Minimum Level Non-Planar Patterns for Trees

J. Joseph Fowler and Stephen G. Kobourov

Department of Computer Science, The University of Arizona

The 15th International Symposium on Graph Drawing (GD 2007)

Outline

- Background

Outline

- Background
  - Motivation
  - Definitions
  - Previous Work

Outline

- Background
- Previous Patterns

Outline

- Background
- Previous Patterns
- New Patterns

Outline

- Background
  - Motivation
  - Definitions
  - Previous Work
- Previous Patterns
- New Patterns

Motivation

- Originally wanted to use patterns for ULP characterization

Motivation

- Originally wanted to use patterns for ULP characterization
  - None of the ULP trees matched any of the existing patterns

Motivation

- Originally wanted to use patterns for ULP characterization
  - None of the ULP trees matched any of the existing patterns
    - Good – showed they were level planar

Motivation

- Originally wanted to use patterns for ULP characterization
  - None of the ULP trees matched any of the existing patterns
  - T_8 matched one of the exisiting patterns

Motivation

- Originally wanted to use patterns for ULP characterization
  - None of the ULP trees matched any of the existing patterns
  - T_8 matched one of the exisiting patterns
    - Good – showed T_8 was level non-planar

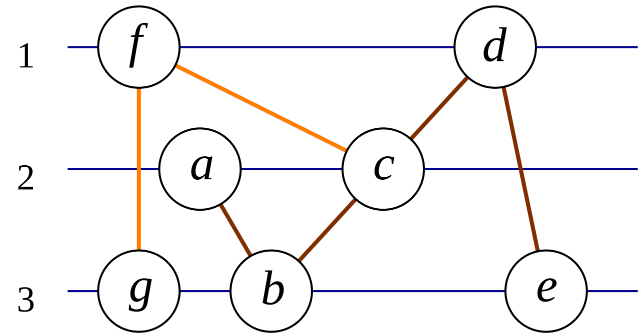
Motivation

- Originally wanted to use patterns for ULP characterization
  - None of the ULP trees matched any of the existing patterns
  - T_8 matched one of the exisiting patterns
  - T_9 did not match any of the exisiting patterns

Motivation

- Originally wanted to use patterns for ULP characterization
  - None of the ULP trees matched any of the existing patterns
  - T_8 matched one of the exisiting patterns
  - T_9 did not match any of the exisiting patterns
    - Not good – either T_9 was level planar (it's not) or the existing patterns were incomplete

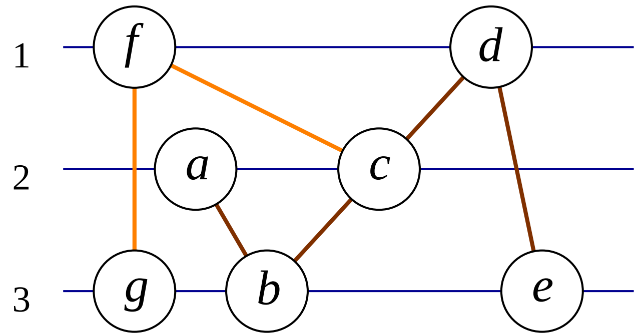
Definitions – Level Planar Graphs



3−Level Graph

- A k-*level graph* $G(V, E, \phi)$

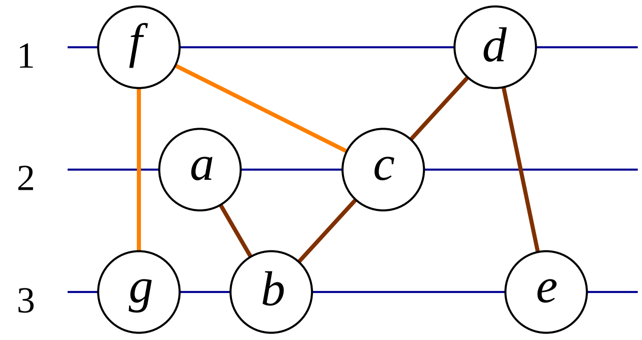
Definitions – Level Planar Graphs



3−Level Graph

- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$

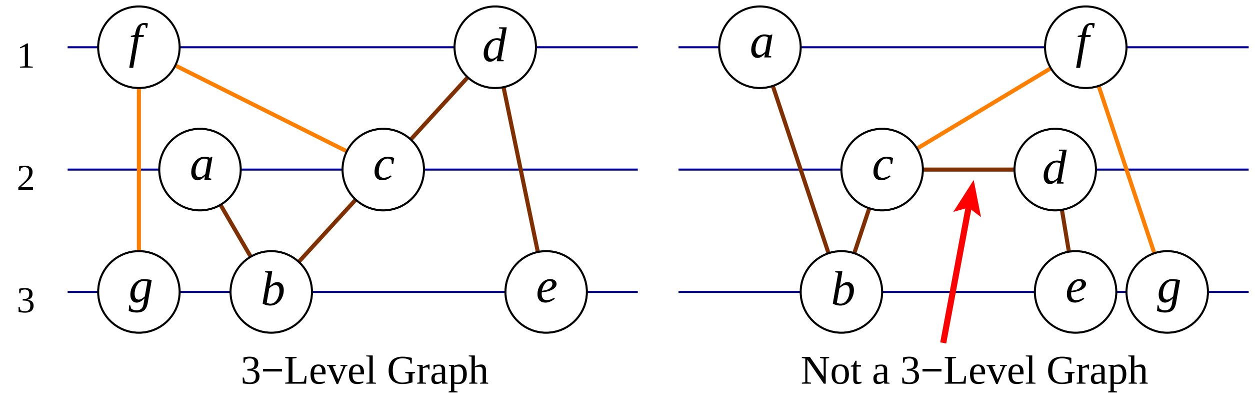
Definitions – Level Planar Graphs



3−Level Graph

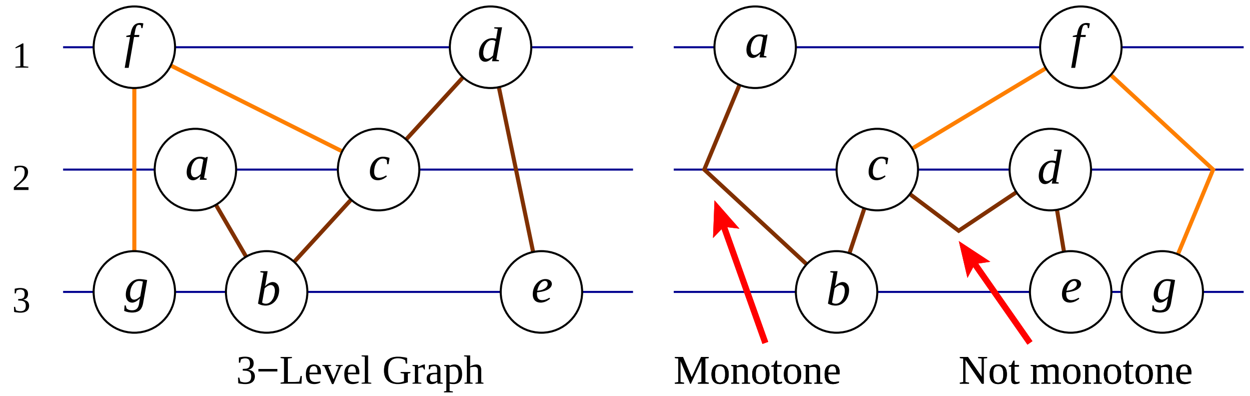
- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \to [1..k]$
    - Can have multiple vertices per level

Definitions – Level Planar Graphs



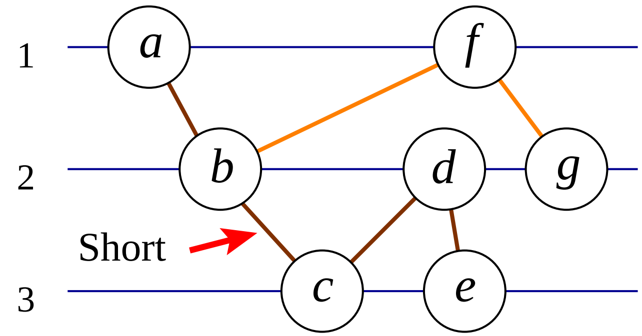
- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$
    - Cannot have an edge between two vertices in same level

Definitions – Level Planar Graphs



- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$
  - Edges are y-monotone

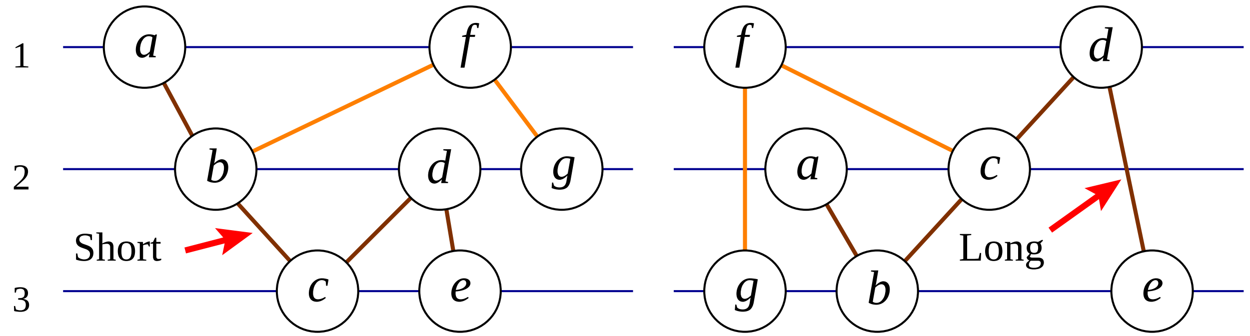
Definitions – Level Planar Graphs



Proper Level Graph

- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$
  - Edges are y-monotone
    - G is *proper* if all edges are short spanning one level

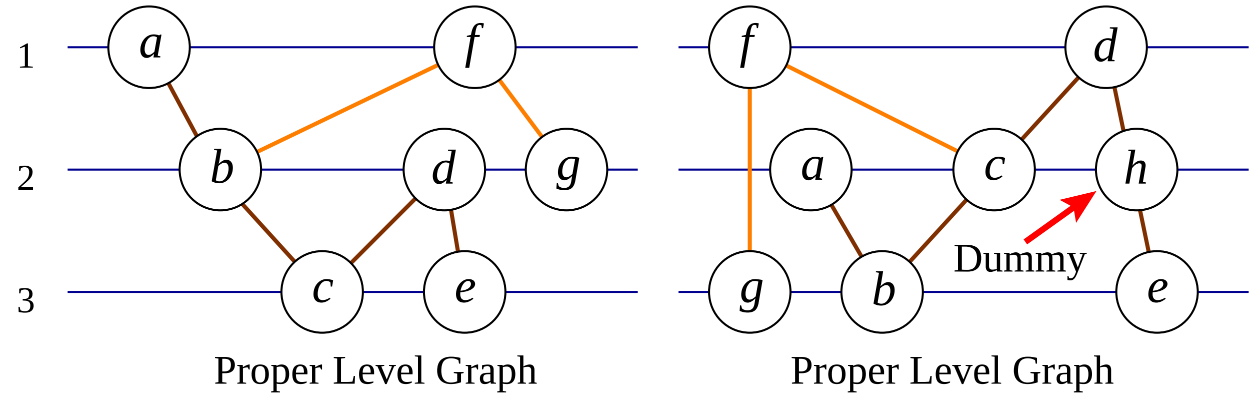
Definitions – Level Planar Graphs



Proper Level Graph   Improper Level Graph

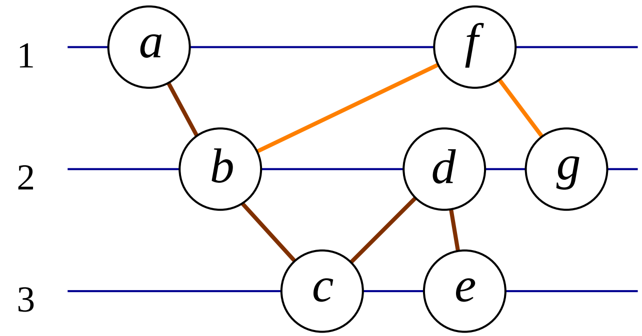
- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$
  - Edges are y-monotone
    - Otherwise G is *improper* with *long* edges

Definitions – Level Planar Graphs



- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$
  - Edges are y-monotone
    - Any level graph can be made proper by adding dummy vertices to long edges

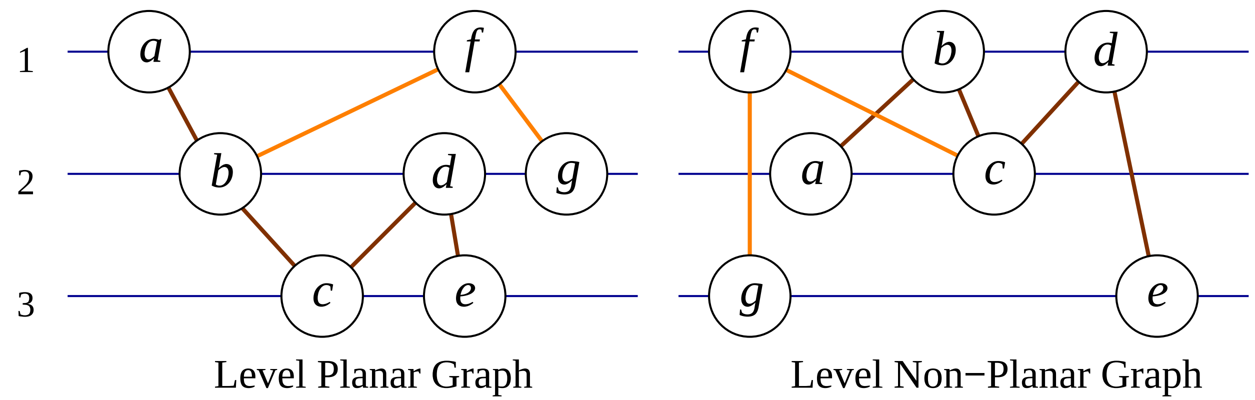
Definitions – Level Planar Graphs



Level Planar Graph

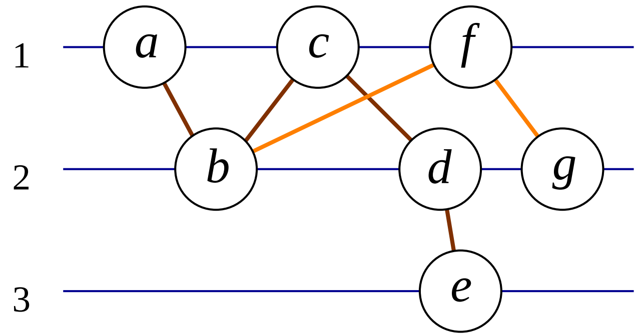
- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \to [1..k]$
  - Edges are y-monotone
- G is *level planar if*

Definitions – Level Planar Graphs



- A k-*level graph* $G(V, E, \phi)$
  - Has n vertices where $n \geq k$ with a *leveling* $\phi : V \rightarrow [1..k]$
  - Edges are y-monotone
- G is *level planar if*
  - G can be drawn without crossings AND each vertex remains on its level

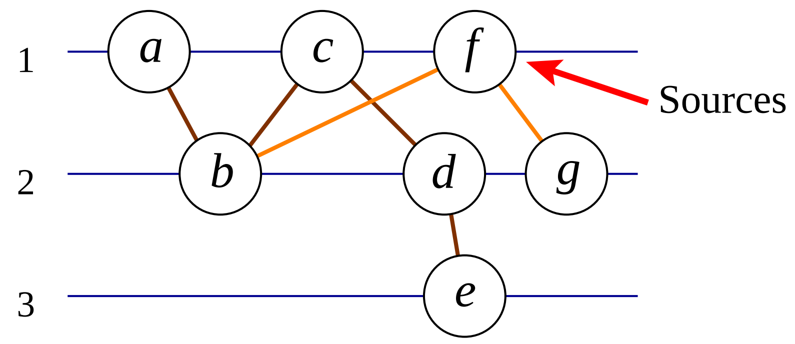
Definitions – Hierarchies



Hierarchy

- *Hierarchies* are proper level graphs

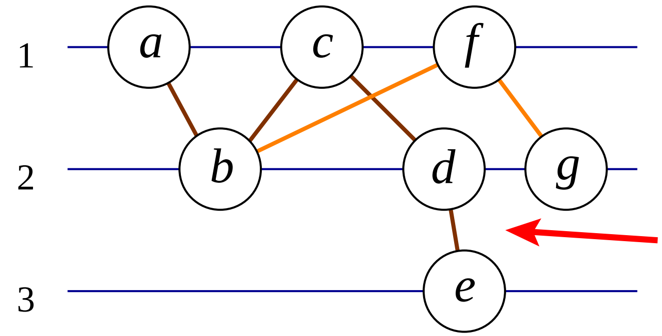
Definitions – Hierarchies



Hierarchy

- *Hierarchies* are proper level graphs
  - All source vertices are in top level

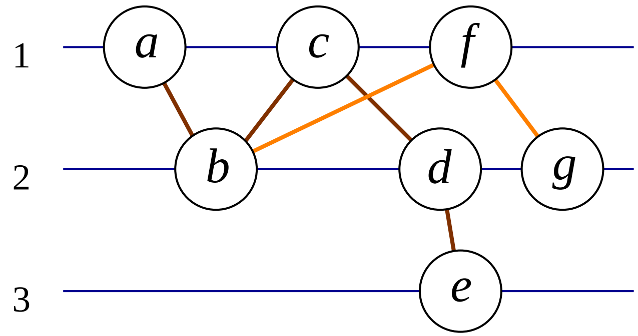
Definitions – Hierarchies



Hierarchy

- *Hierarchies* are proper level graphs
  - All source vertices are in top level
    - Exists a y-monotone path from the source to every other vertex

Definitions – Hierarchies



Hierarchy

- *Hierarchies* are proper level graphs
  - All source vertices are in top level
  - All edges are directed from higher to lower levels

Previous Work

- $O(n)$ time algorithms for level graphs
  - Jünger, Leipert, and Mutzel gave a level planarity testing algorithm in 1998
  - Jünger and Leipert achieved level planar embedding in 1999
  - Eades, Feng, Lin, and Nagamochi devised a straight-line level planar drawing algorithm given an embedding in 1997
- Characterizations of level graphs
  - Di Battista and Nardelli characterized hierarchies in 1988
  - Healy, Kuusik, and Leipert found minimal LNP subgraph patterns in 2000

Outline – Previous Patterns

- Background
- Previous Patterns

Outline – Previous Patterns

- Background
- Previous Patterns
  - Hierarchy Patterns

Outline – Previous Patterns

- Background
- Previous Patterns
  - Hierarchy Patterns
  - Minimum Level Non-Planar (MLNP) Patterns

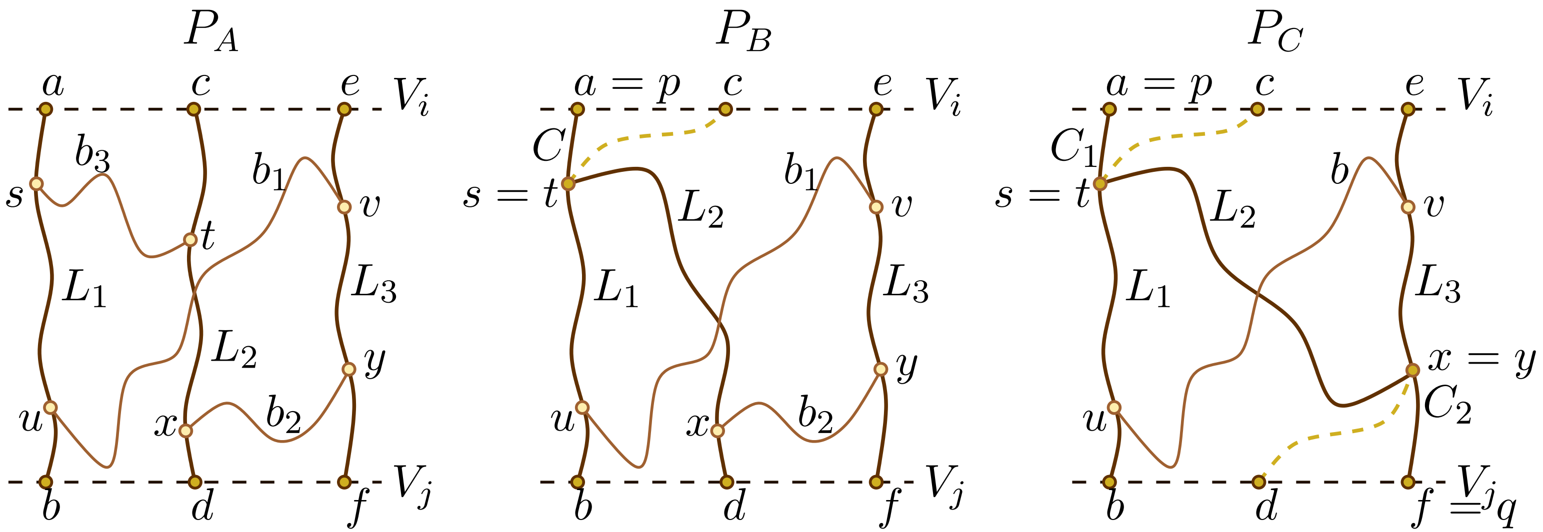
Outline – Previous Patterns

- Background
- Previous Patterns
  - Hierarchy Patterns
  - Minimum Level Non-Planar (MLNP) Patterns
    - Matches larger, more general class of level graphs

Outline – Previous Patterns

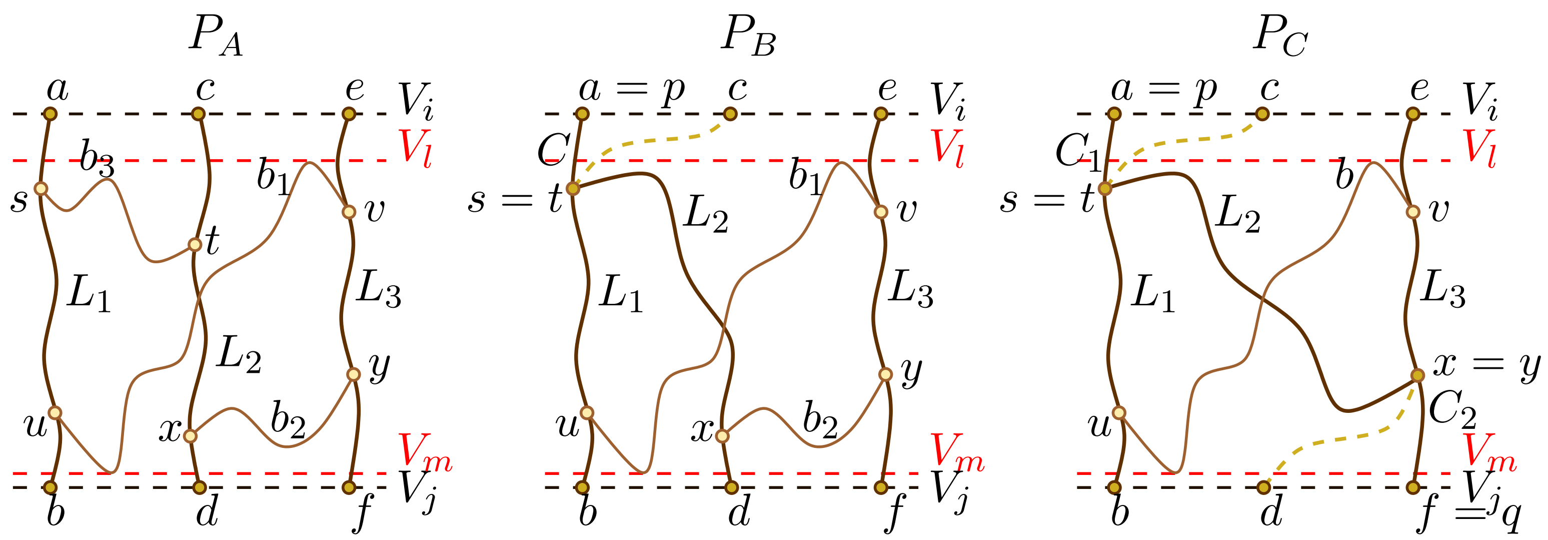
- Background
- Previous Patterns
  - Hierarchy Patterns
  - Minimum Level Non-Planar (MLNP) Patterns
    - Matches larger, more general class of level graphs
    - Minimal unlike hierarchy patterns

Hierarchy Patterns



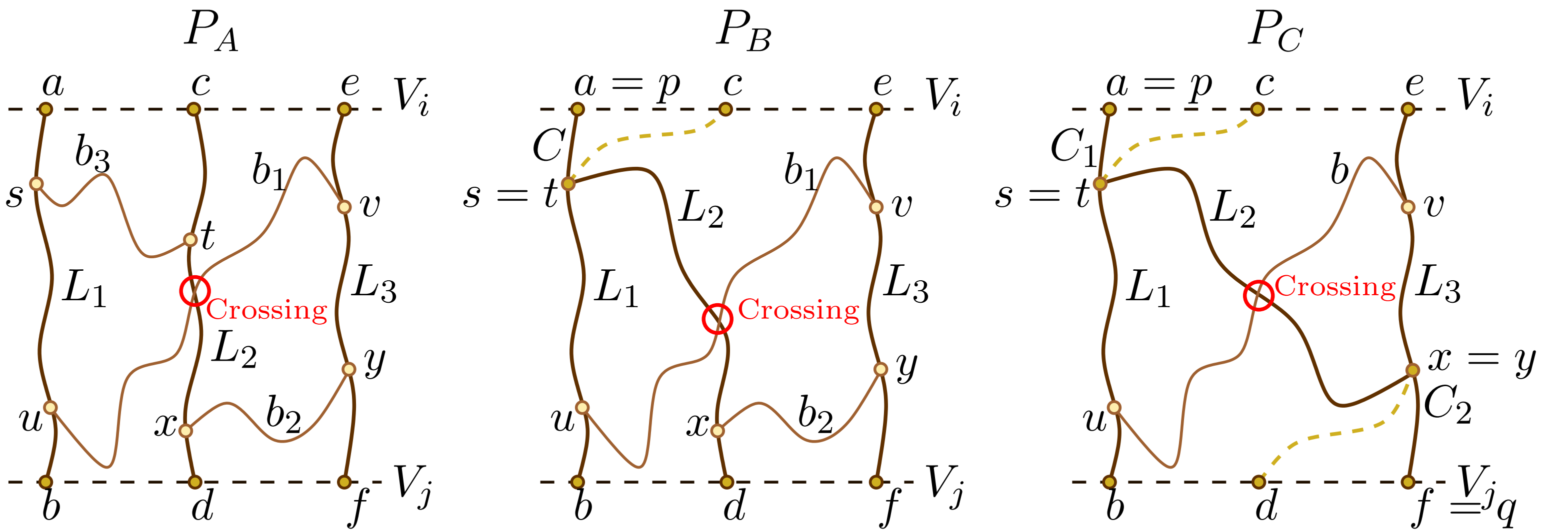
- Three patterns for hierarchies

Hierarchy Patterns

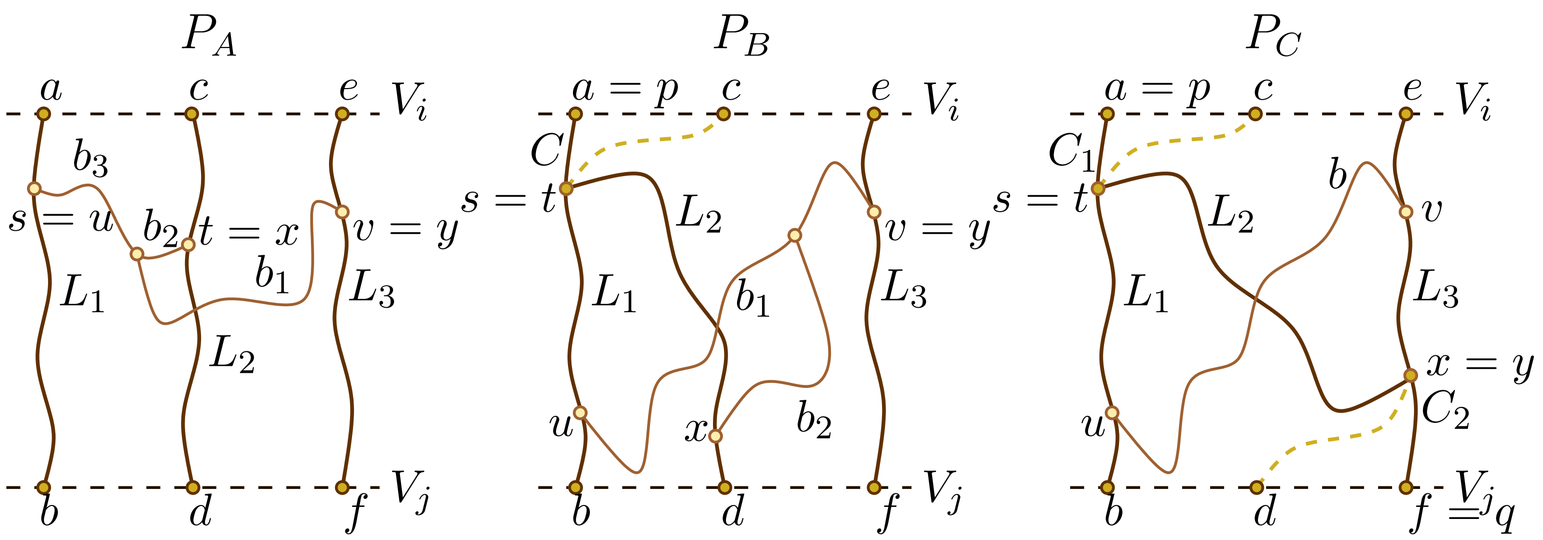


- Three patterns for hierarchies
  - Not necessarily minimal
    - May exist l, m such that $i \leq l < m \leq j$ and $|l - m| < |i - j|$

Hierarchy Patterns

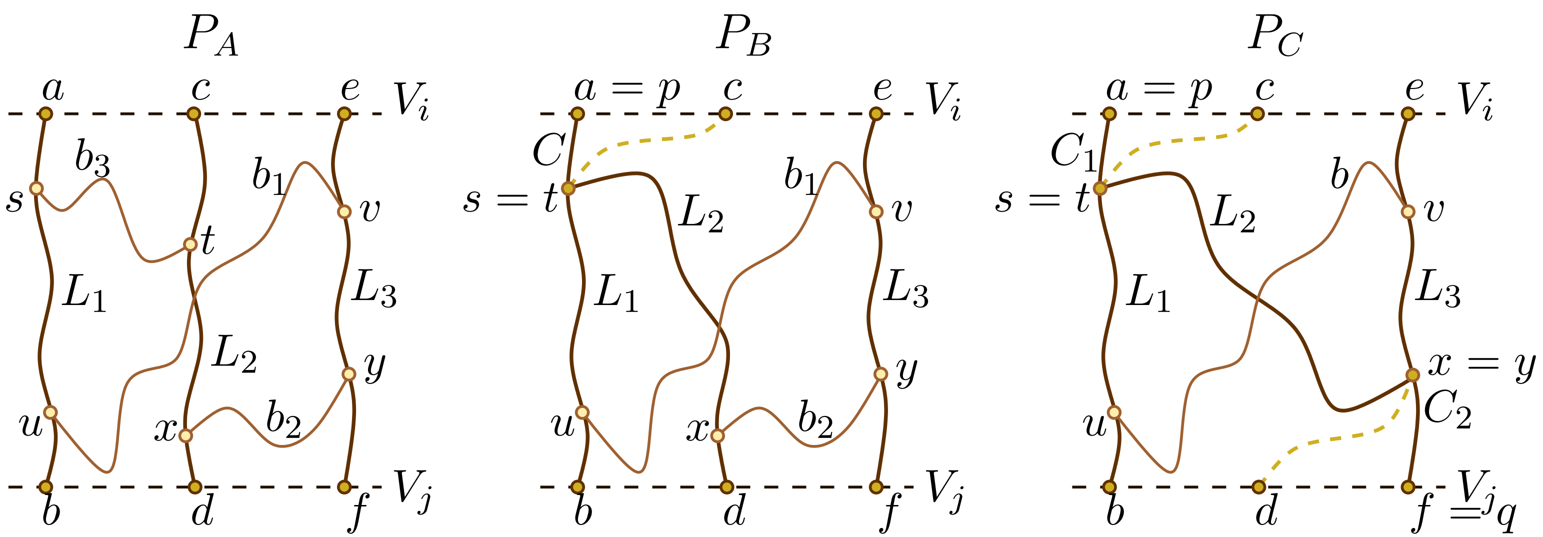


- Three patterns for hierarchies
  - Not necessarily minimal
  - P_A consists of three disjoint linking paths and three pairwise bridges
    - Bridges do *not* share vertices with linking paths except at endpoints



- Three patterns for hierarchies
  - Not necessarily minimal
  - P_A consists of three disjoint linking paths and three pairwise bridges
    - Bridges *can* share vertices with each other

Hierarchy Patterns



- Three patterns for hierarchies
  - Not necessarily minimal
  - P_A consists of three disjoint linking paths and three pairwise bridges
  - P_B and P_C are special cases of P_A

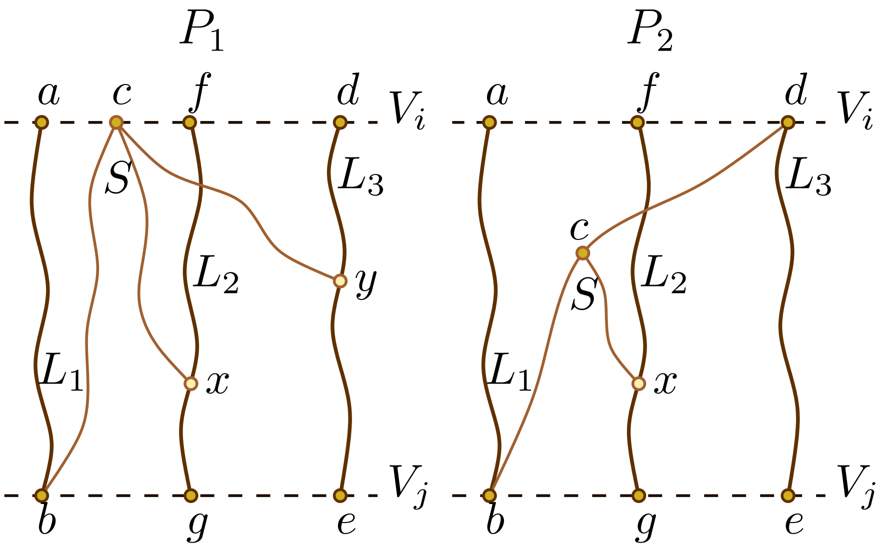
Previous MLNP Tree Patterns

- Minimum Level Non-Planar (MLNP) patterns are for more general class of level graphs

Previous MLNP Tree Patterns

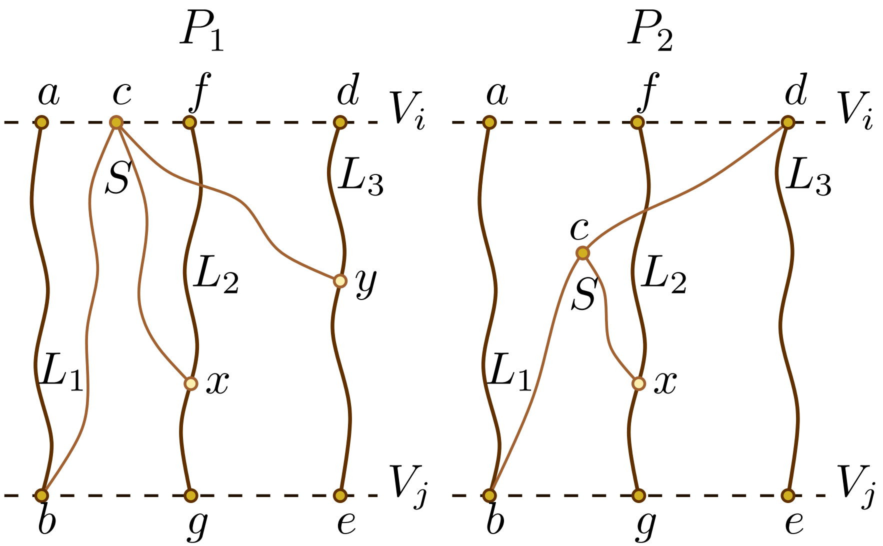
- However we only consider more restricted class of level trees

Previous MLNP Tree Patterns



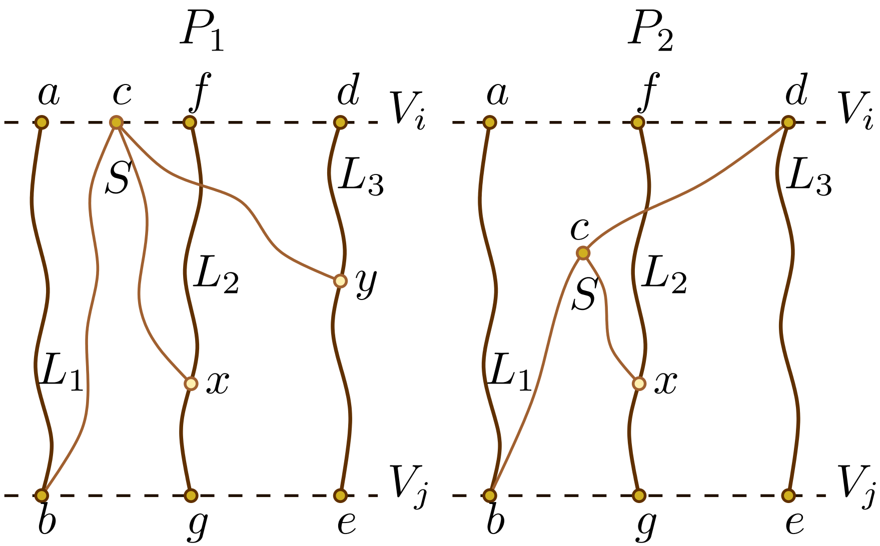
- Two patterns P_1 and P_2 previously given for trees

Previous MLNP Tree Patterns



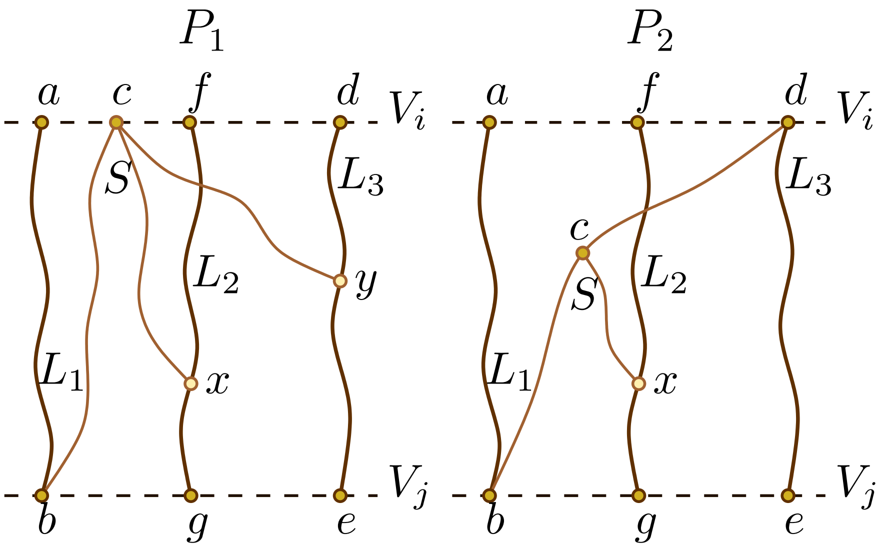
- Two patterns P_1 and P_2 previously given for trees
  - Healy *et al*. claimed these sufficient

Previous MLNP Tree Patterns



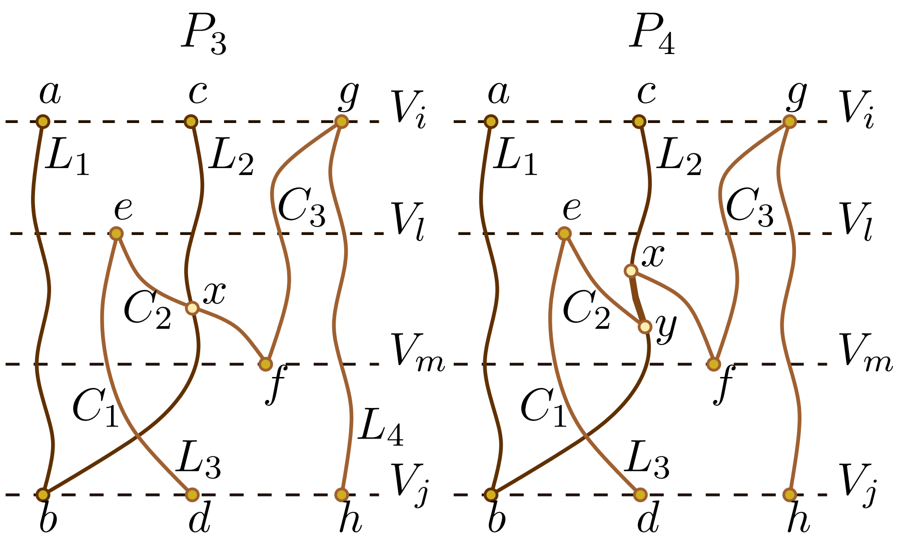
- Two patterns P_1 and P_2 previously given for trees
  - Healy *et al*. claimed these sufficient
  - Both are special cases of P_A

Previous MLNP Tree Patterns



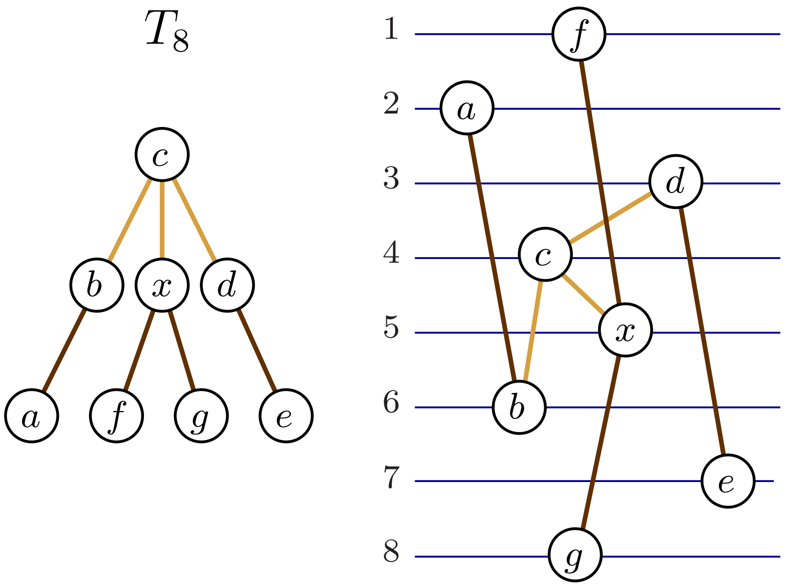
- Two patterns P_1 and P_2 previously given for trees
  - Healy *et al*. claimed these sufficient
  - Both are special cases of P_A
  - Neither have degree-4 vertex—cannot match T_9

Two New MLNP Tree Patterns



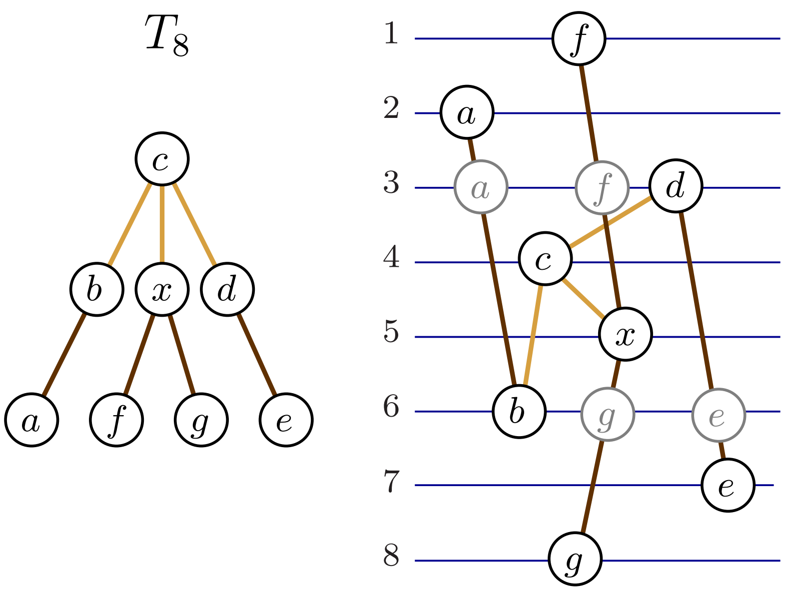
- Need two more patterns P_3 and P_4 based on T_9

Deriving P_2 from Forbidden Tree T_8



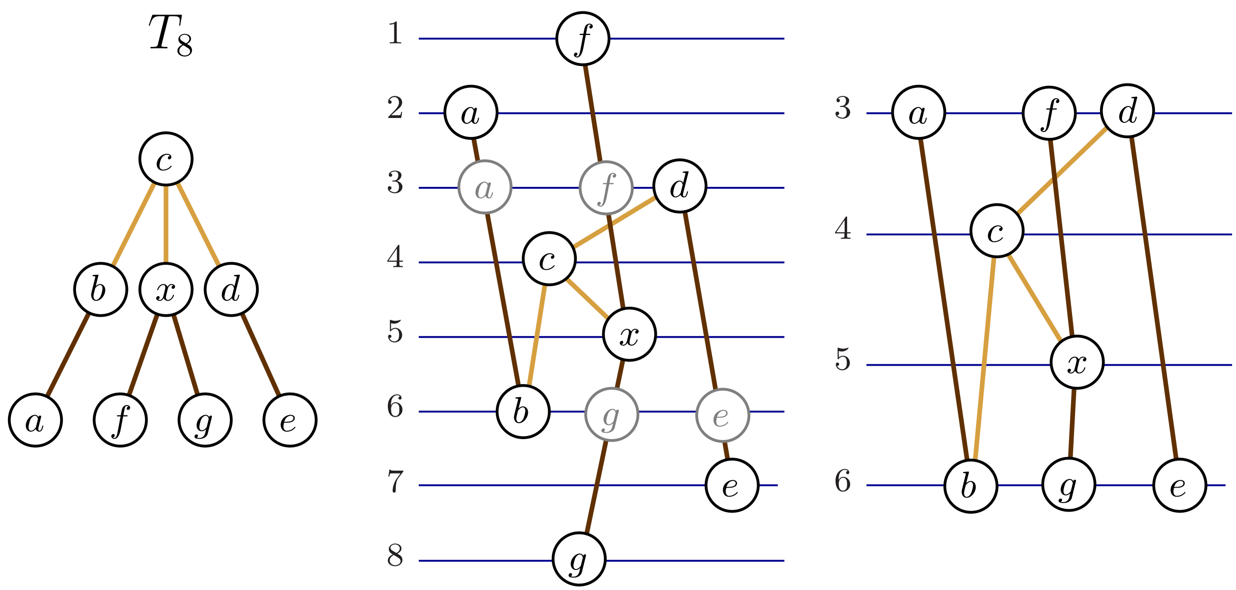
- Start with level non-planar leveling for T_8

Deriving P_2 from Forbidden Tree T_8



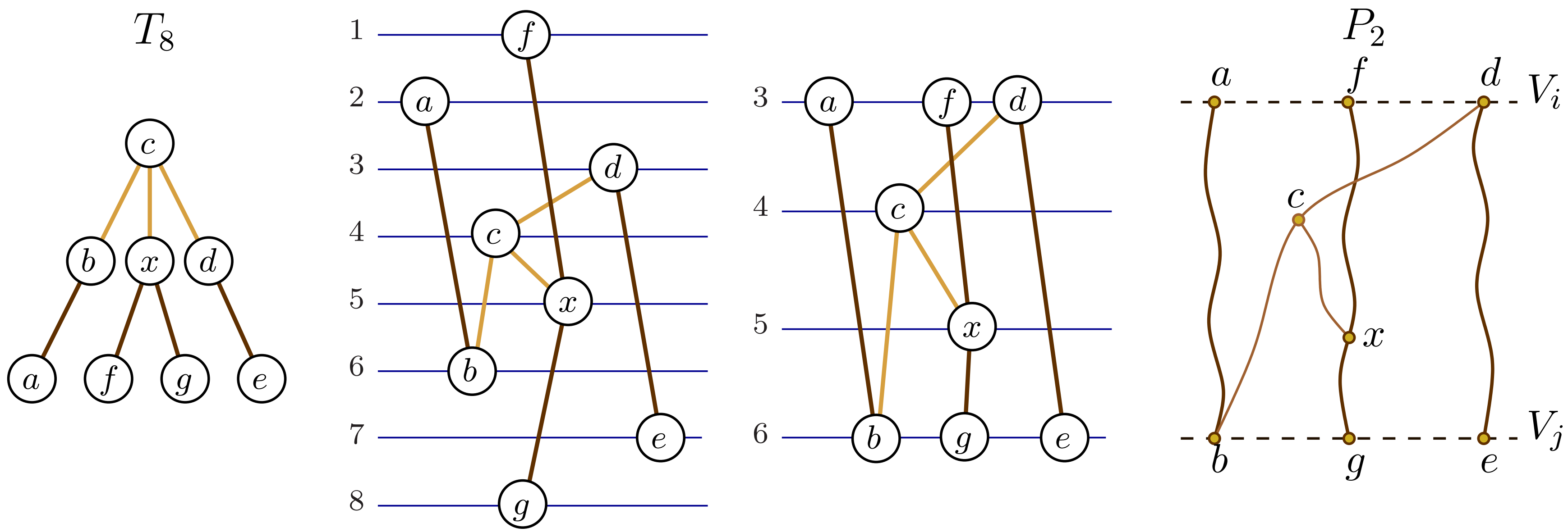
- Start with level non-planar leveling for T_8
  - Add dummy vertices to levels 3 and 6

Deriving P_2 from Forbidden Tree T_8



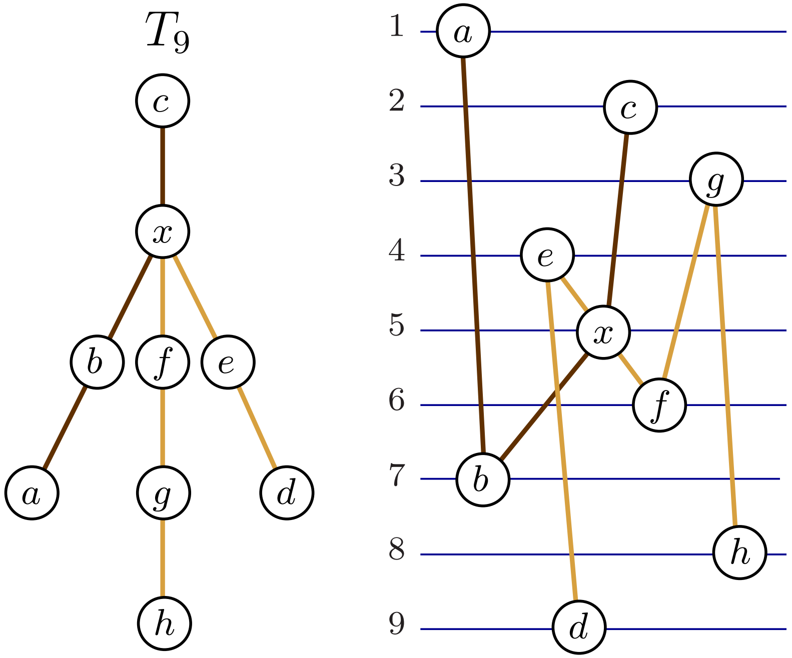
- Start with level non-planar leveling for T_8
  - Extract a proper subtree between levels 3 and 6

Deriving P_2 from Forbidden Tree T_8



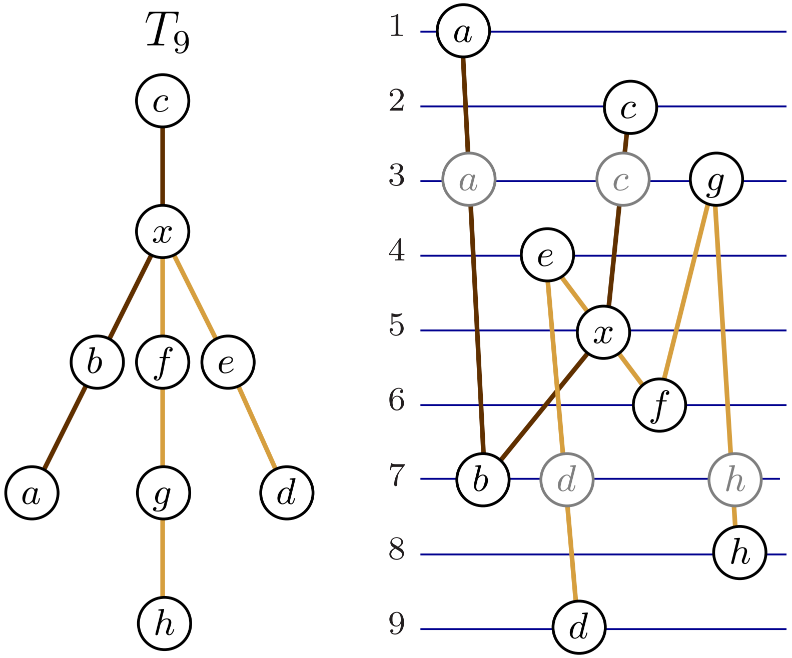
- Start with level non-planar leveling for T_8
  - Extract a proper subtree between levels 3 and 6
- Generalize into a pattern

Deriving P_3 from Forbidden Tree T_9



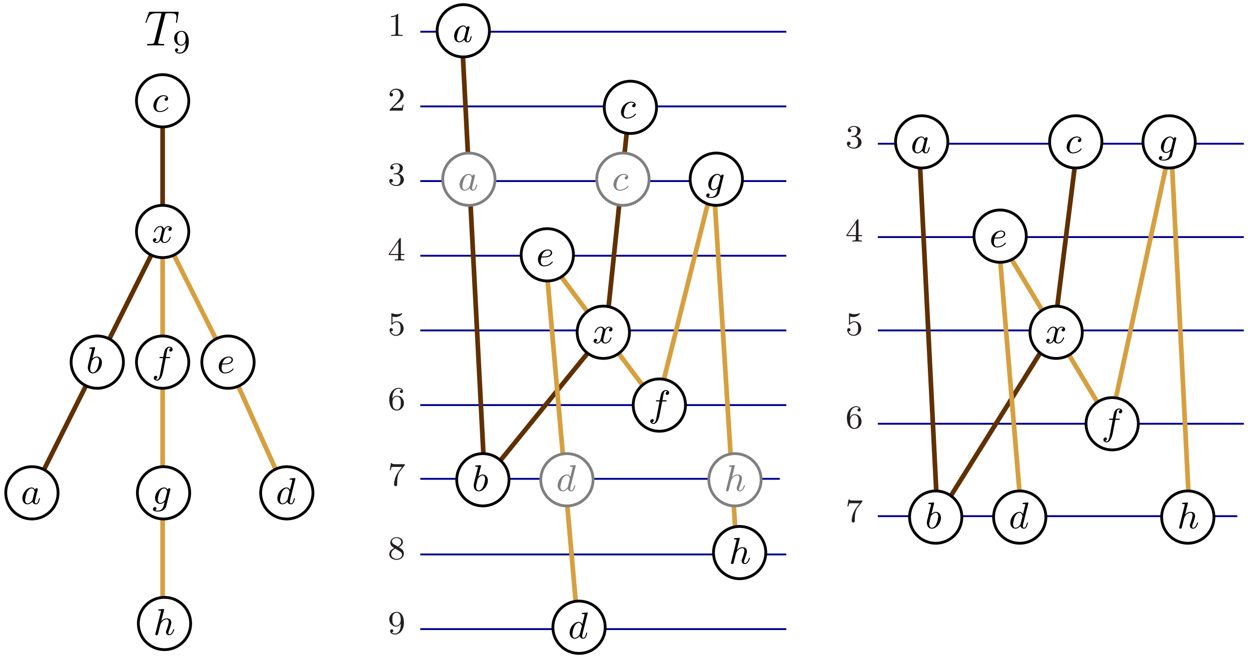
- Start with level non-planar leveling for T_9

Deriving P_3 from Forbidden Tree T_9



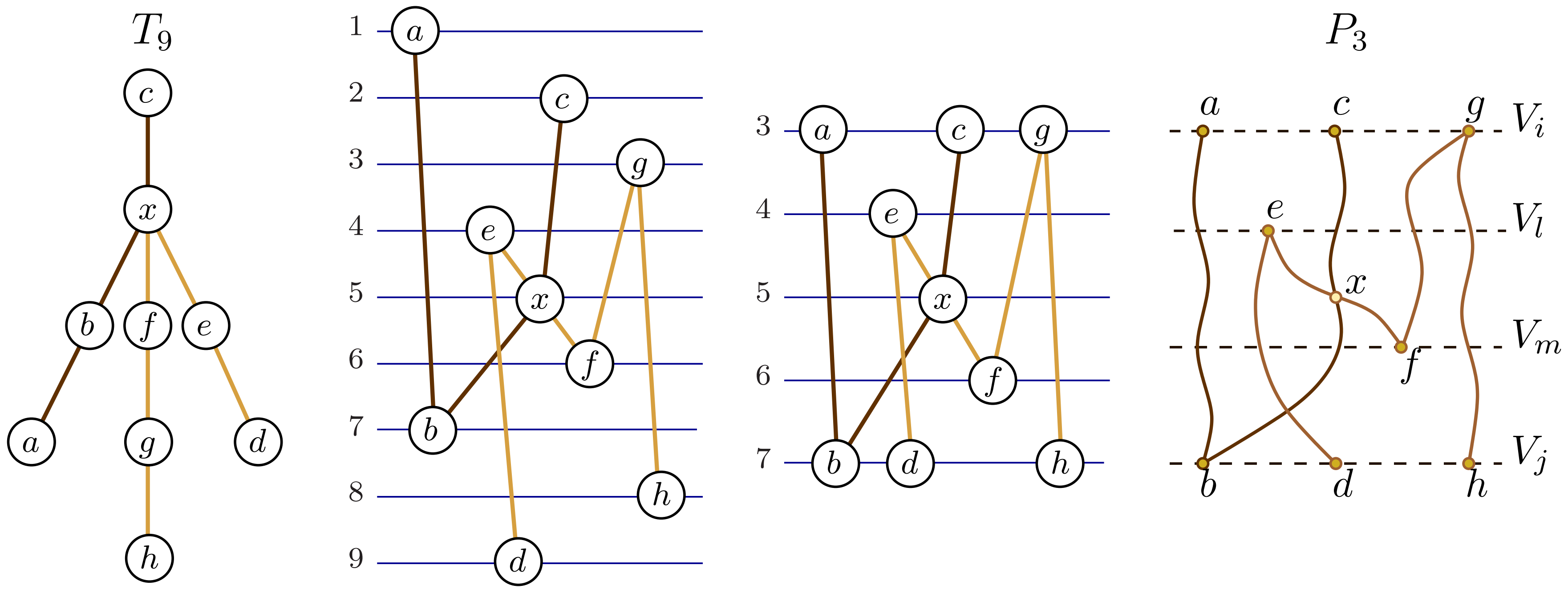
- Start with level non-planar leveling for T_9
  - Add dummy vertices to levels 3 and 7

Deriving P_3 from Forbidden Tree T_9



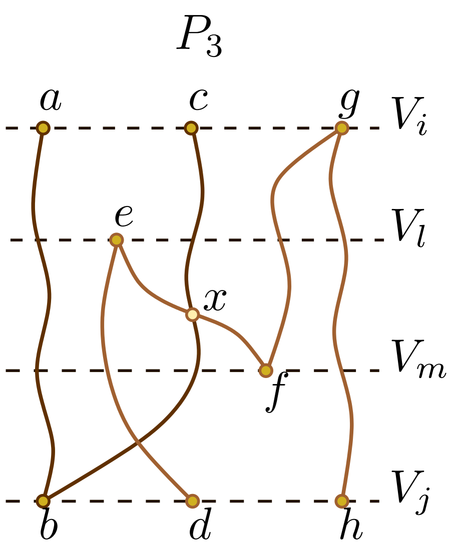
- Start with level non-planar leveling for T_9
  - Extract a proper subtree between levels 3 and 7

Deriving P_3 from Forbidden Tree T_9



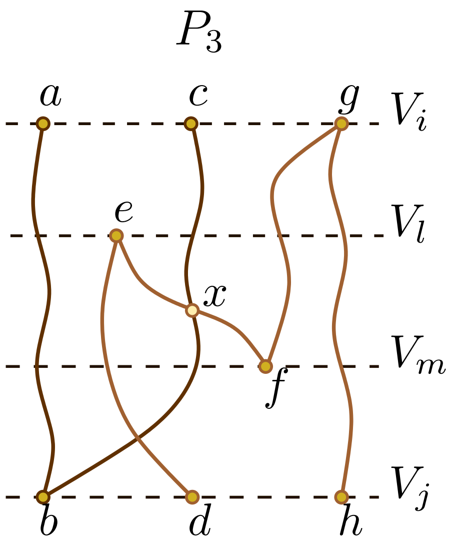
- Start with level non-planar leveling for T_9
  - Extract a proper subtree between levels 3 and 7
- Generalize into a pattern

Deriving P_4 from P_3



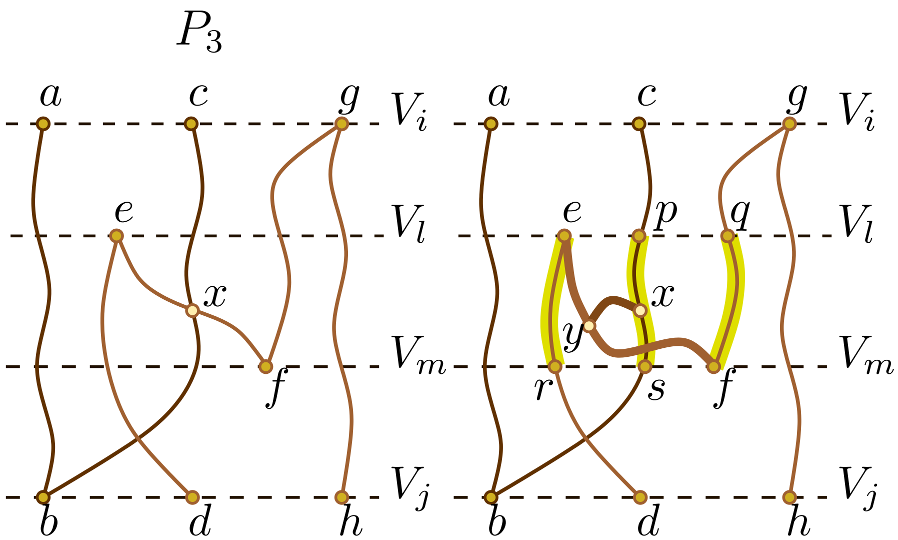
- Start with MLNP pattern P_3

Deriving P_4 from P_3



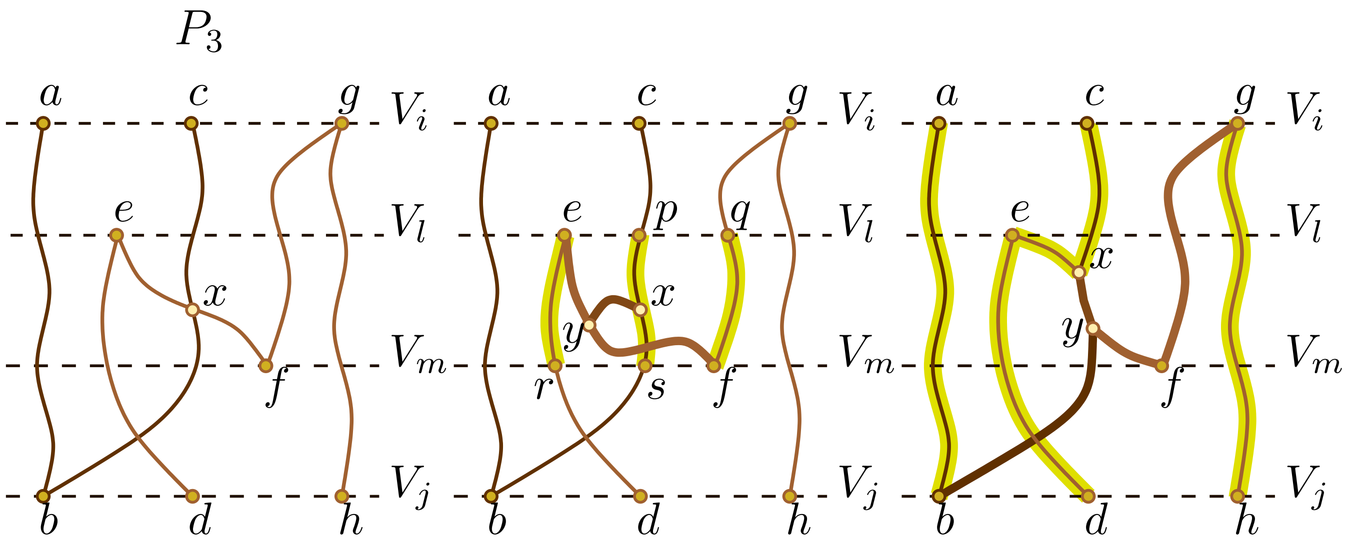
- Split degree-4 vertex into two degree-3 vertices

Deriving P_4 from P_3



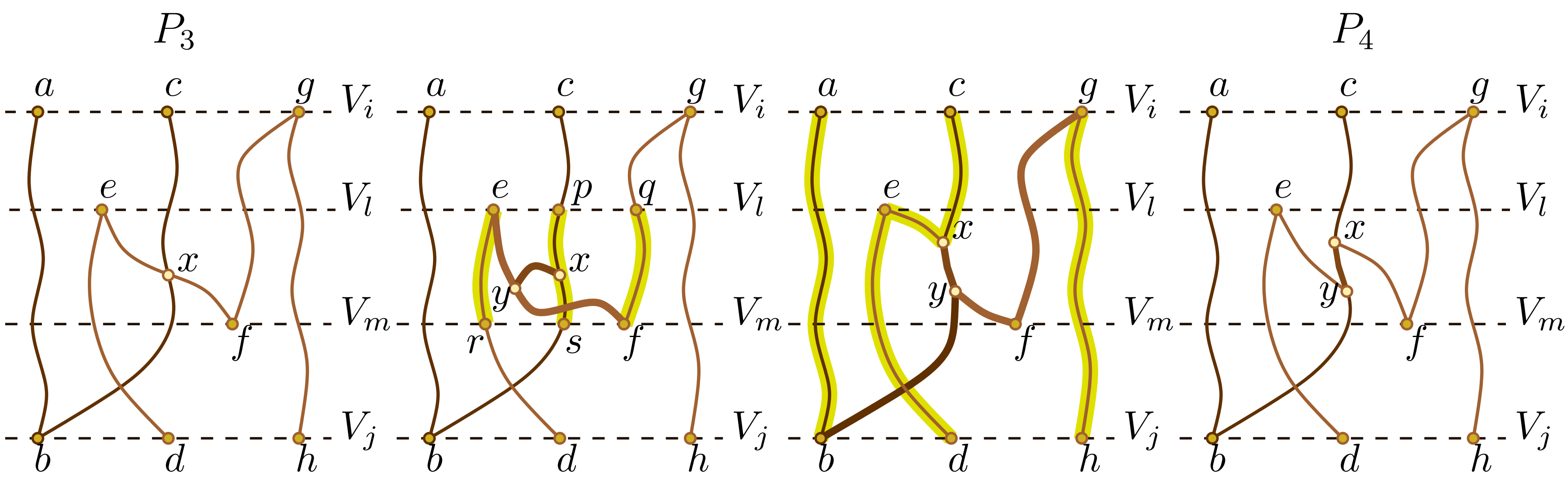
- Split degree-4 vertex into two degree-3 vertices
  - First way gives P_2 between V_l and V_m

Deriving P_4 from P_3



- Split degree-4 vertex into two degree-3 vertices
  - First way gives P_2 between V_l and V_m
  - Second way gives P_2 between V_i and V_j

Deriving P_4 from P_3



- Split degree-4 vertex into two degree-3 vertices
  - First way gives P_2 between V_l and V_m
  - Second way gives P_2 between V_i and V_j
  - Third way gives P_4

Outline – New Patterns

- Background
- Previous Patterns
- New Patterns

Outline – New Patterns

- Background
- Previous Patterns
- New Patterns
  - Three things to prove

Outline – New Patterns

- Background
- Previous Patterns
- New Patterns
  - Three things to prove
    - Minimality

Outline – New Patterns

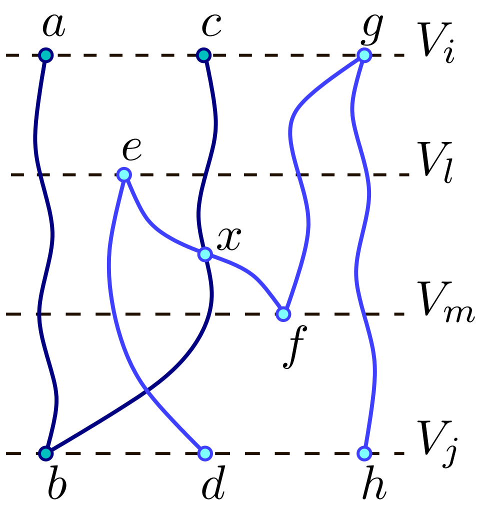
- Background
- Previous Patterns
- New Patterns
  - Three things to prove
    - Minimality
    - Necessity

Outline – New Patterns

- Background
- Previous Patterns
- New Patterns
  - Three things to prove
    - Minimality
    - Necessity
    - Sufficiency

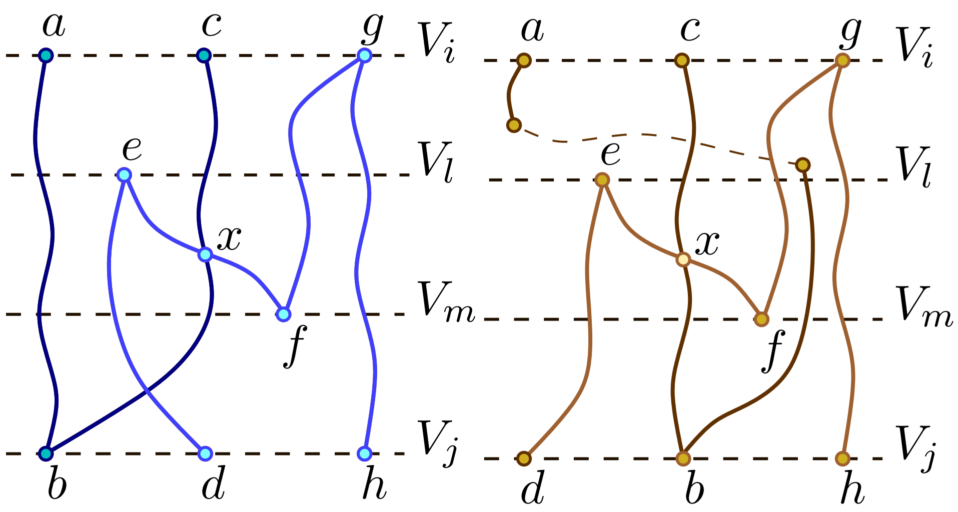
Minimality of P_3

- Start with MLNP pattern P_3
  - P_3 has at least one crossing



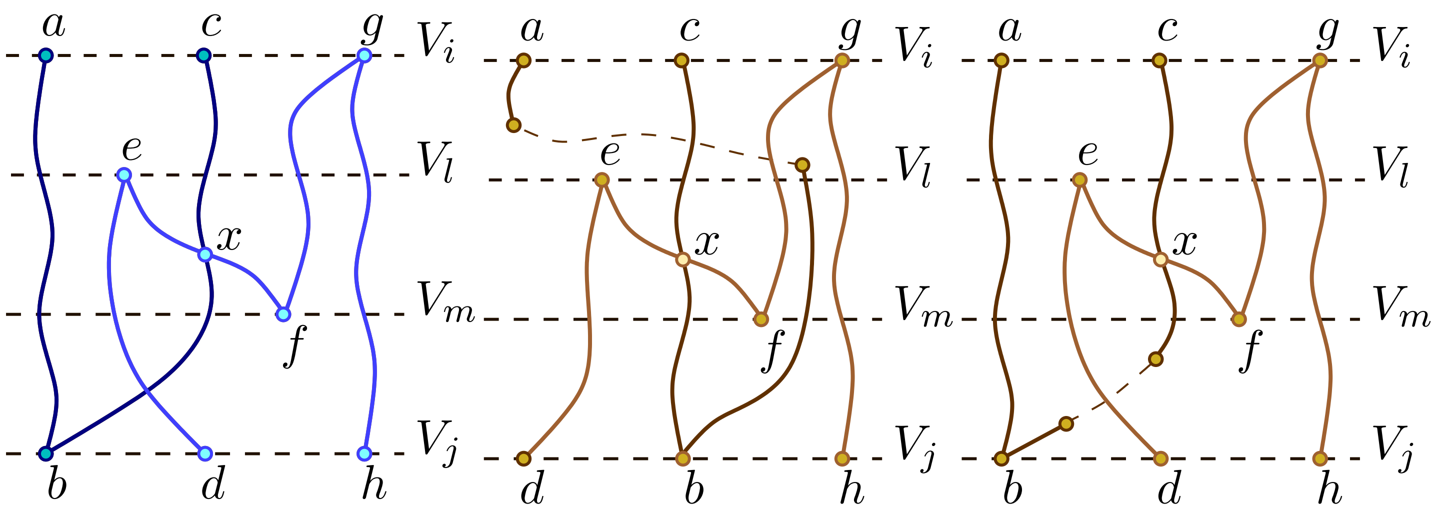
Minimality of P_3

- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $a \rightsquigarrow b$

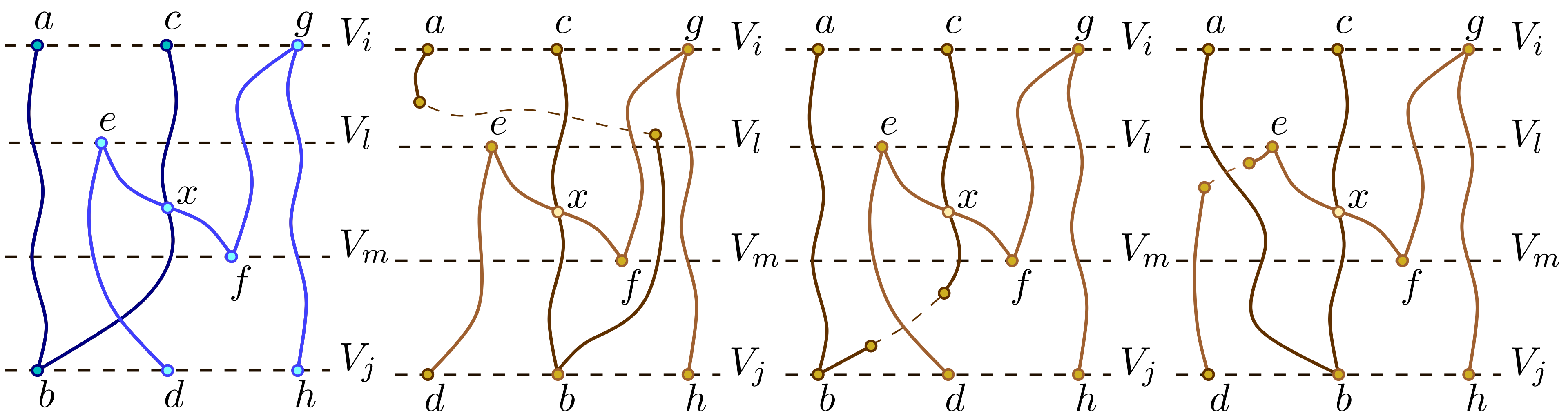


Minimality of P_3

- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $b \rightsquigarrow x$

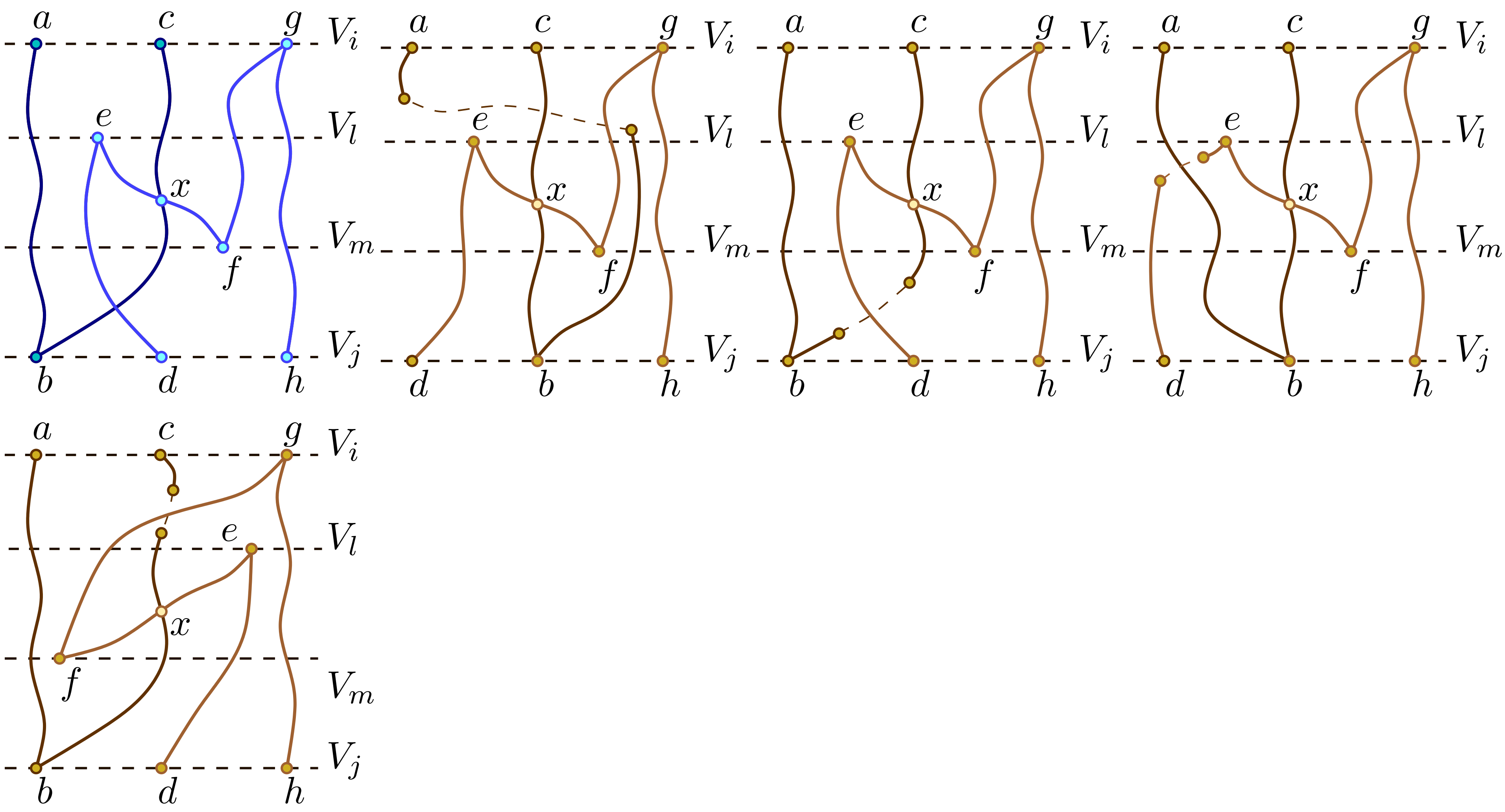


- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $d \rightsquigarrow x$



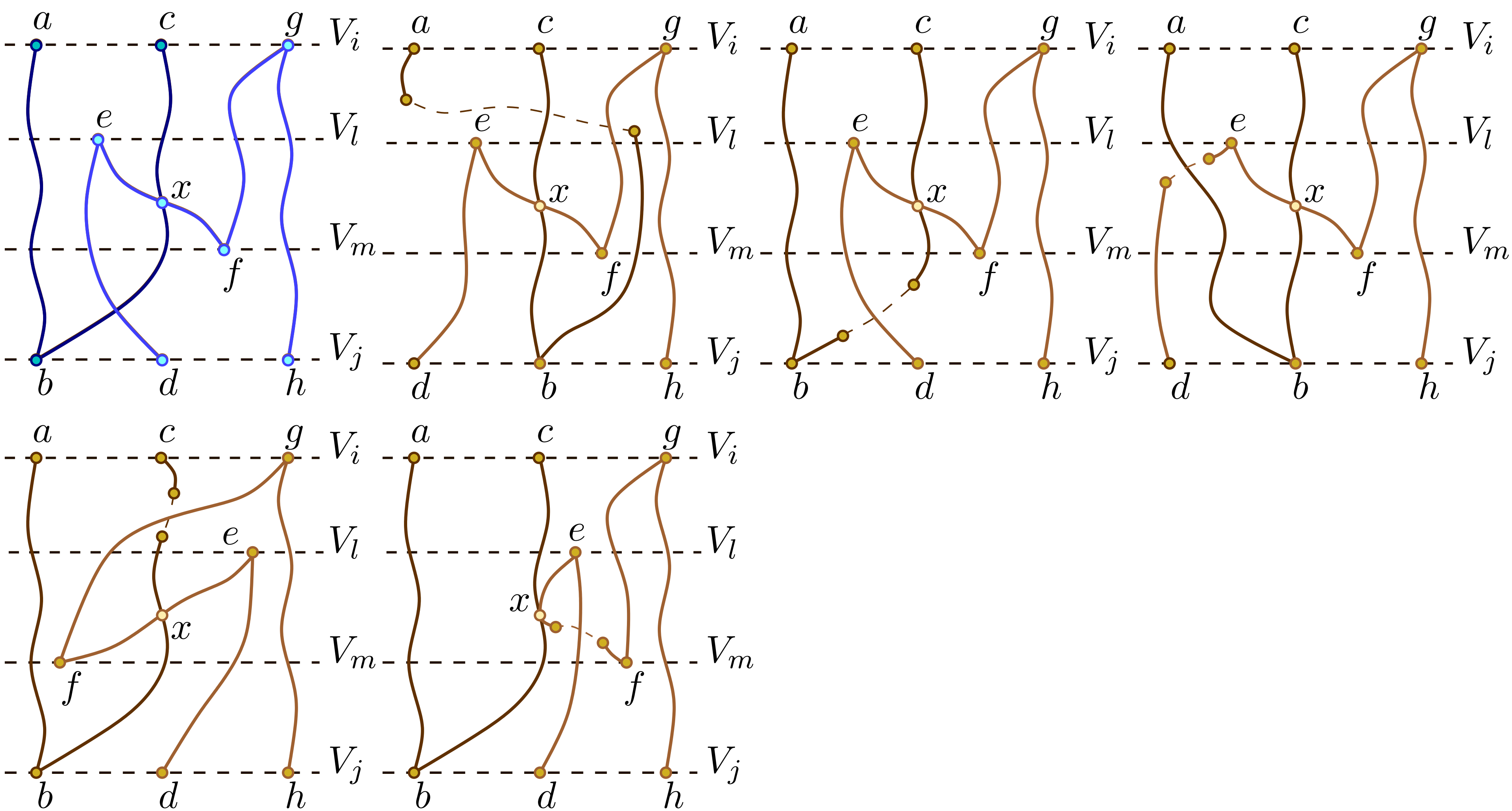
Minimality of P_3

- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $x \rightsquigarrow c$



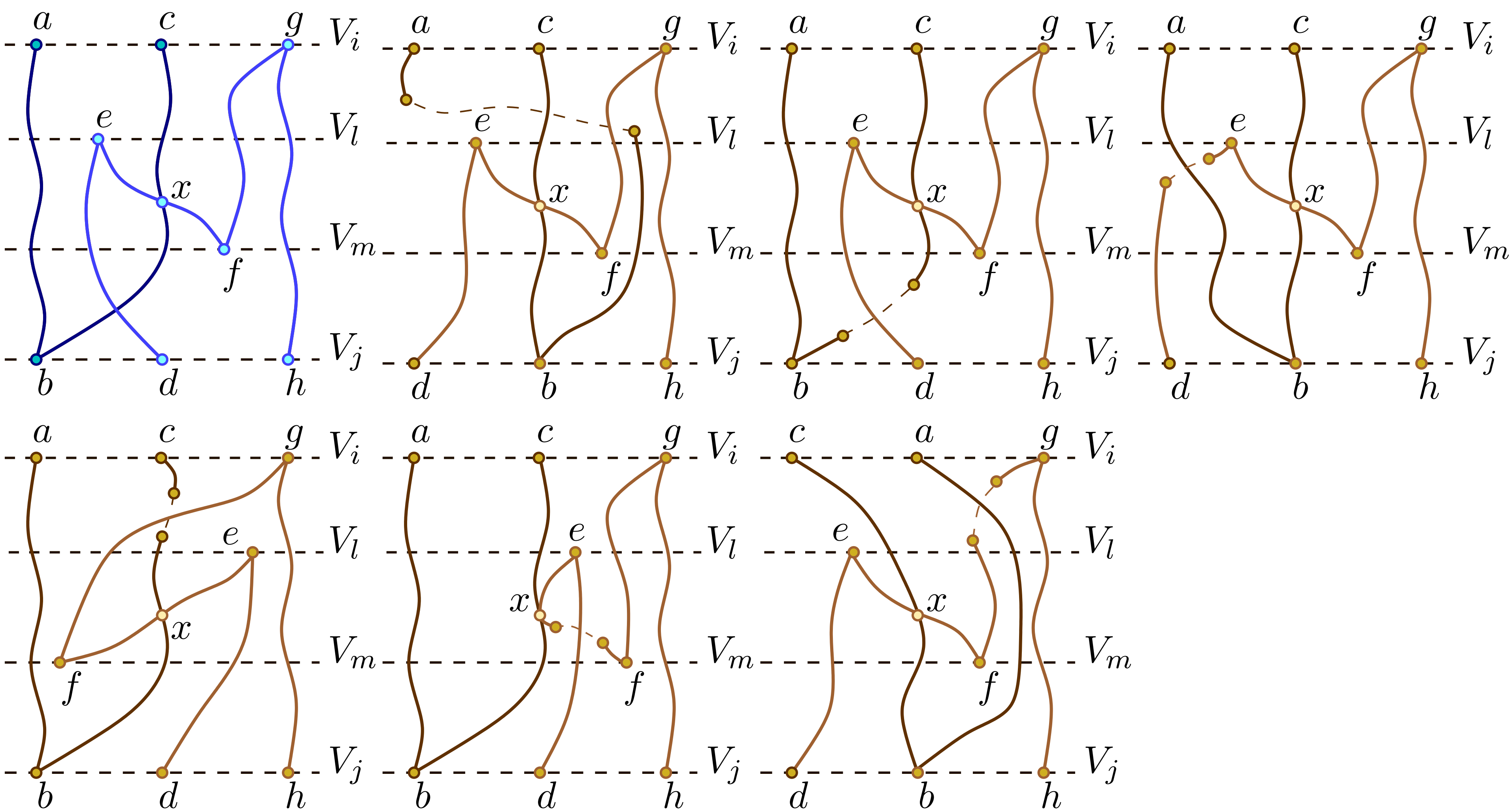
Minimality of P_3

- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $x \rightsquigarrow f$



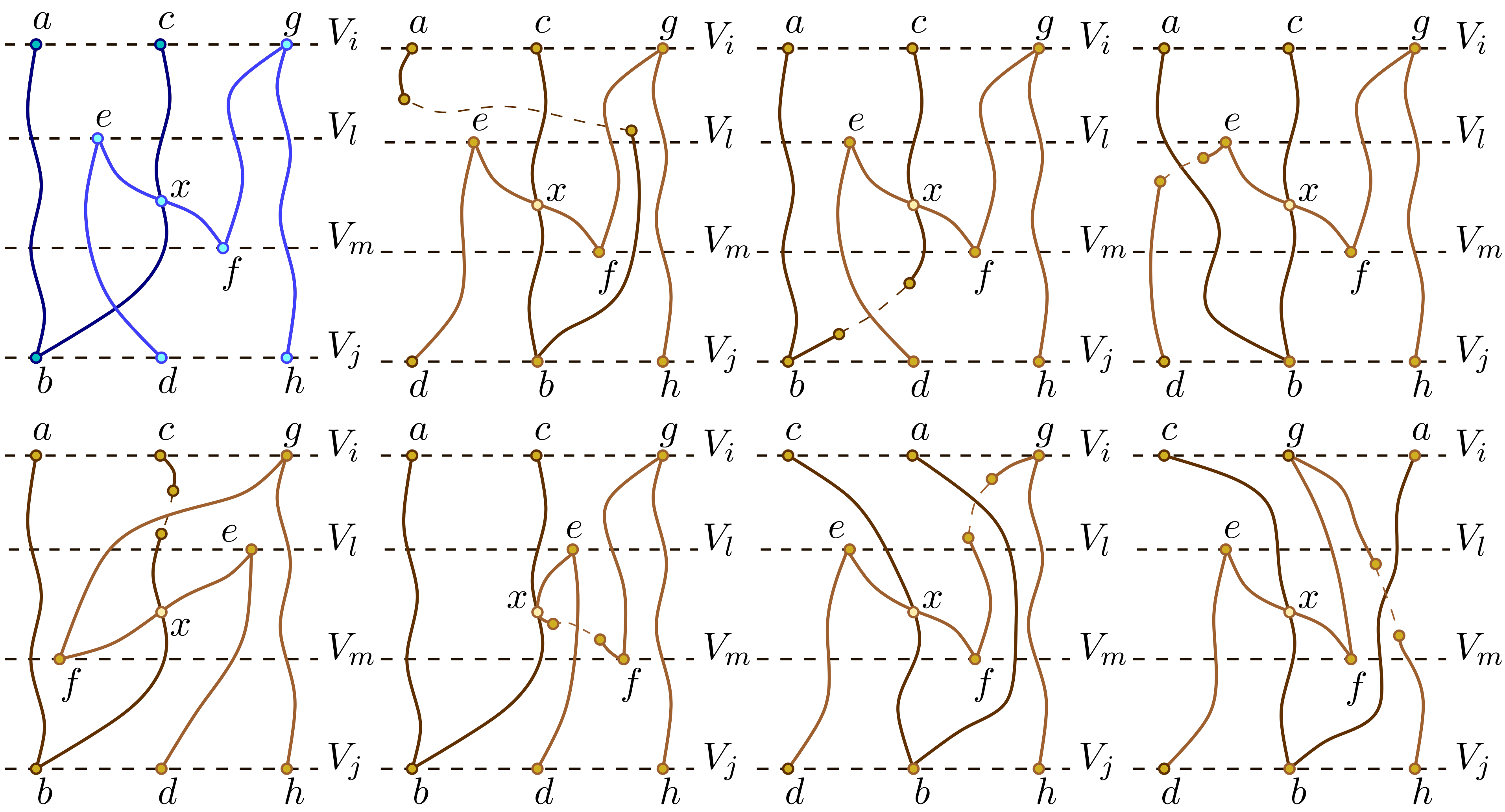
Minimality of P_3

- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $f \rightsquigarrow g$



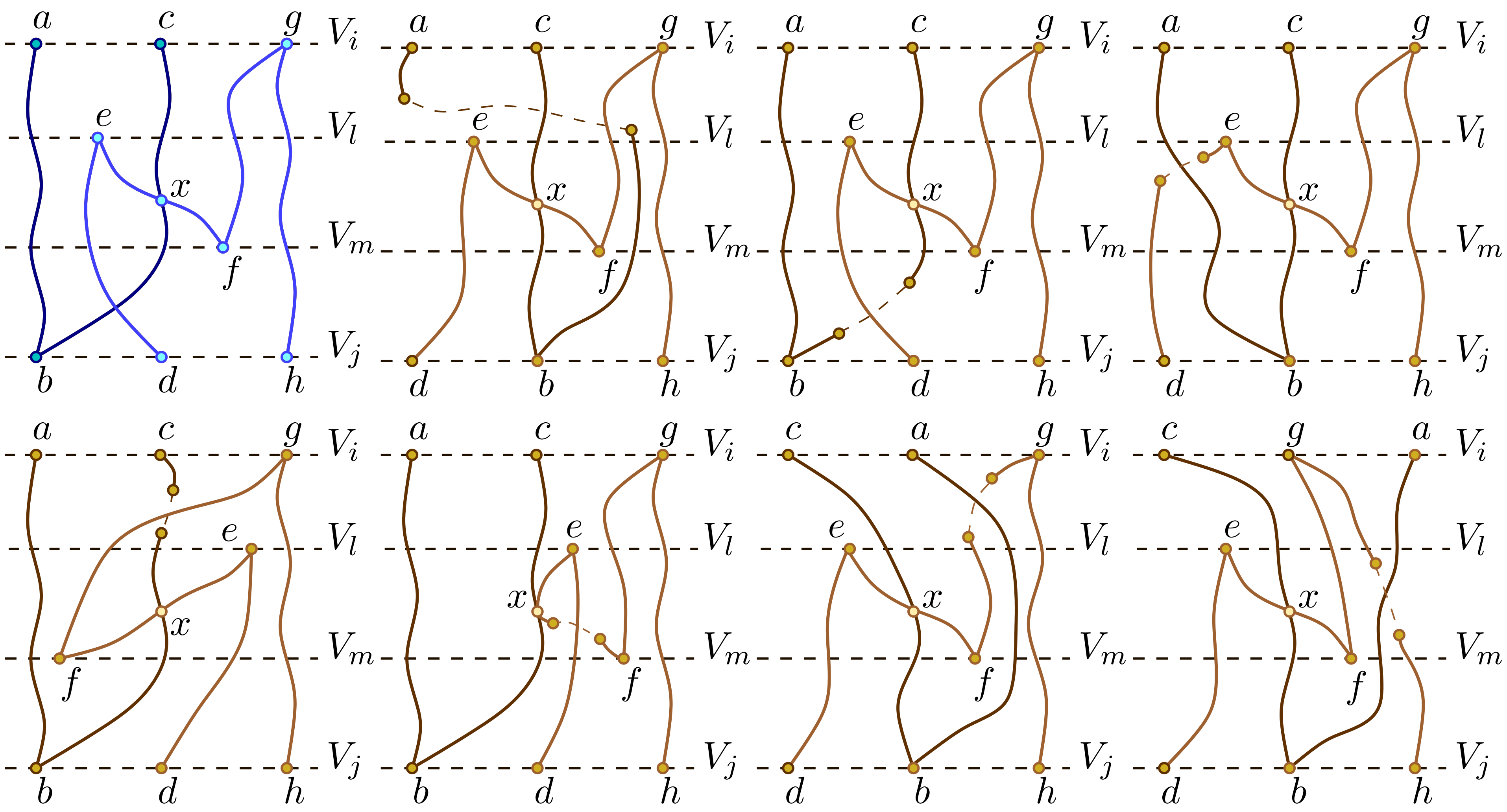
Minimality of P_3

- Next consider seven distinct ways of cutting an edge of P_3
  - Edge along $g \rightsquigarrow h$

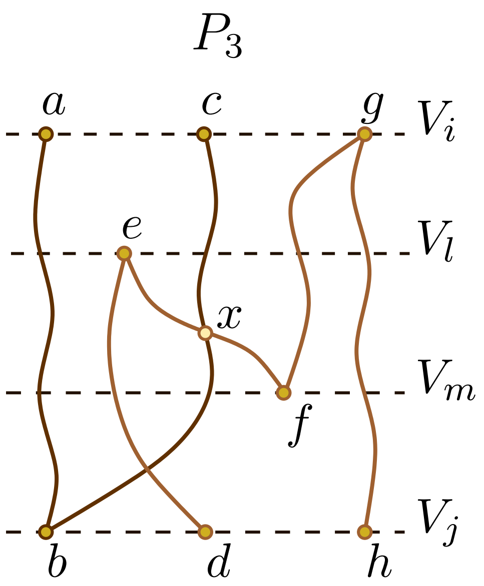


Minimality of P_3

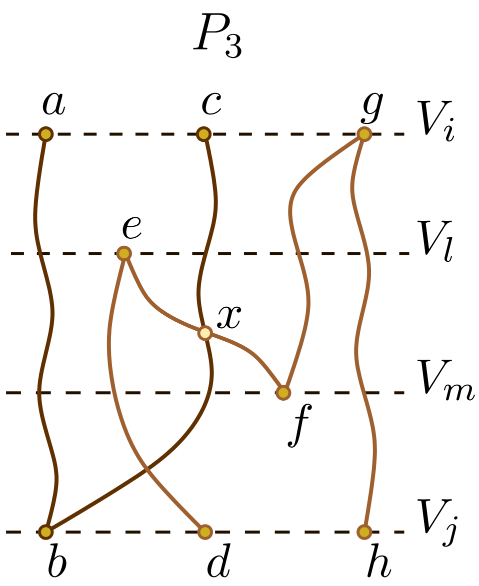
- Next consider seven distinct ways of cutting an edge of P_3
  - All seven ways allow for a level planar embedding



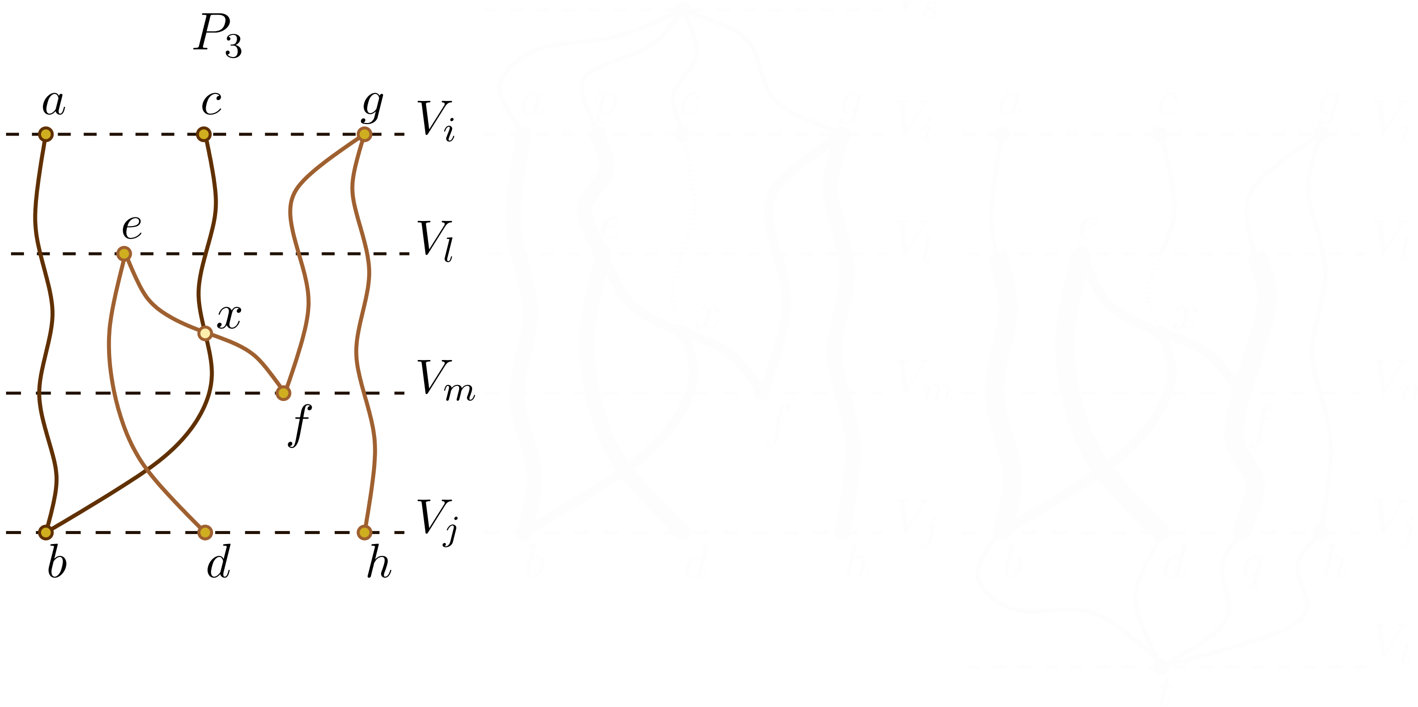
Necessity – P_3 is Not a Hierarchy Pattern



- Nonplanarity of P_3 same as argument for T_9

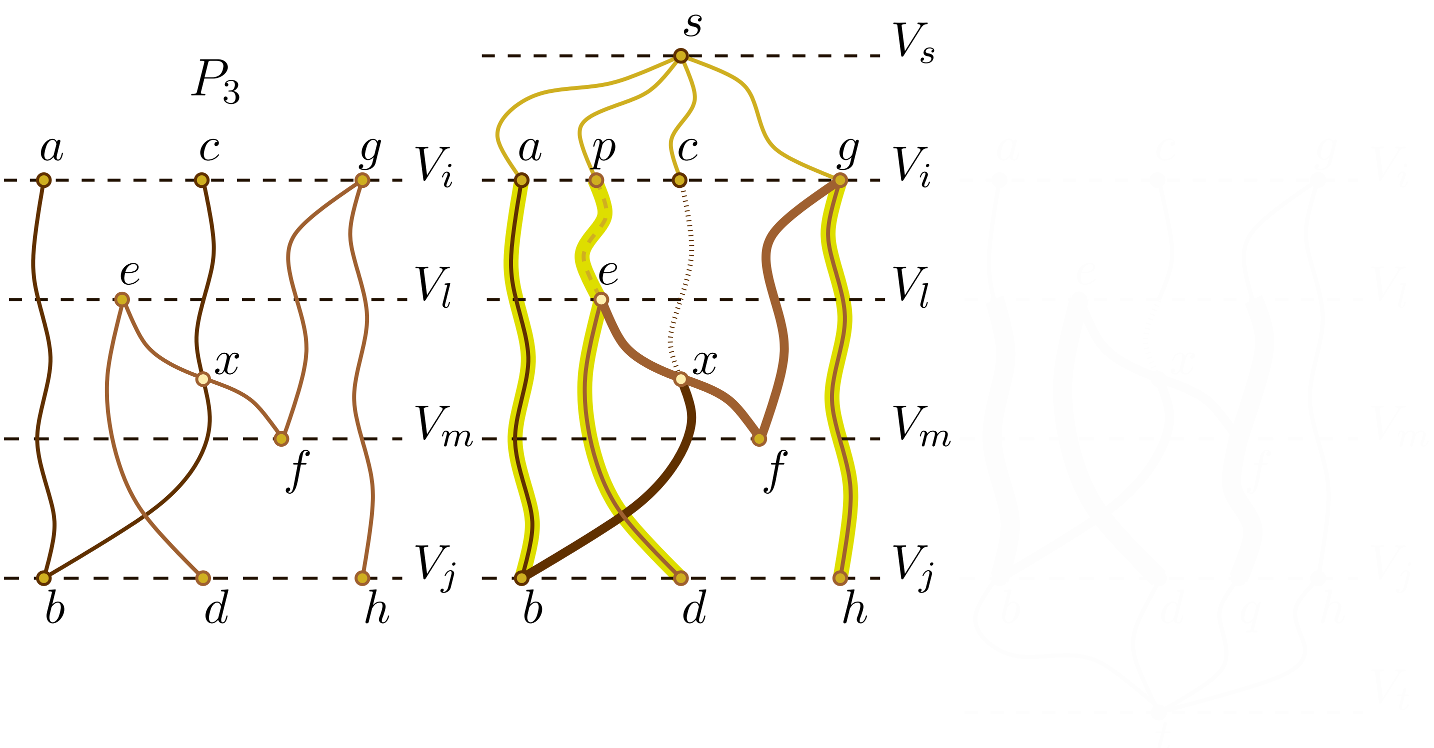


- Nonplanarity of P_3 same as argument for T_9
- Augmenting Pattern P_3 to Hierarchy



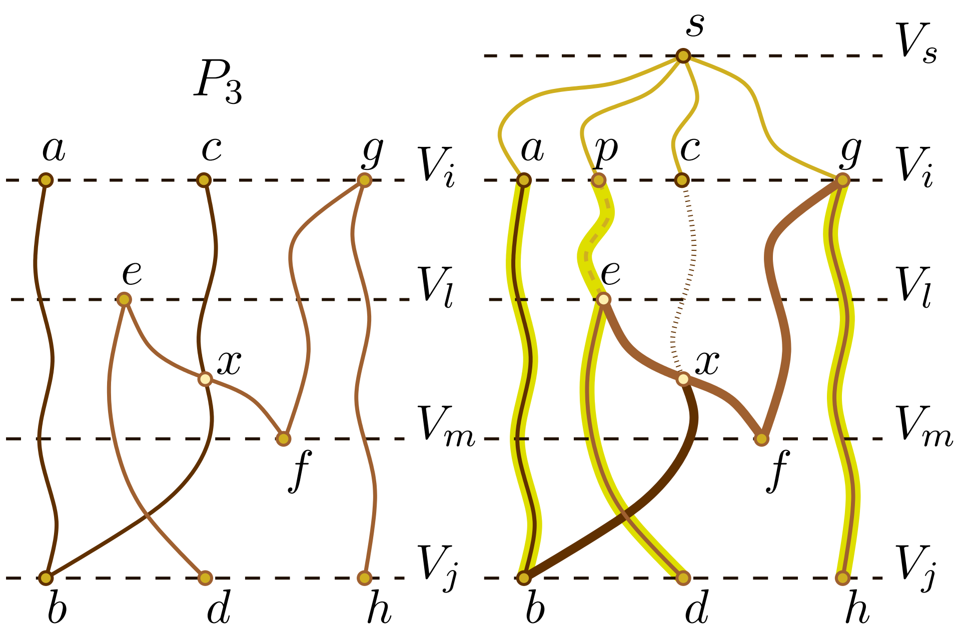
- Nonplanarity of P_3 same as argument for T_9
- Augmenting Pattern P_3 to Hierarchy
  - Start with MLNP pattern P_3

Necessity – P_3 is Not a Hierarchy Pattern



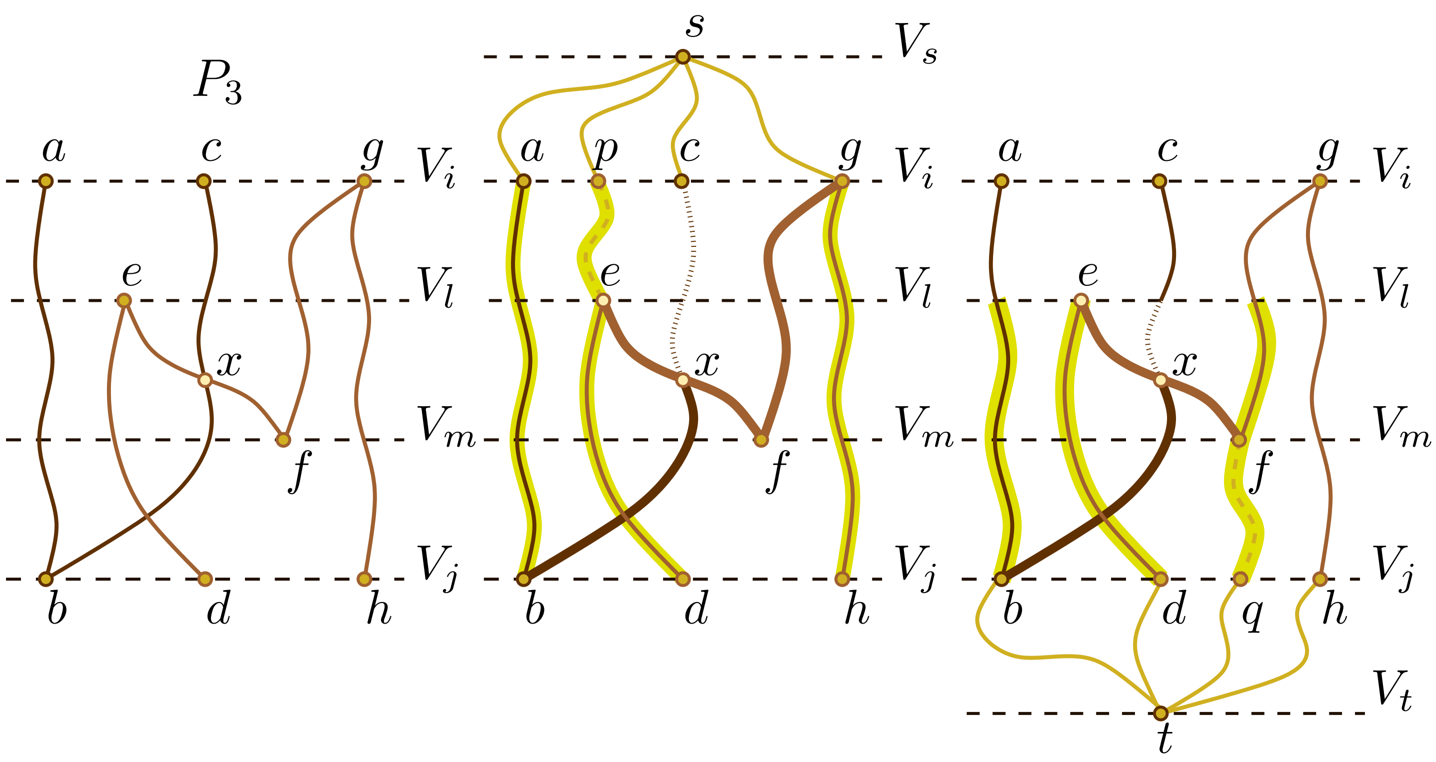
- Nonplanarity of P_3 same as argument for T_9
- Augmenting Pattern P_3 to Hierarchy
  - Augment to a hierarchy from above

Necessity – P_3 is Not a Hierarchy Pattern



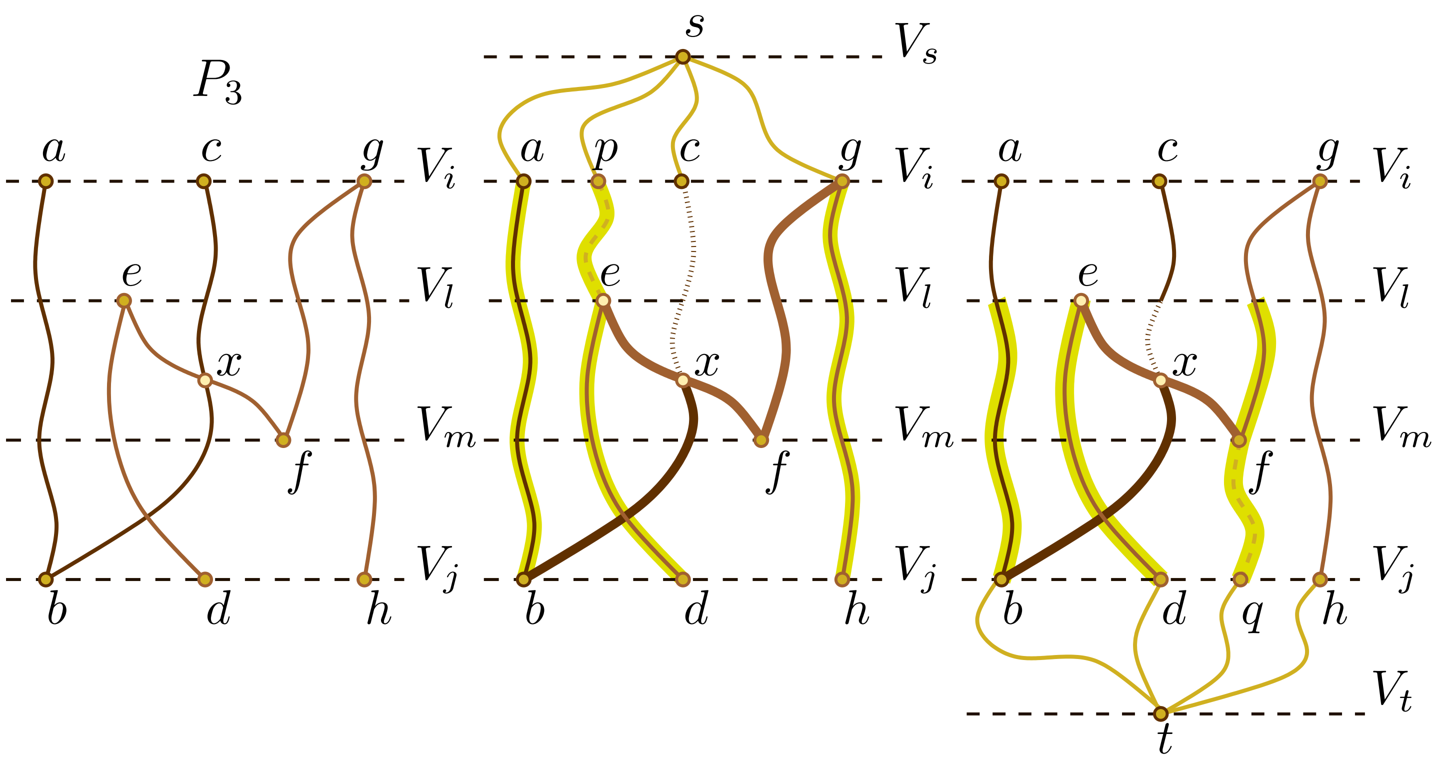
- Nonplanarity of P_3 same as argument for T_9
- Augmenting Pattern P_3 to Hierarchy
  - Augment to a hierarchy from above
    - Contains P_2 between V_i and V_j

Necessity – P_3 is Not a Hierarchy Pattern



- Nonplanarity of P_3 same as argument for T_9
- Augmenting Pattern P_3 to Hierarchy
  - Augment to a hierarchy from below

Necessity – P_3 is Not a Hierarchy Pattern



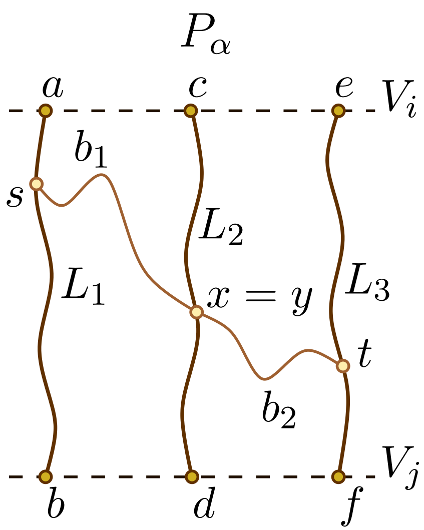
- Nonplanarity of P_3 same as argument for T_9
- Augmenting Pattern P_3 to Hierarchy
  - Augment to a hierarchy from below
    - Contains P_2 between V_l and V_j

Sufficiency – Minimal Requirements

- Three MLNP pattern prototypes - P_A minus a bridge

Sufficiency – Minimal Requirements

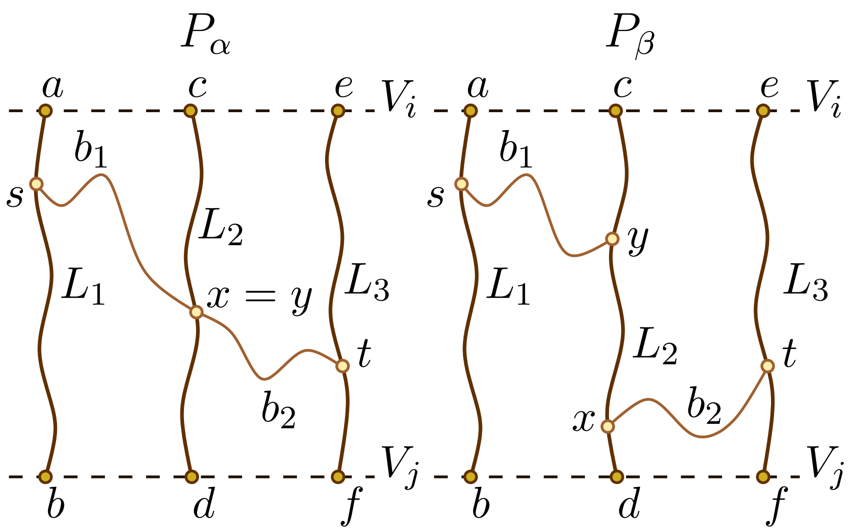
- Three MLNP pattern prototypes - P_A minus a bridge



  - Pattern P_α
    - $x = y$

Sufficiency – Minimal Requirements

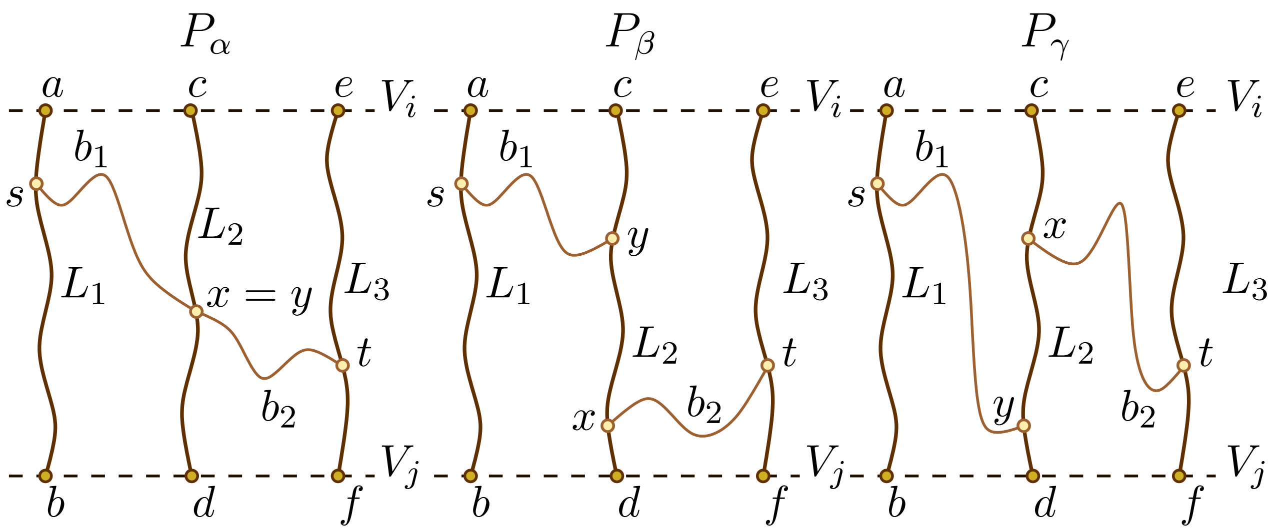
- Three MLNP pattern prototypes - P_A minus a bridge



  - Pattern P_β
    - $\phi(x) < \phi(y)$

Sufficiency – Minimal Requirements

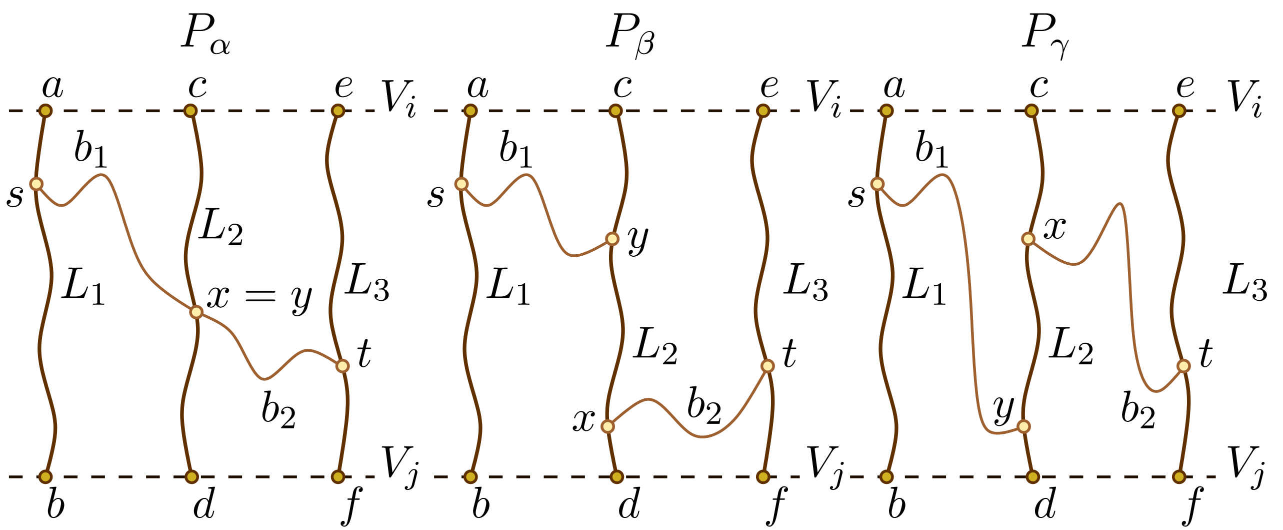
- Three MLNP pattern prototypes - P_A minus a bridge



  - Pattern P_γ
    - $\phi(x) > \phi(y)$

Sufficiency – Minimal Requirements

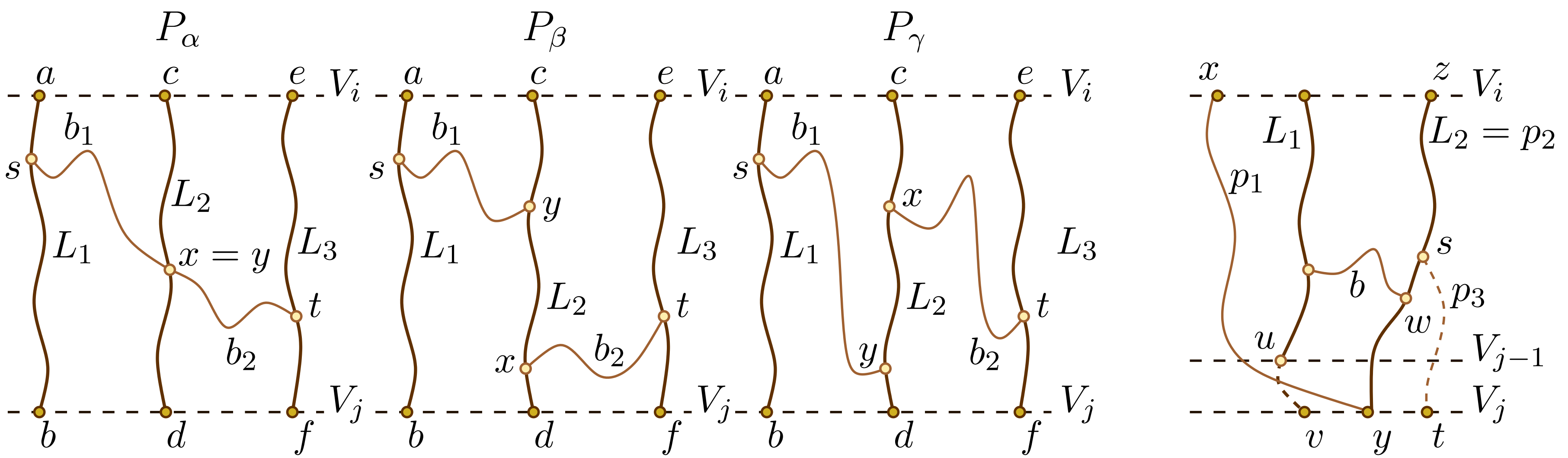
- Three MLNP pattern prototypes - P_A minus a bridge



  - All have three disjoints linking paths
    - Why?

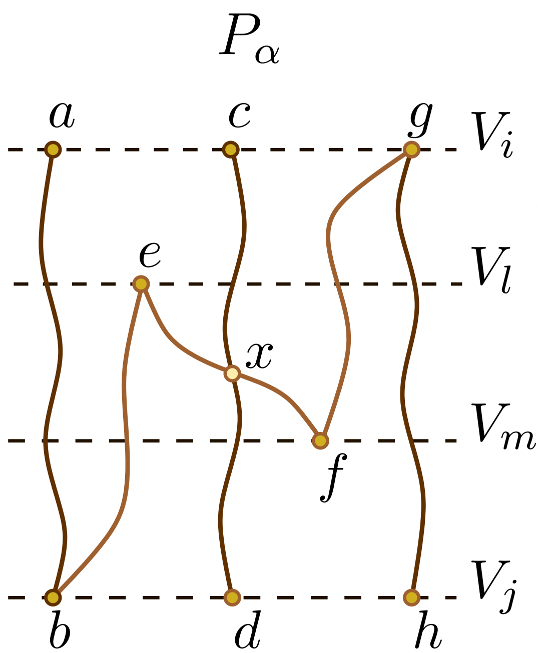
Sufficiency – Minimal Requirements

- Three MLNP pattern prototypes - P_A minus a bridge



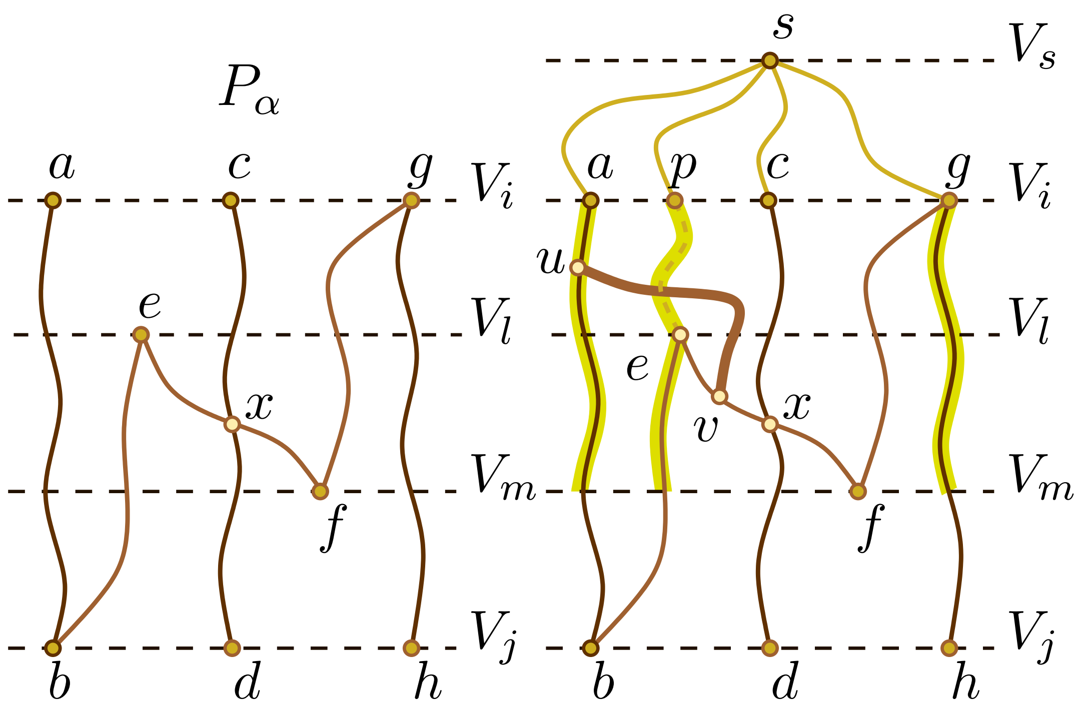
  - All have three disjoints linking paths
    - Two disjoints path cannot force a crossing

Sufficiency – Showing Completeness



- Augment P_α to get P_3
  - Start with P_α

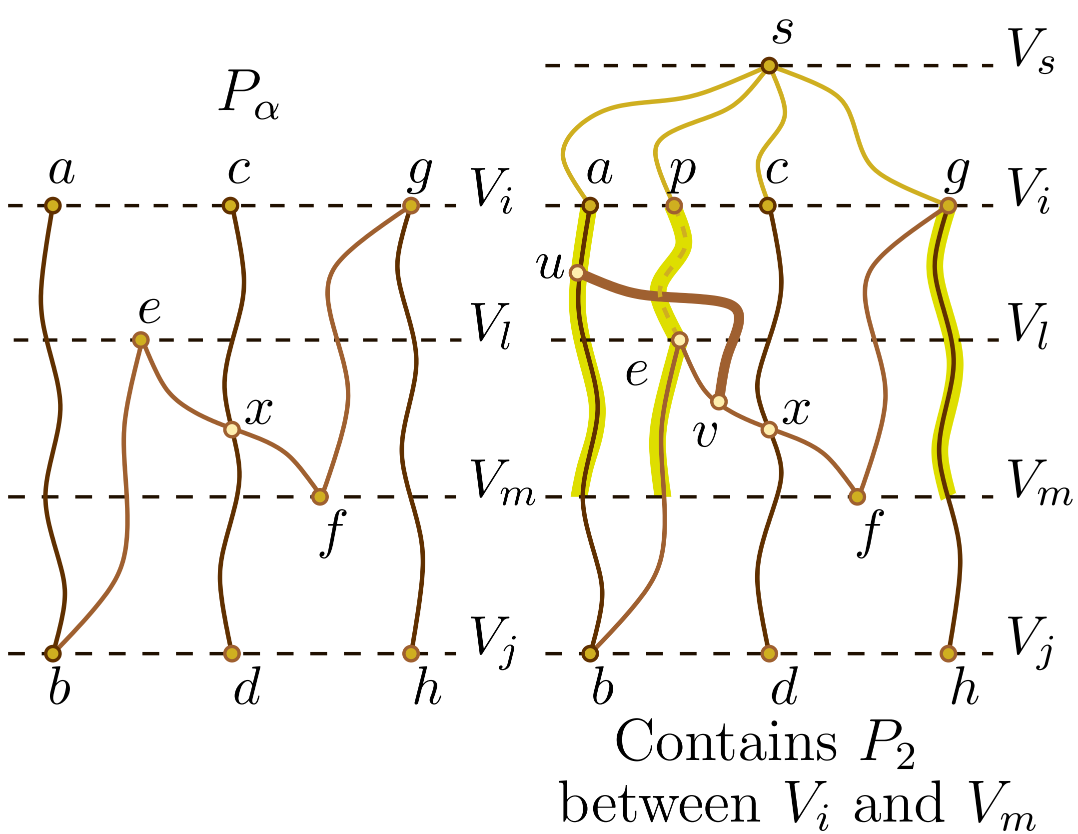
Sufficiency – Showing Completeness



Contains P_2
between V_i and V_m

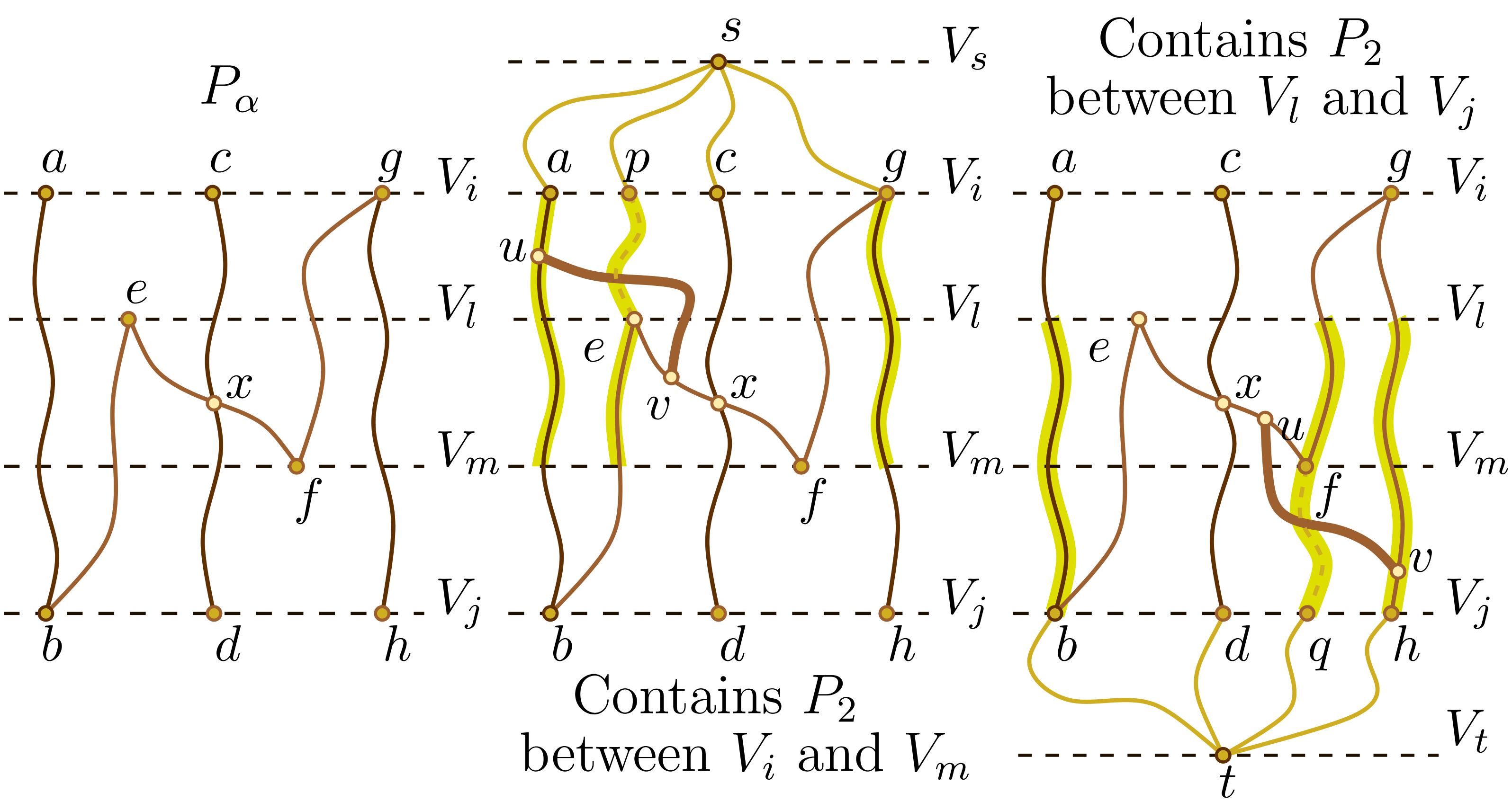
- Augment P_α to get P_3
  - Try augmenting to a hierarchy from above

Sufficiency – Showing Completeness



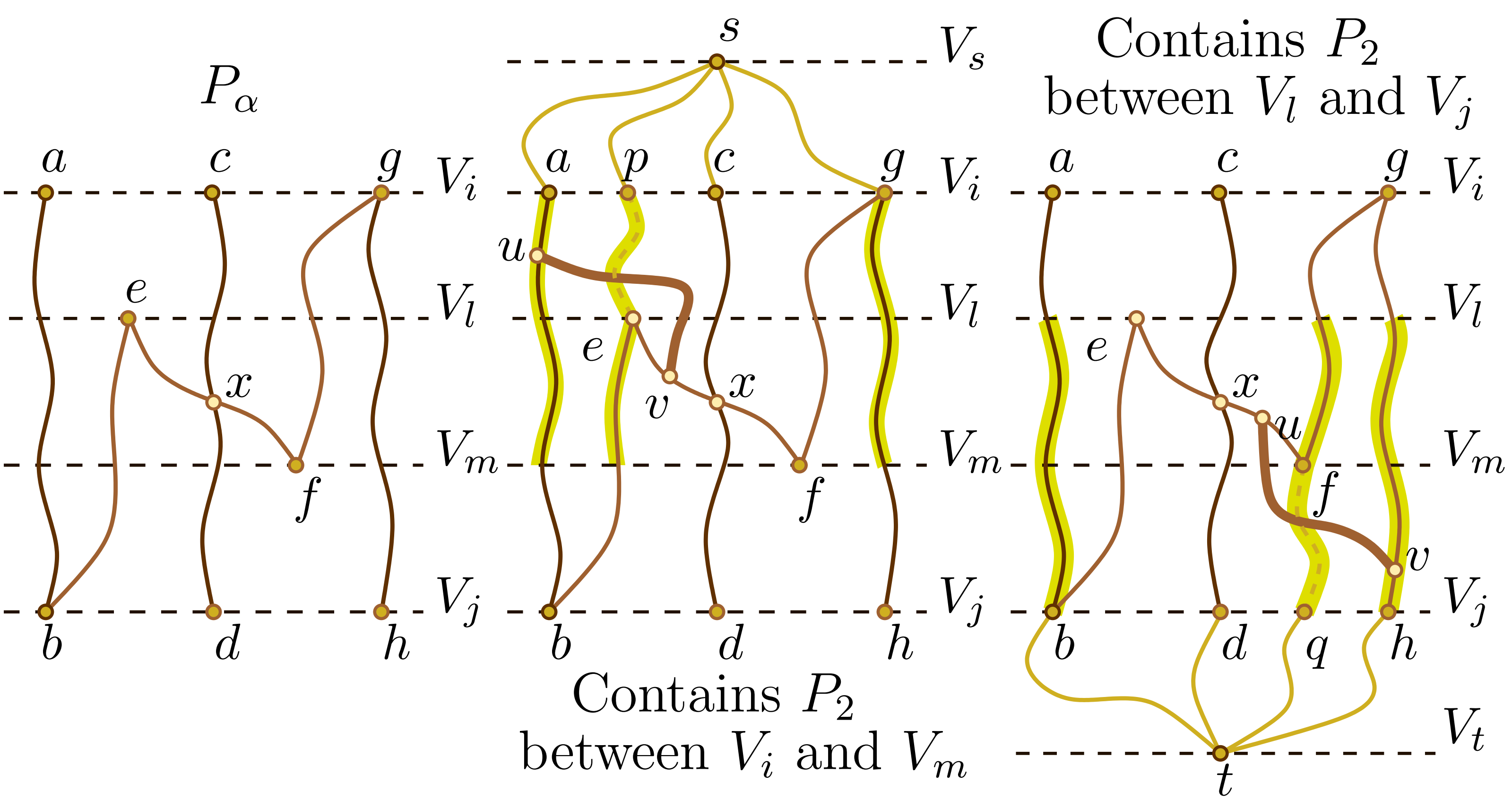
- Augment P_α to get P_3
  - Try augmenting to a hierarchy from above
    - Must have a cycle – cannot match a tree

Sufficiency – Showing Completeness



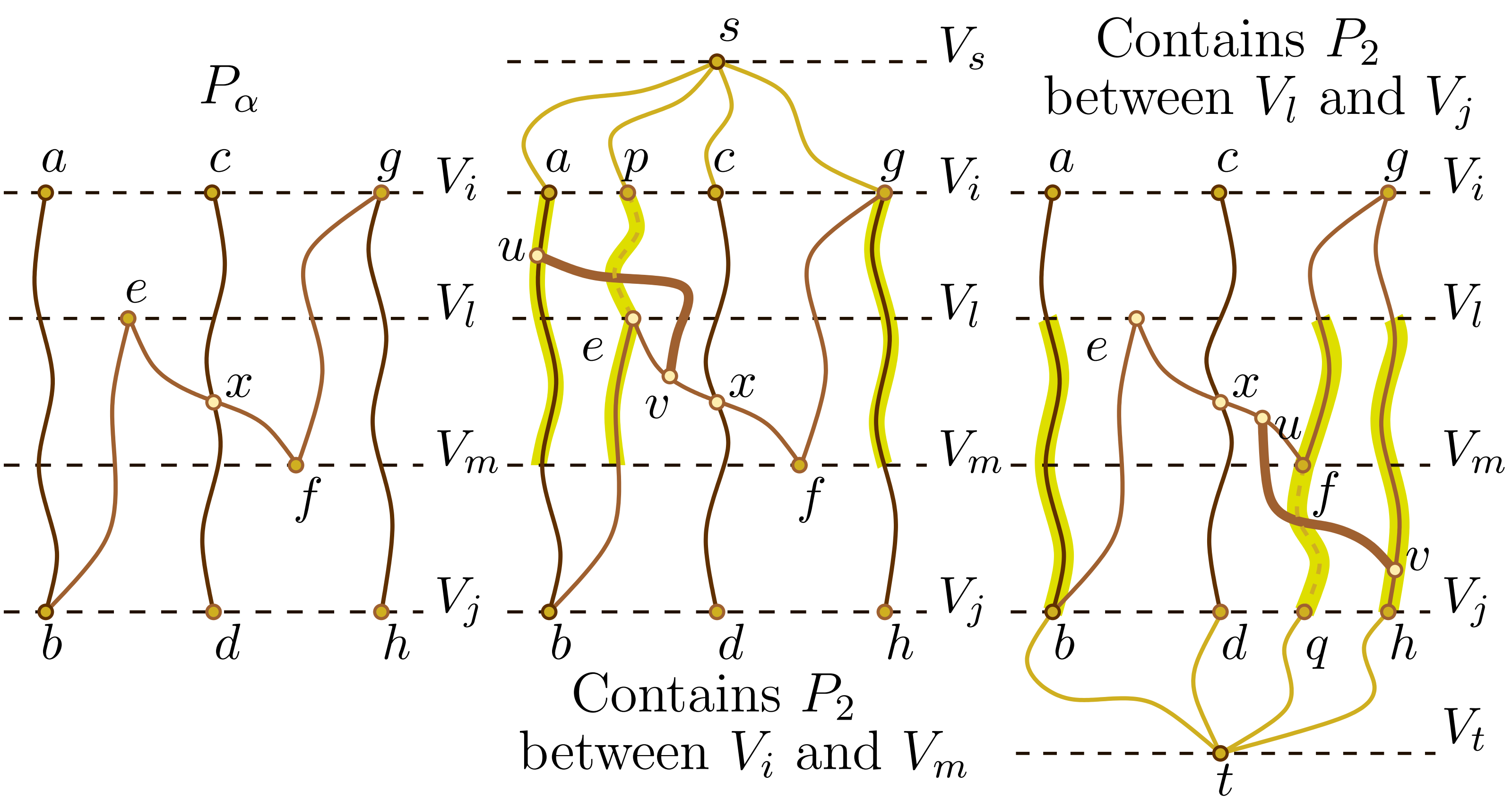
- Augment P_α to get P_3
  - Try augmenting to a hierarchy from below

Sufficiency – Showing Completeness



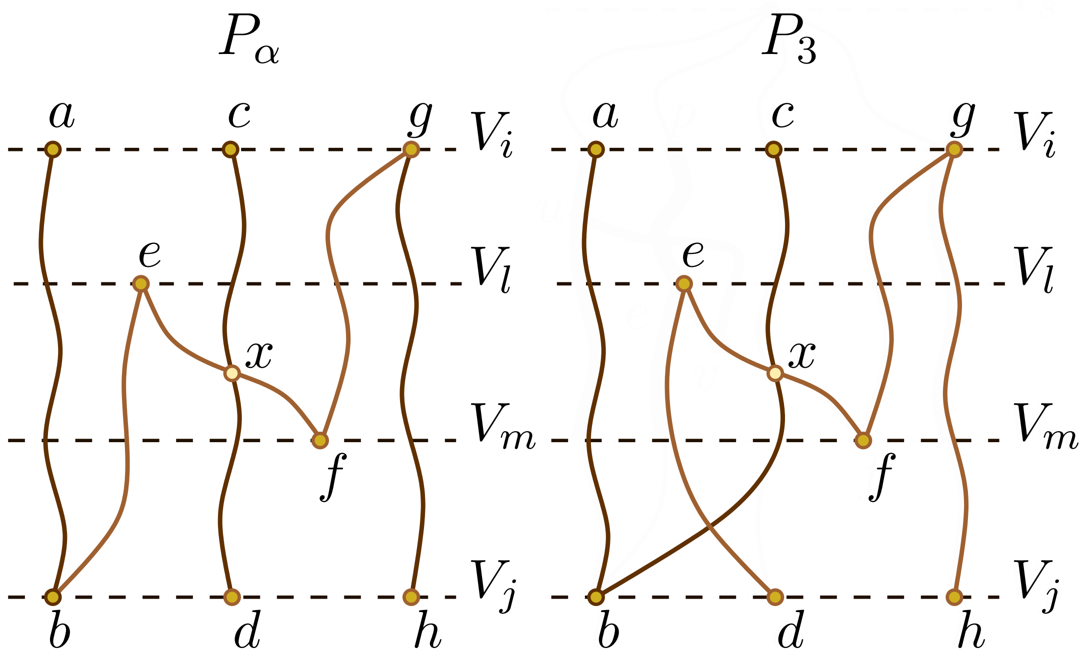
- Augment P_α to get P_3
  - Try augmenting to a hierarchy from below
    - Again has a cycle – again cannot match a tree

Sufficiency – Showing Completeness



- Augment P_α to get P_3
  - P_α must be whole pattern

Sufficiency – Showing Completeness



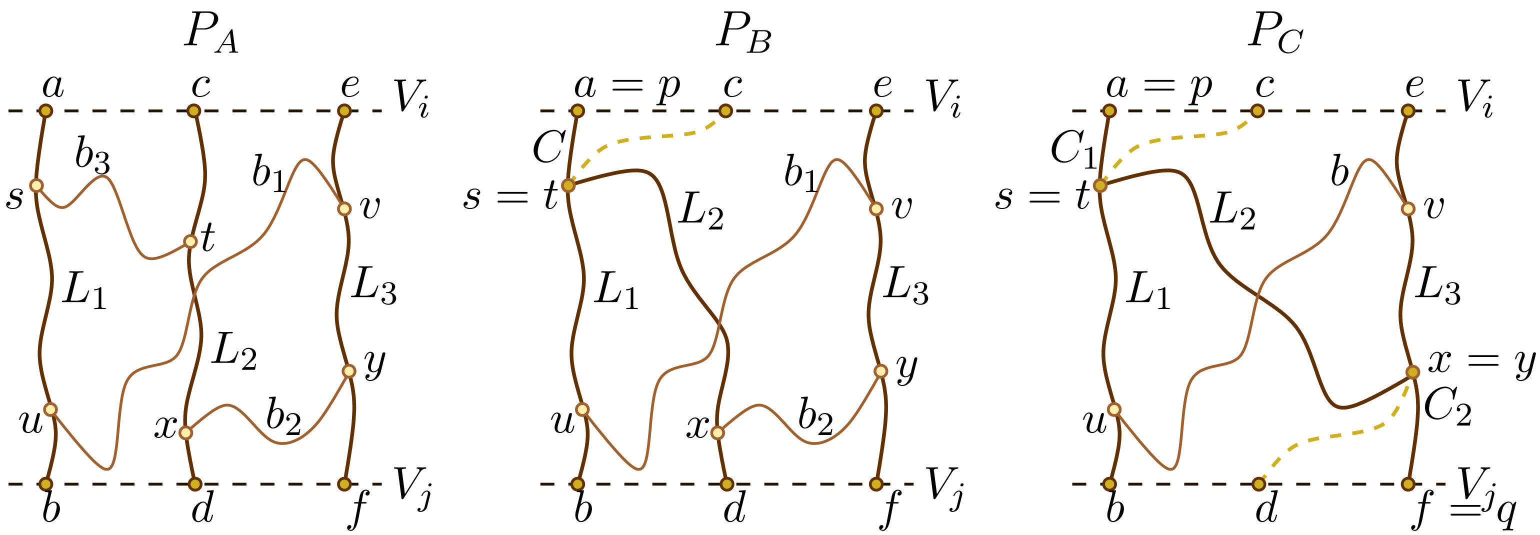
- Augment P_α to get P_3
  - P_α must be whole pattern
    - P_3 is only way for P_α to be MLNP

Summary

- Background
- Previous Patterns

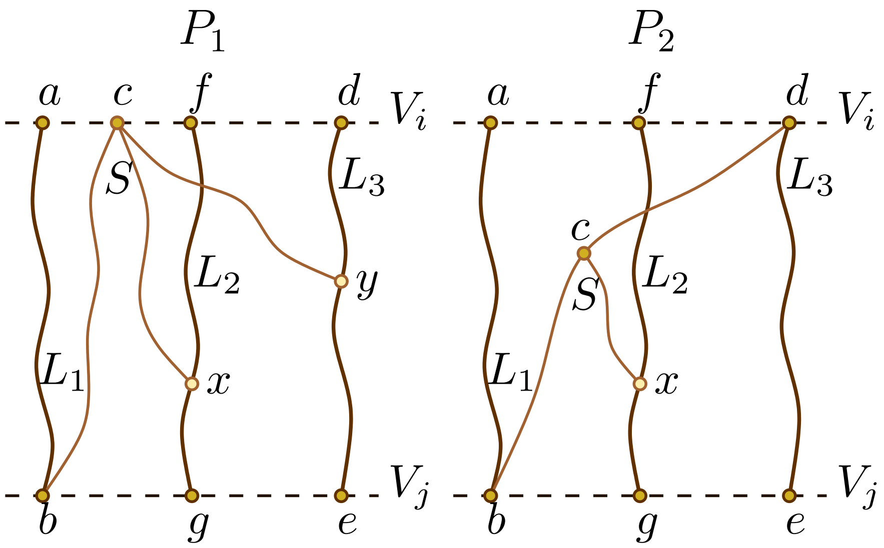
Summary

- Background
- Previous Patterns
  - Hierarchy Patterns



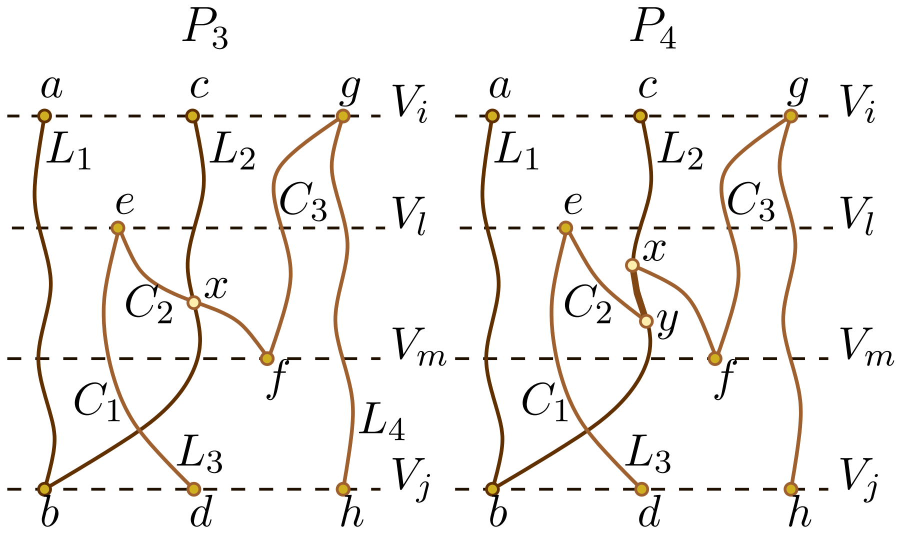
Summary

- Background
- Previous Patterns
  - Hierarchy Patterns
  - Minimal Minimum Level Non-Planar (MLNP) Patterns for Trees



Summary

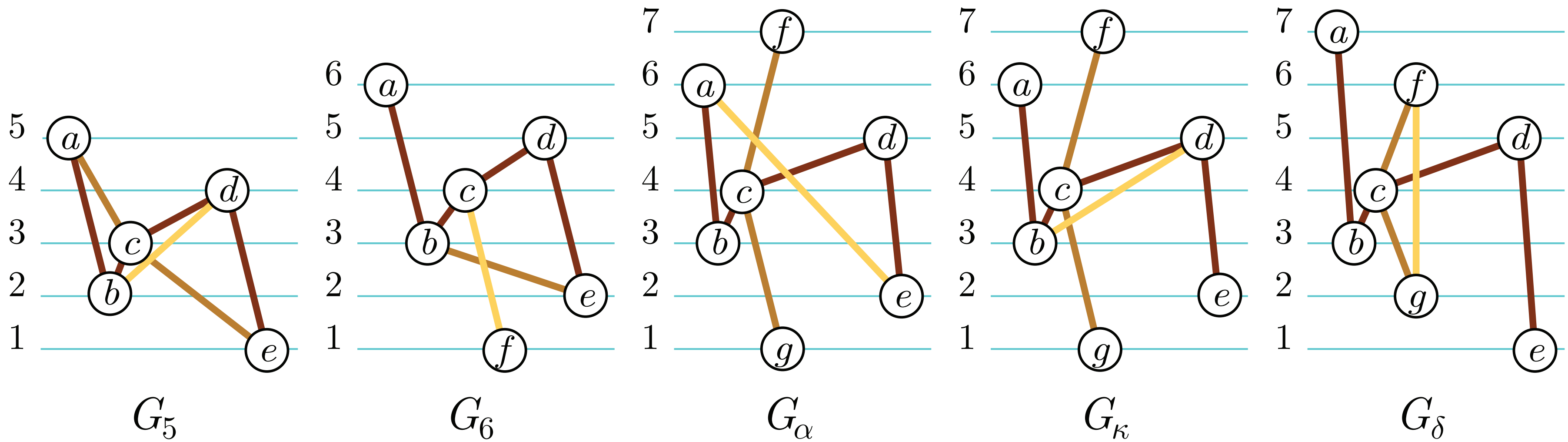
- Background
- Previous Patterns
- New Patterns



Future Work

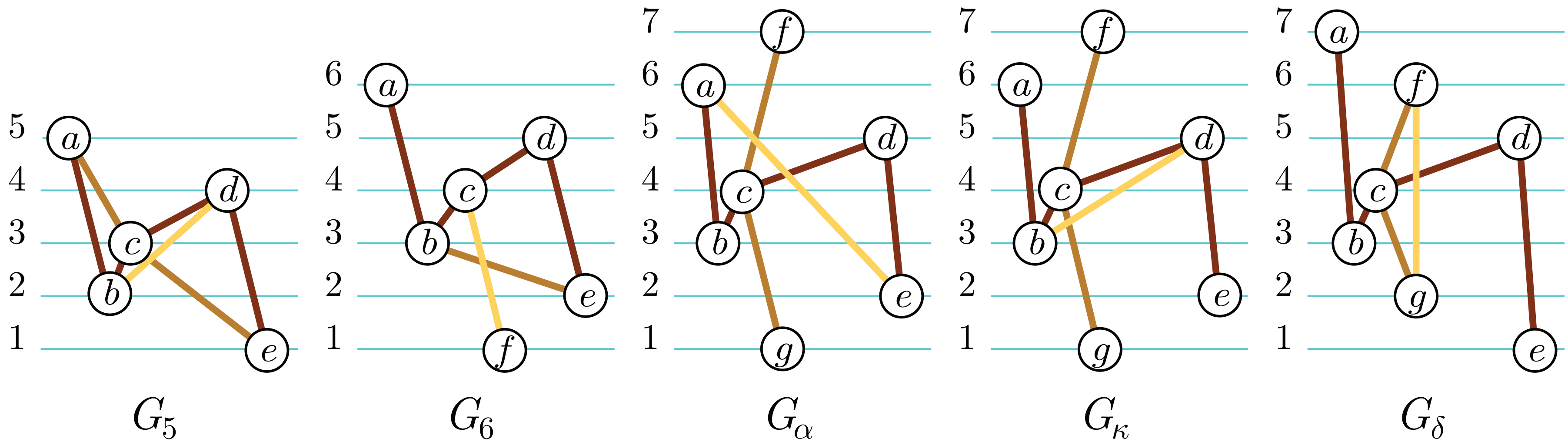
- Find all patterns for level planar graphs with cycles

Future Work



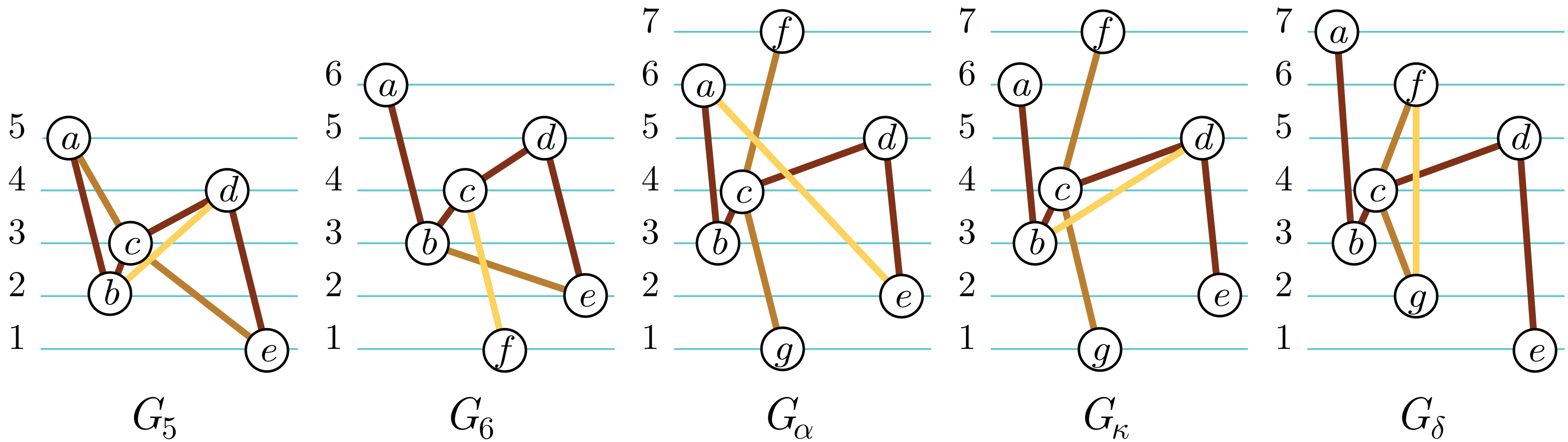
- Find all patterns for level planar graphs with cycles
  - Four of the five forbidden ULP graphs with cycles will yield new patterns

Future Work



- Find all patterns for level planar graphs with cycles
  - Four of the five forbidden ULP graphs with cycles will yield new patterns
    - G_5, G_α, G_κ, and G_δ have degree-4 vertices like T_9

Future Work



- Find all patterns for level planar graphs with cycles
  - Four of the five forbidden ULP graphs with cycles will yield new patterns
    - G_5, G_α, G_κ, and G_δ have degree-4 vertices like T_9
    - None of the three HLNP patterns match any of these four

Thank You!