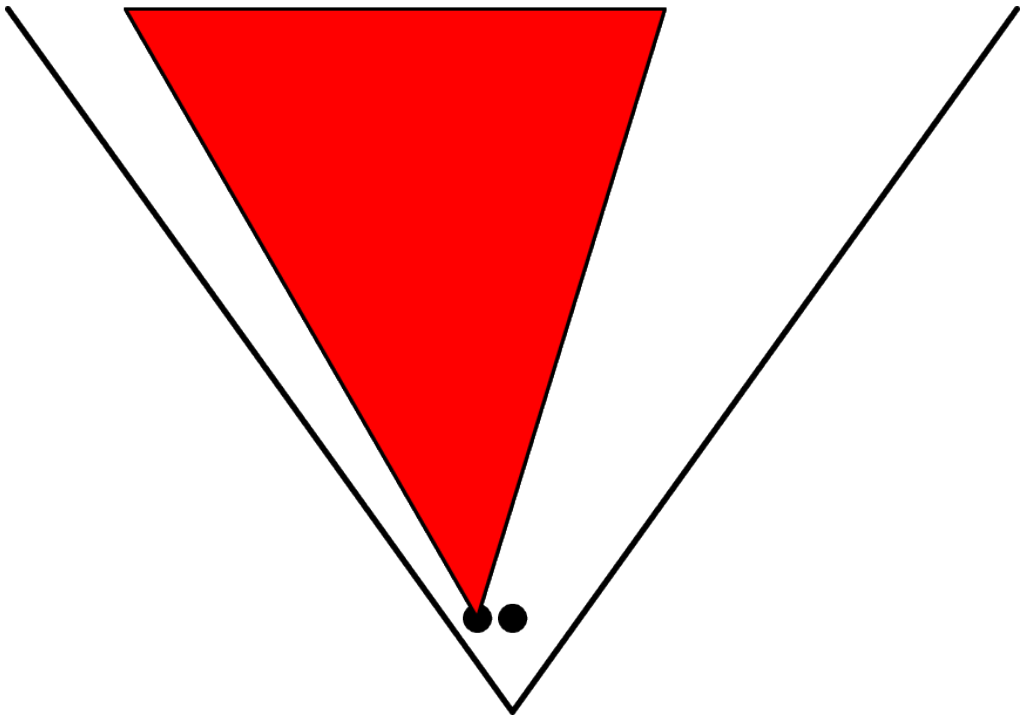
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



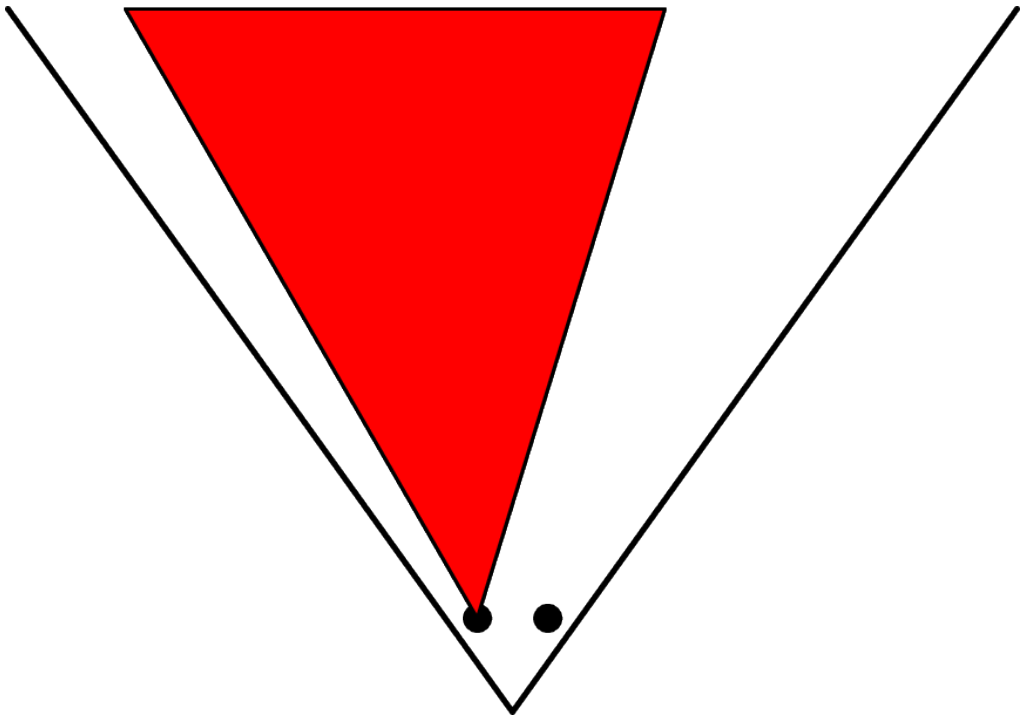
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



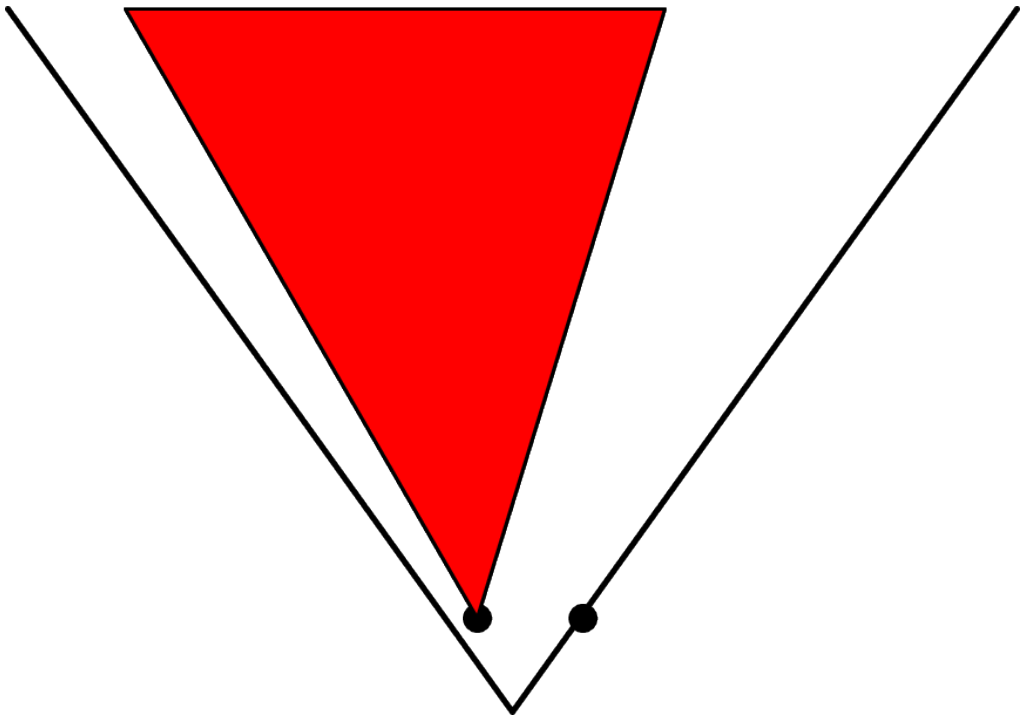
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



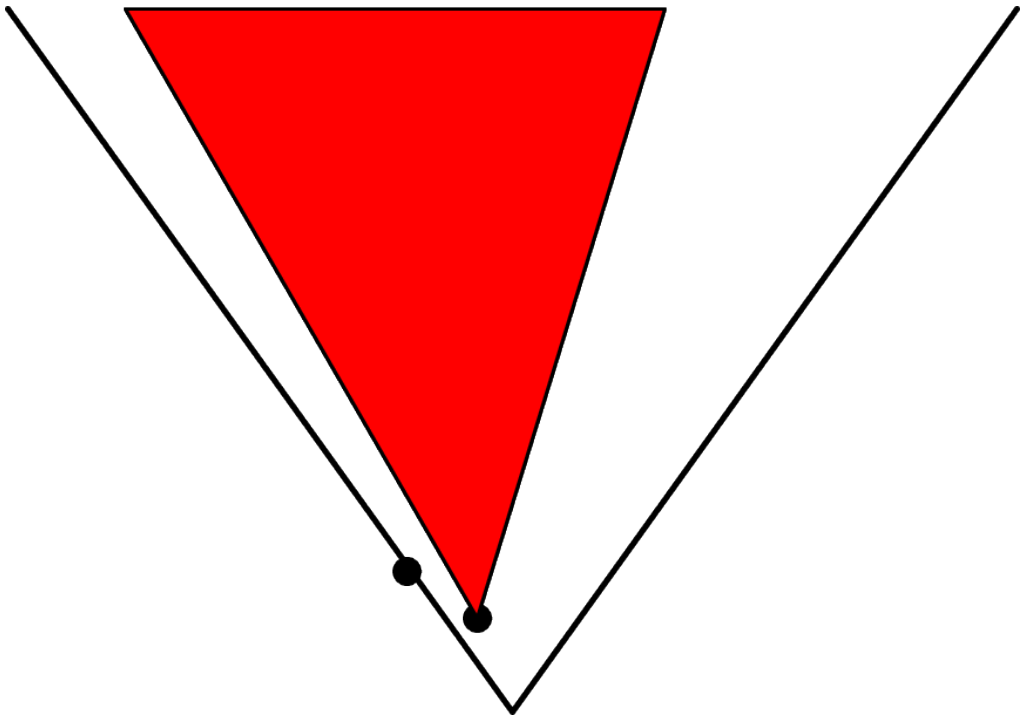
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



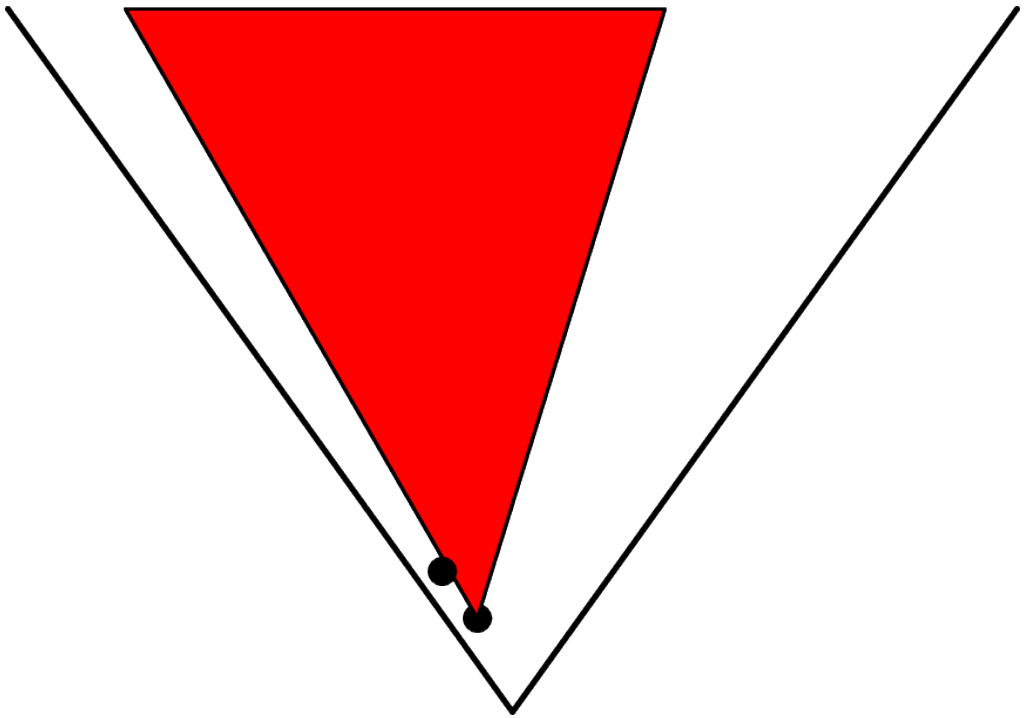
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



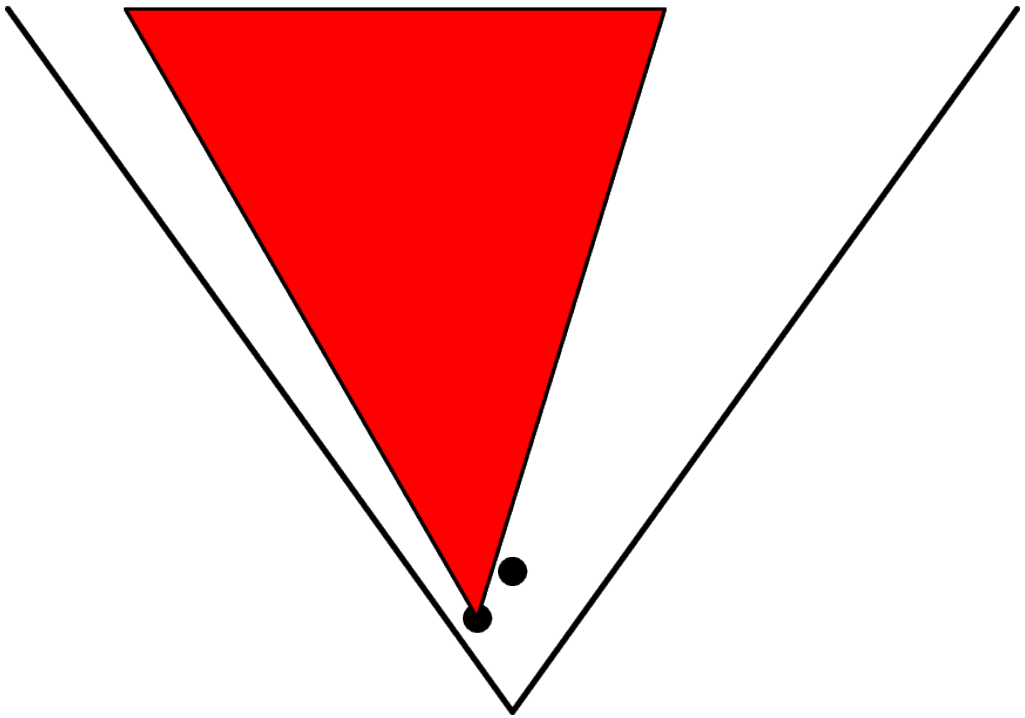
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



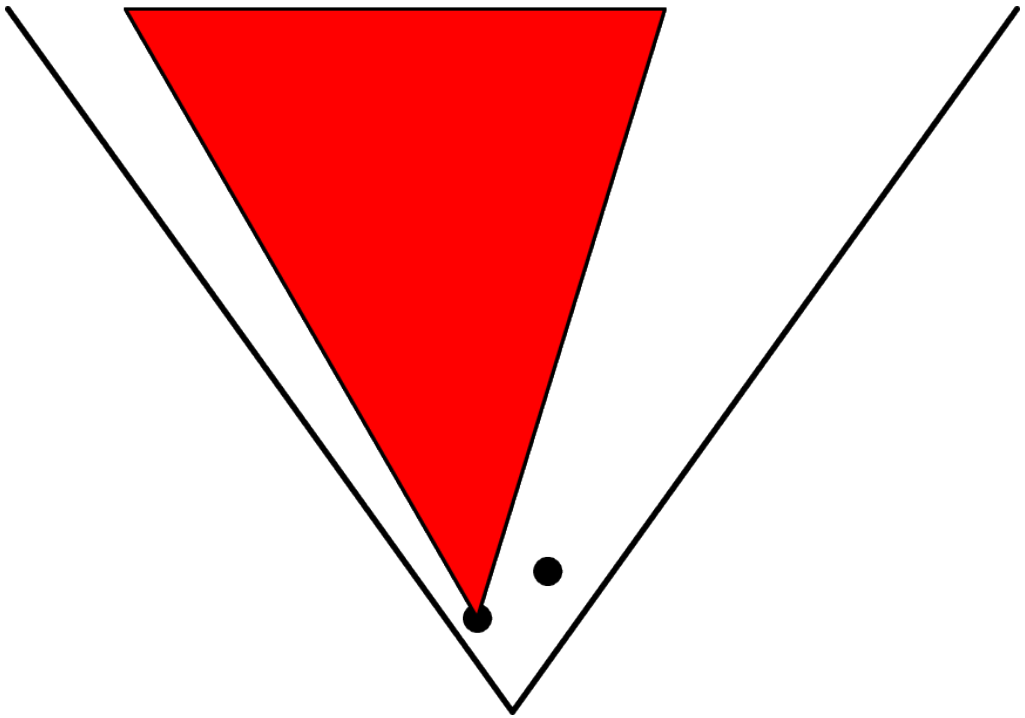
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



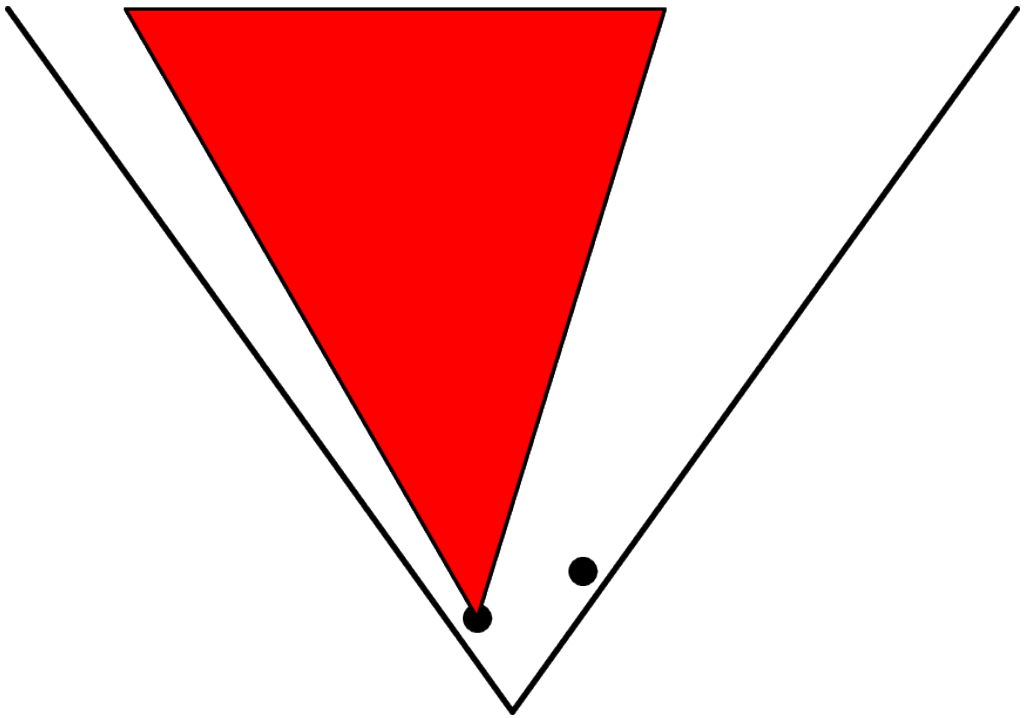
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



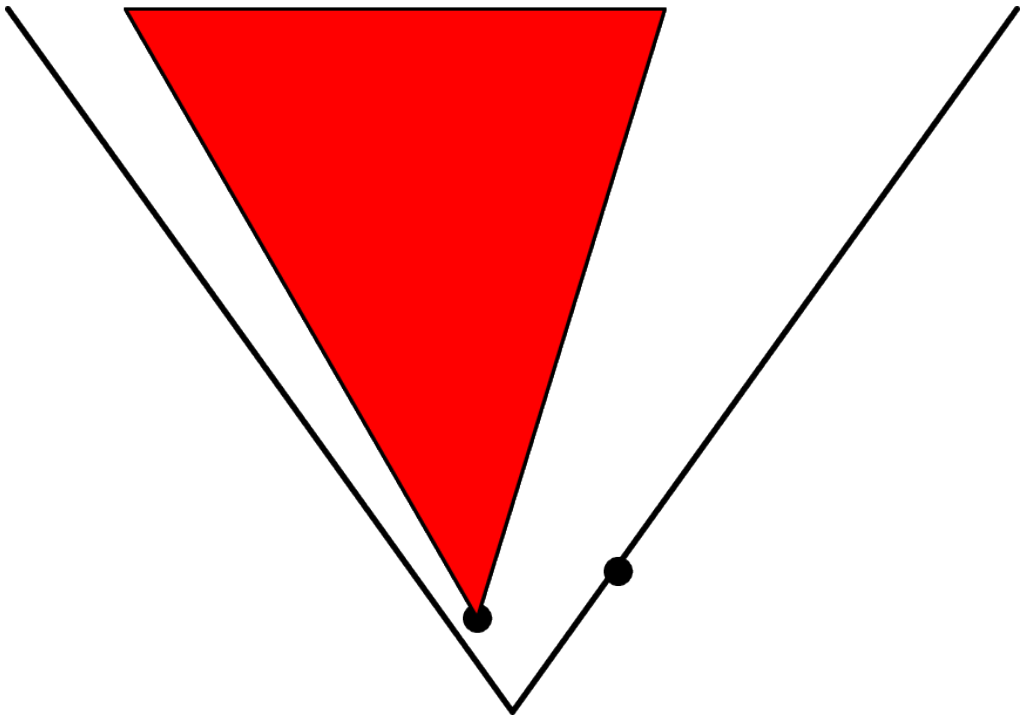
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



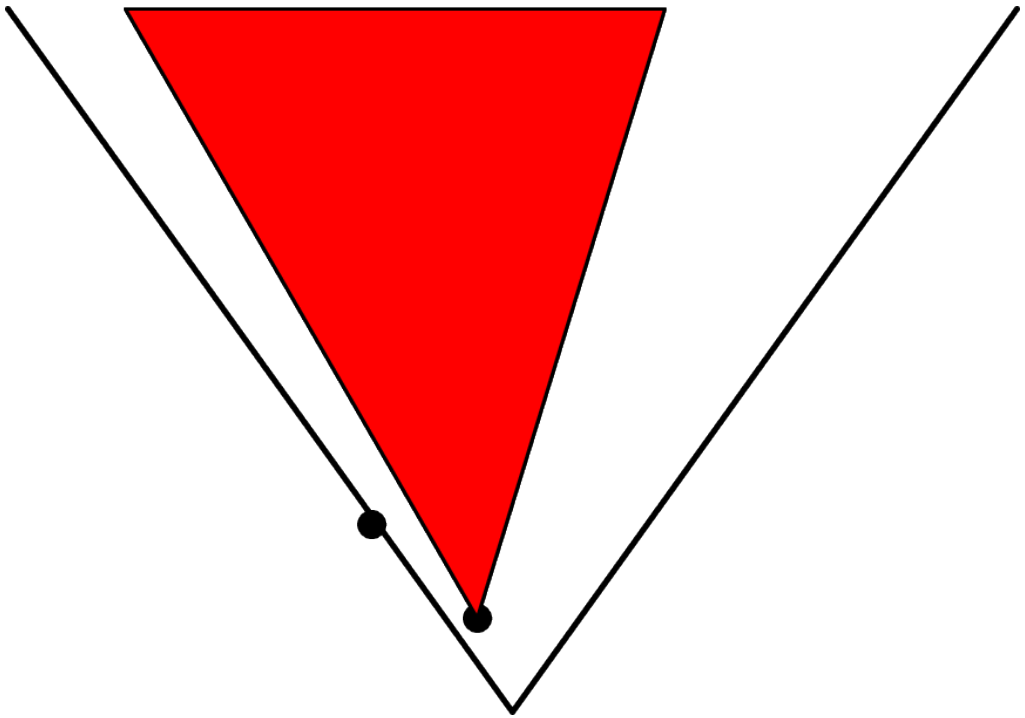
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



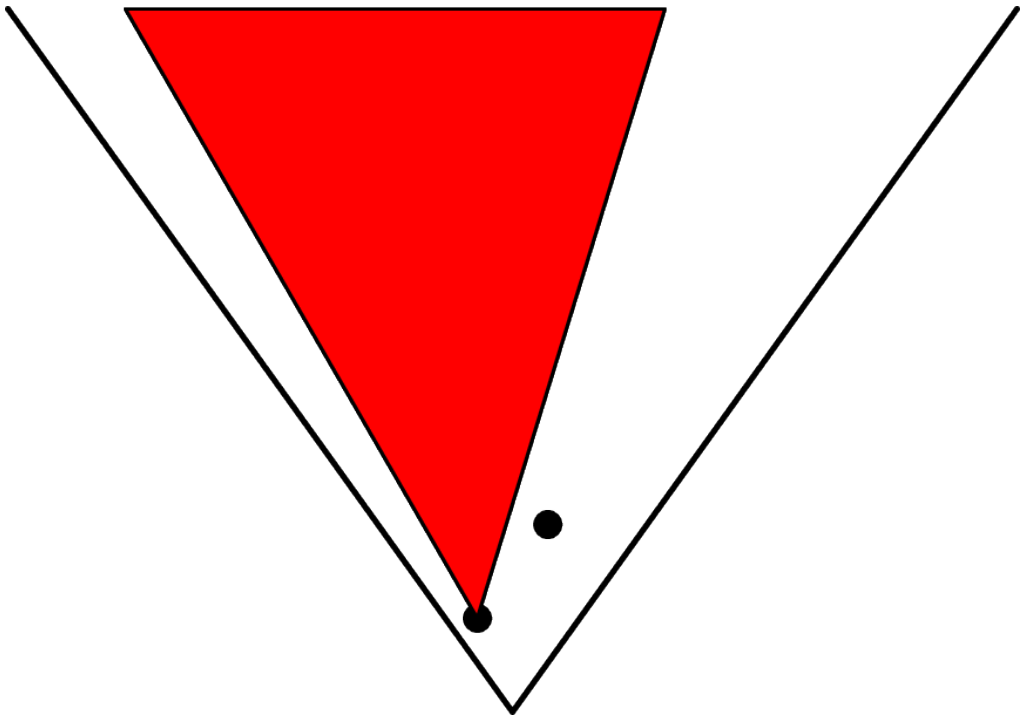
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



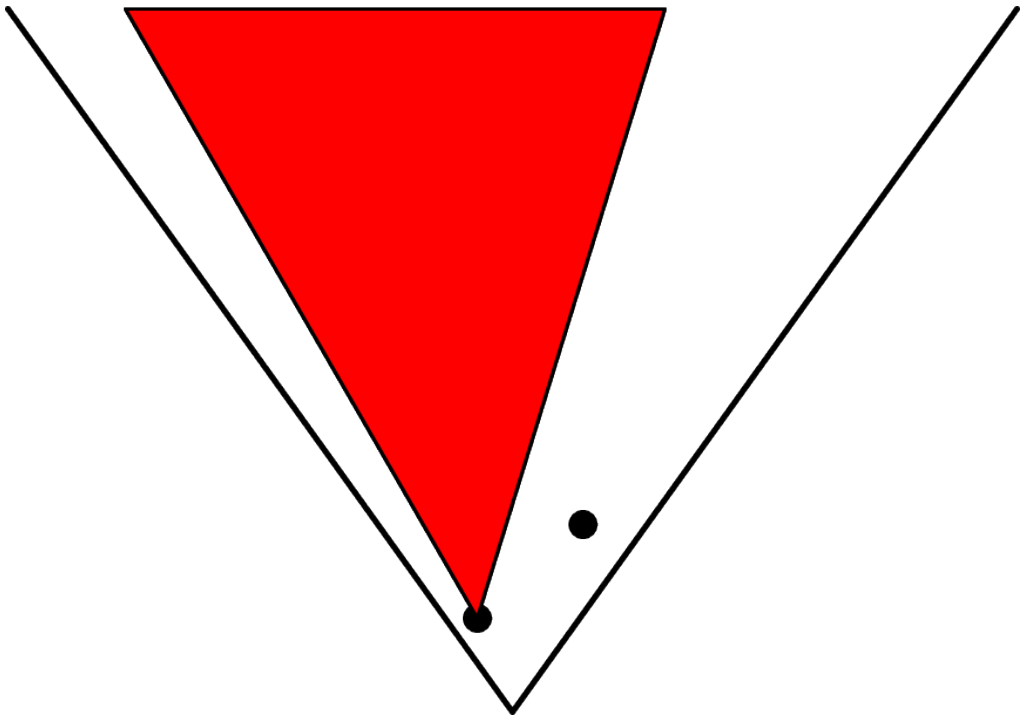
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



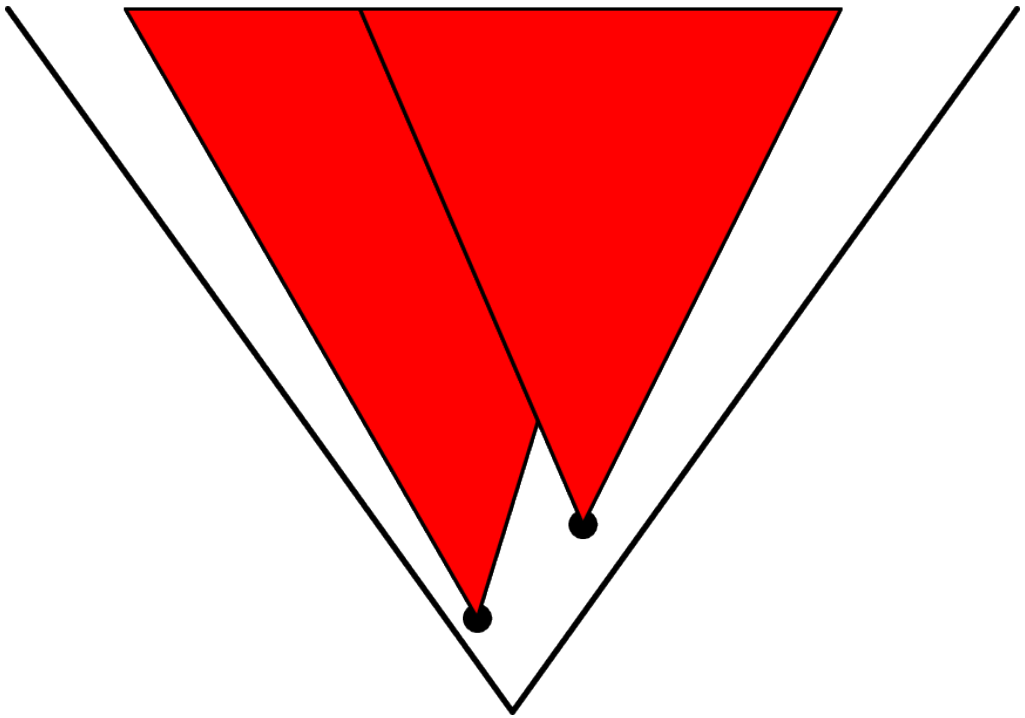
Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



Is this sub-structure in the data sample?

2D Schema of a Semilattice of Constraints



Is this sub-structure in the data sample?

Formalizing Learning Criteria (De Raedt 2008)

On what grounds can the learner prefer one set of constraints over another?

Criteria specifying whether a given set of constraints is acceptable w.r.t. data.

We want constraints:

- whose largest forbidden substructure is of size k
- which cover the data, i.e. $D \subseteq L(G)$
- which are more specific than all the other constraints G' that cover the data, so $L(G) \subseteq L(G')$
- which forbid structures S that are substructures of structures S' forbidden by other grammars G' that satisfy (1,2,3)
  - For all $S' \in G'$, there exists $S \in G$ such that $S \sqsubseteq S'$.

Conclusion and Open Problems

Today's Results

- Possible constraints are structured as a semilattice.
- Hayes and Wilson are right to look for shorter, more general constraints, but there is richer structure in the space for them and us to take advantage of.

Things To Do

- Prove that Bottom-Up Induction always satisfies the learning criteria.
- Determine the trade-off between data sparsity and time complexity. We hypothesize sparser data should yield faster generalization.
- Extend results to learning phonological transformations.

Thanks!

Special thanks to **Jim Rogers** and **Rmi Eyraud** for immensely helpful discussions