Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG
Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems
Institute for Informatics

Simplification of Polyline Bundles

Joachim Spoerhase
Aalto University, Finland

Sabine Storandt
Universität Konstanz, Germany

Johannes Zink
Universität Würzburg, Germany

Motivation

Motivation

- Maps often consist of polylines

Motivation

- Maps often consist of polylines

Motivation

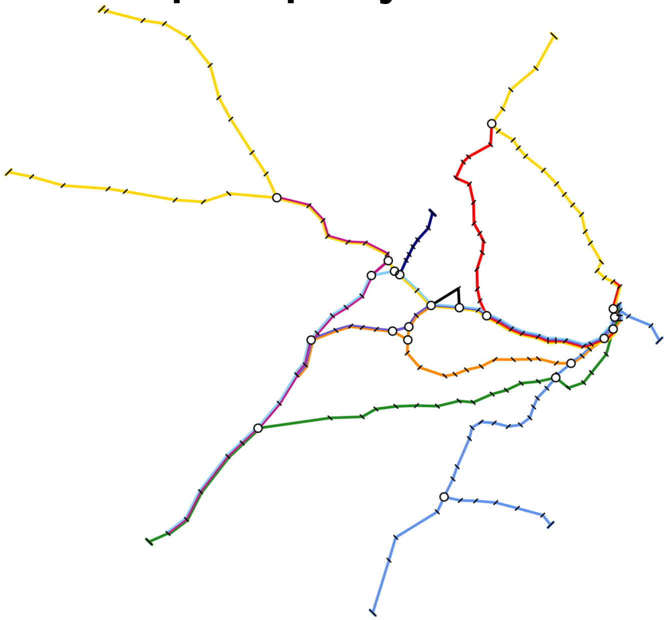
- Maps often consist of polylines

- Multiple polylines share edges and vertices sectionwise

Motivation

- Maps often consist of polylines

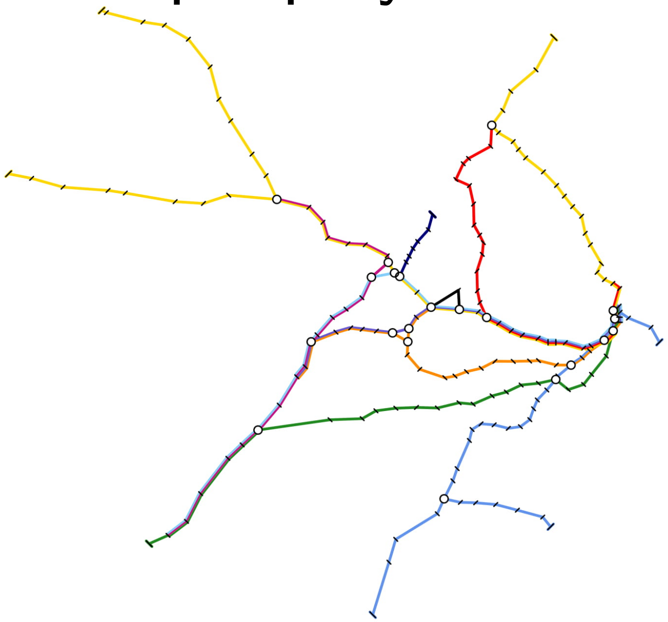
- Multiple polylines share edges and vertices sectionwise



Motivation

- Maps often consist of polylines

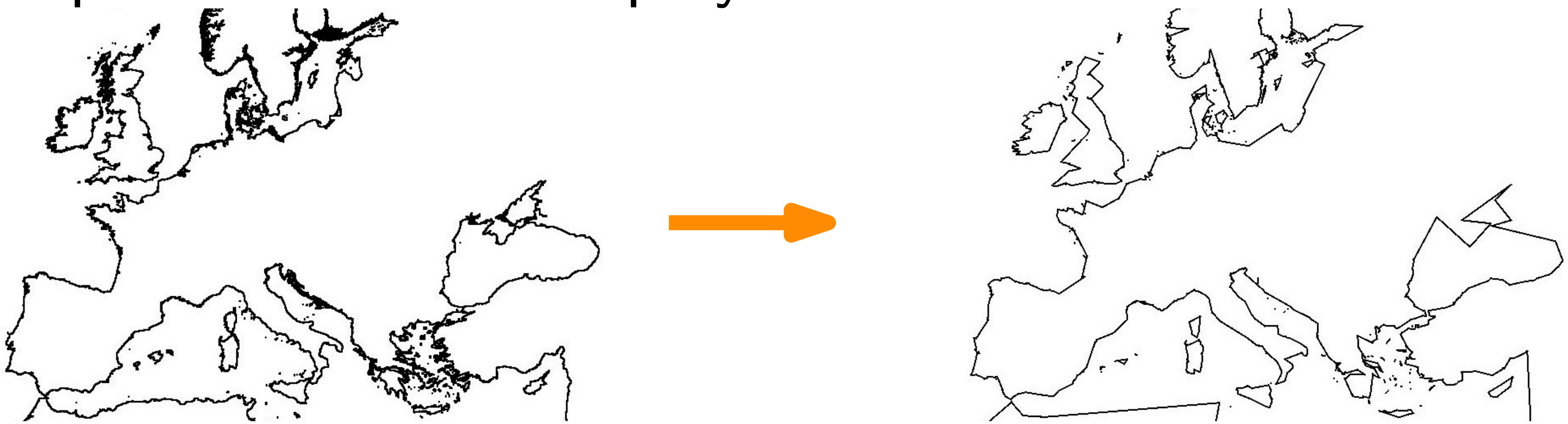
- Multiple polylines share edges and vertices sectionwise



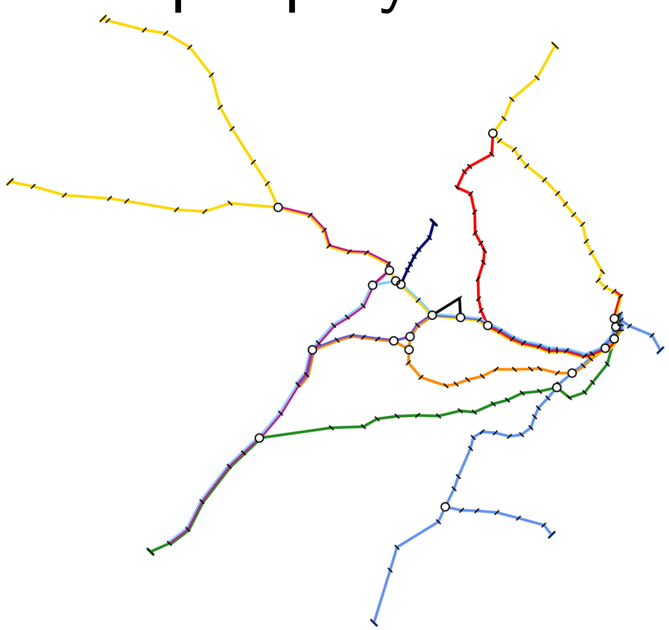
- Reduce full data for zooming or schematization

Motivation

- Maps often consist of polylines



- Multiple polylines share edges and vertices sectionwise

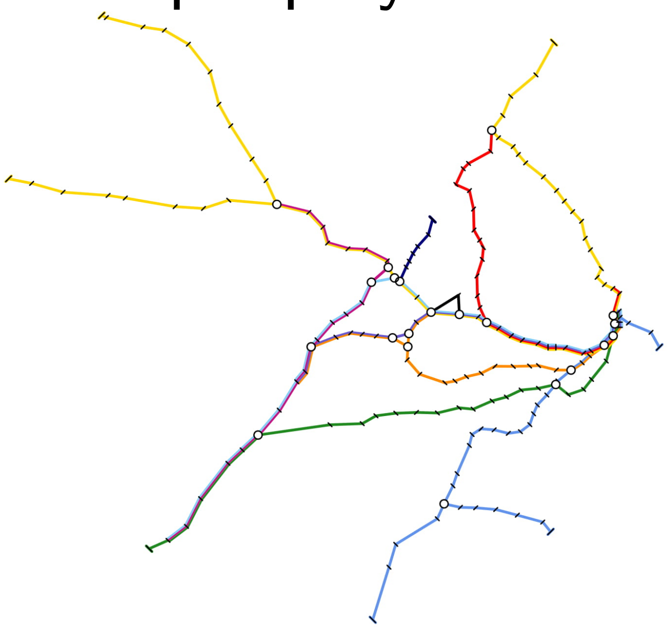


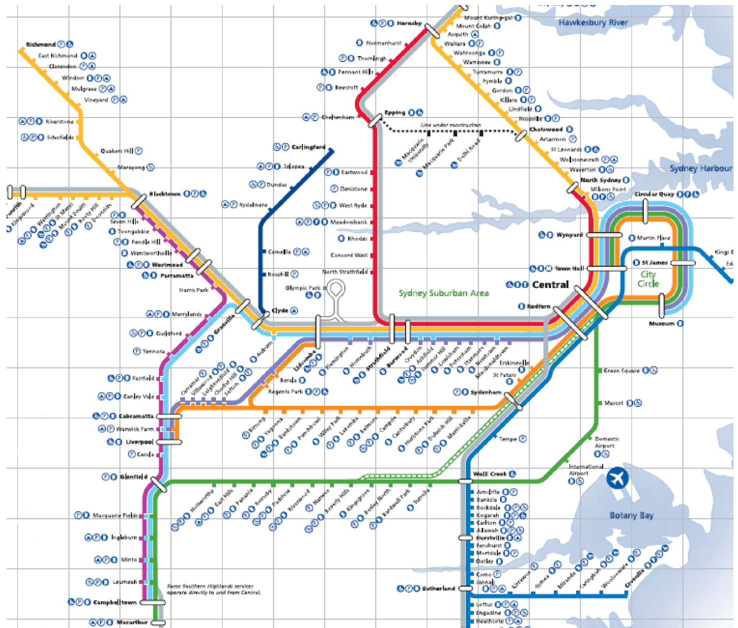
- Reduce full data for zooming or schematization

Motivation

- Maps often consist of polylines

- Multiple polylines share edges and vertices sectionwise





- Reduce full data for zooming or schematization

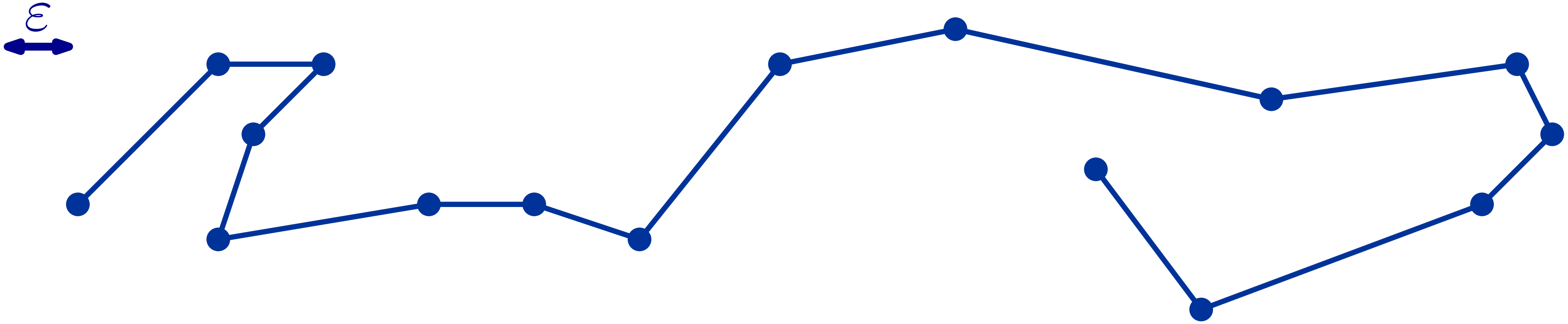
Introduction: Simplifying a Polyline

Introduction: Simplifying a Polyline

Given:
- polyline L as a sequence of points in the plane
- distance threshold ε

Introduction: Simplifying a Polyline

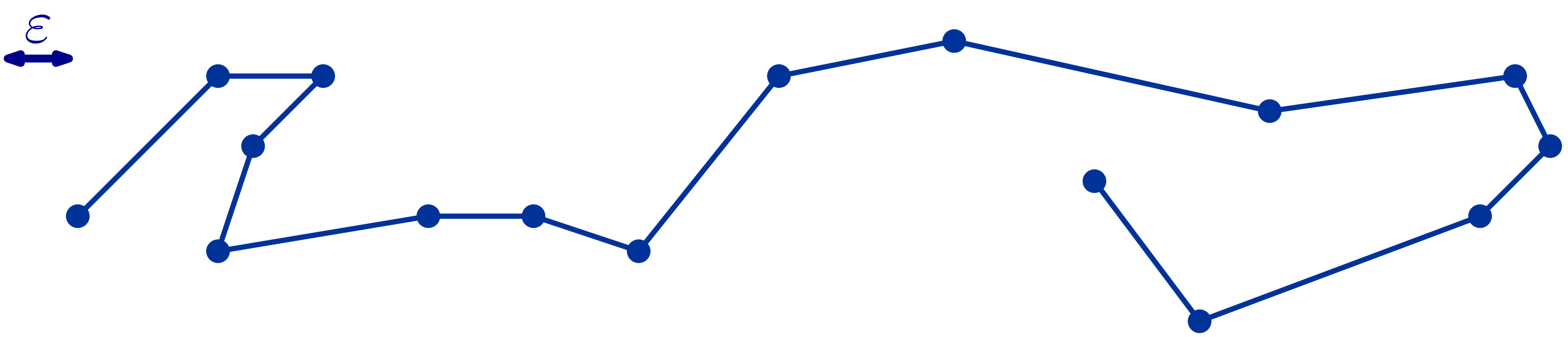
Given:
- polyline L as a sequence of points in the plane
- distance threshold ε



Introduction: Simplifying a Polyline

Given:
- polyline L as a sequence of points in the plane
- distance threshold ε

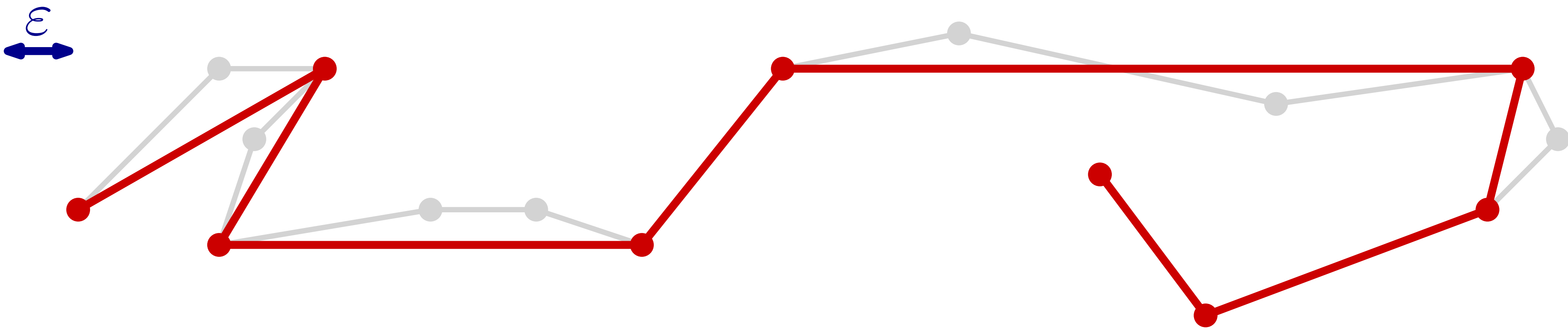
Goal: Find a minimum size subsequence L' of L, such that the segment-wise undirected Hausdorff distance between L' and L does not exceed ε.



Introduction: Simplifying a Polyline

Given:
- polyline L as a sequence of points in the plane
- distance threshold ε

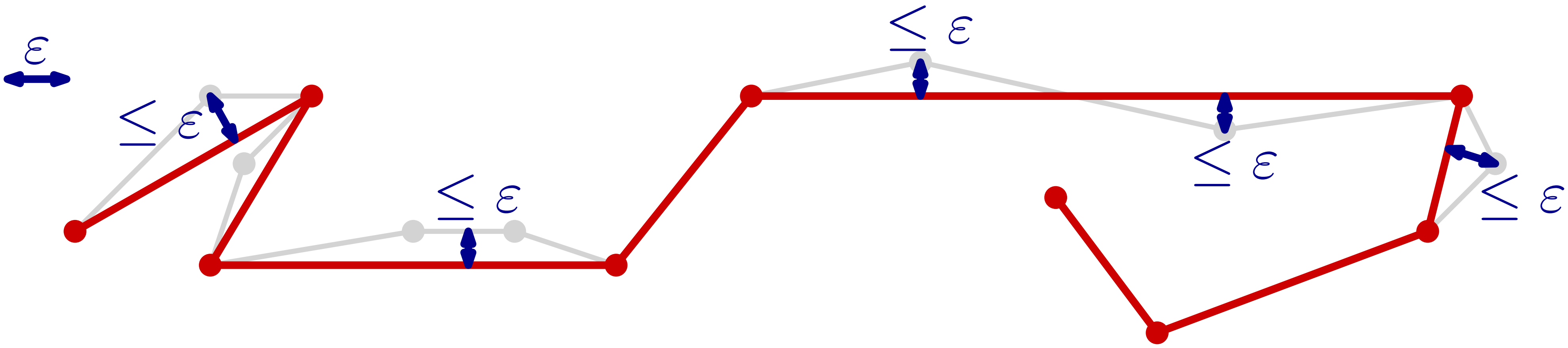
Goal: Find a minimum size subsequence L' of L, such that the segment-wise undirected Hausdorff distance between L' and L does not exceed ε.



Introduction: Simplifying a Polyline

Given:
- polyline L as a sequence of points in the plane
- distance threshold ε

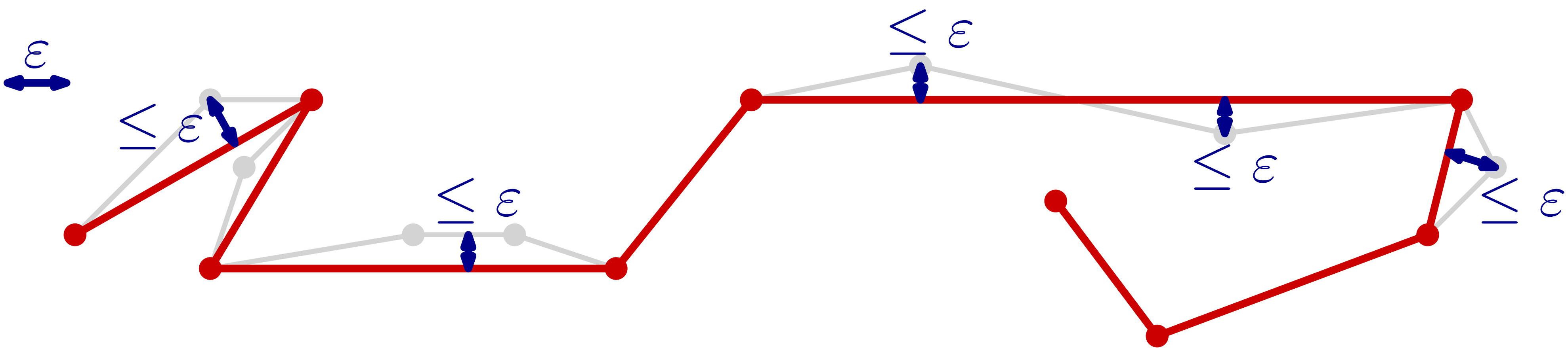
Goal: Find a minimum size subsequence L' of L, such that the segment-wise undirected Hausdorff distance between L' and L does not exceed ε.



Introduction: Simplifying a Polyline

Given:
- polyline L as a sequence of points in the plane
- distance threshold ε

Goal: Find a minimum size subsequence L' of L, such that the segment-wise undirected Hausdorff distance between L' and L does not exceed ε.

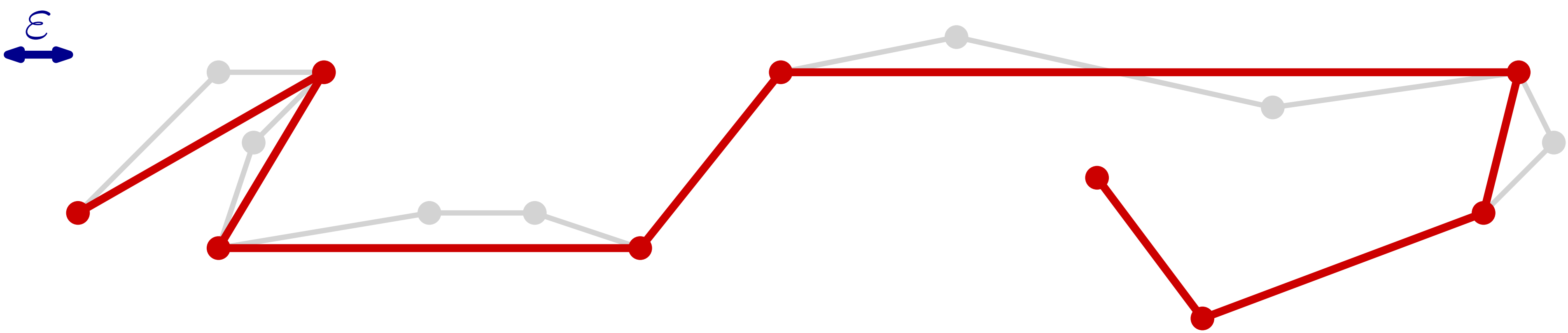


- Can be solved efficiently in $O(|L|^2)$ time.
  [Imai, Iri '88], [Chan, Chin '96]

Introduction: Simplifying a Polyline

Given:
- polyline L as a sequence of points in the plane
- distance threshold ε

Goal: Find a minimum size subsequence L' of L, such that the segment-wise undirected Hausdorff distance between L' and L does not exceed ε.



- Can be solved efficiently in $O(|L|^2)$ time.
  [Imai, Iri '88], [Chan, Chin '96]

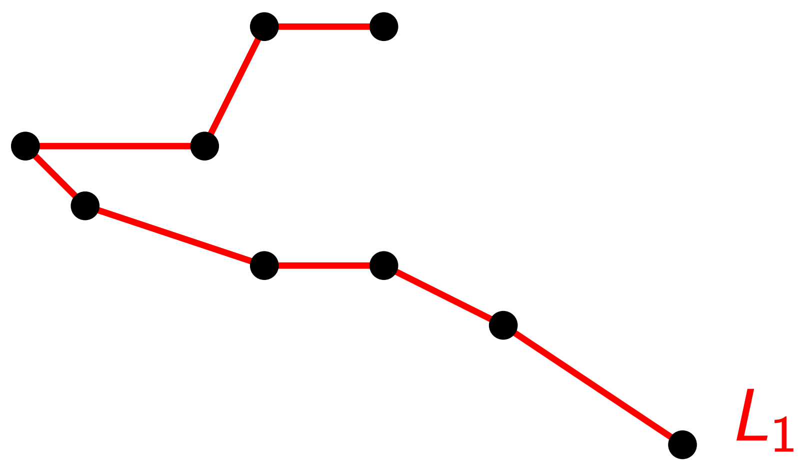
Our Generalization

Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

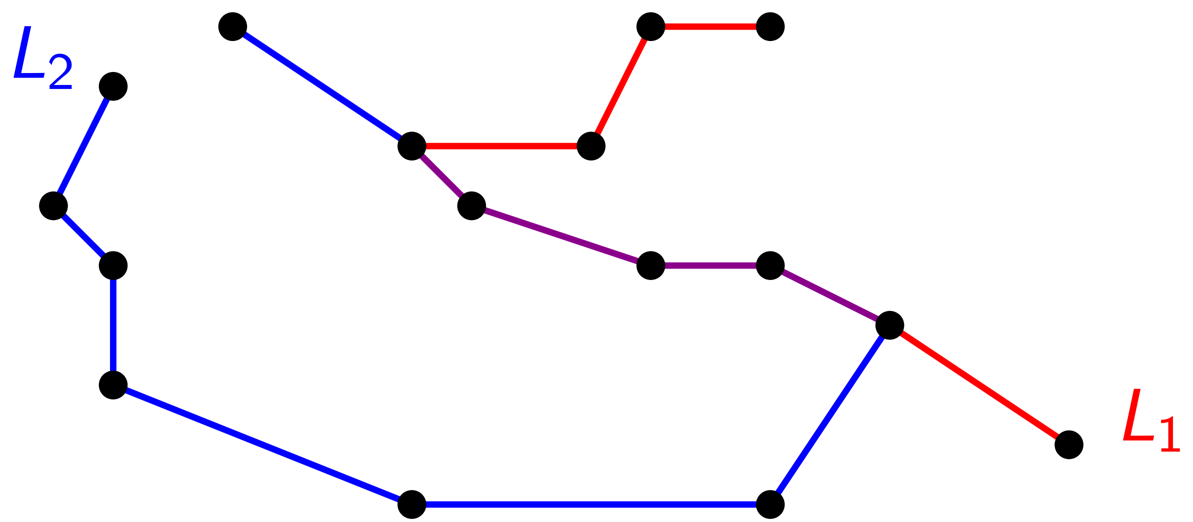
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges



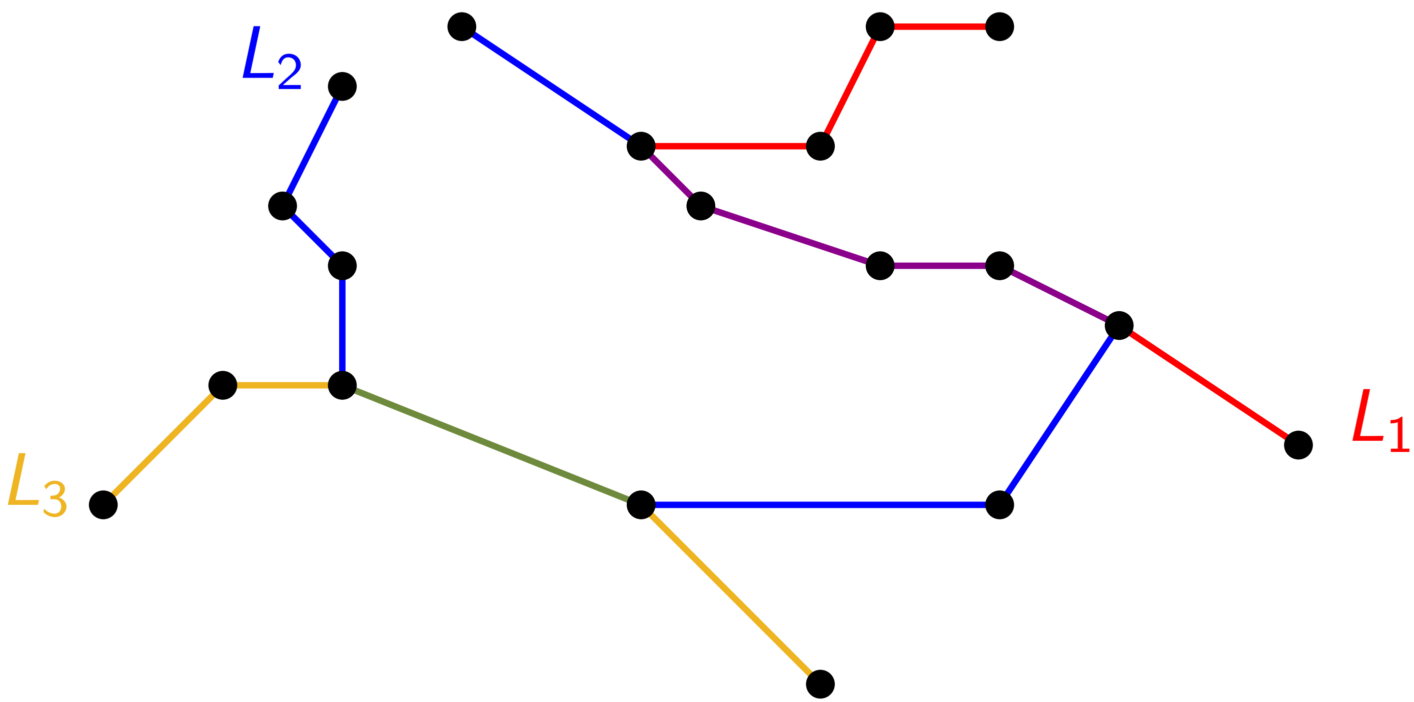
Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges



Our Generalization

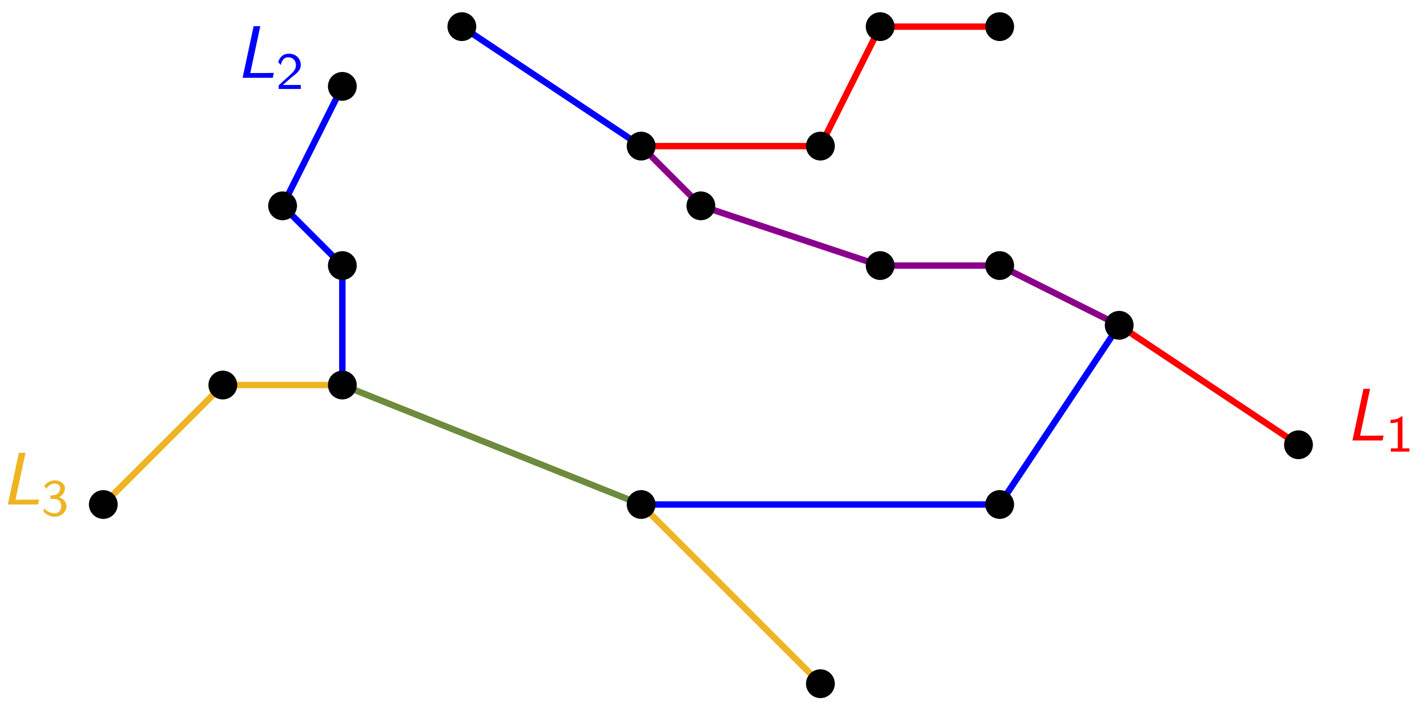
- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges



Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

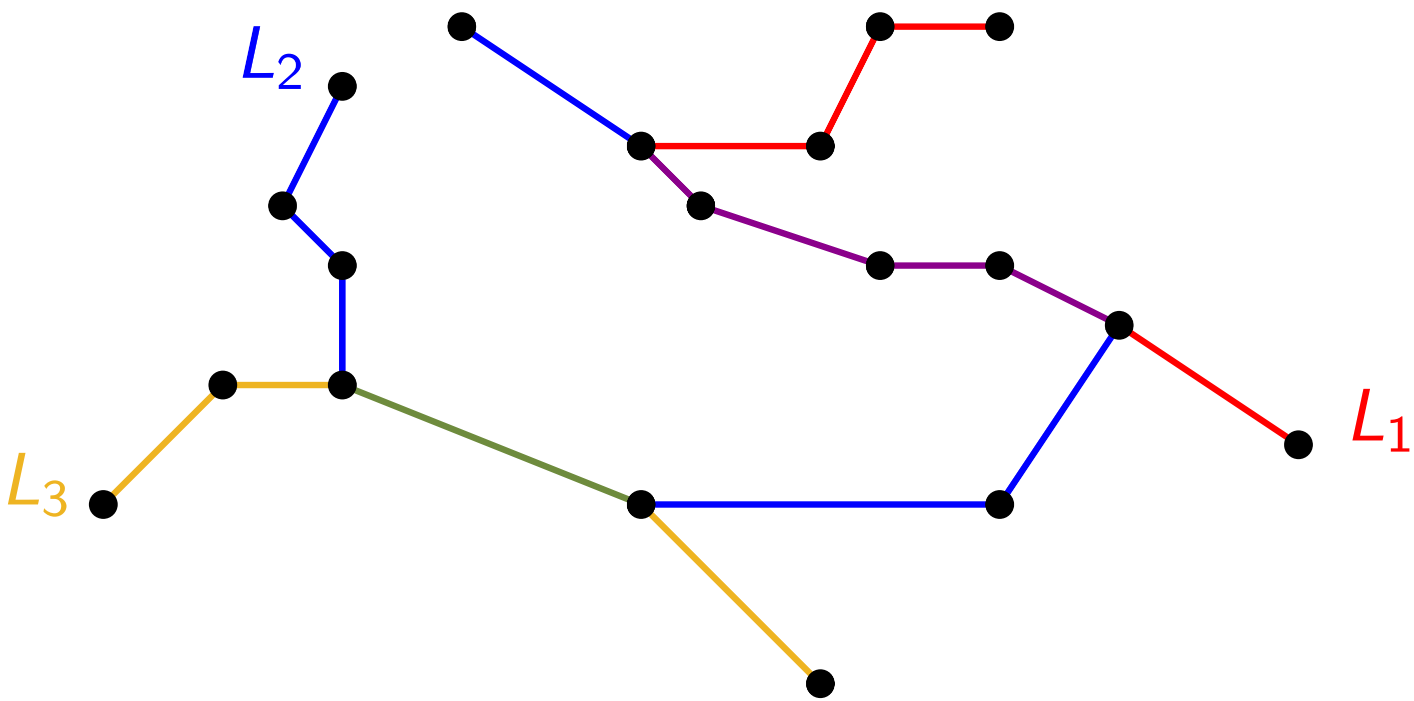
We call the union of their vertices V



Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal $\textsc{Min-Vertices}$: find a $V^* \subseteq V$

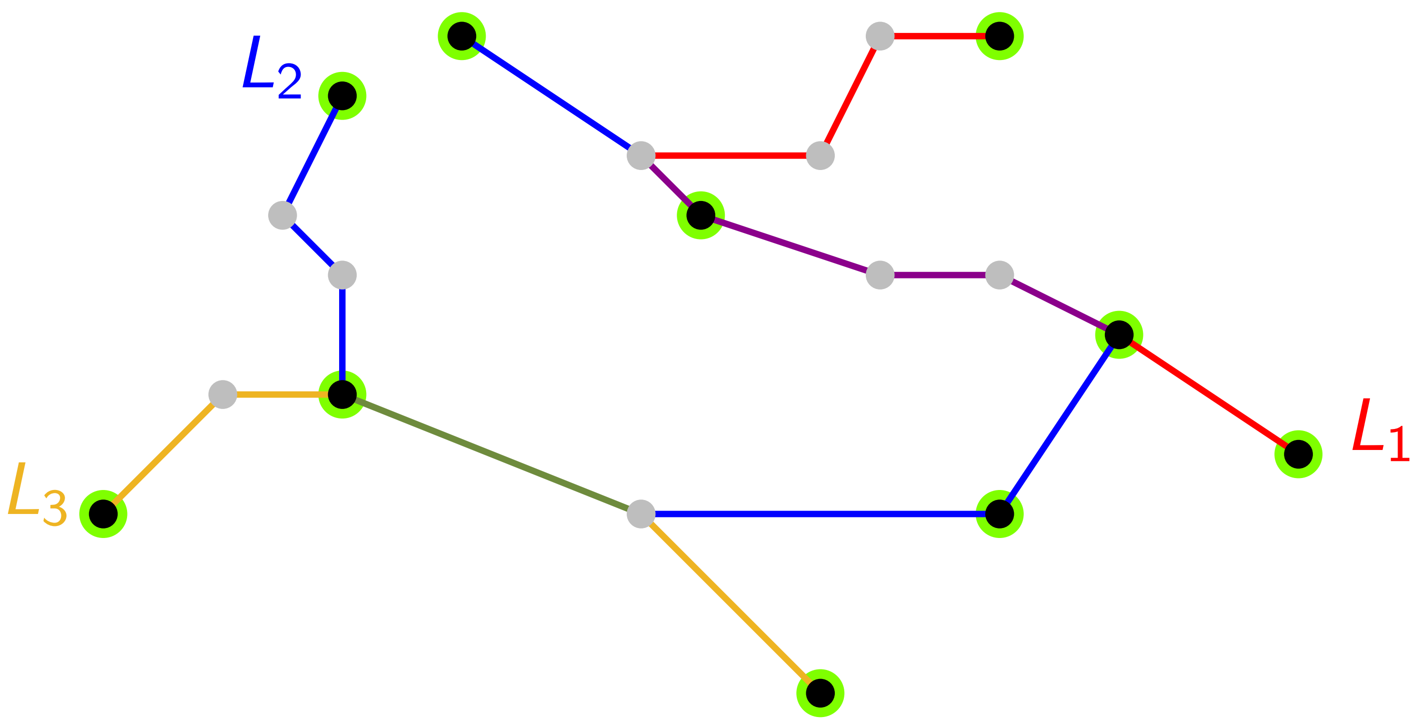
We call the union of their vertices V



Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal Min-Vertices: find a $V^* \subseteq V$

We call the union of their vertices V

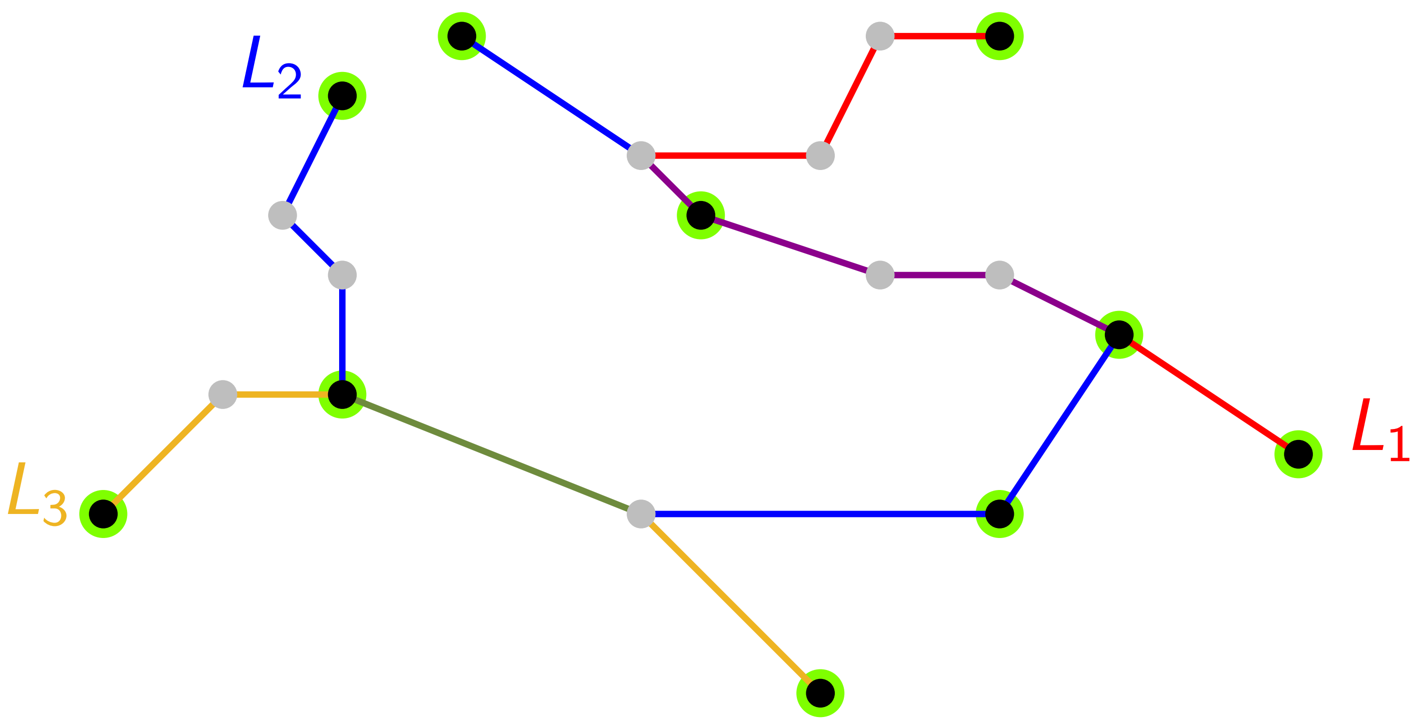


Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices V

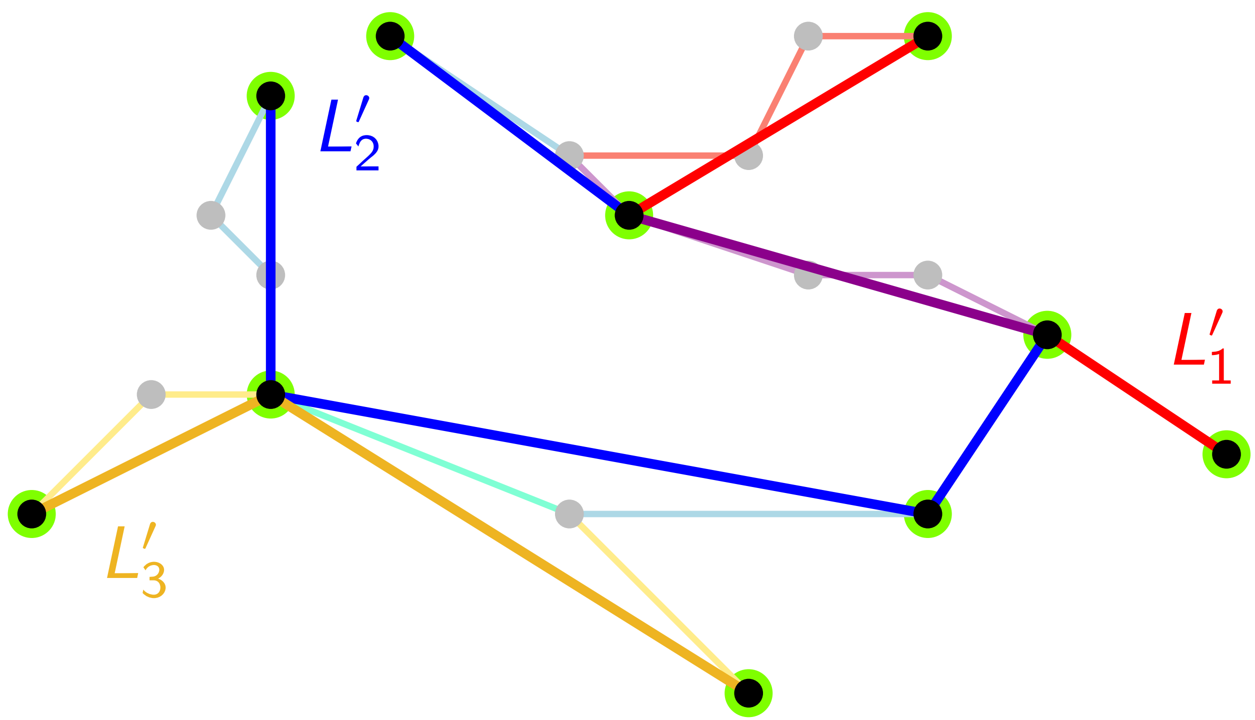
- Goal Min-Vertices: find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$



Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges
- Goal Min-Vertices:
  find a V^* $\subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$

We call the union of their vertices V

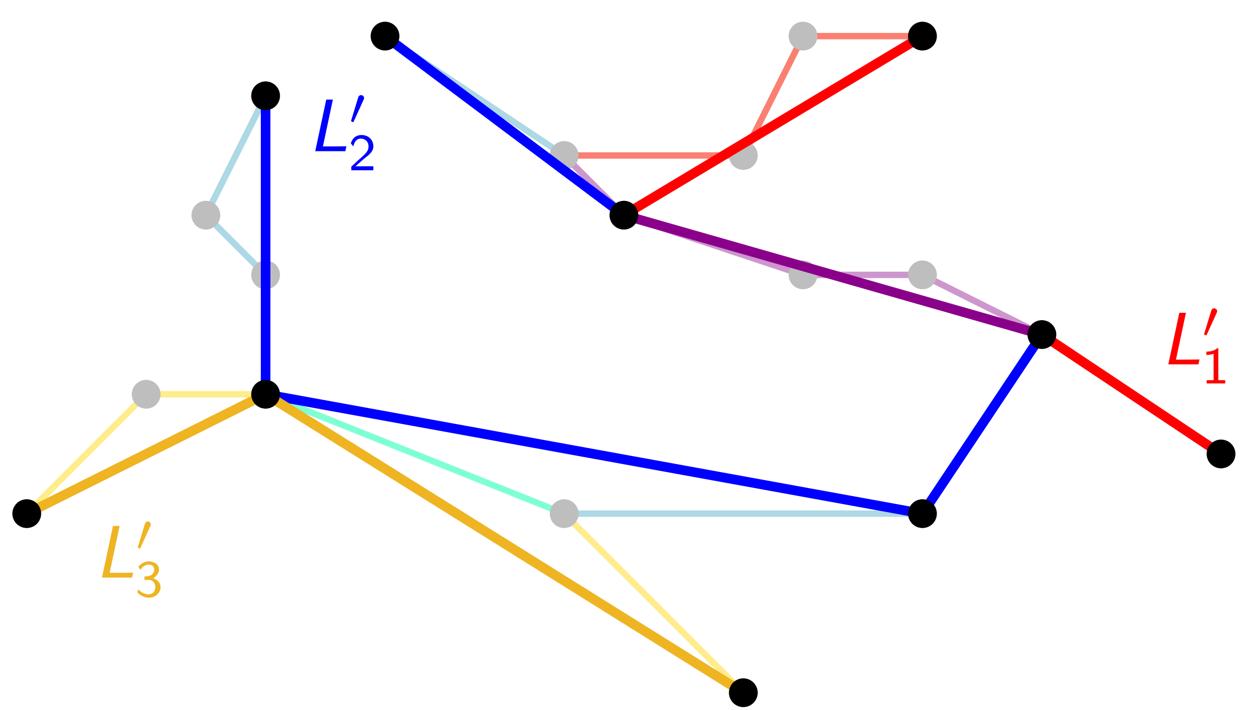


Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices V

- Goal Min-Vertices:
find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no L'_i and L_i exceeding the maximum distance

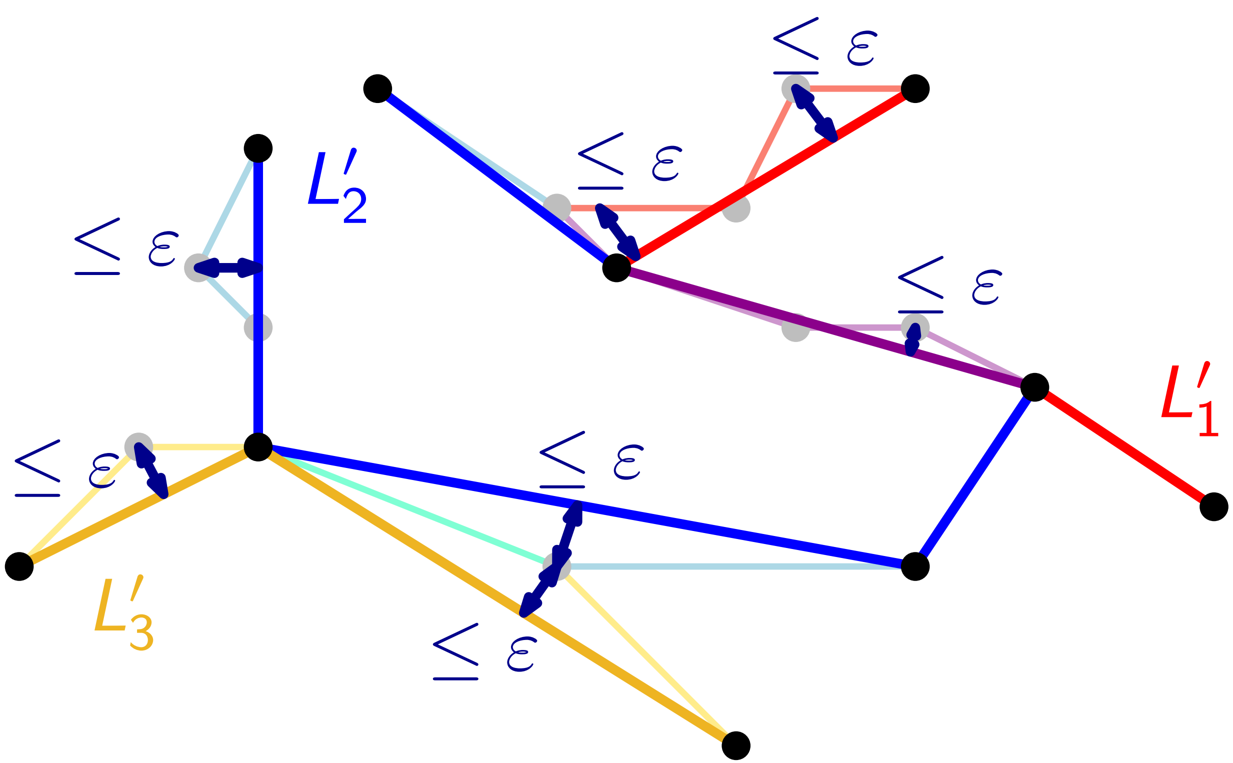


Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices V

- Goal MIN-VERTICES:
  find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no L'_i and L_i exceeding the maximum distance

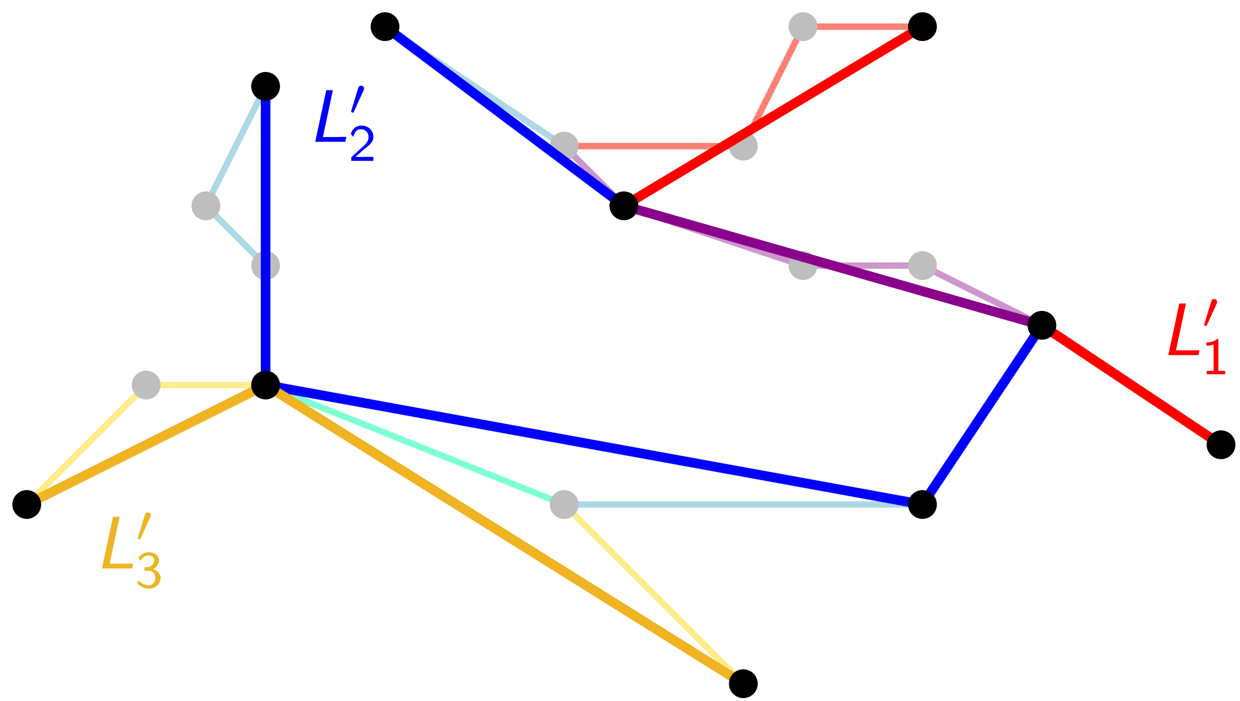


Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices V

- Goal Min-Vertices:
  find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no L'_i and L_i exceeding the maximum distance and the number of preserved vertices $|V^*|$ is minimum.

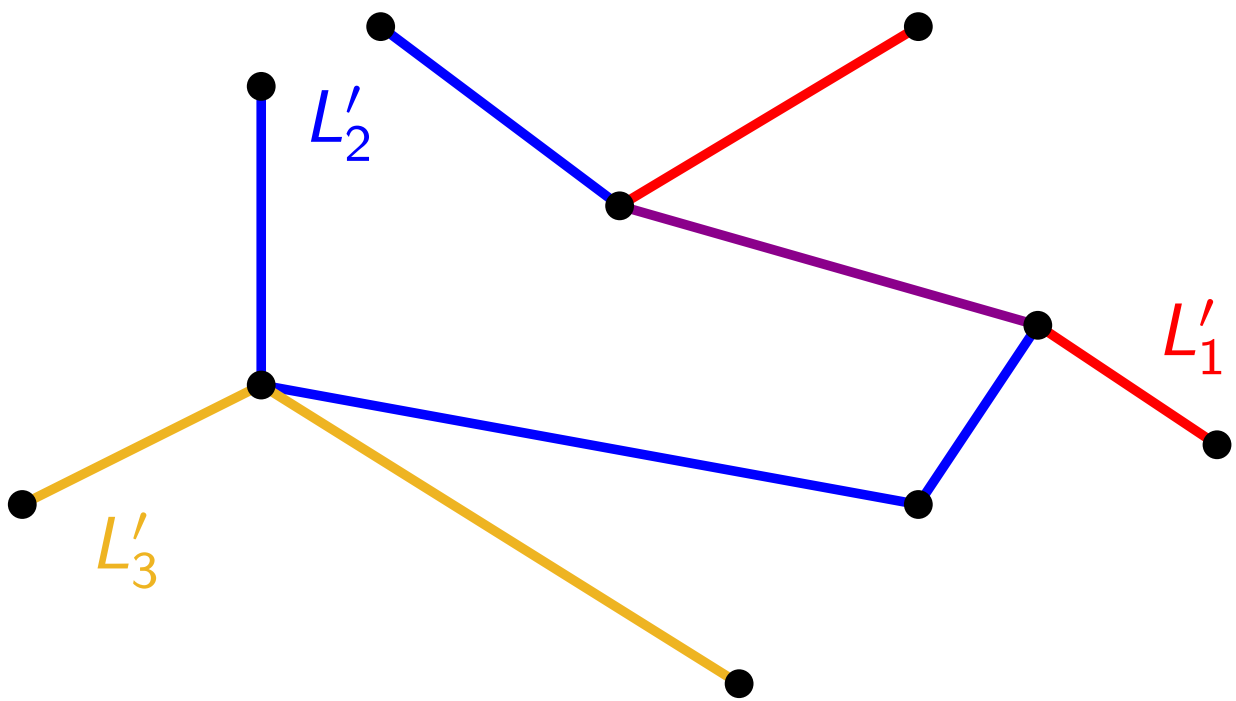


Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices V

- Goal Min-Vertices: find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no L'_i and L_i exceeding the maximum distance and the number of preserved vertices $|V^*|$ is minimum.

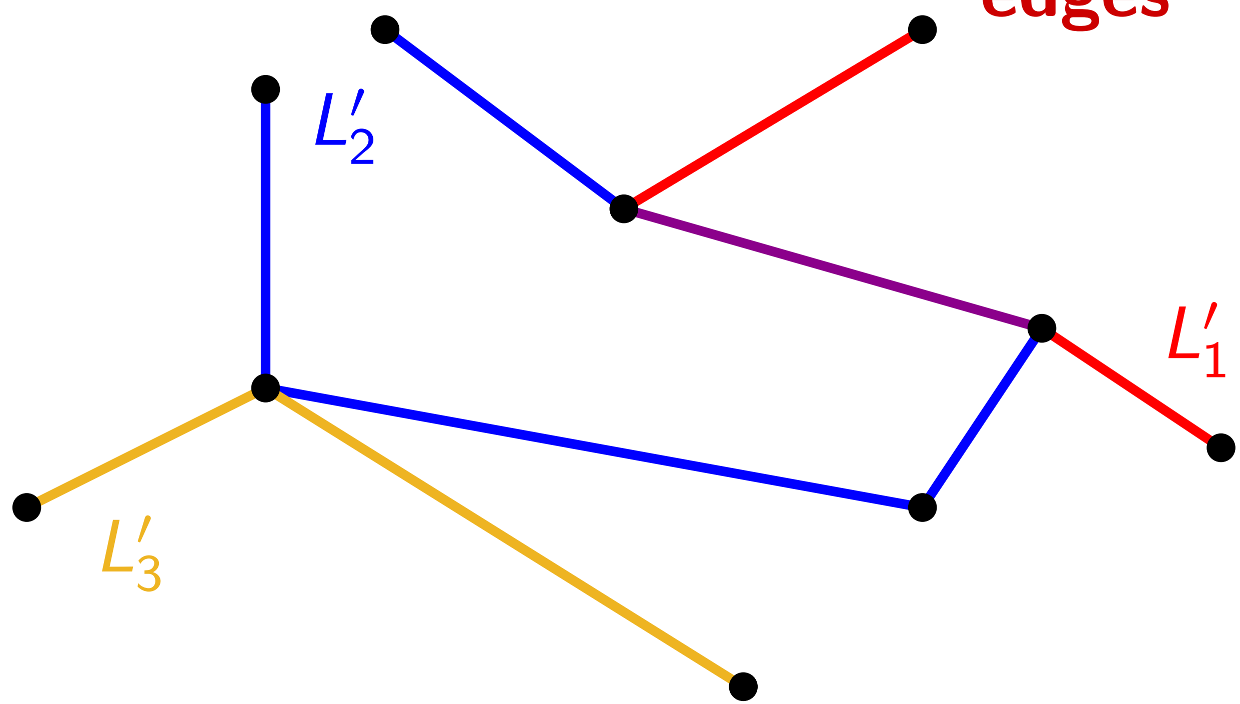


Our Generalization

- Given: a set $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ of polylines possibly sharing vertices and edges

We call the union of their vertices V

- Goal $\textsc{Min-}\cancel{\textsc{Vertices}}$ $\textsc{Edges}$:
  find a $V^* \subseteq V$ inducing polylines $\{L'_1, \ldots, L'_\ell\}$ on $\mathcal{L}$, such that there is no L'_i and L_i exceeding the maximum distance and the number of preserved ~~vertices~~ $\cancel{|V^*|}$ **edges** is minimum.



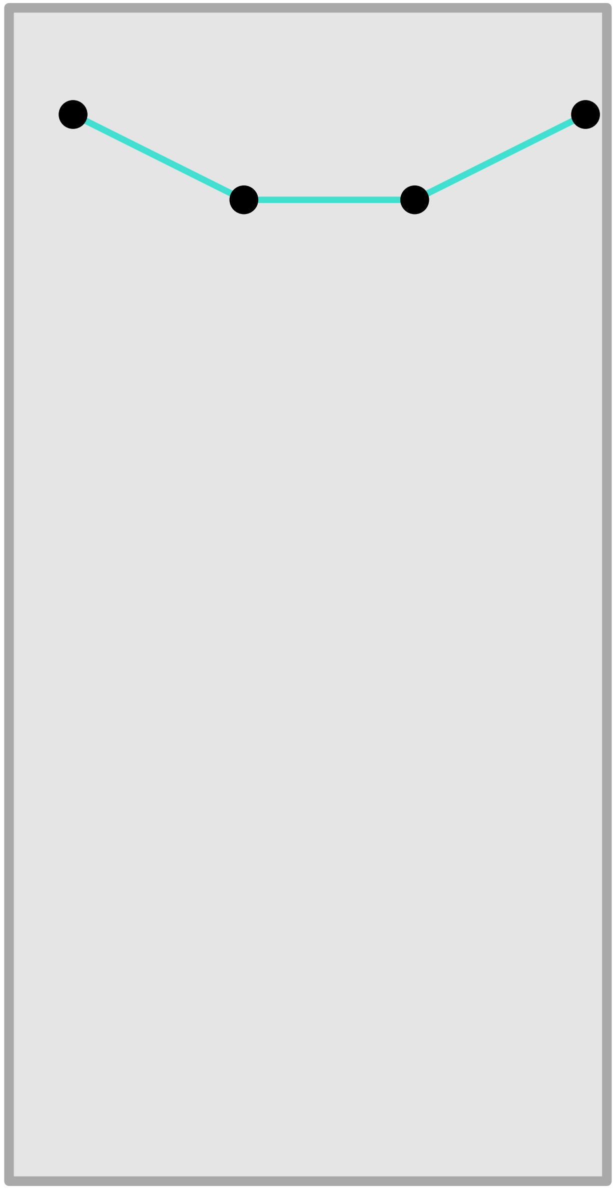
Aren't Both Goals the Same?

Aren't Both Goals the Same?

No! Here is a counterexample:

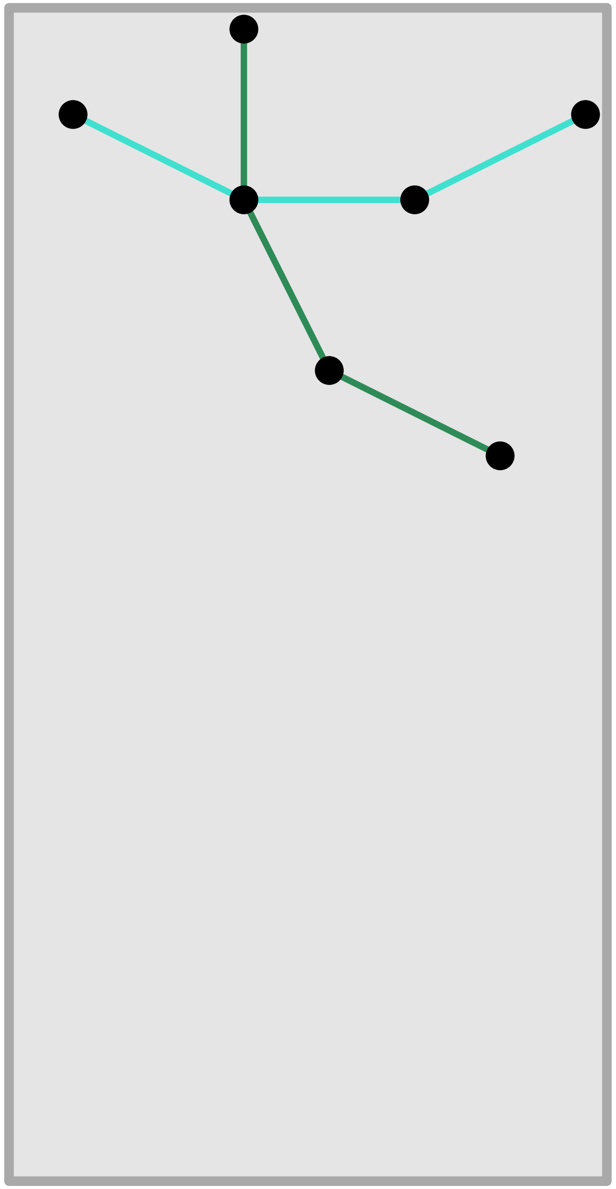
Aren't Both Goals the Same?

No! Here is a counterexample:



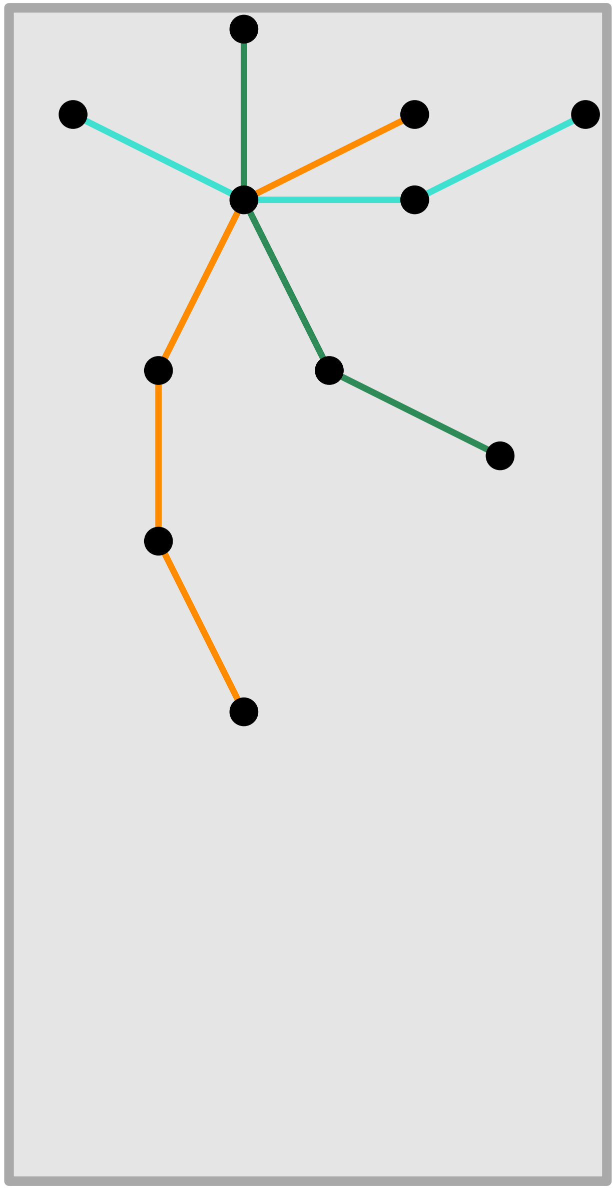
Aren't Both Goals the Same?

No! Here is a counterexample:



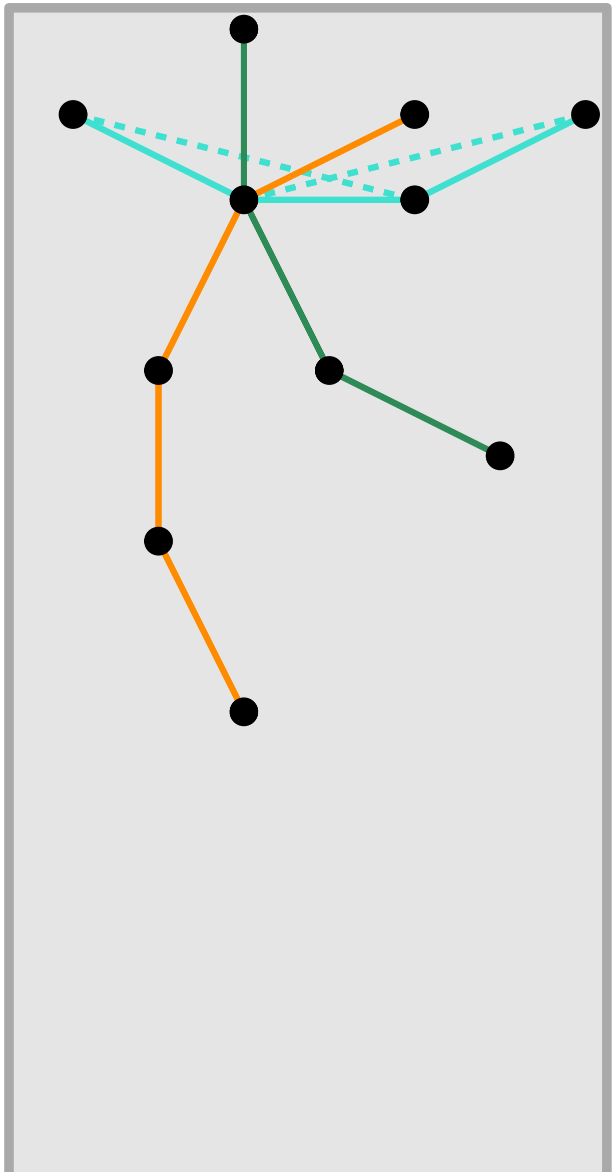
Aren't Both Goals the Same?

No! Here is a counterexample:



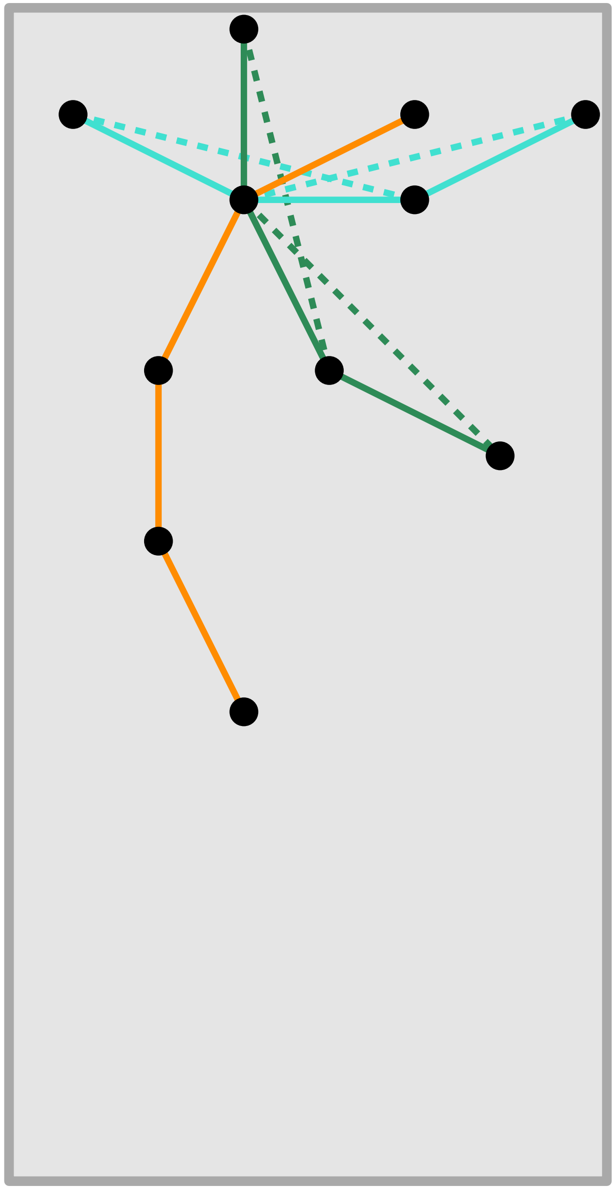
Aren't Both Goals the Same?

No! Here is a counterexample:



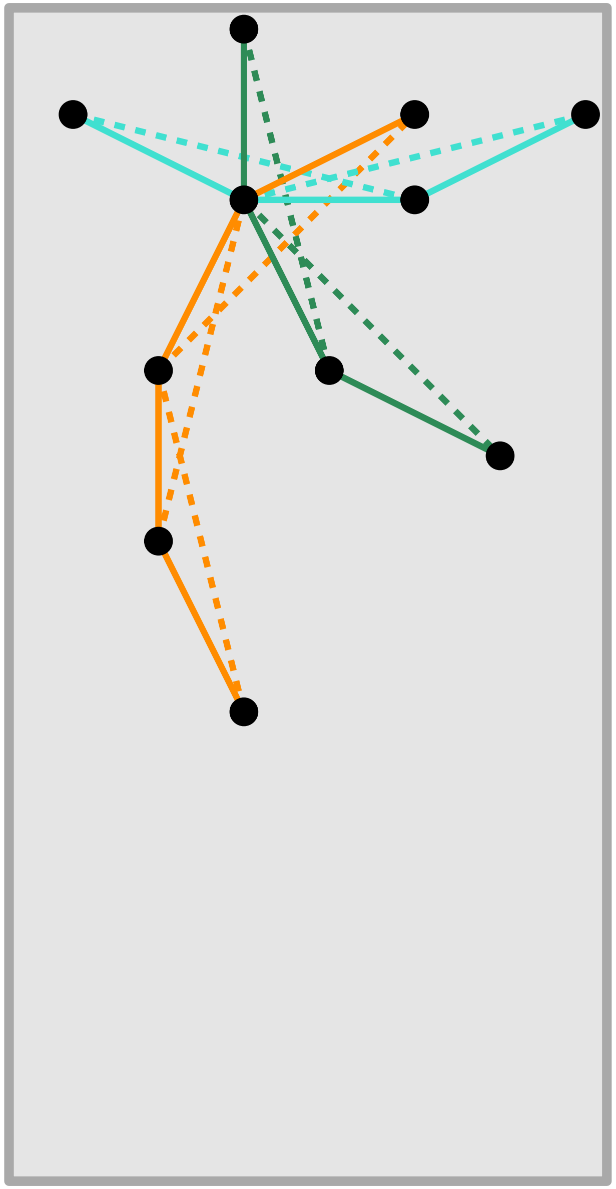
Aren't Both Goals the Same?

No! Here is a counterexample:



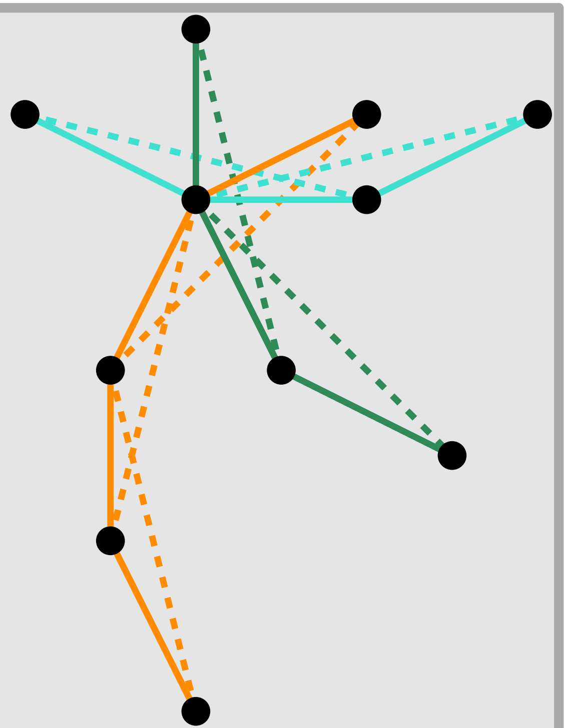
Aren't Both Goals the Same?

No! Here is a counterexample:



Aren't Both Goals the Same?

No! Here is a counterexample:

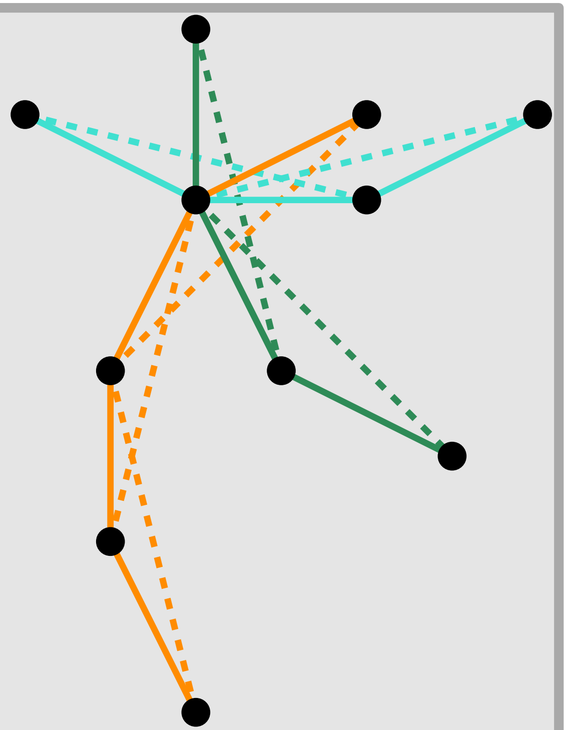


base bundle

11 vertices
10 edges

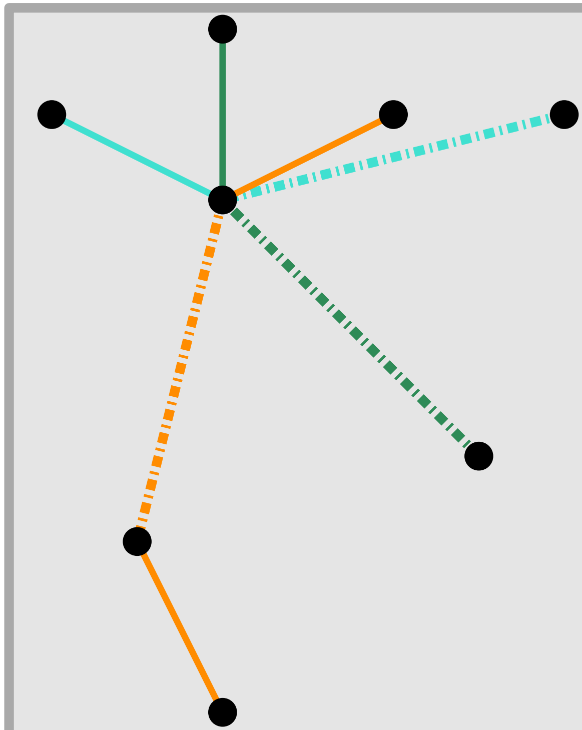
Aren't Both Goals the Same?

No! Here is a counterexample:



base bundle

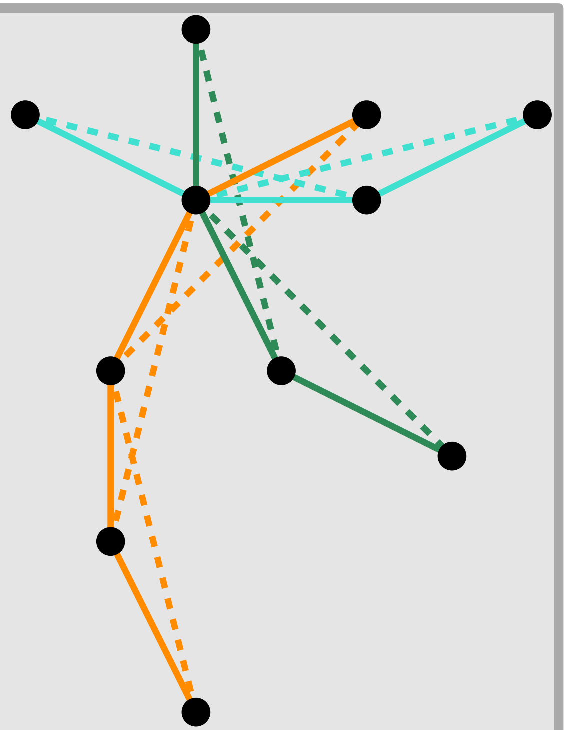
11 vertices
10 edges



optimal for
MIN-VERTICES
8 vertices
7 edges

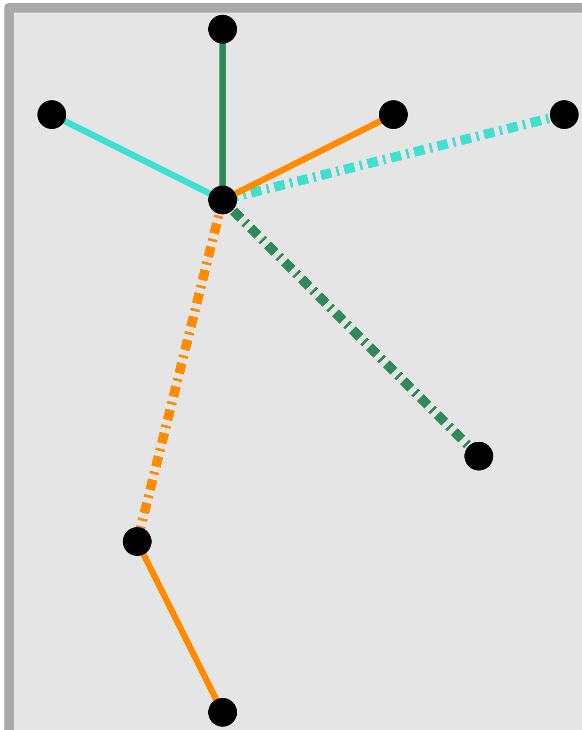
Aren't Both Goals the Same?

No! Here is a counterexample:

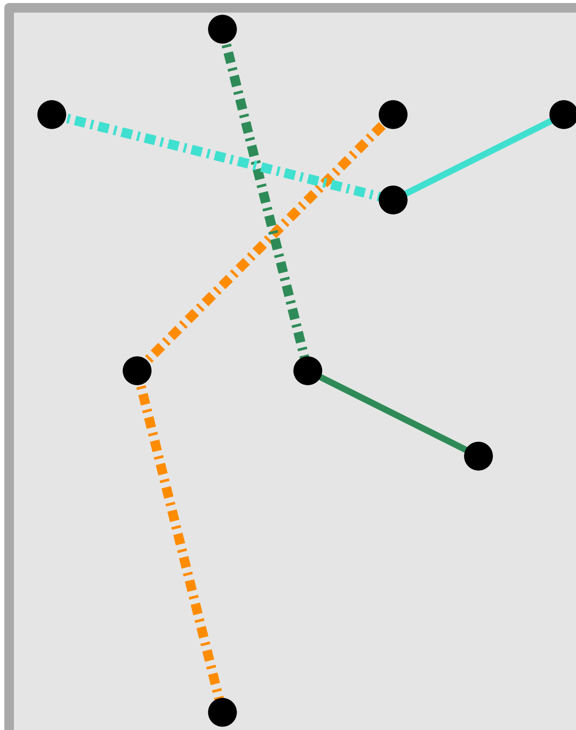


base bundle

11 vertices
10 edges



optimal for
MIN-VERTICES
8 vertices
7 edges



optimal for
MIN-EDGES
9 vertices
6 edges

Why Not Simplifying Independently?

Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

Why Not Simplifying Independently?

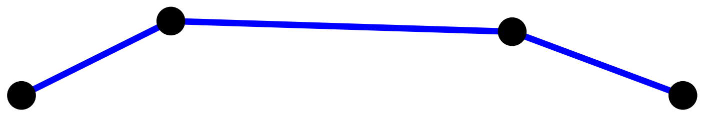
A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality

Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

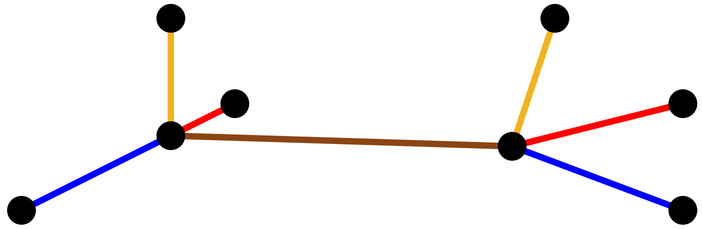
A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality

Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

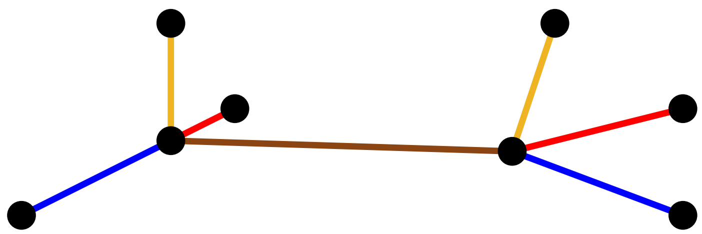
$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality

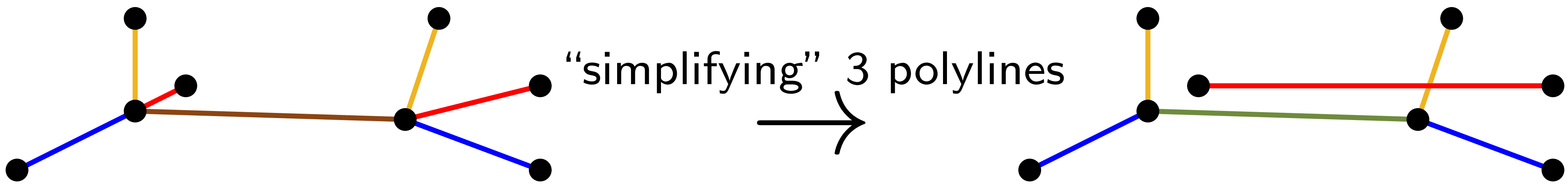


$\xrightarrow{\text{“simplifying” 3 polylines}}$

Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

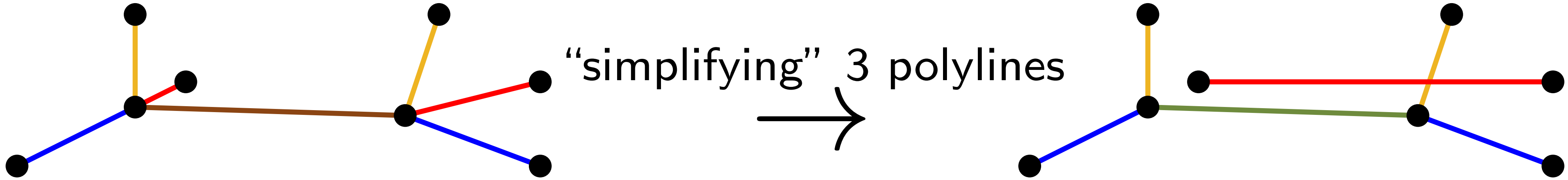
$\Rightarrow$ misleading picture of the reality



Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality

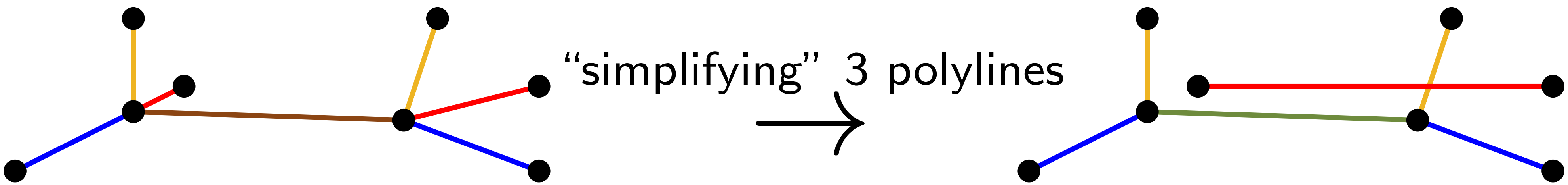


$\Rightarrow$ the total complexity might even increase

Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality

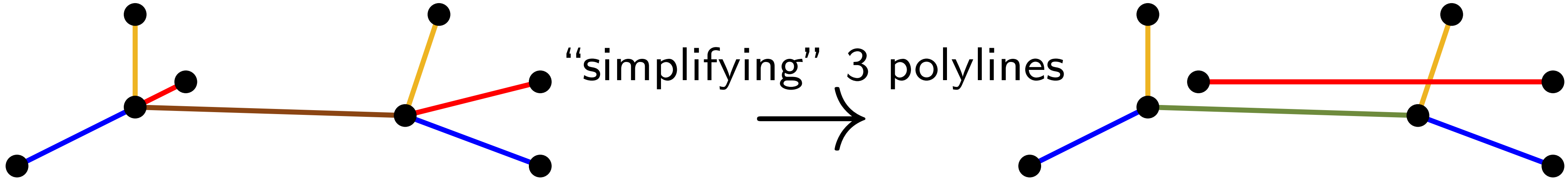


$\Rightarrow$ the total complexity might even increase

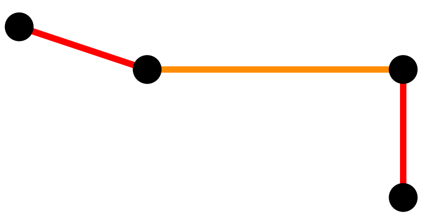
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



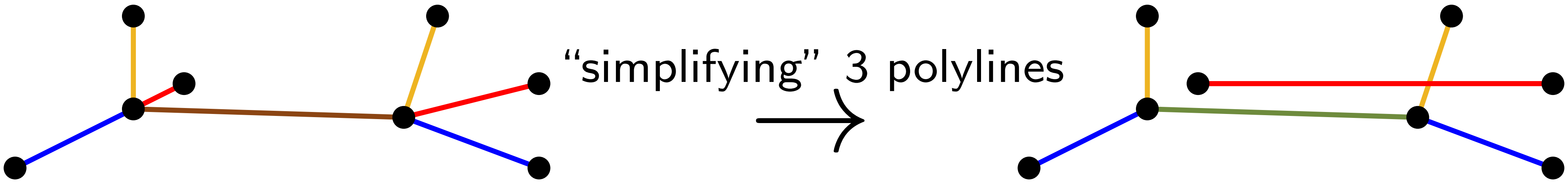
$\Rightarrow$ the total complexity might even increase



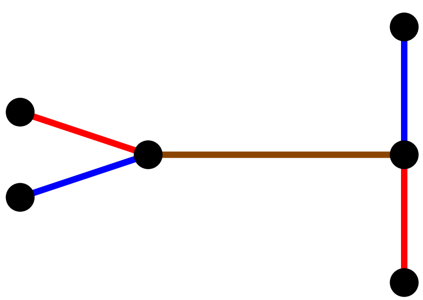
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



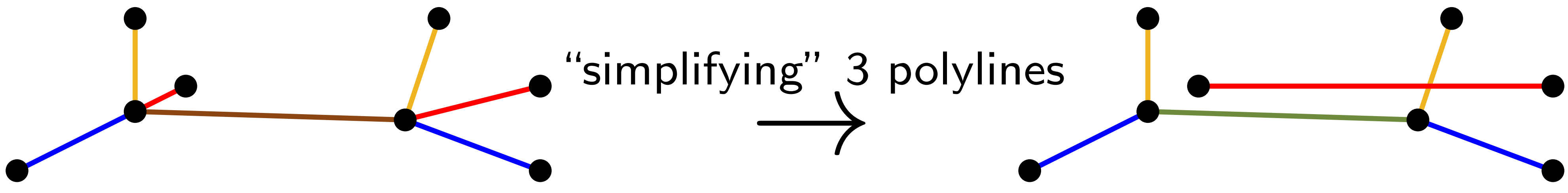
$\Rightarrow$ the total complexity might even increase



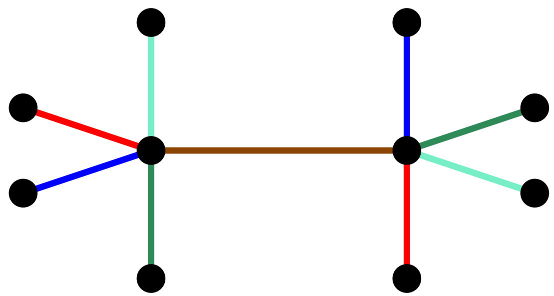
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



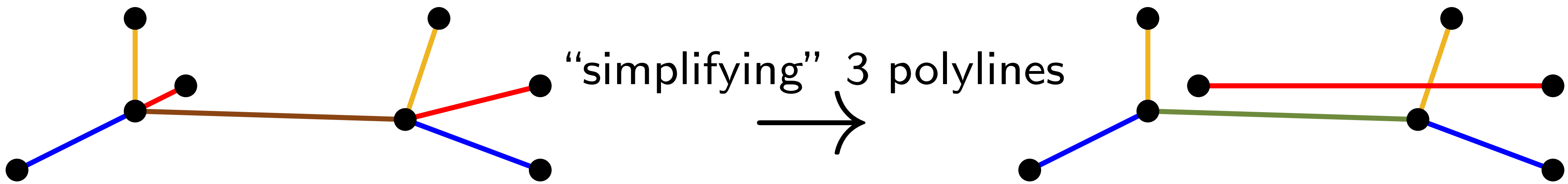
$\Rightarrow$ the total complexity might even increase



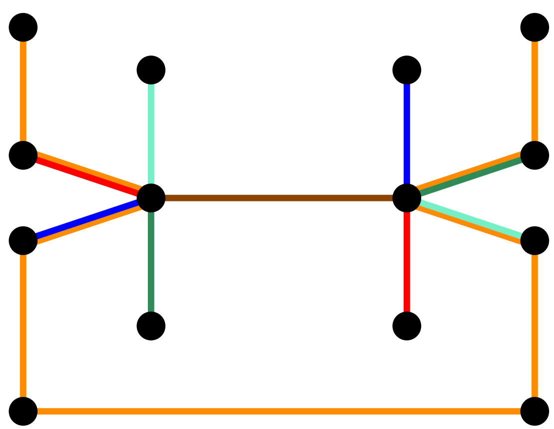
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase

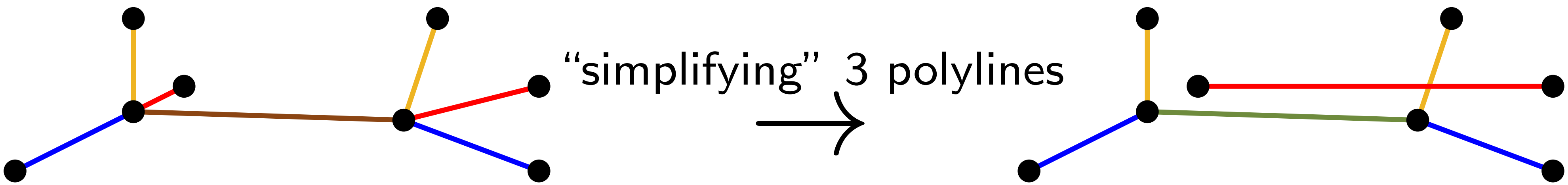


14 vertices, 14 edges

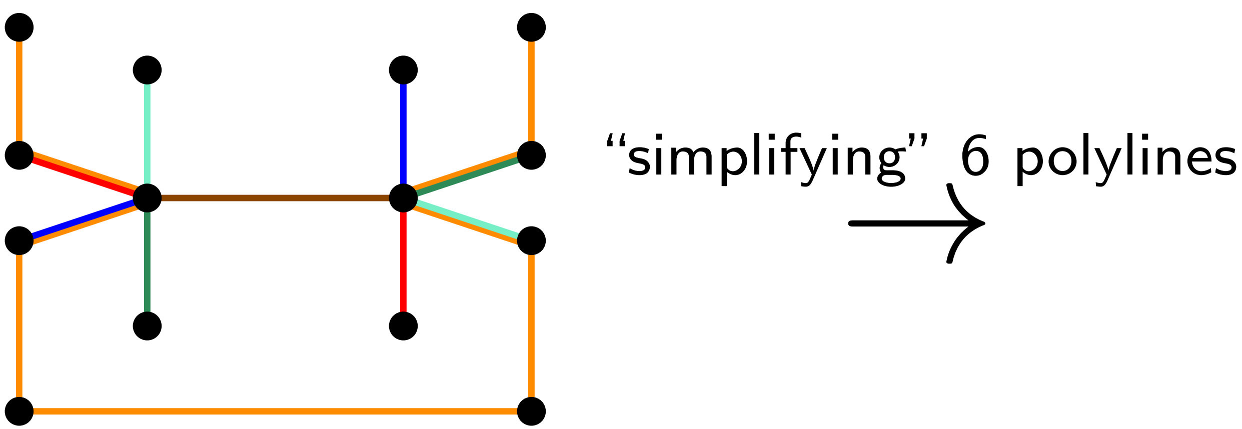
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase

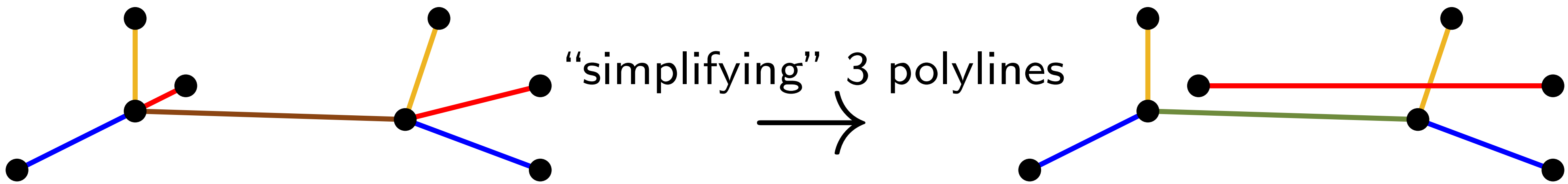


14 vertices, 14 edges

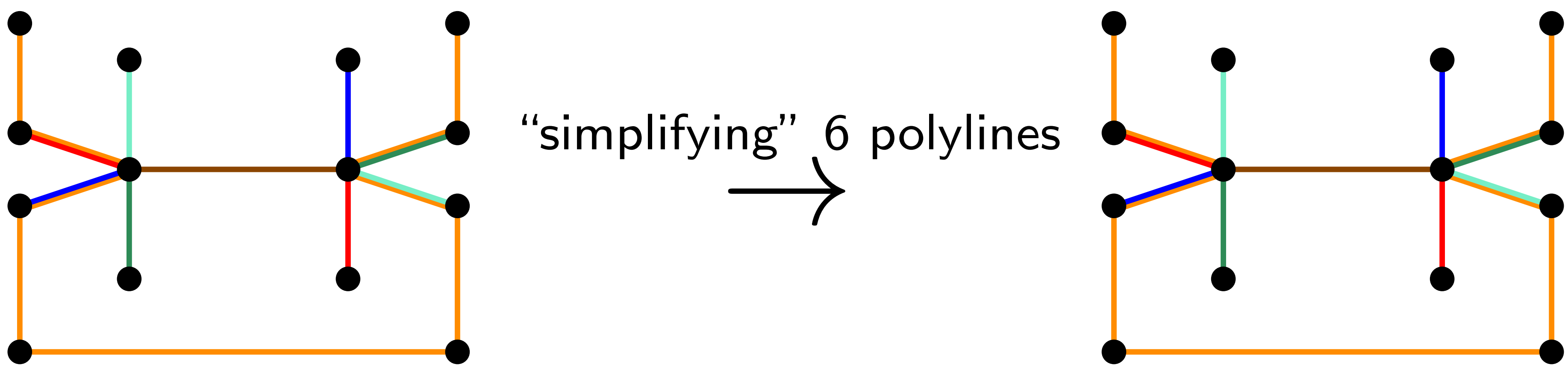
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase

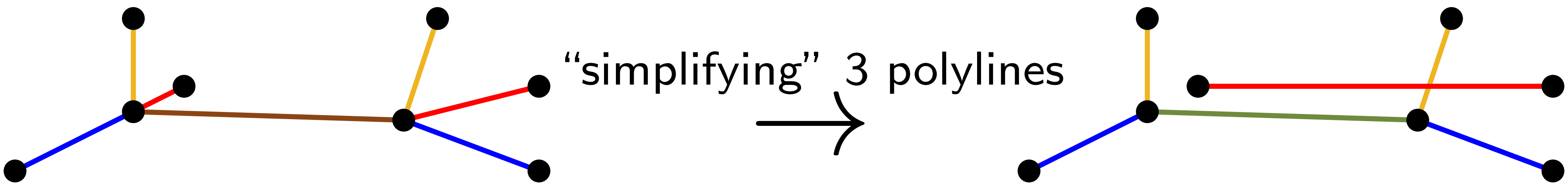


14 vertices, 14 edges

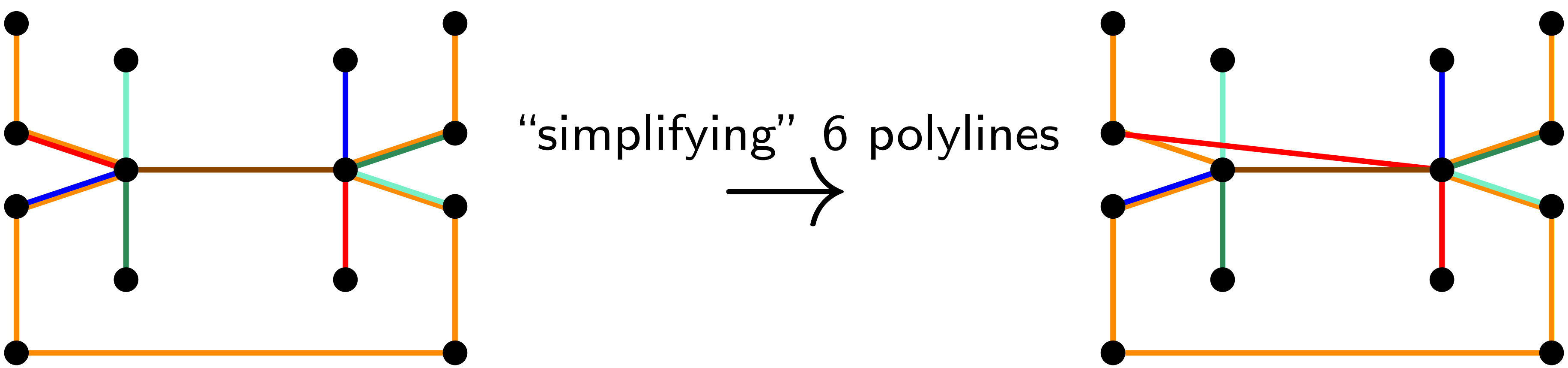
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase

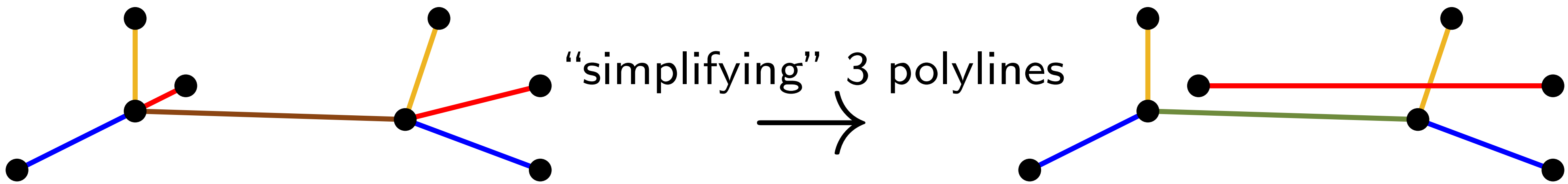


14 vertices, 14 edges

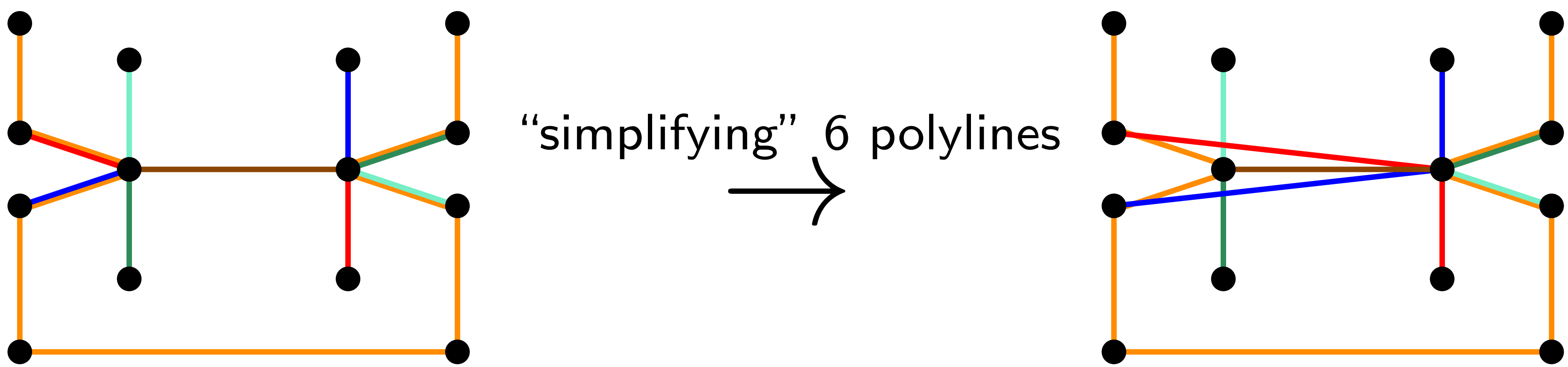
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase

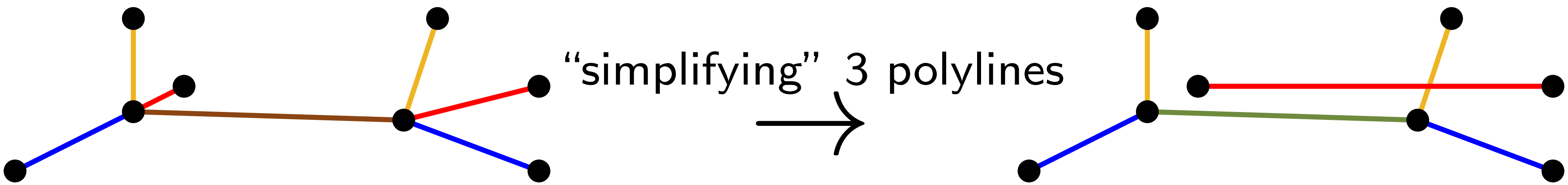


14 vertices, 14 edges

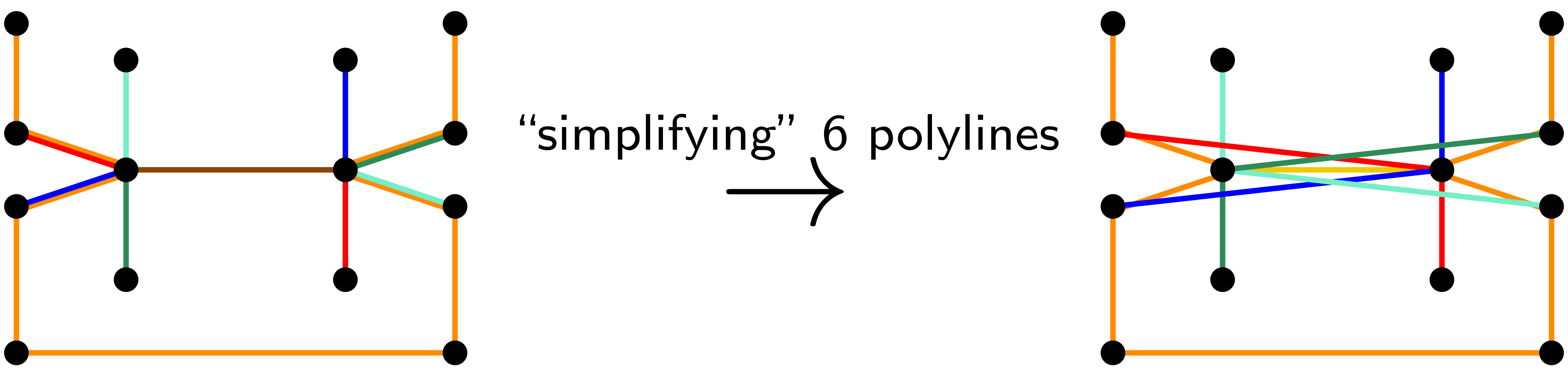
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase

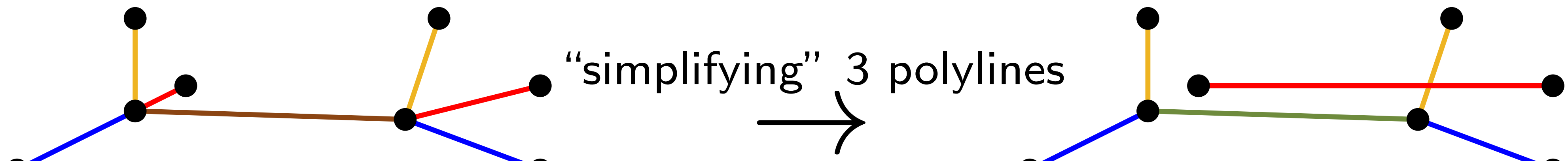


14 vertices, 14 edges

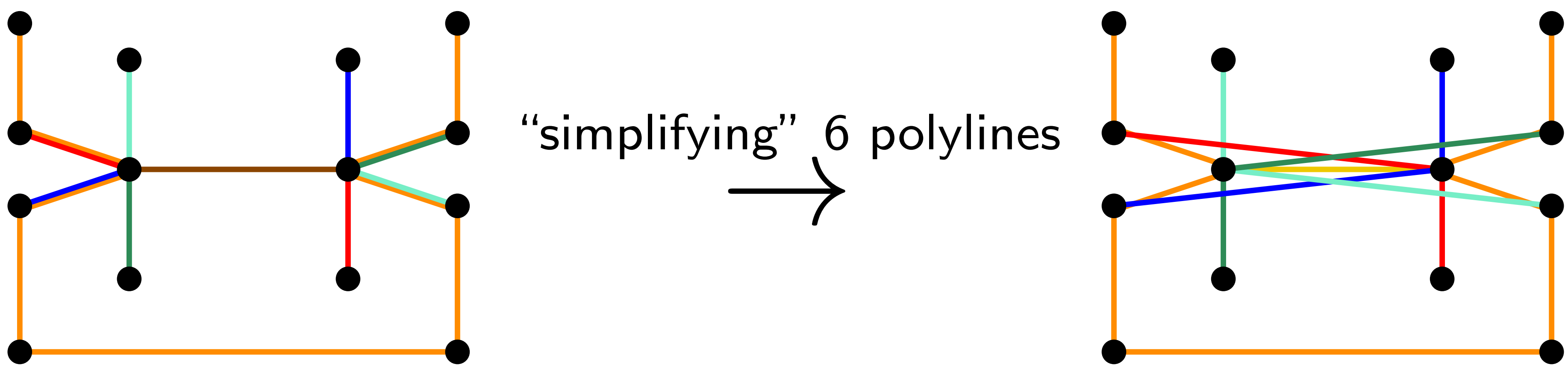
Why Not Simplifying Independently?

A vertex or edge might once be kept and once be discarded

$\Rightarrow$ misleading picture of the reality



$\Rightarrow$ the total complexity might even increase



14 vertices, 14 edges

14 vertices, 18 edges

NP-Hardness

NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$ even for 2 polylines.

NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals Min-Vertices and Min-Edges even for 2 polylines.

Proof Sketch:

NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$ even for 2 polylines.

Proof Sketch:

- We reduce from $\textsc{Max-2-SAT}$.

NP-Hardness

Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$ even for 2 polylines.

Proof Sketch:

- We reduce from $\textsc{Max-2-SAT}$.
- We use a polyline as a gadget for each literal in each clause. We connect them serially. $\rightarrow$ first polyline

NP-Hardness

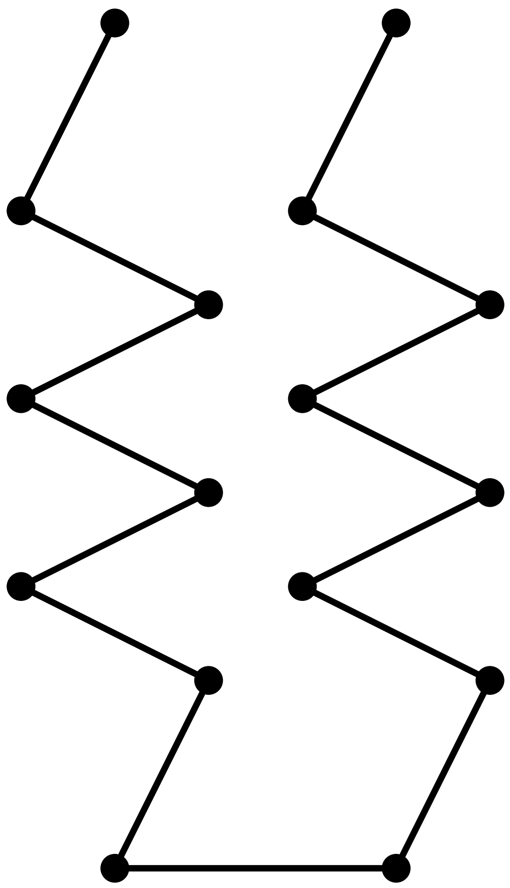
Theorem 1:
Simplifying a bundle of polylines is NP-hard for the goals MIN-VERTICES and MIN-EDGES even for 2 polylines.

Proof Sketch:

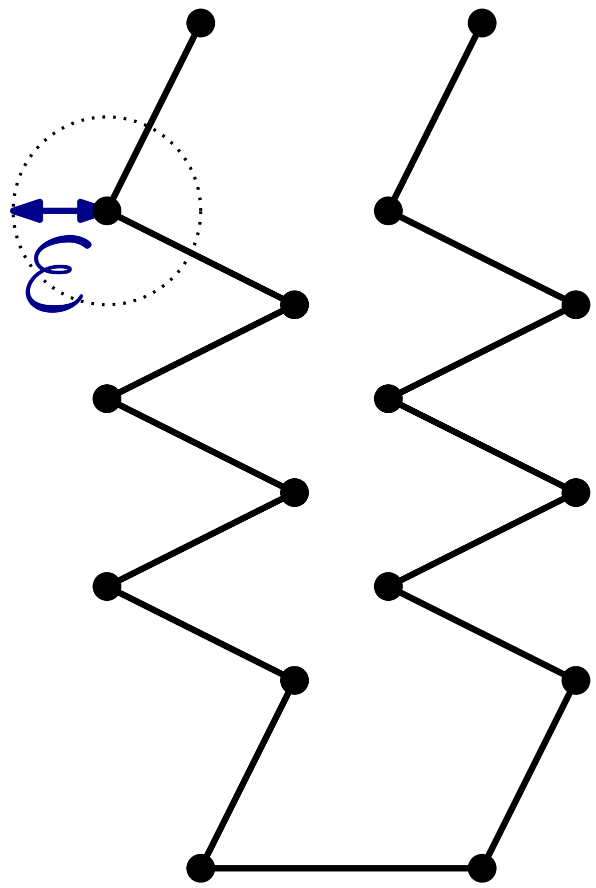
- We reduce from MAX-2-SAT.
- We use a polyline as a gadget for each literal in each clause. We connect them serially. $\rightarrow$ first polyline
- We use a polyline as a gadget for each variable and each clause. We connect them serially. $\rightarrow$ second polyline

Variable-Gadget

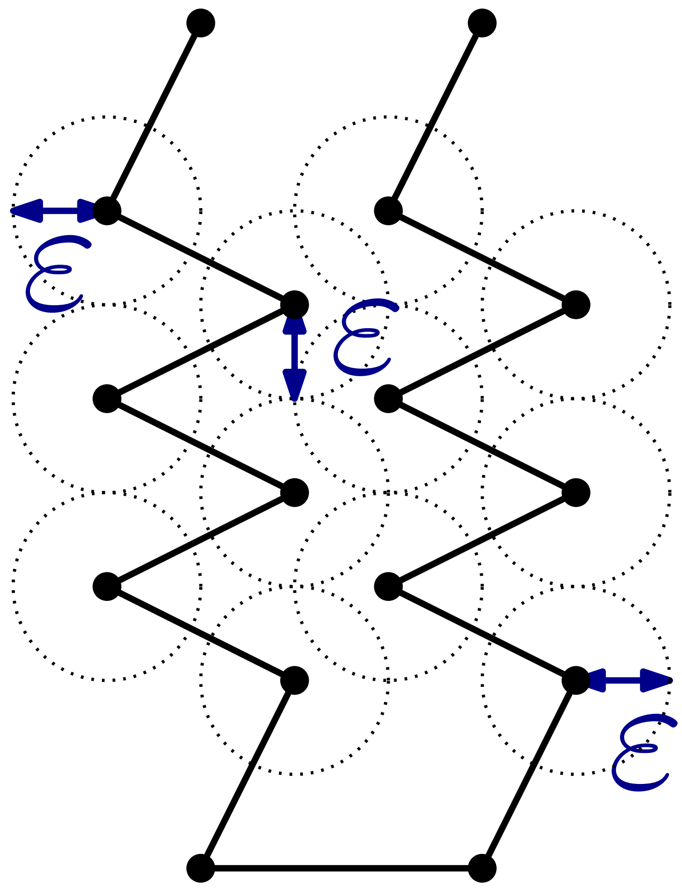
Variable-Gadget



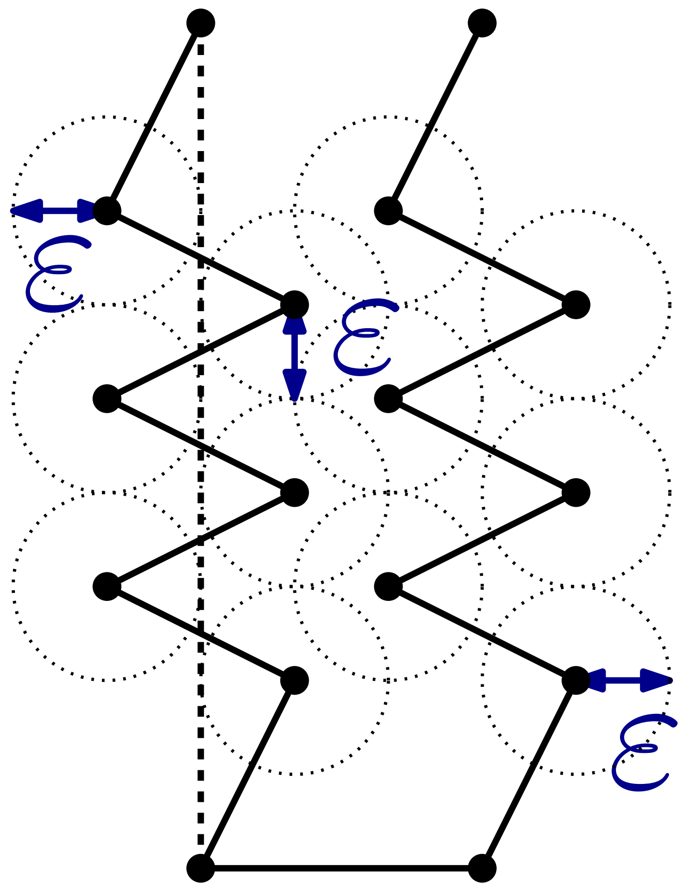
Variable-Gadget



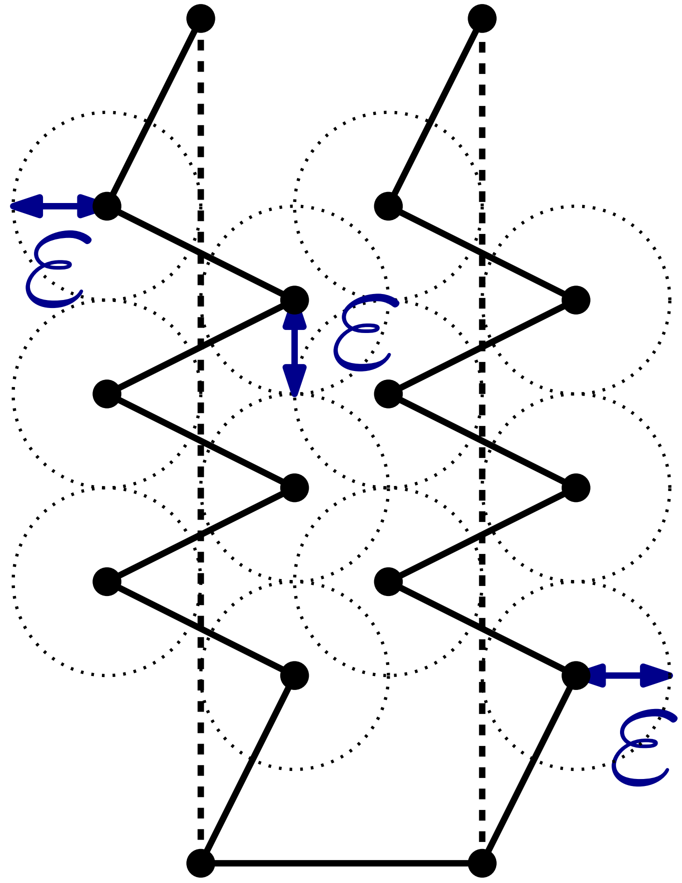
Variable-Gadget



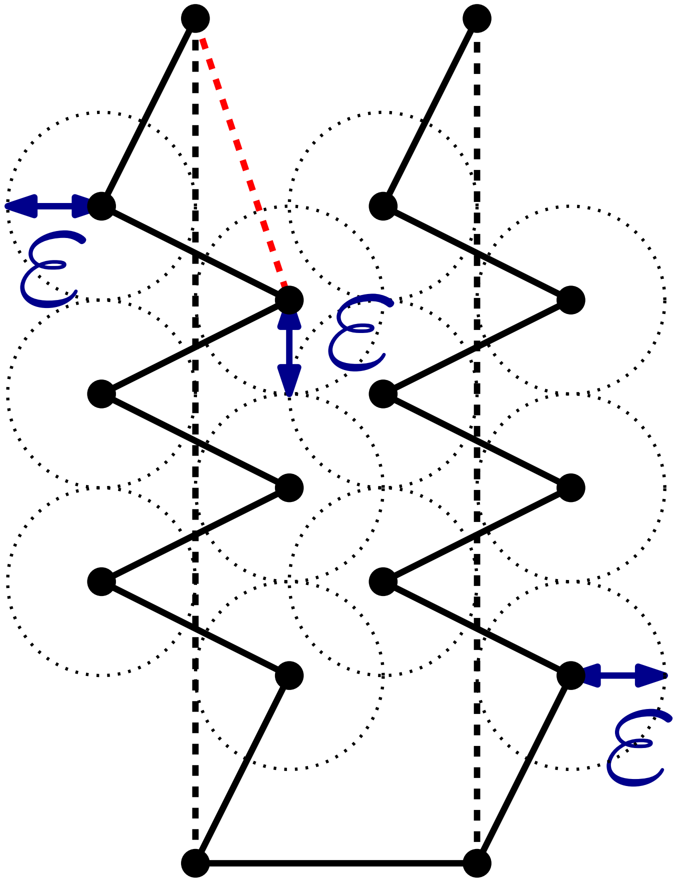
Variable-Gadget



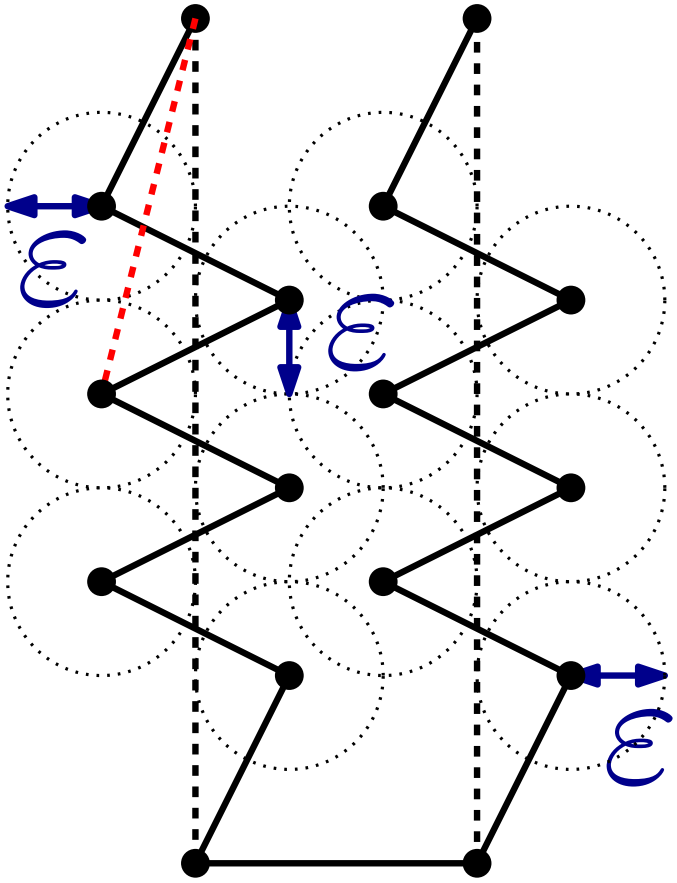
Variable-Gadget



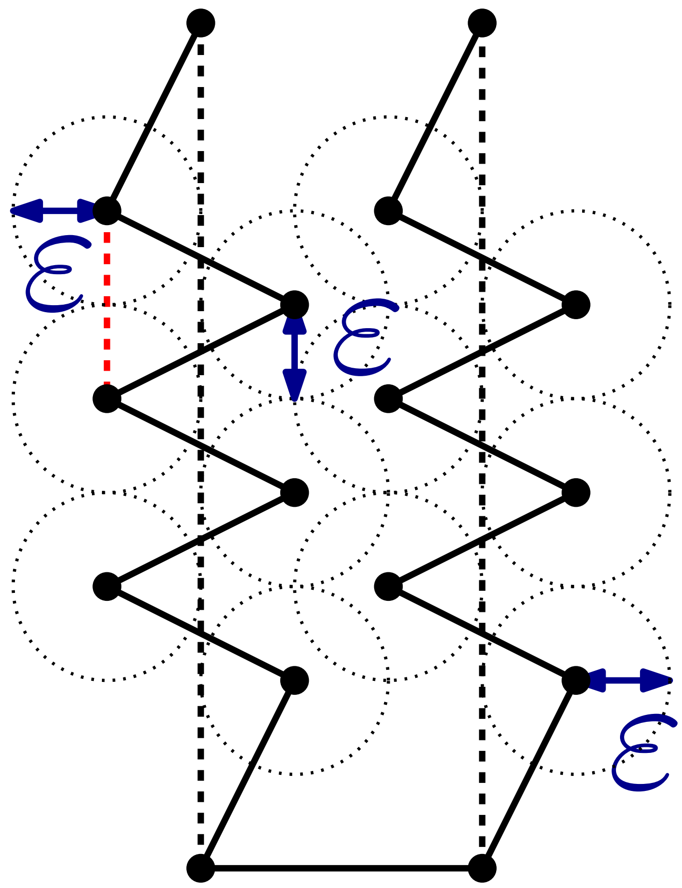
Variable-Gadget



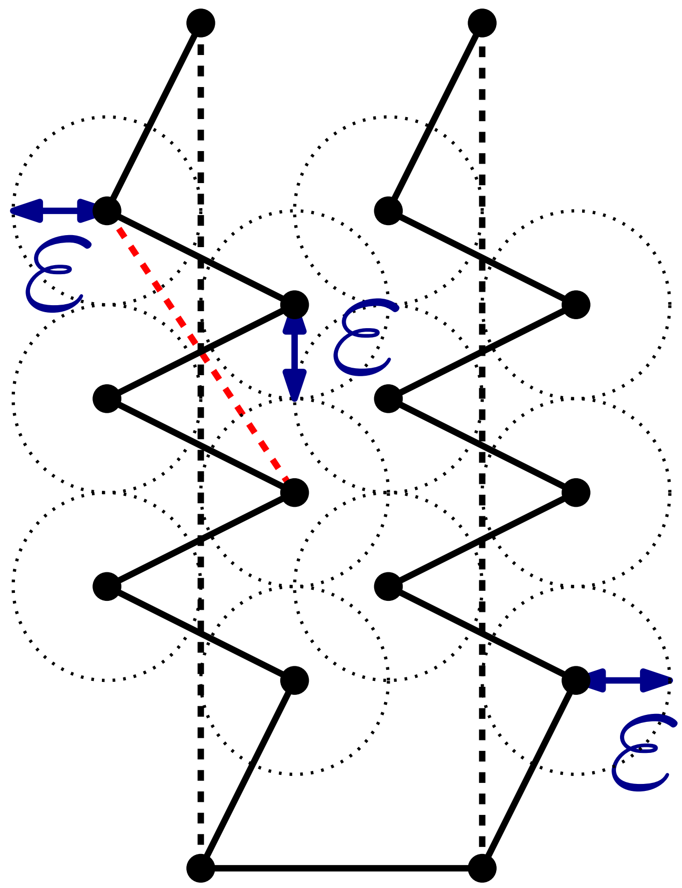
Variable-Gadget



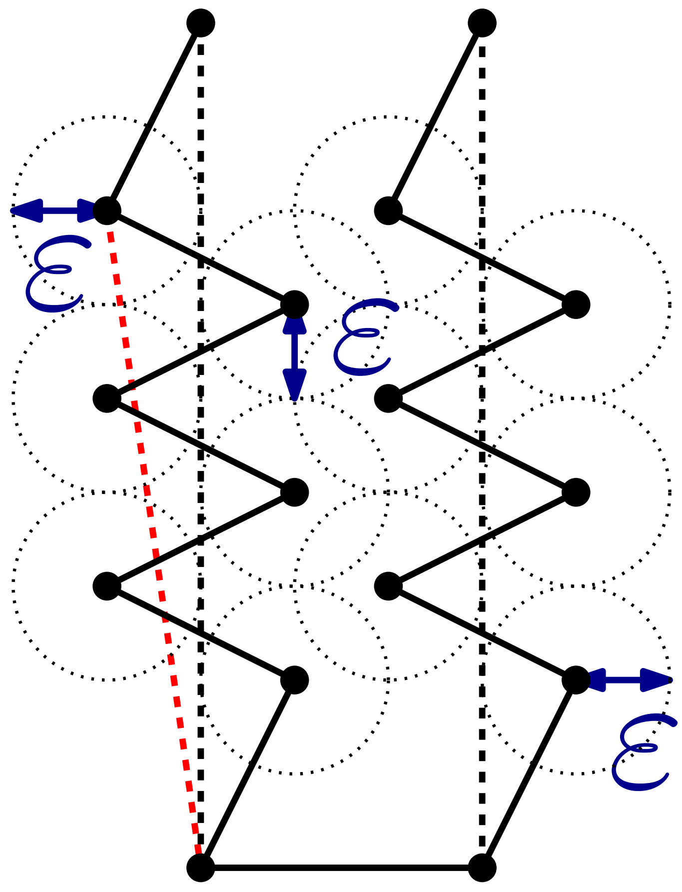
Variable-Gadget



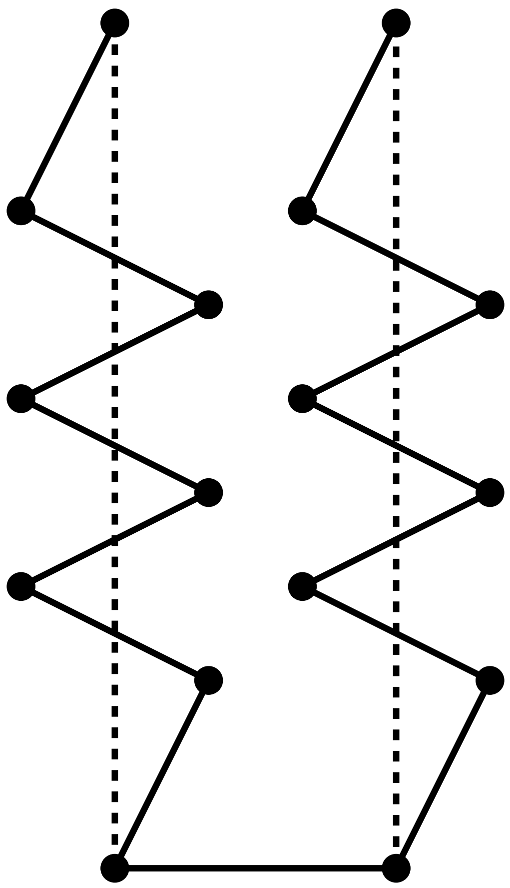
Variable-Gadget



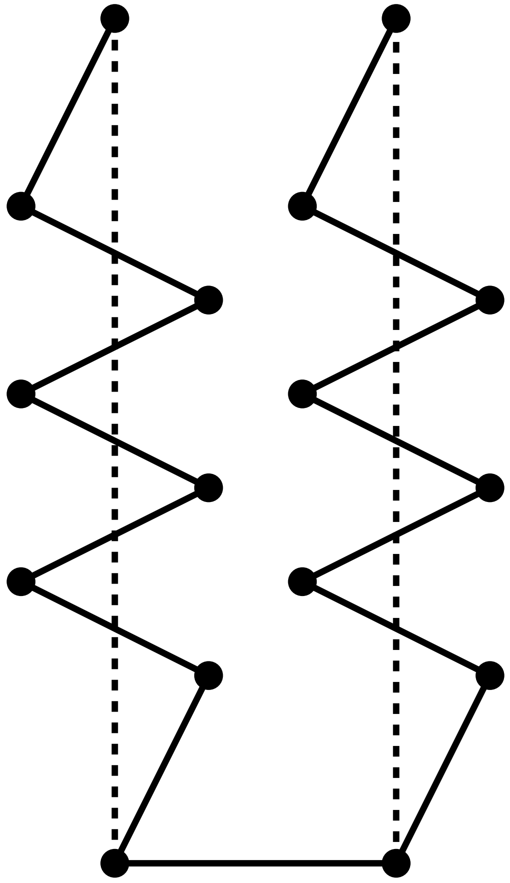
Variable-Gadget



Variable-Gadget

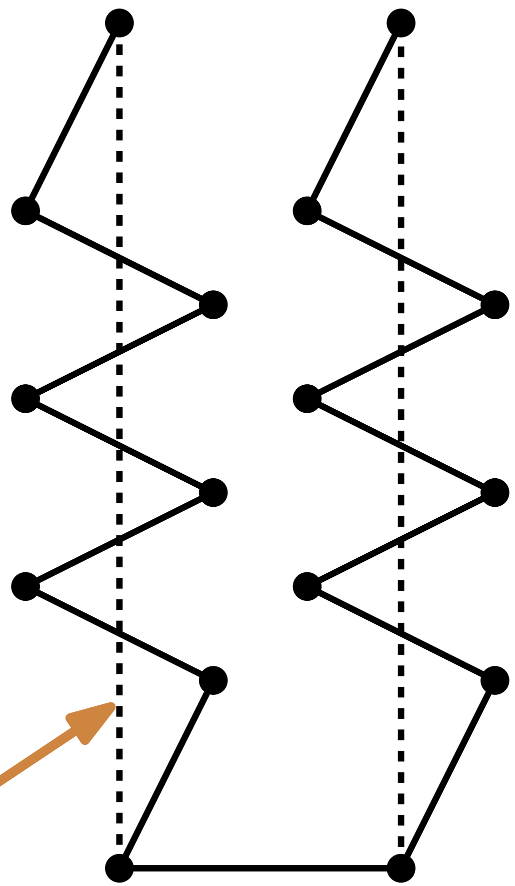


Variable-Gadget



Interpretation:

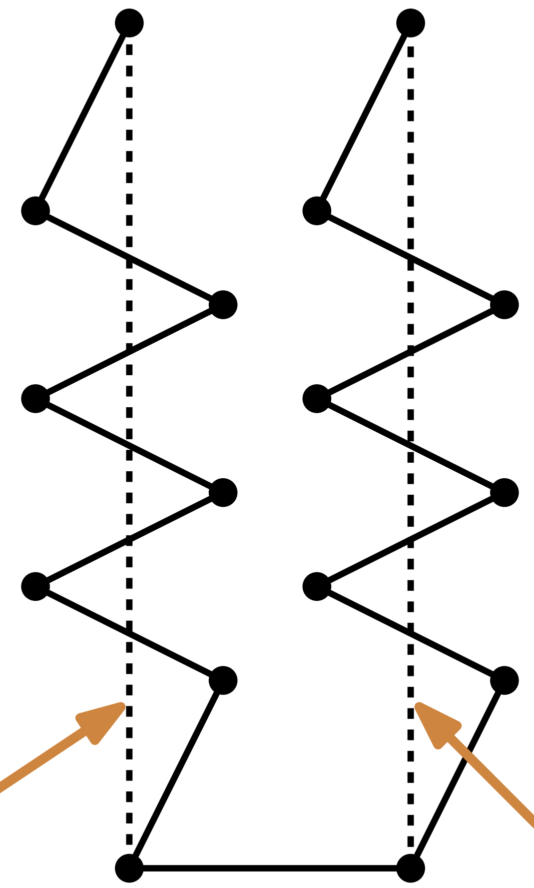
Variable-Gadget



Interpretation:

shortcut taken $\Leftrightarrow$ variable set to true

Variable-Gadget

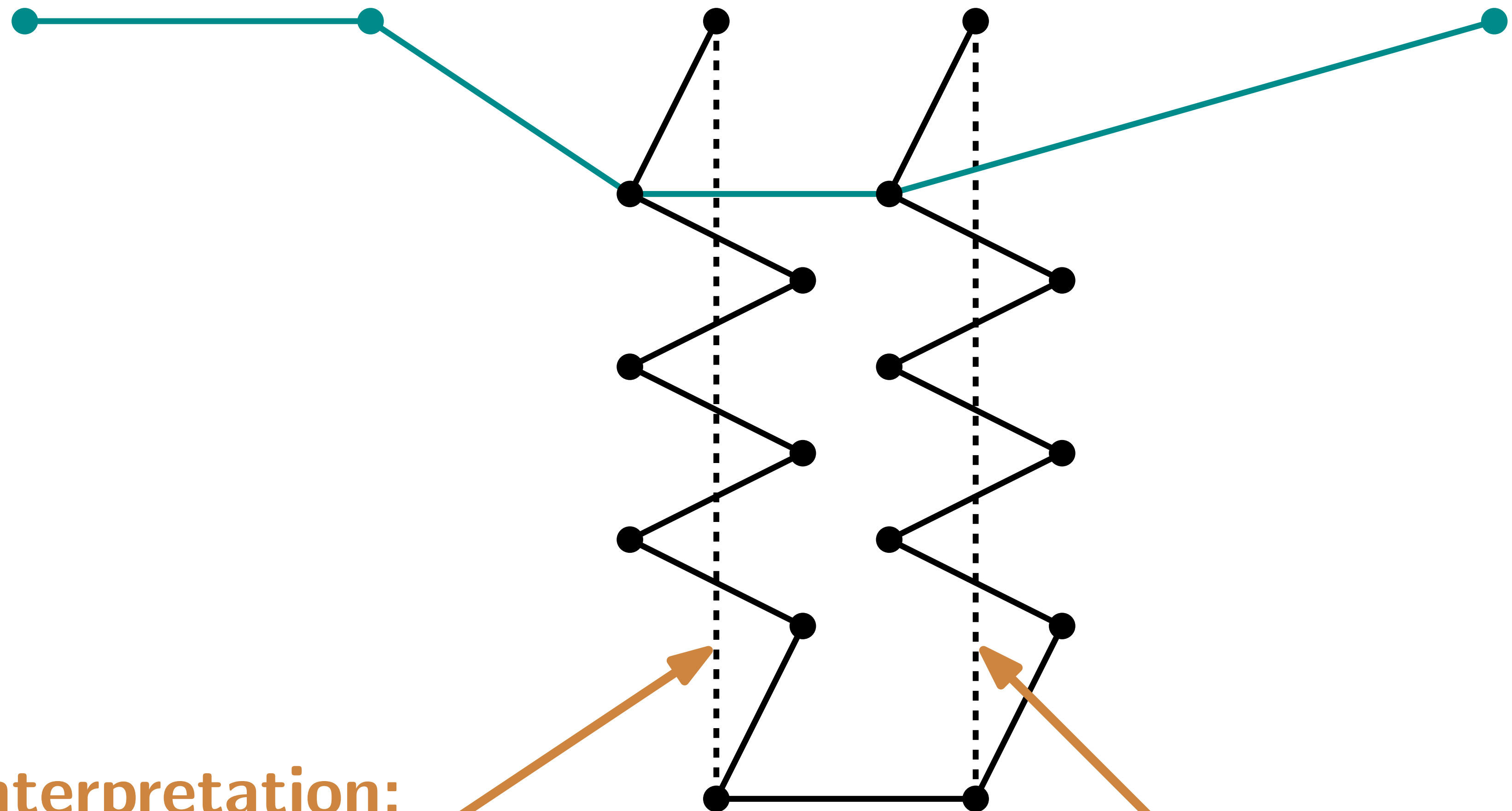


Interpretation:

shortcut taken ⇔ variable set to false

shortcut taken ⇔ variable set to true

Variable-Gadget

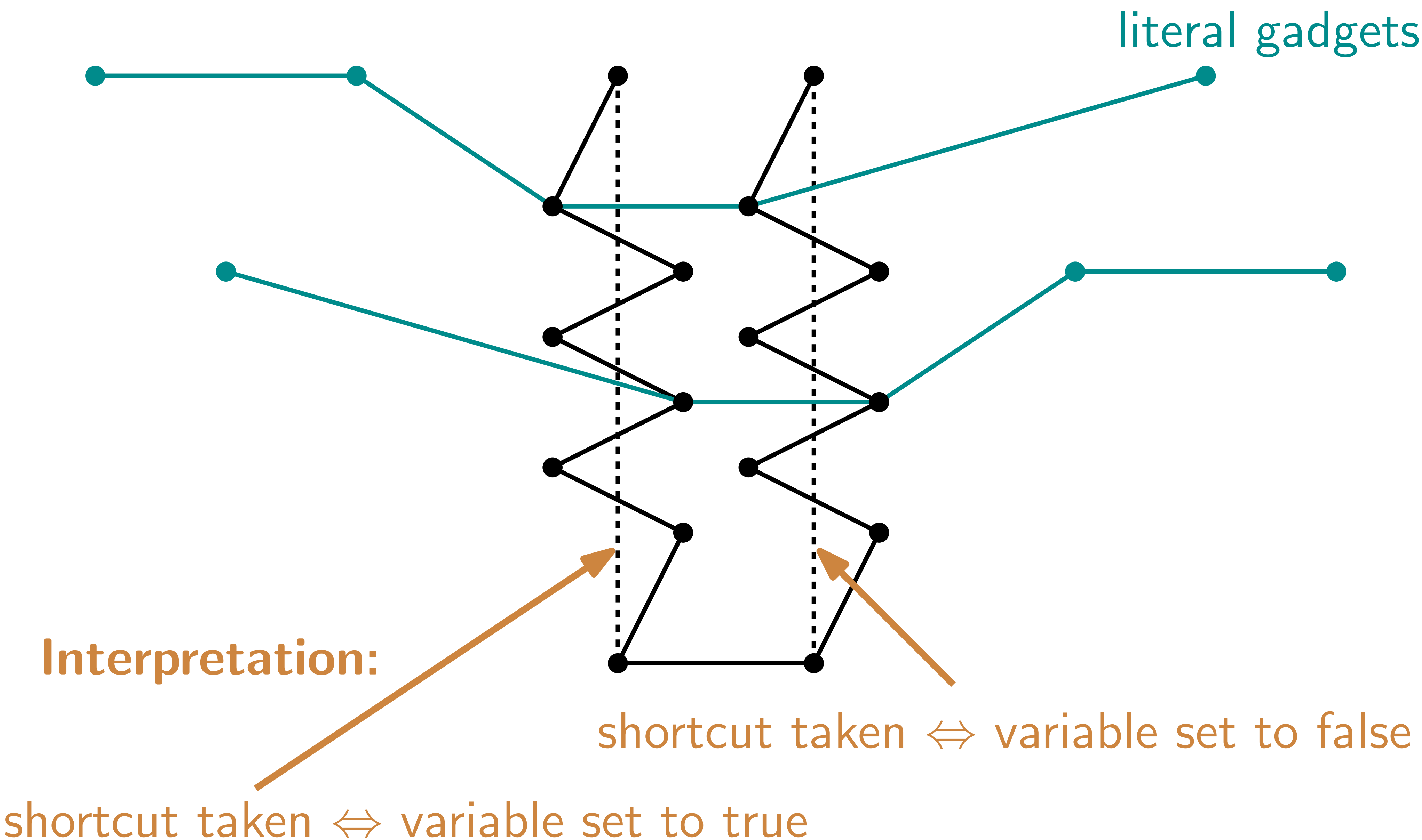


Interpretation:

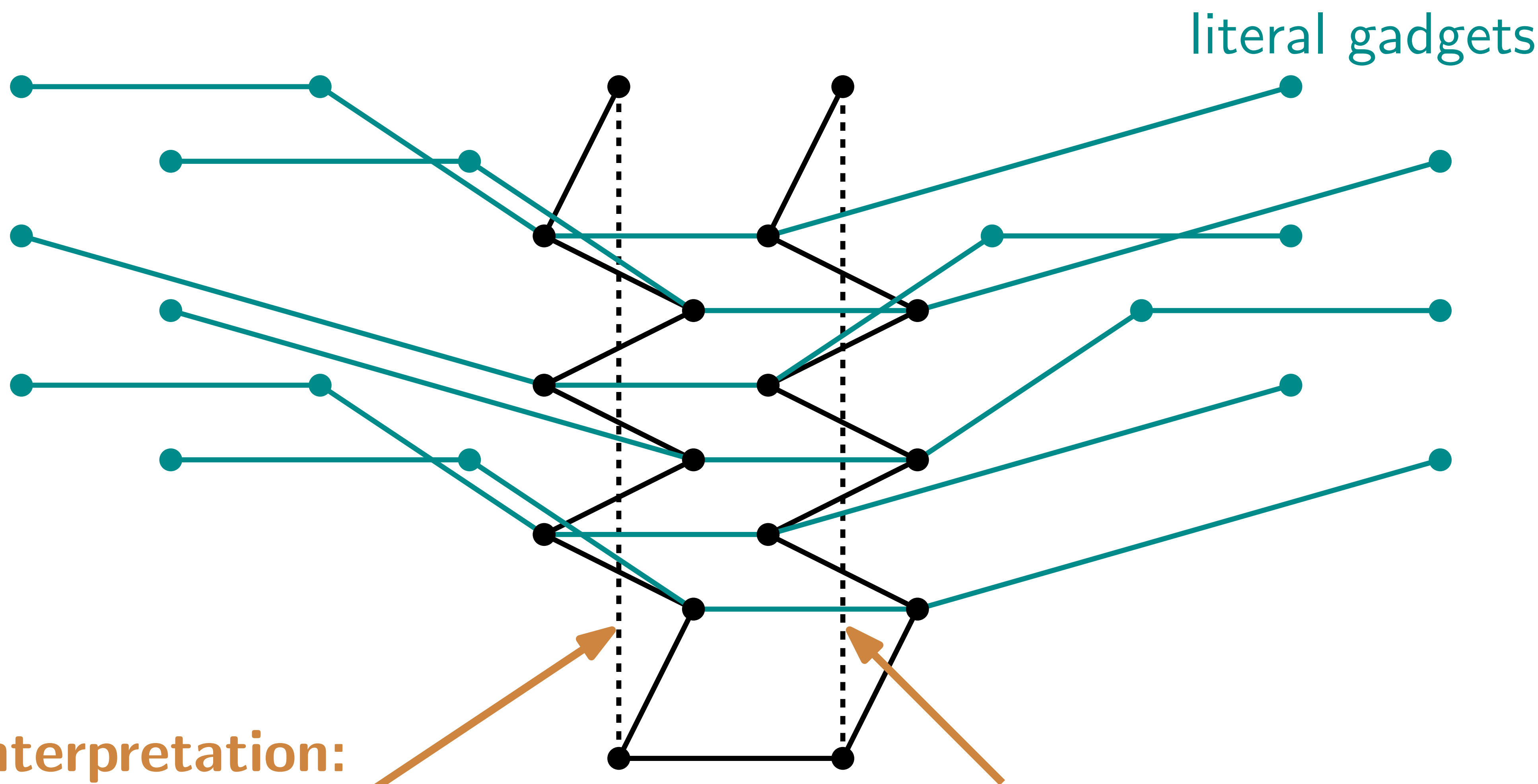
shortcut taken ⇔ variable set to false

shortcut taken ⇔ variable set to true

Variable-Gadget



Variable-Gadget

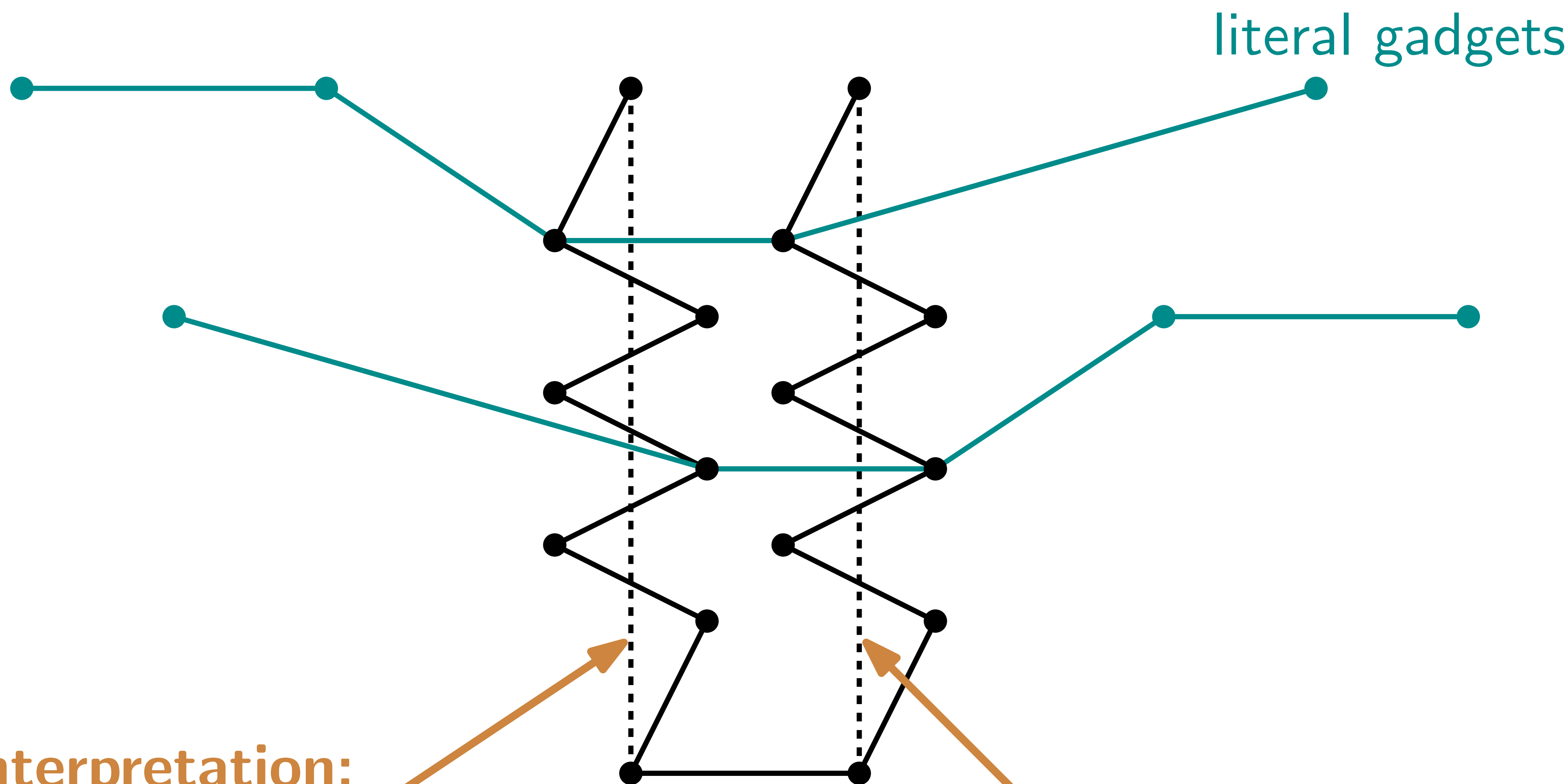


Interpretation:

shortcut taken ⇔ variable set to false

shortcut taken ⇔ variable set to true

Variable-Gadget

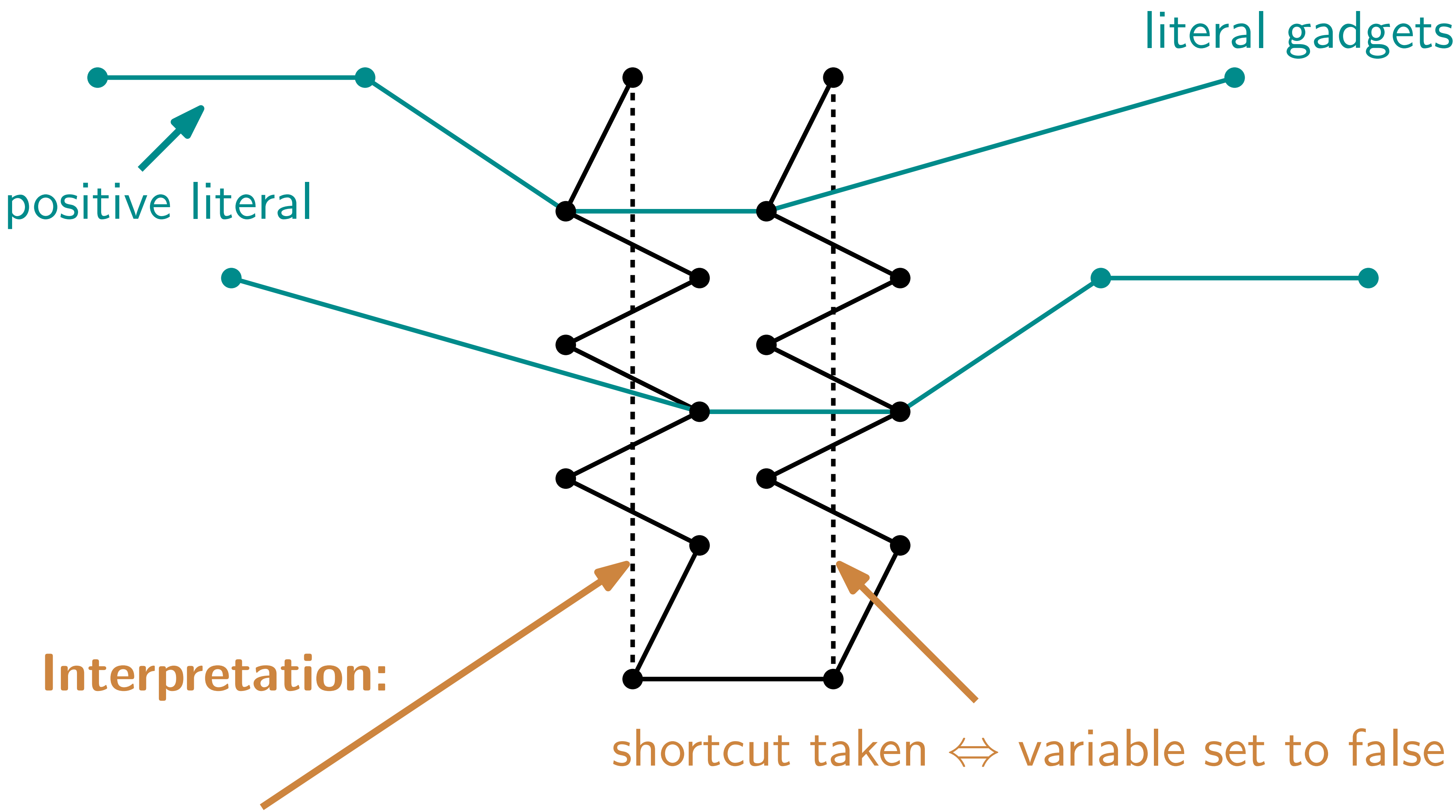


Interpretation:

shortcut taken ⇔ variable set to false

shortcut taken ⇔ variable set to true

Variable-Gadget

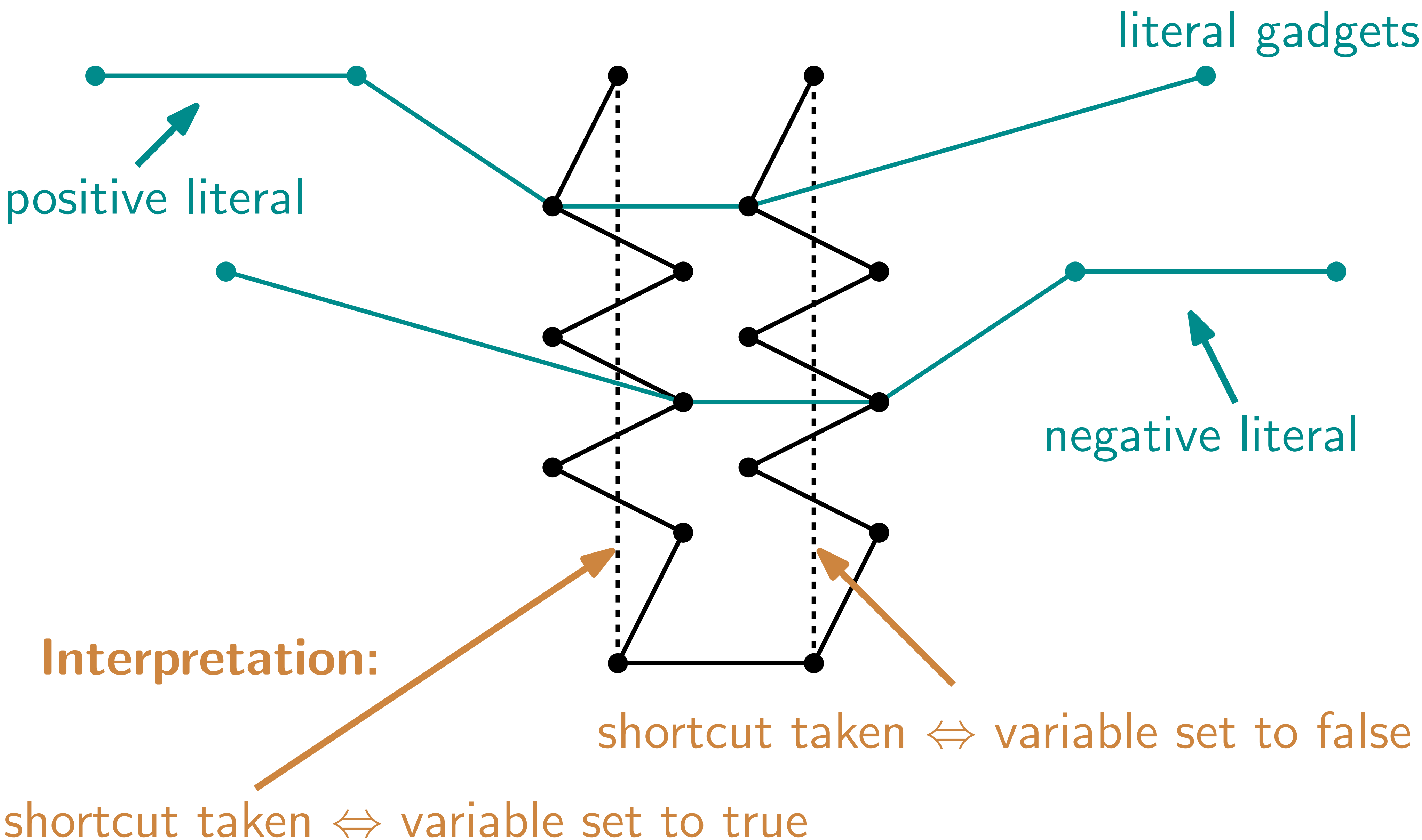


Interpretation:

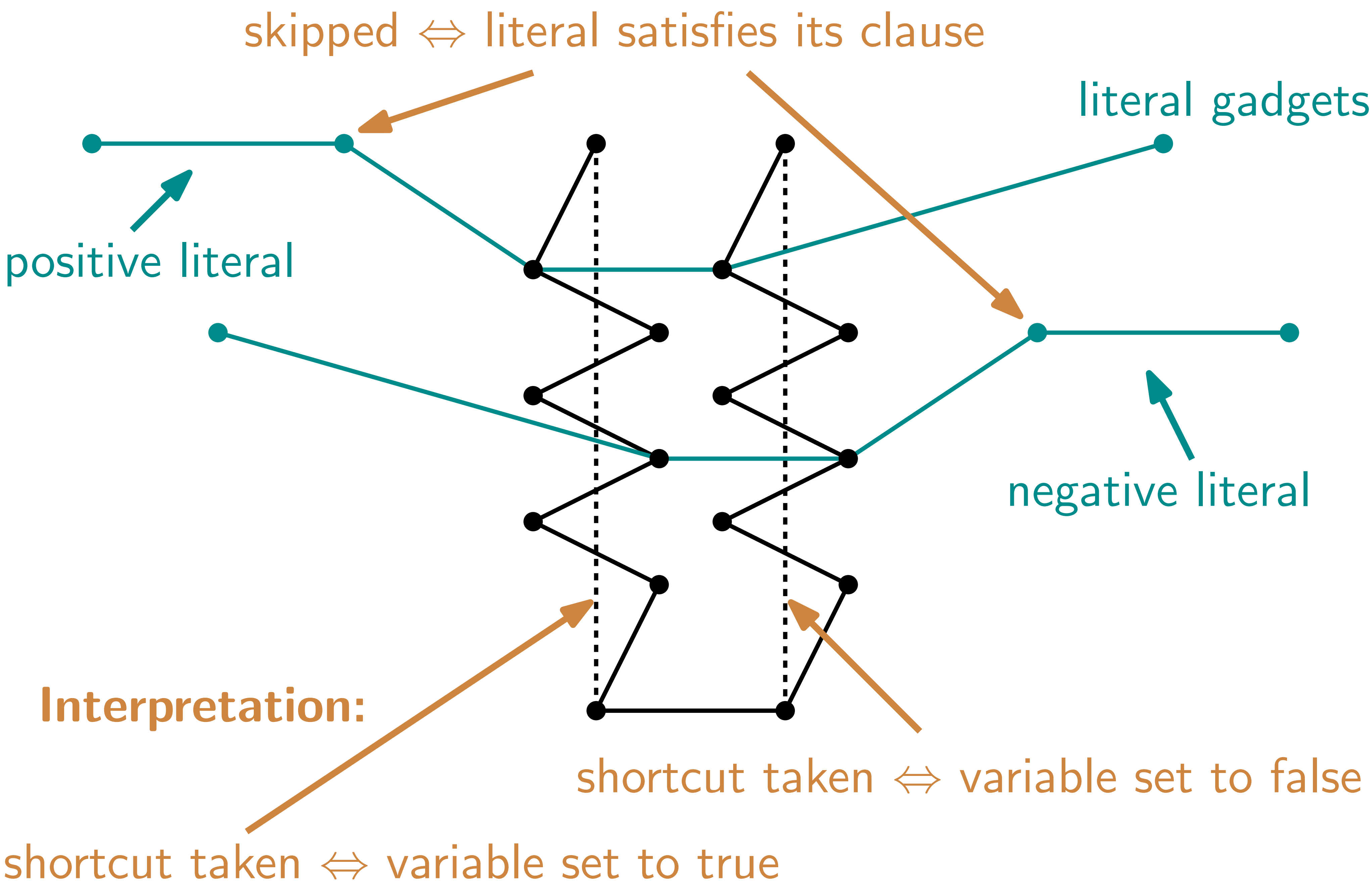
shortcut taken ⇔ variable set to false

shortcut taken ⇔ variable set to true

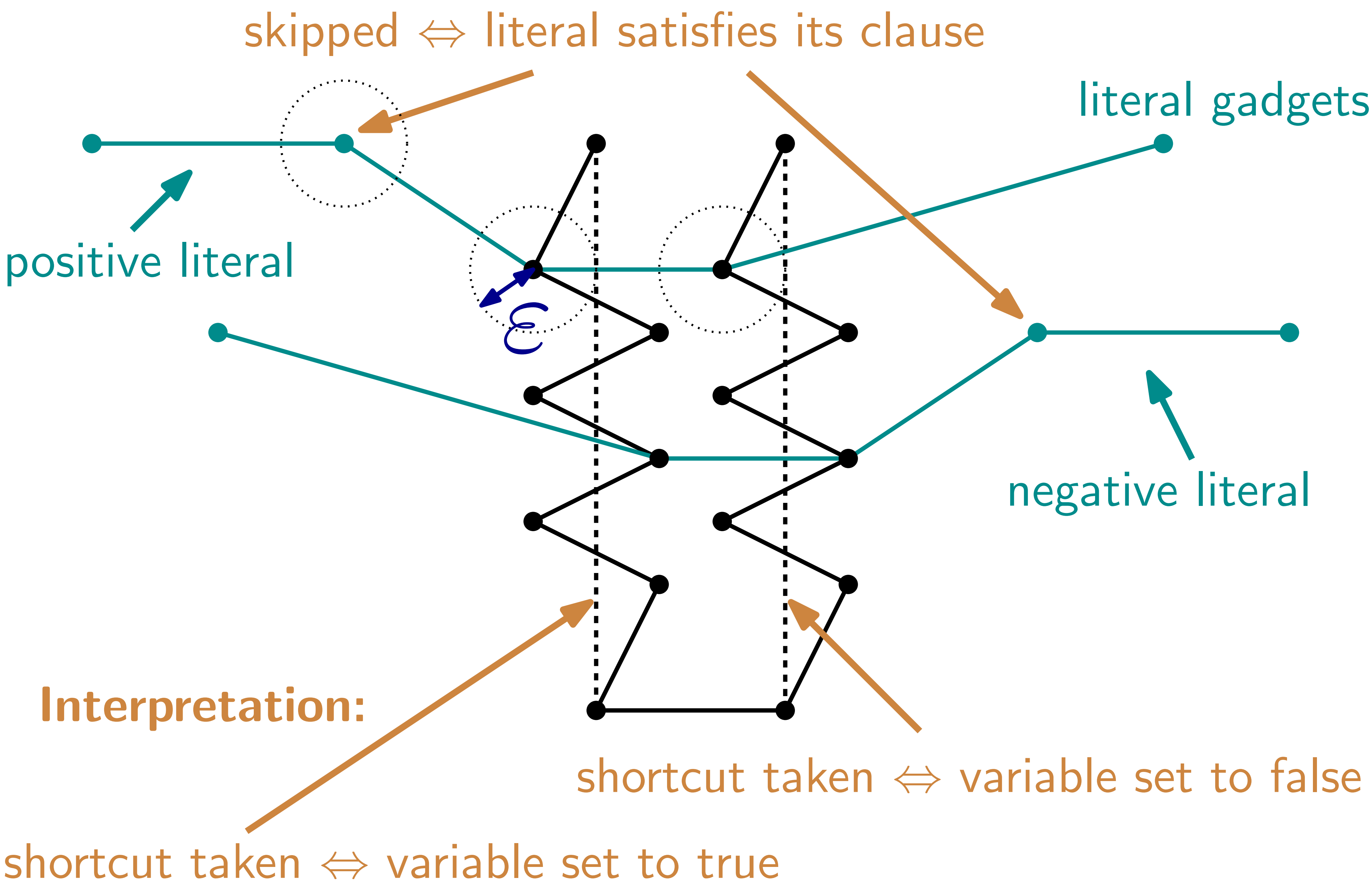
Variable-Gadget



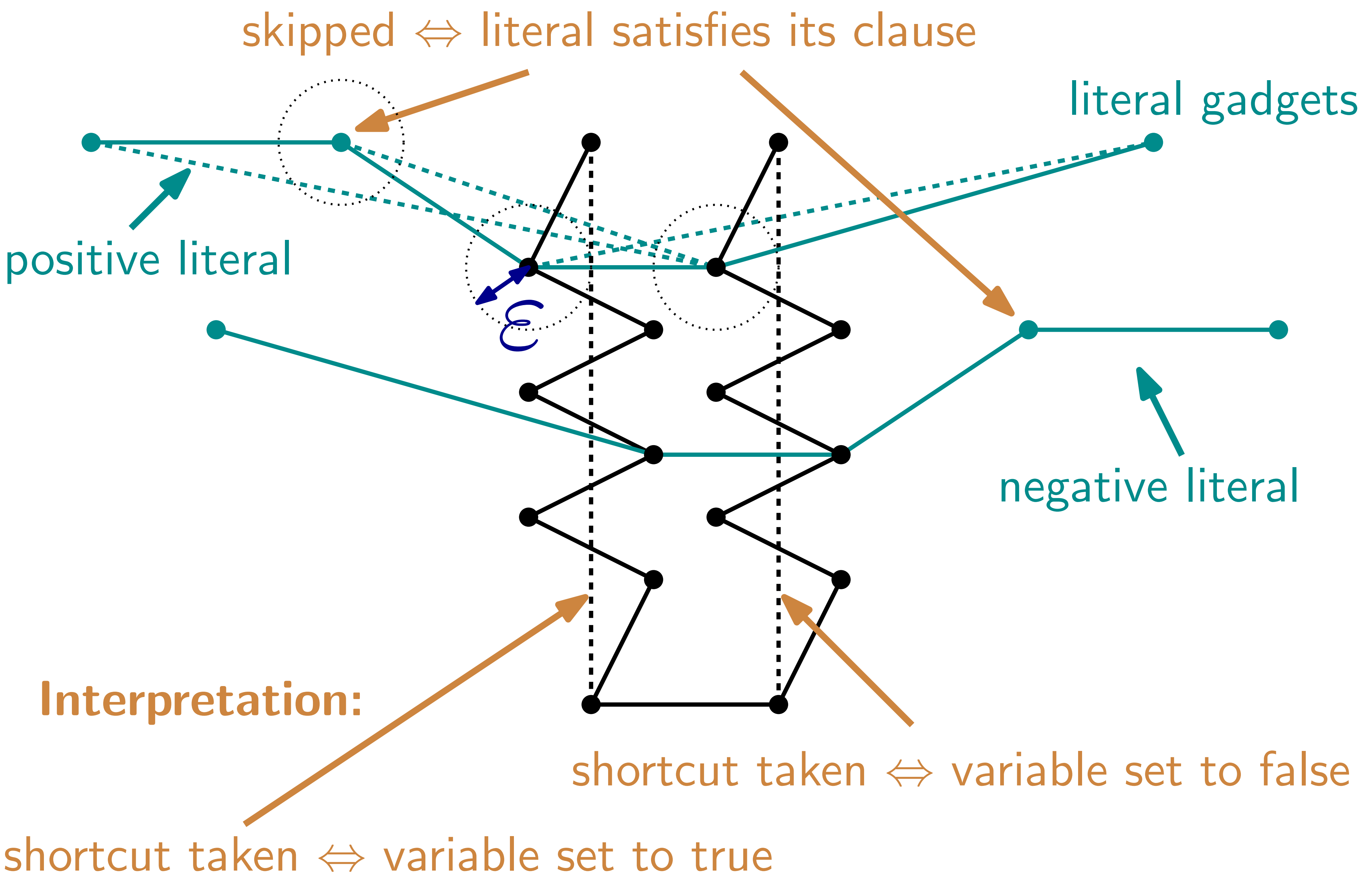
Variable-Gadget



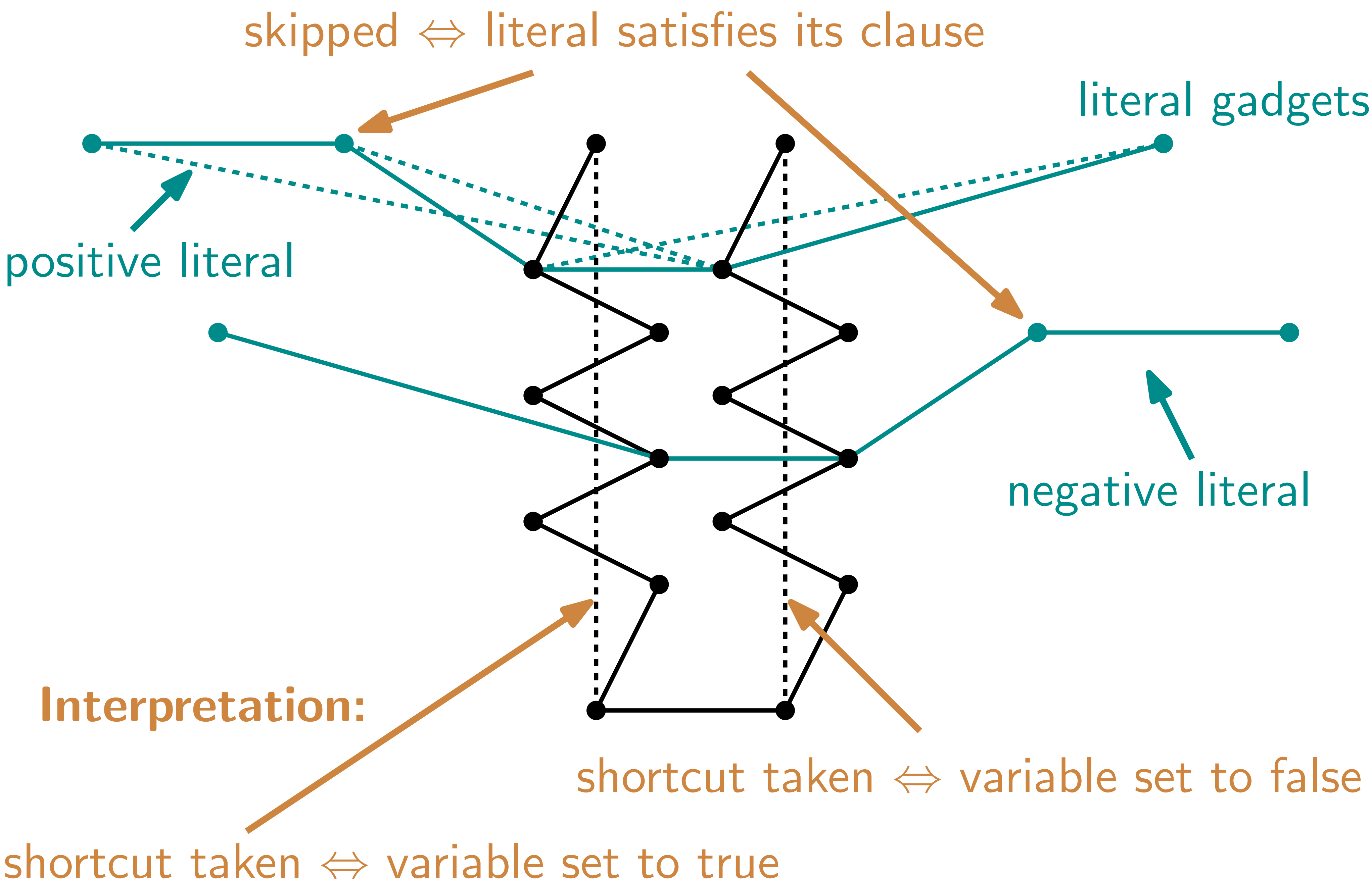
Variable-Gadget



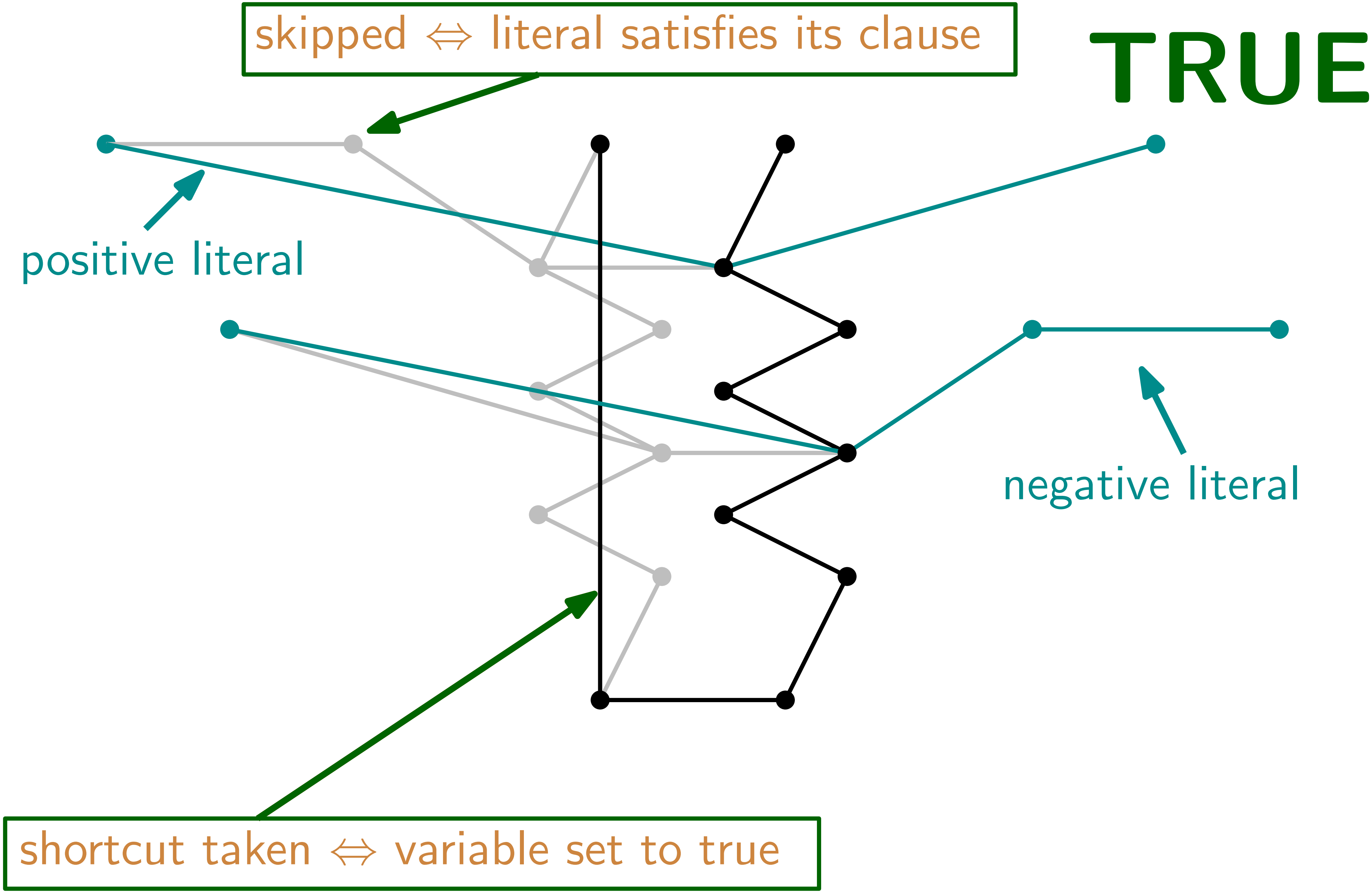
Variable-Gadget



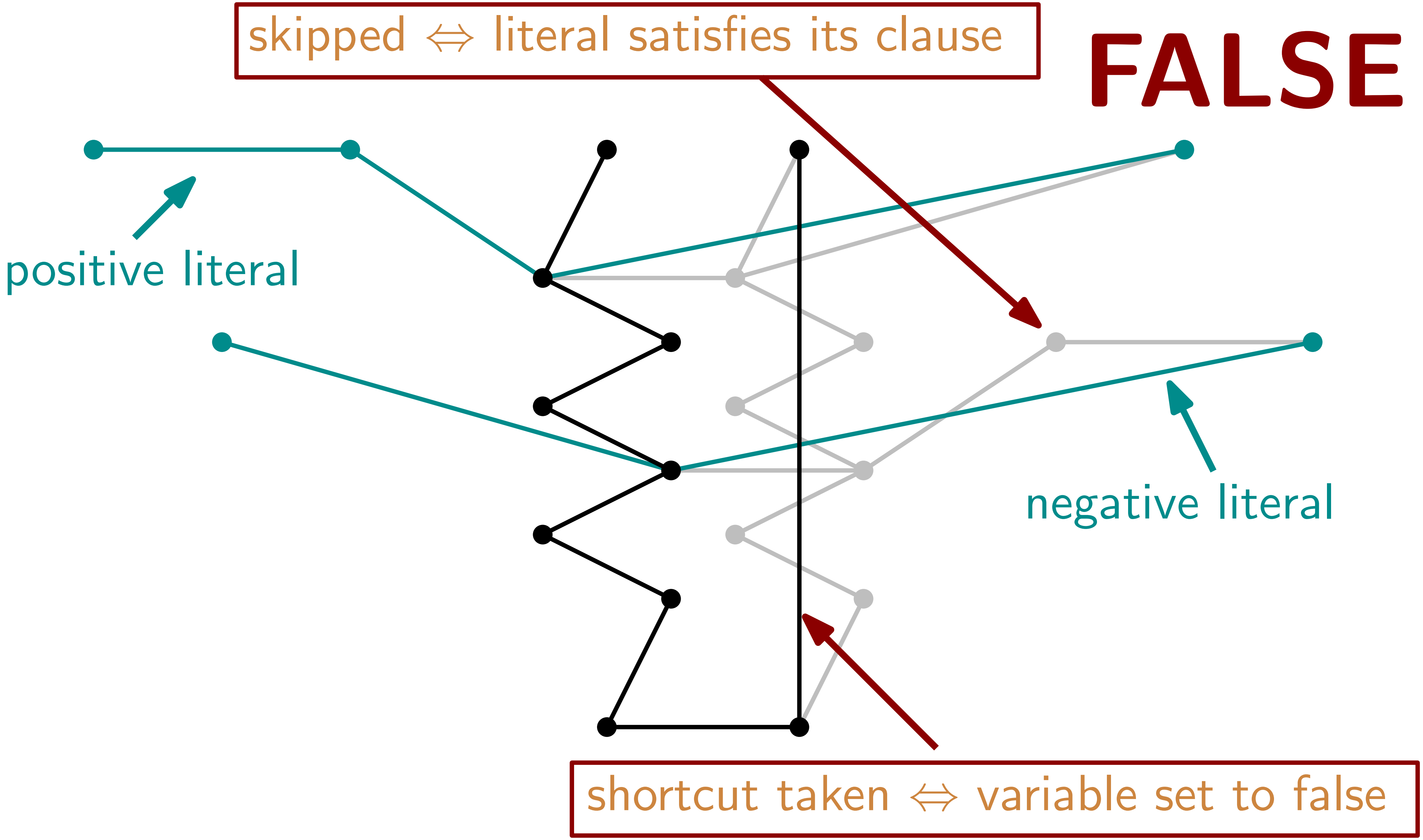
Variable-Gadget



Variable-Gadget

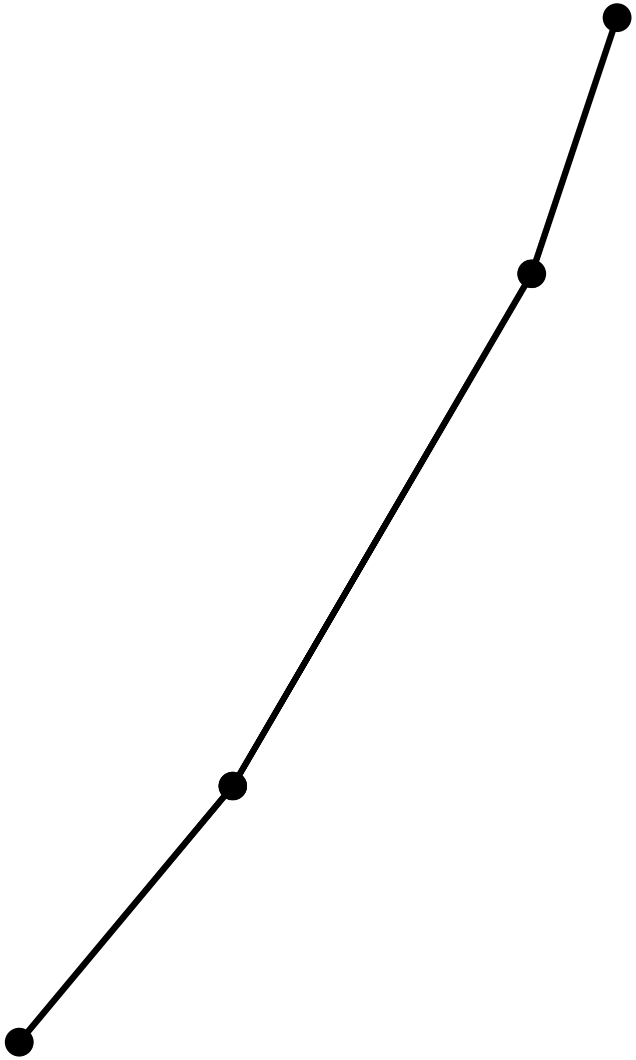


Variable-Gadget

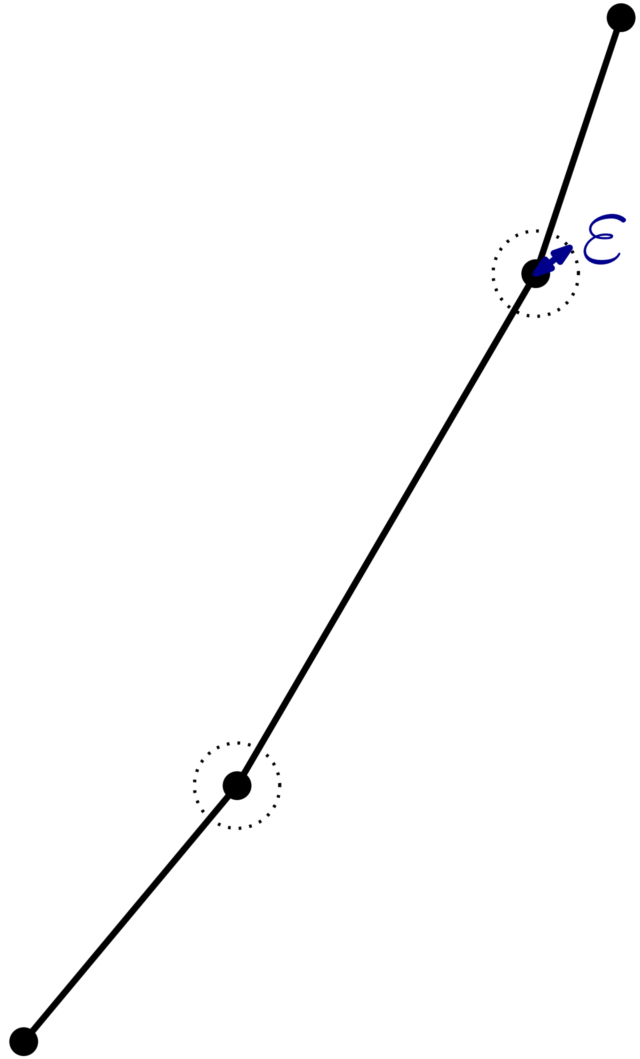


Clause-Gadget

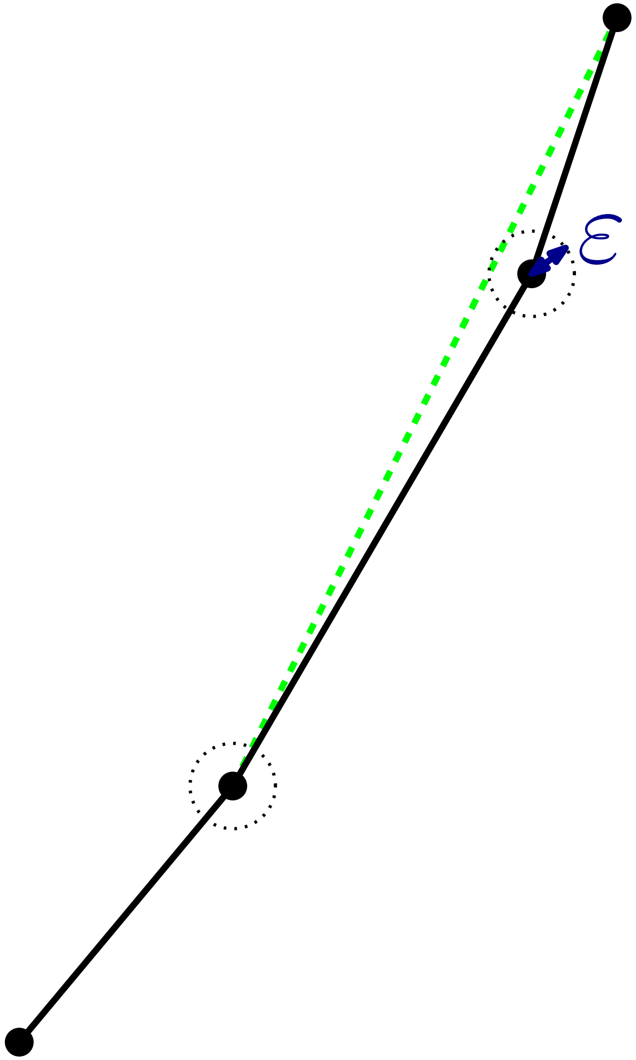
Clause-Gadget



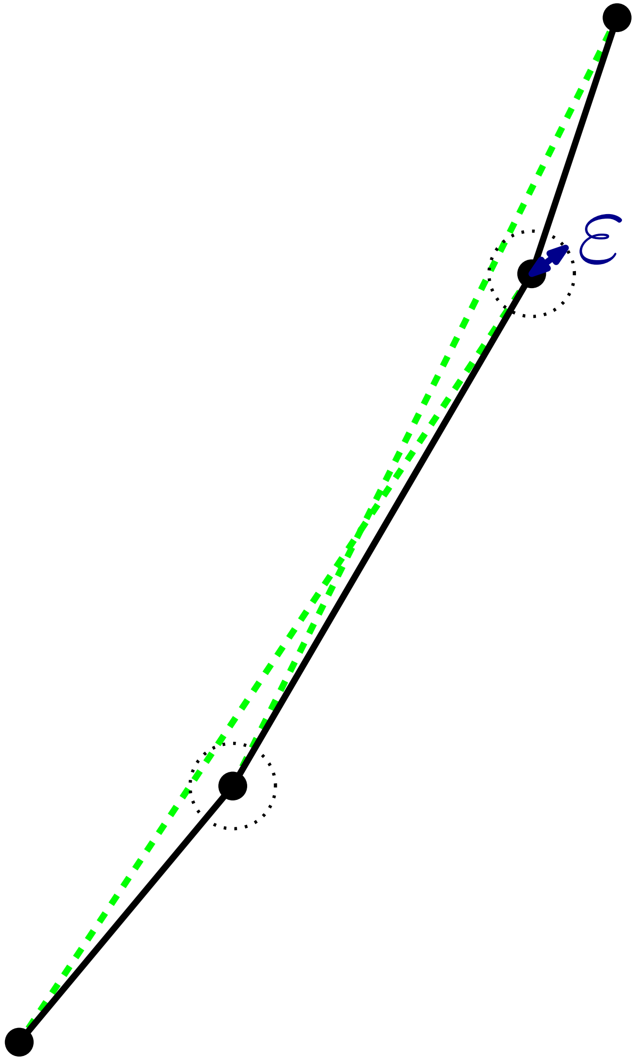
Clause-Gadget



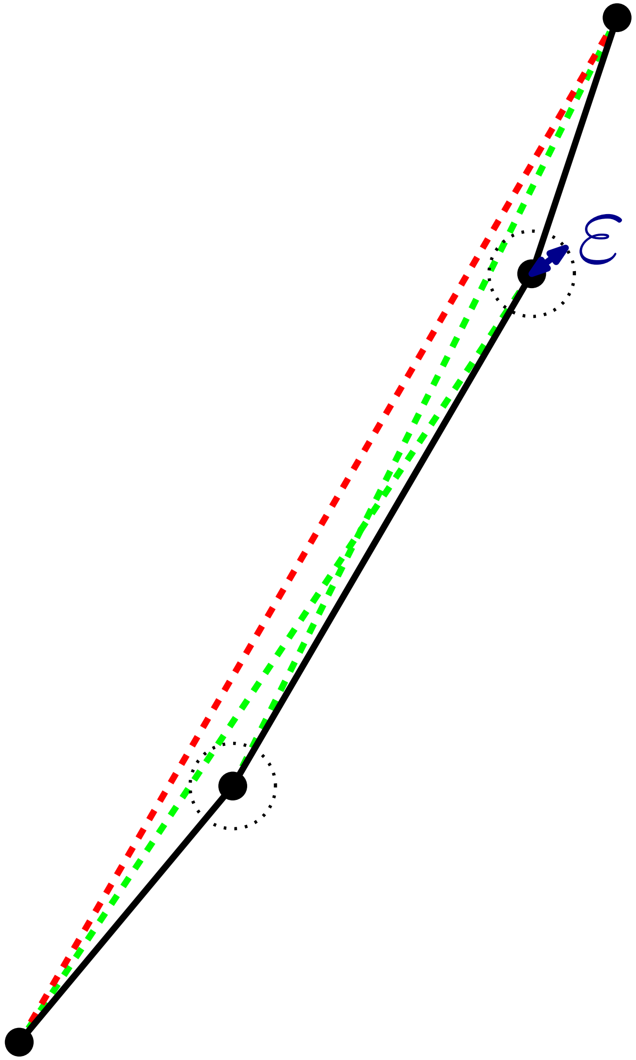
Clause-Gadget



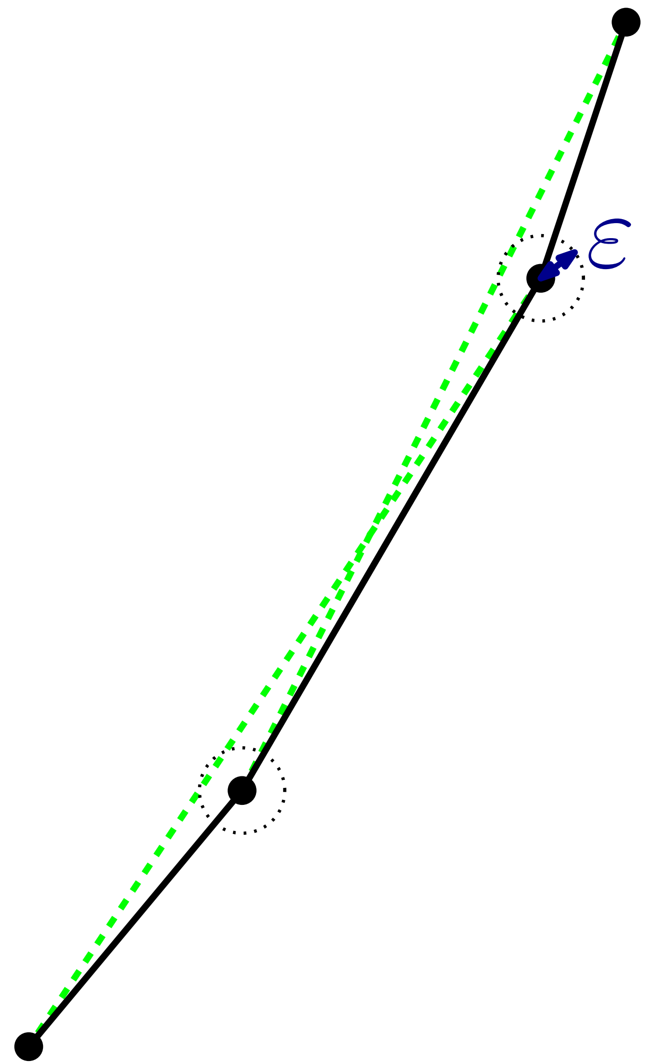
Clause-Gadget



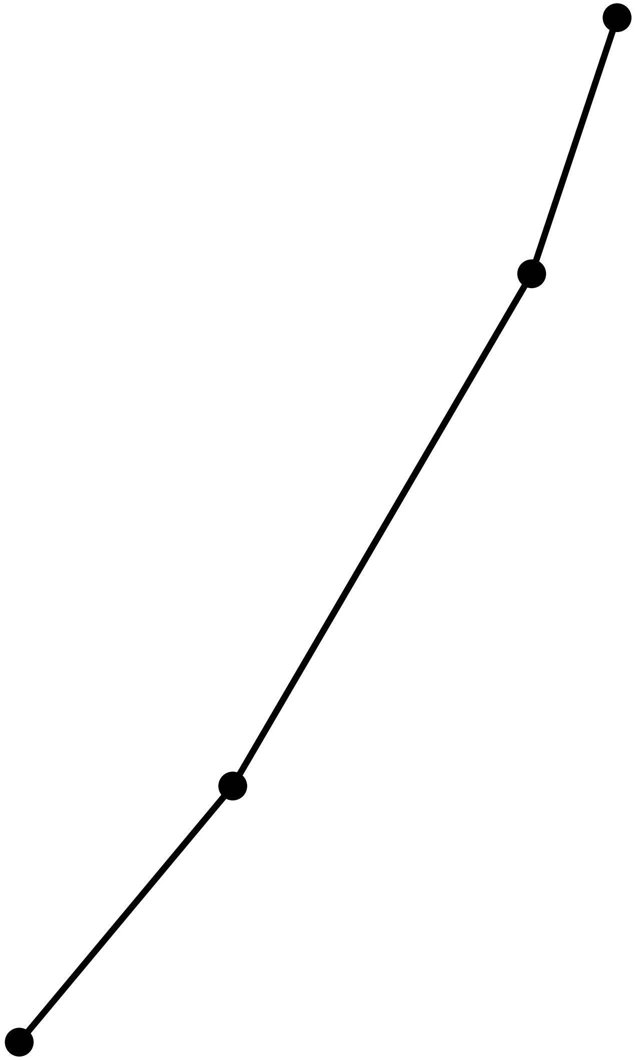
Clause-Gadget



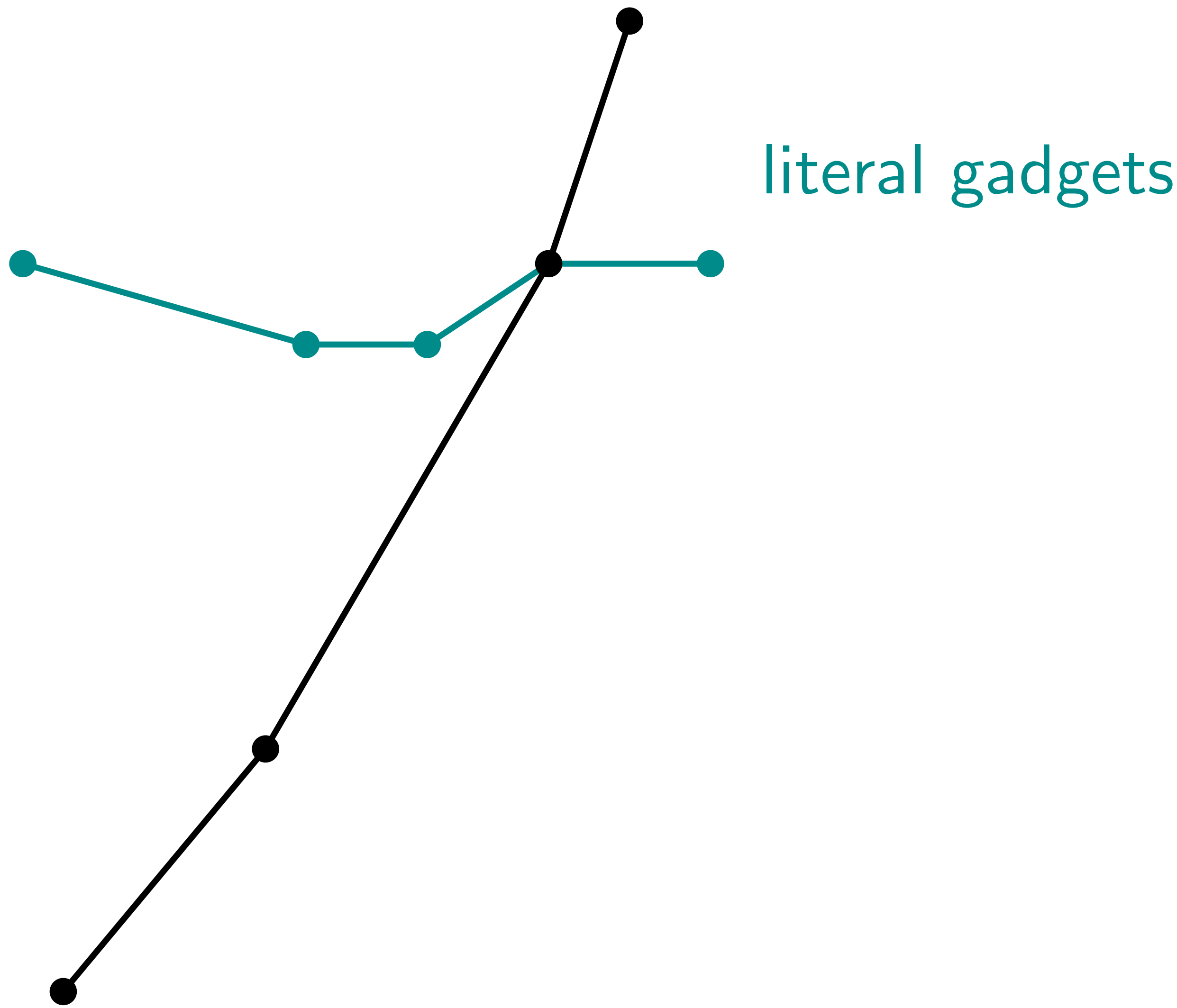
Clause-Gadget



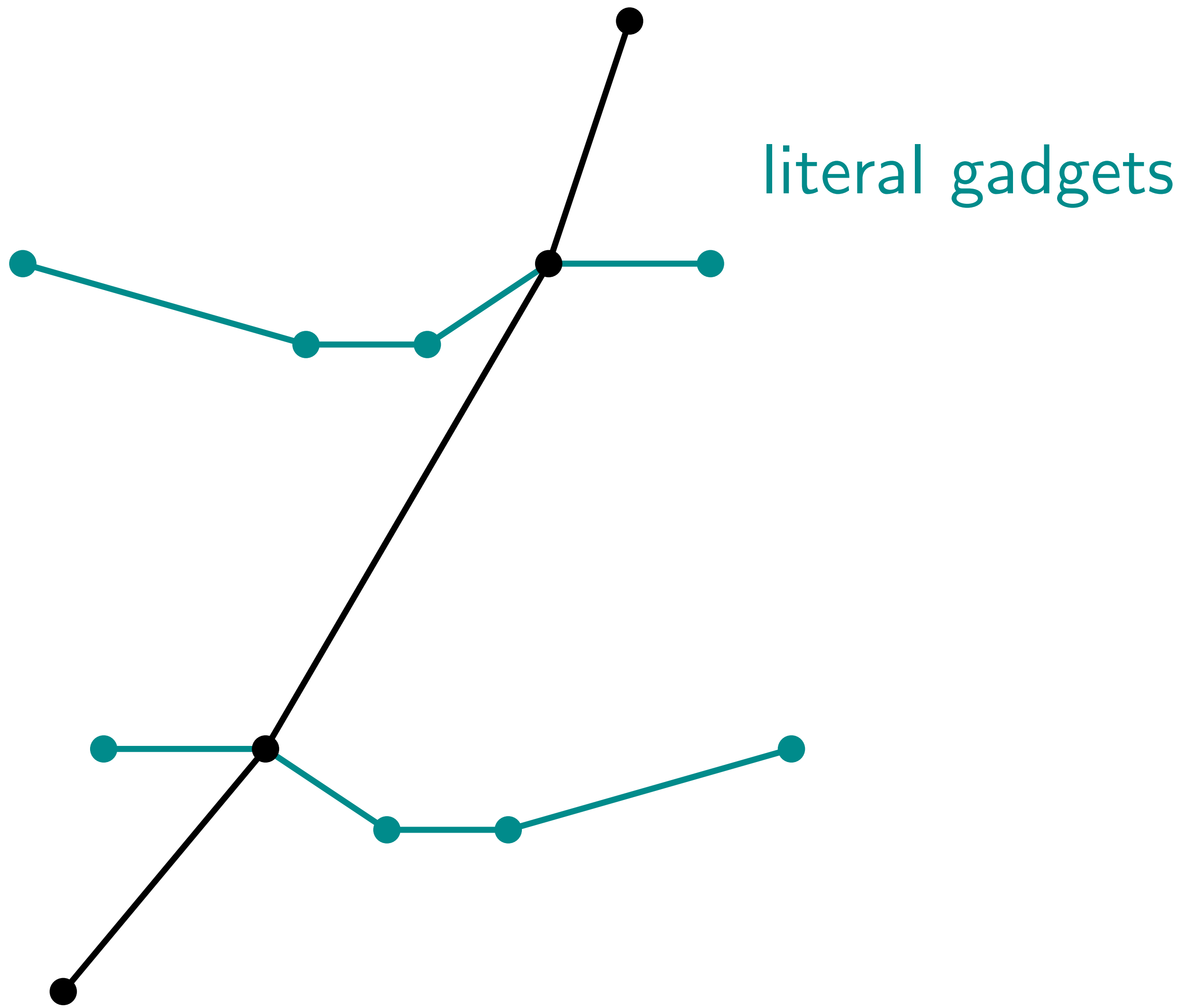
Clause-Gadget



Clause-Gadget

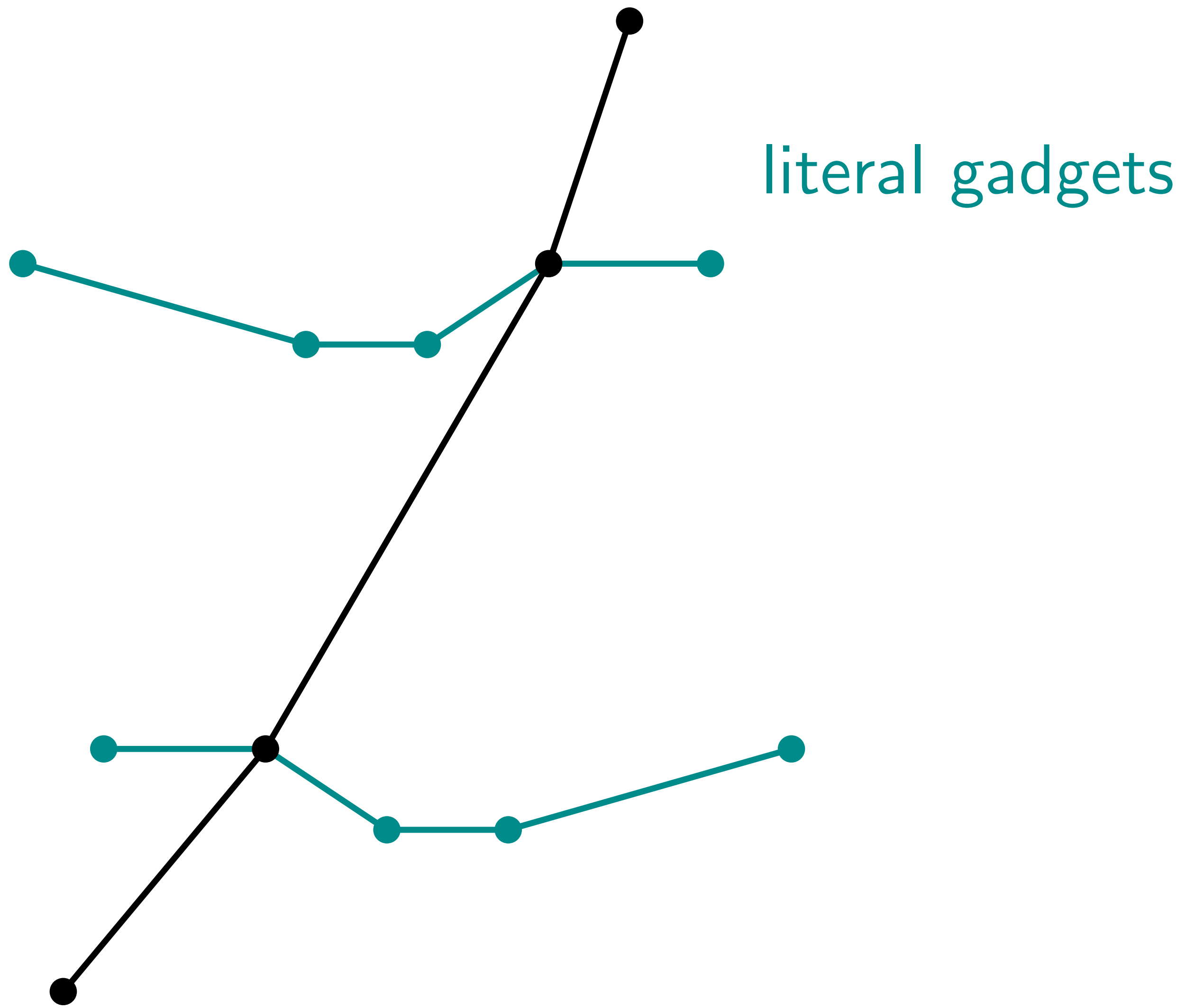


Clause-Gadget



Clause-Gadget

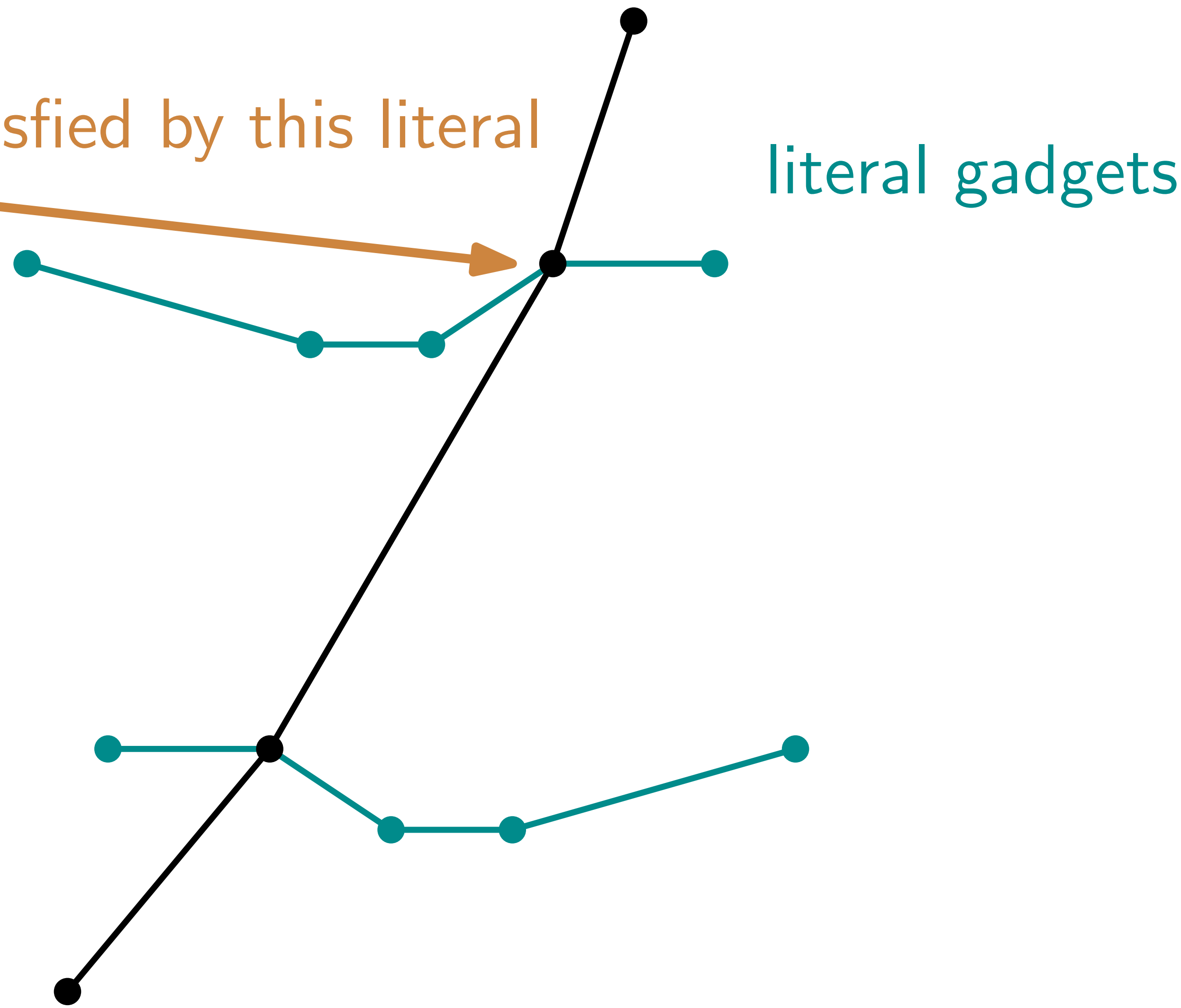
Interpretation:



Clause-Gadget

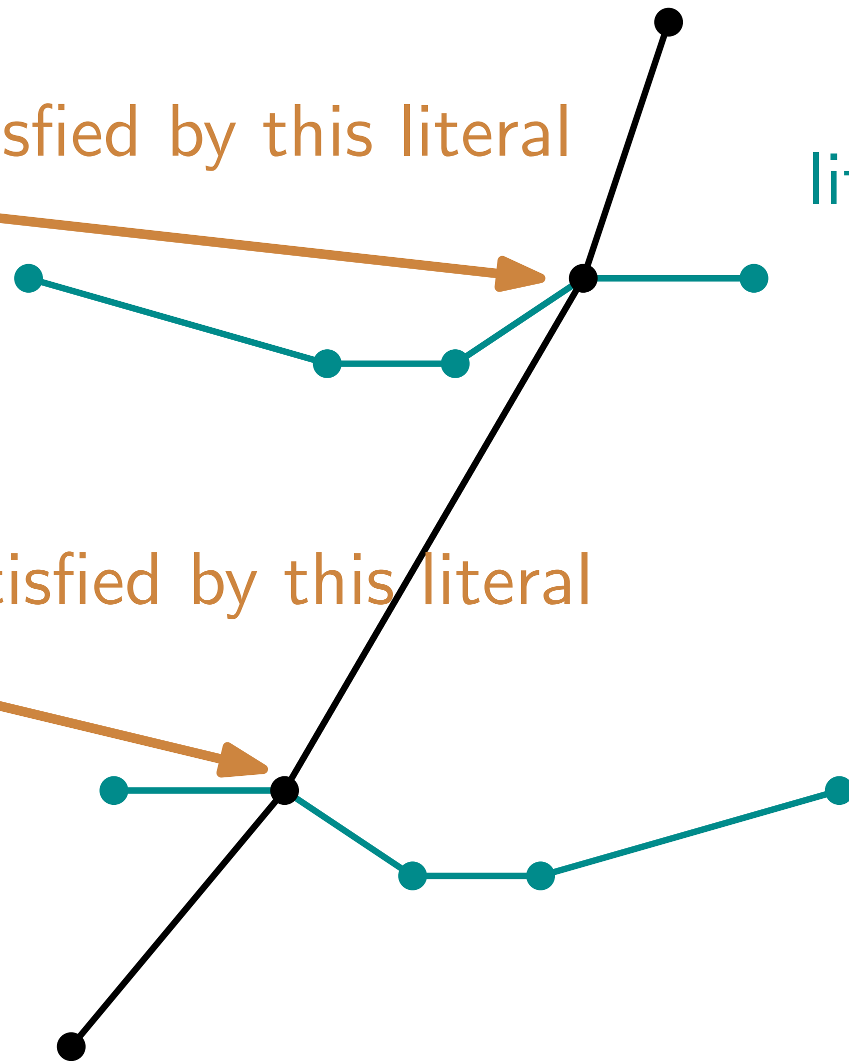
Interpretation:

skipped $\Leftrightarrow$ clause satisfied by this literal



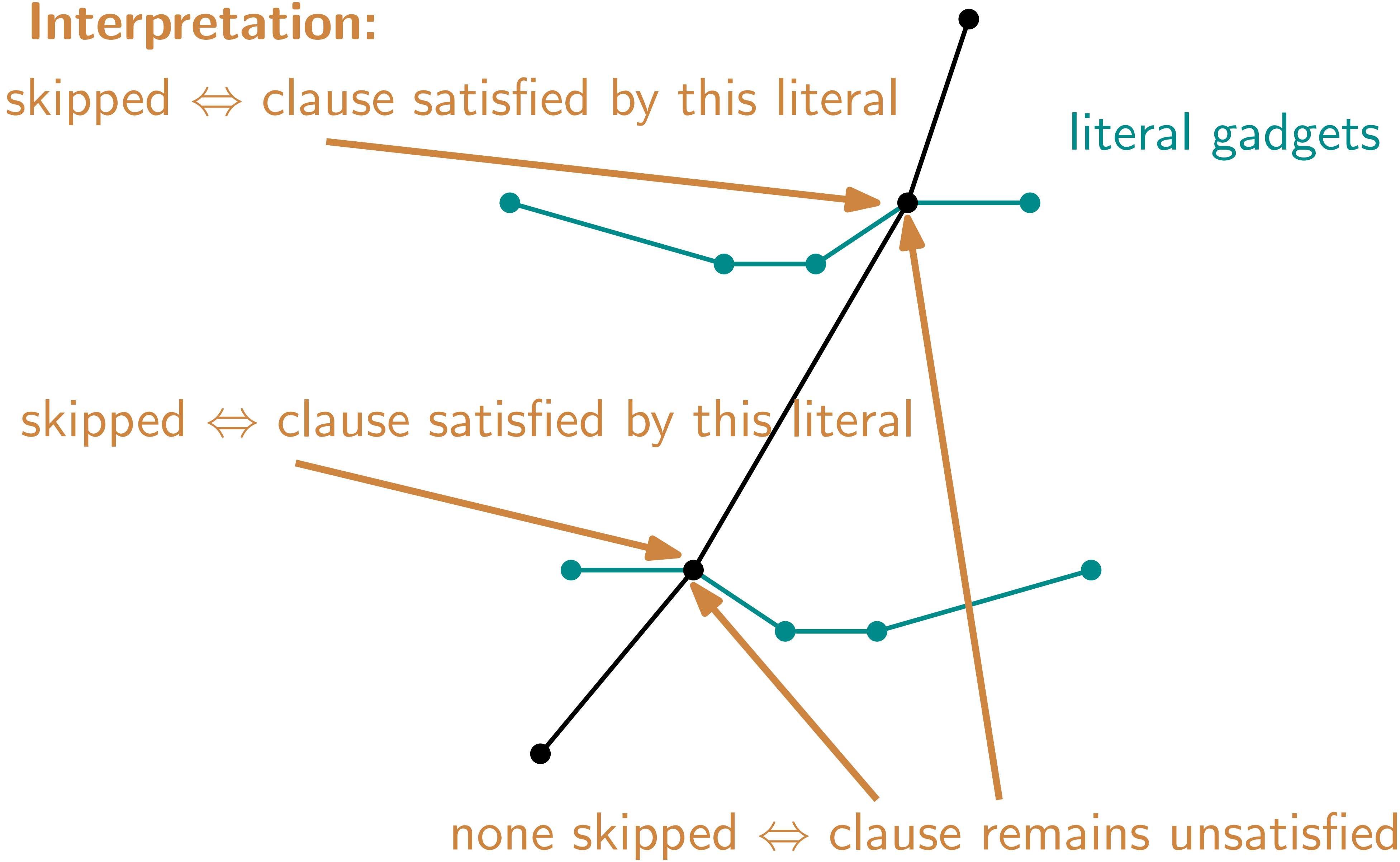
Clause-Gadget

Interpretation:



Clause-Gadget

Interpretation:



Full Example

Full Example

$(x_1 \lor x_2) \land$
$(\neg x_1 \lor x_3) \land$
$(\neg x_3)$

Full Example

$(x_1 \vee x_2) \wedge$

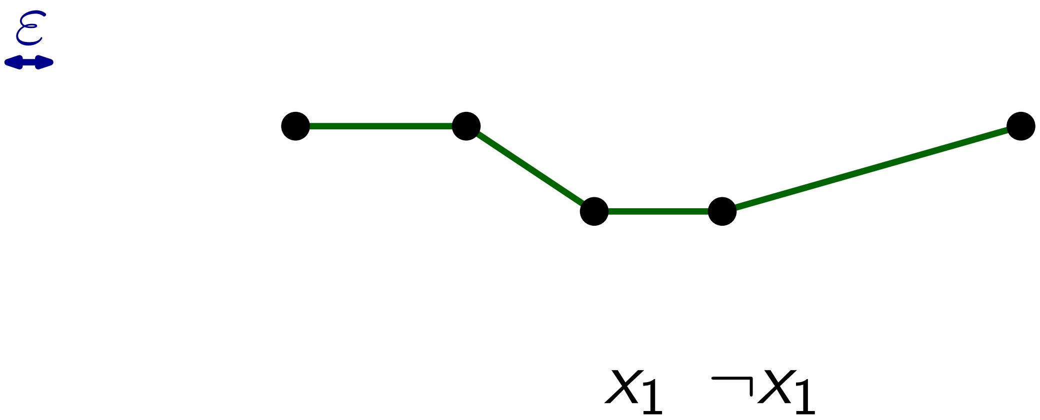
$(\neg x_1 \vee x_3) \wedge$

$(\neg x_3)$

$\xleftrightarrow{\varepsilon}$

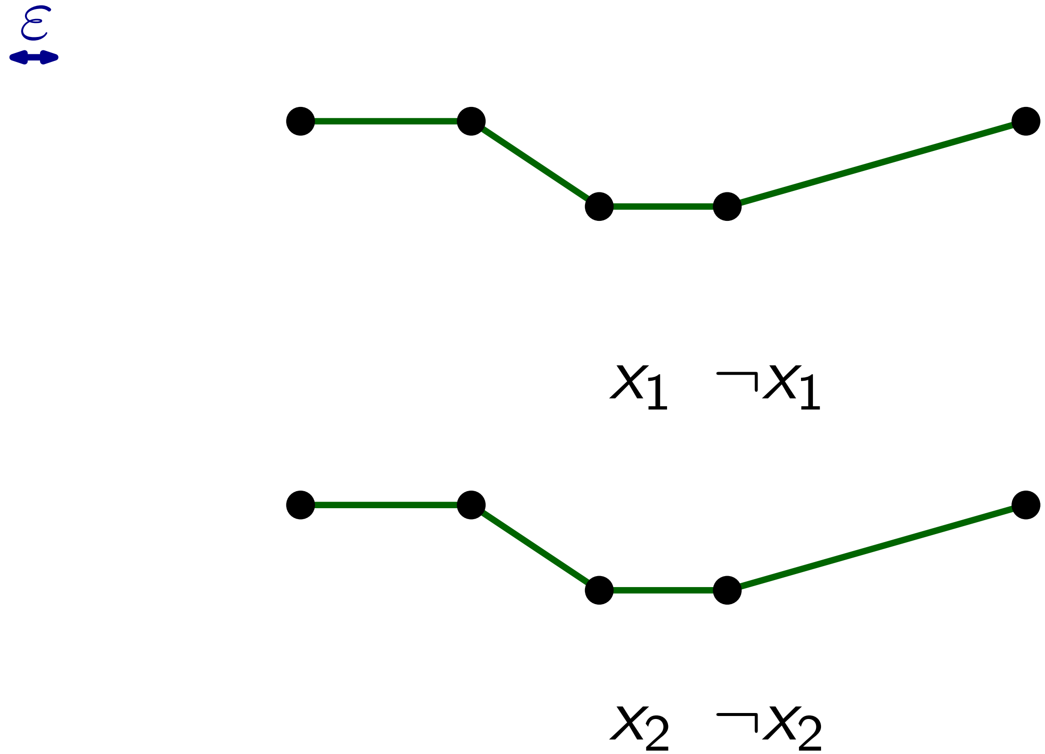
Full Example

$(\boxed{x_1} \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



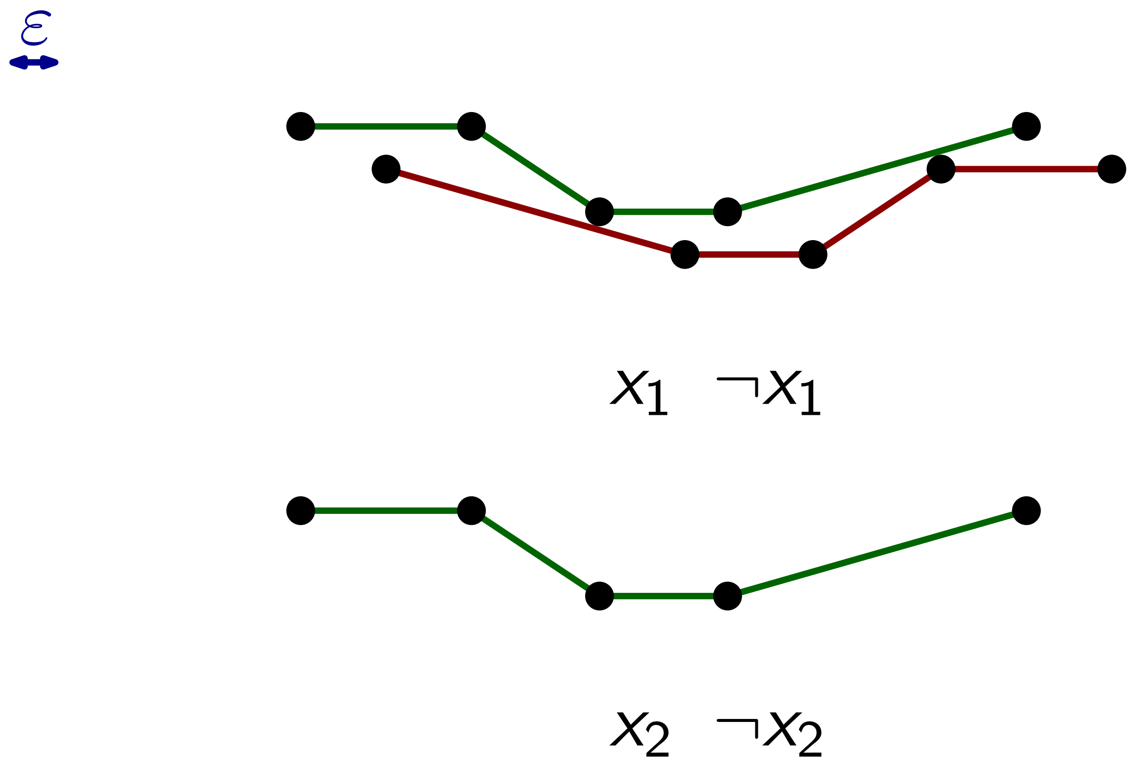
Full Example

$(x_1 \vee \boxed{x_2}) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



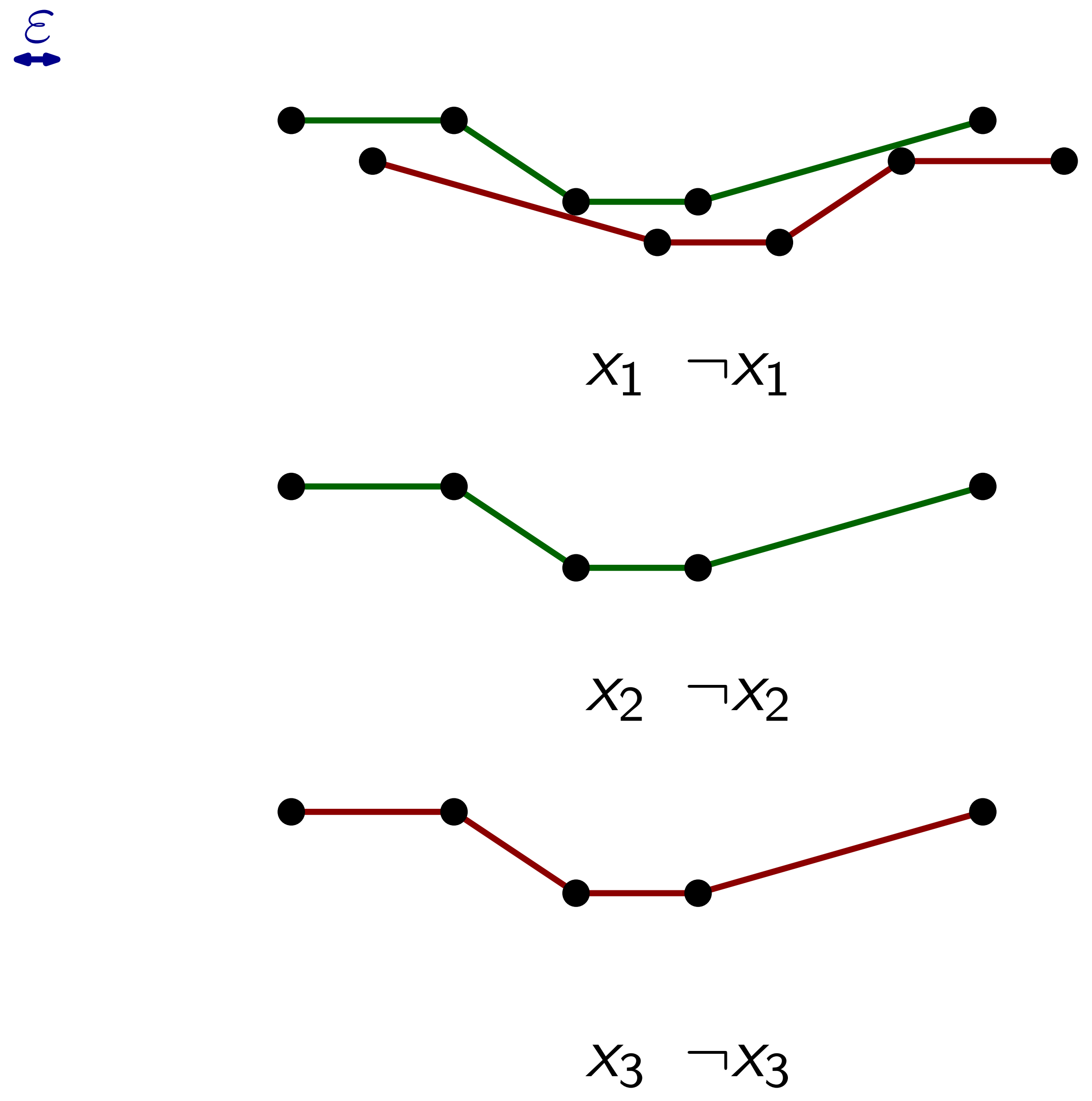
Full Example

$(x_1 \lor x_2)\land$
$(\boxed{\neg x_1} \lor x_3)\land$
$(\neg x_3)$



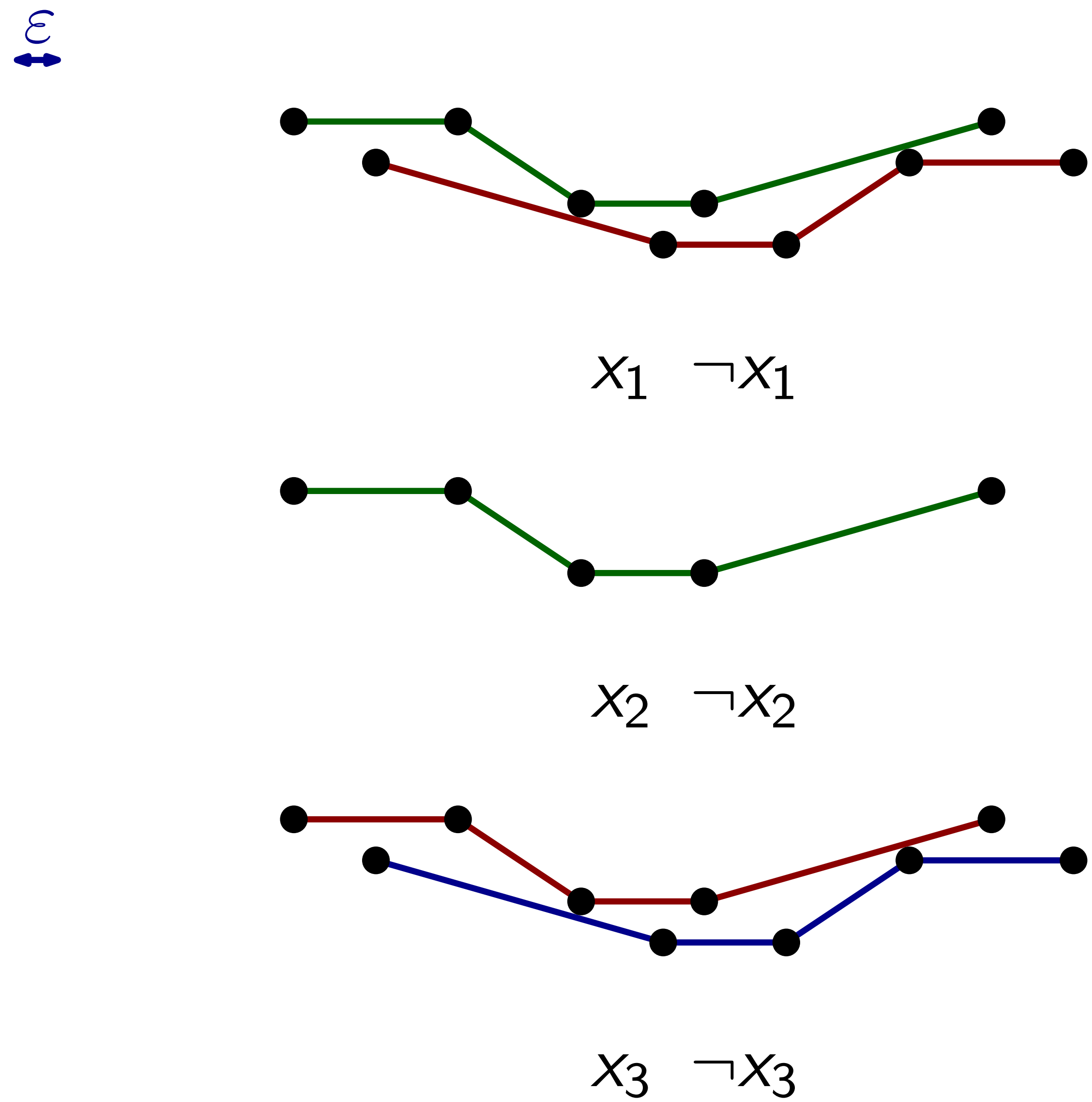
Full Example

$(x_1 \lor x_2) \land$
$(\neg x_1 \lor \boxed{x_3}) \land$
$(\neg x_3)$



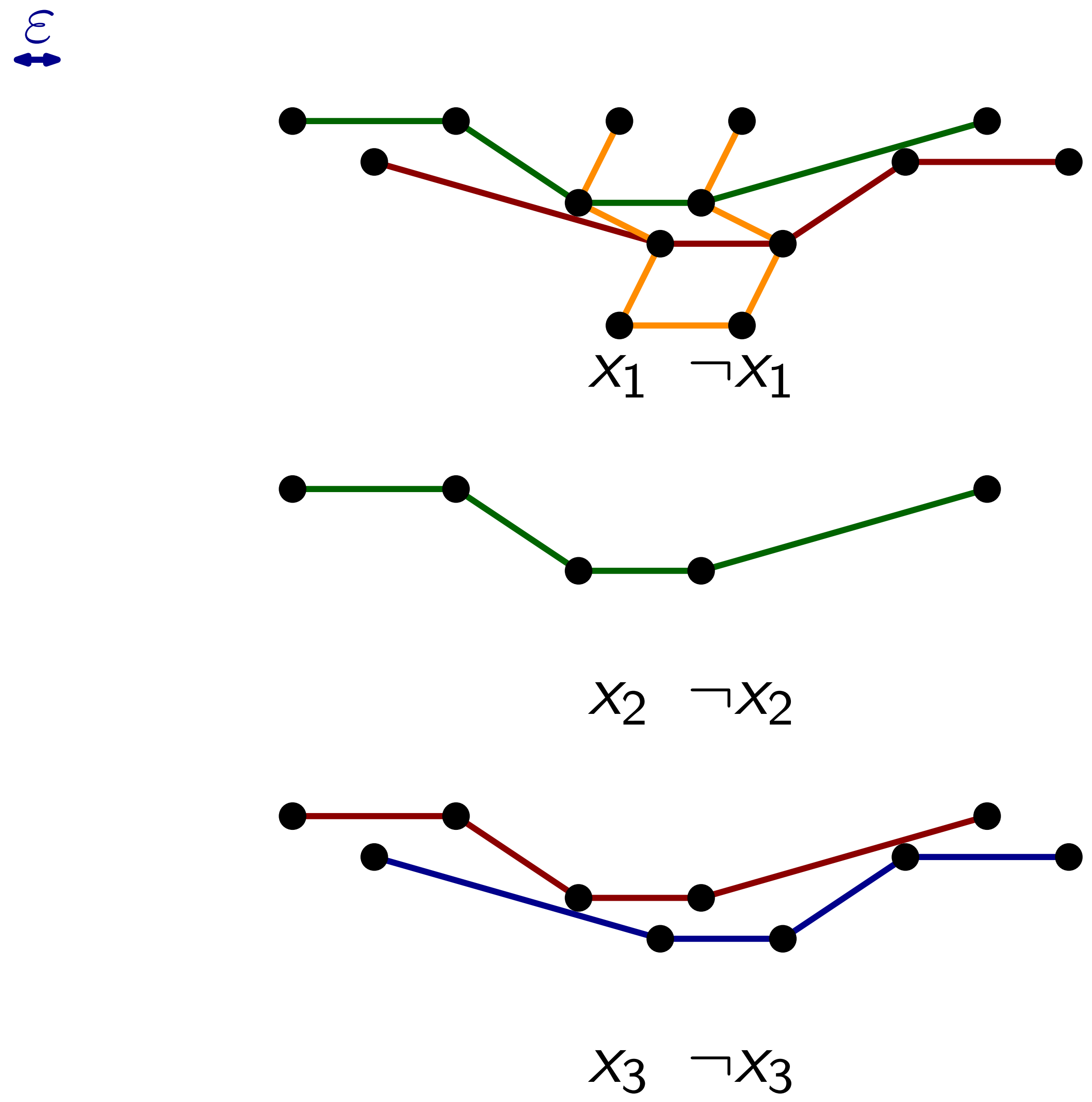
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\boxed{\neg x_3})$



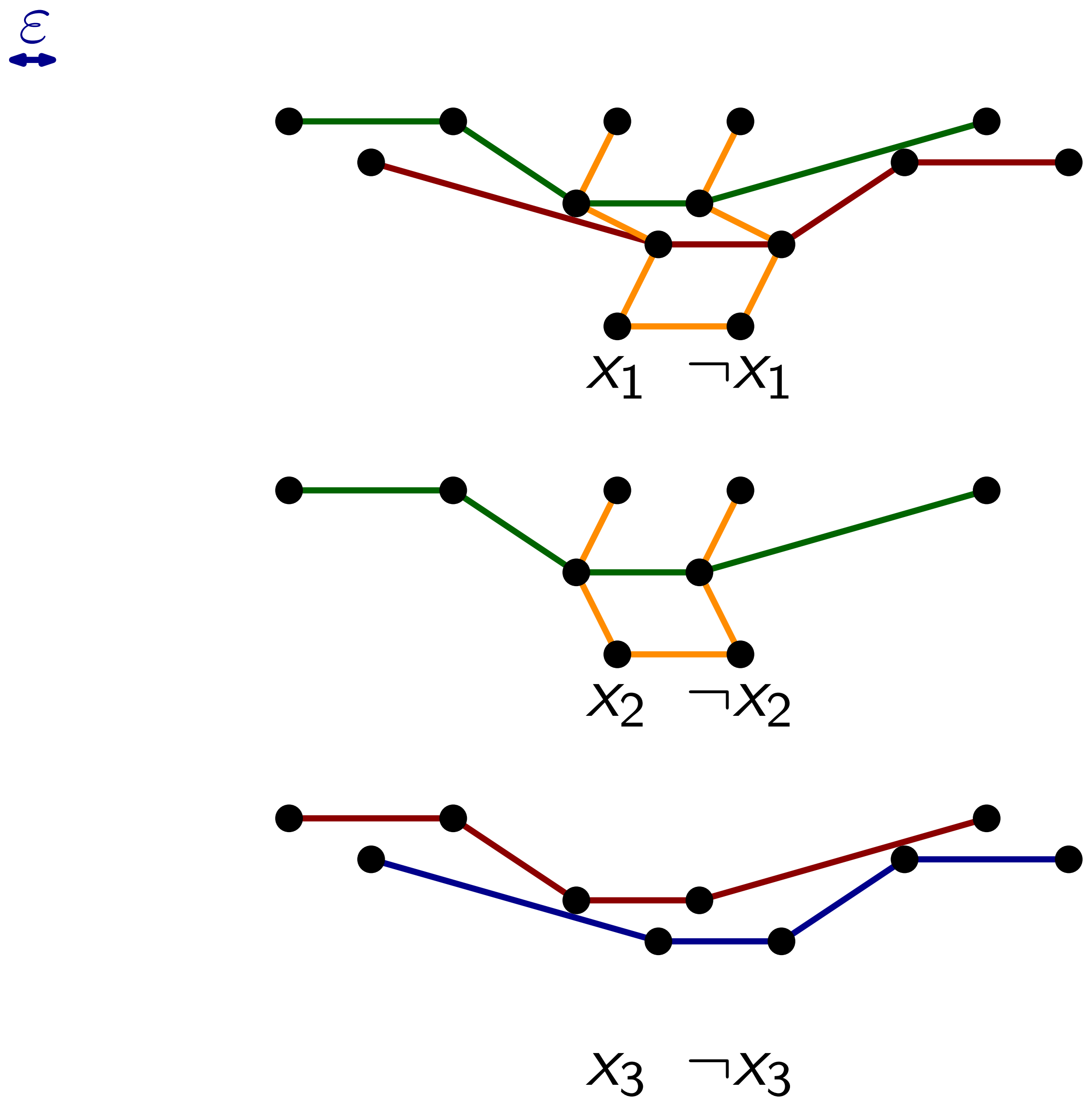
Full Example

$(\boxed{x_1} \vee x_2) \wedge$
$(\boxed{\neg x_1} \vee x_3) \wedge$
$(\neg x_3)$



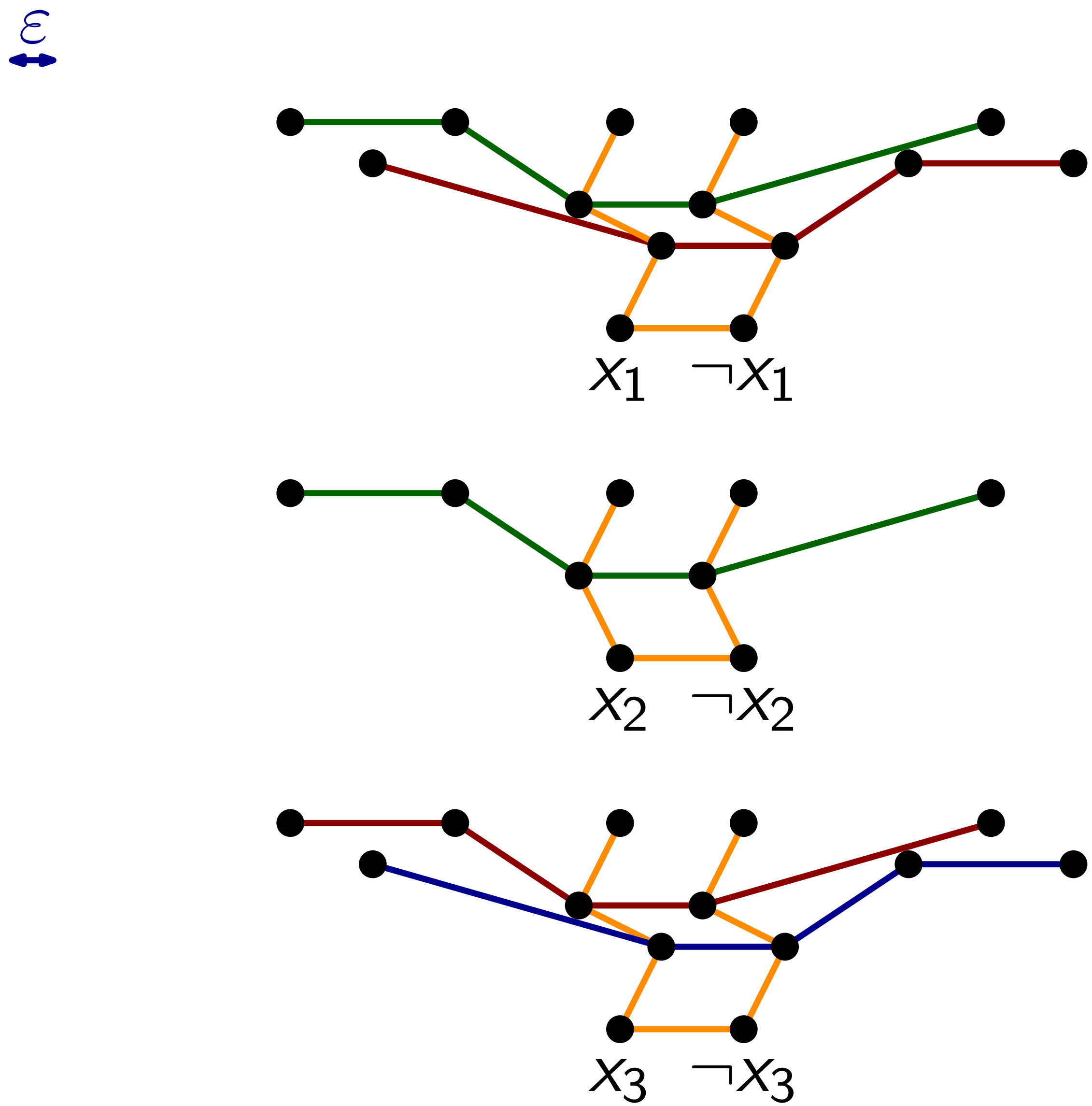
Full Example

$(x_1 \vee \boxed{x_2})\wedge$
$(\neg x_1 \vee x_3)\wedge$
$(\neg x_3)$



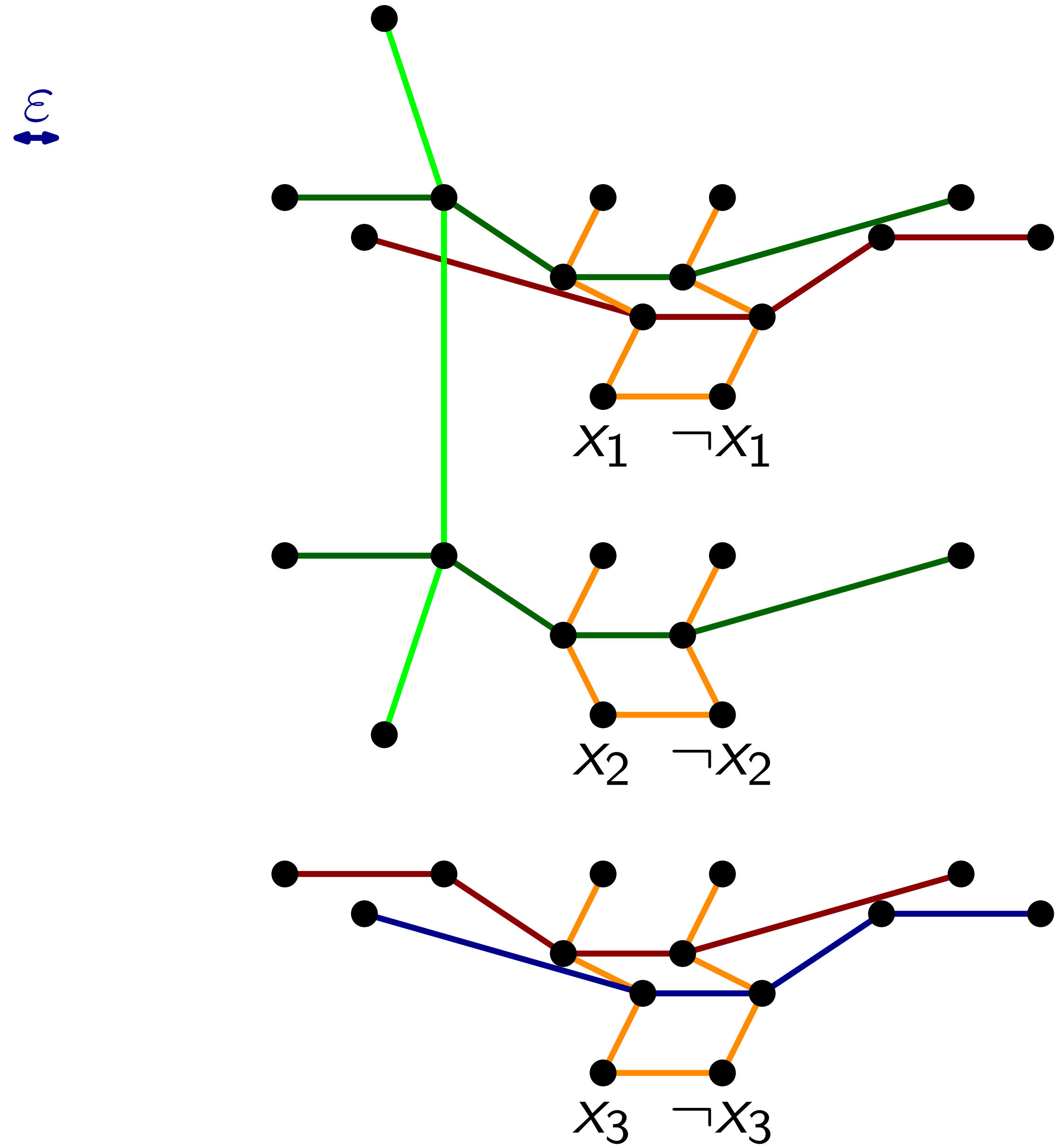
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee \boxed{x_3}) \wedge$
$(\boxed{\neg x_3})$



Full Example

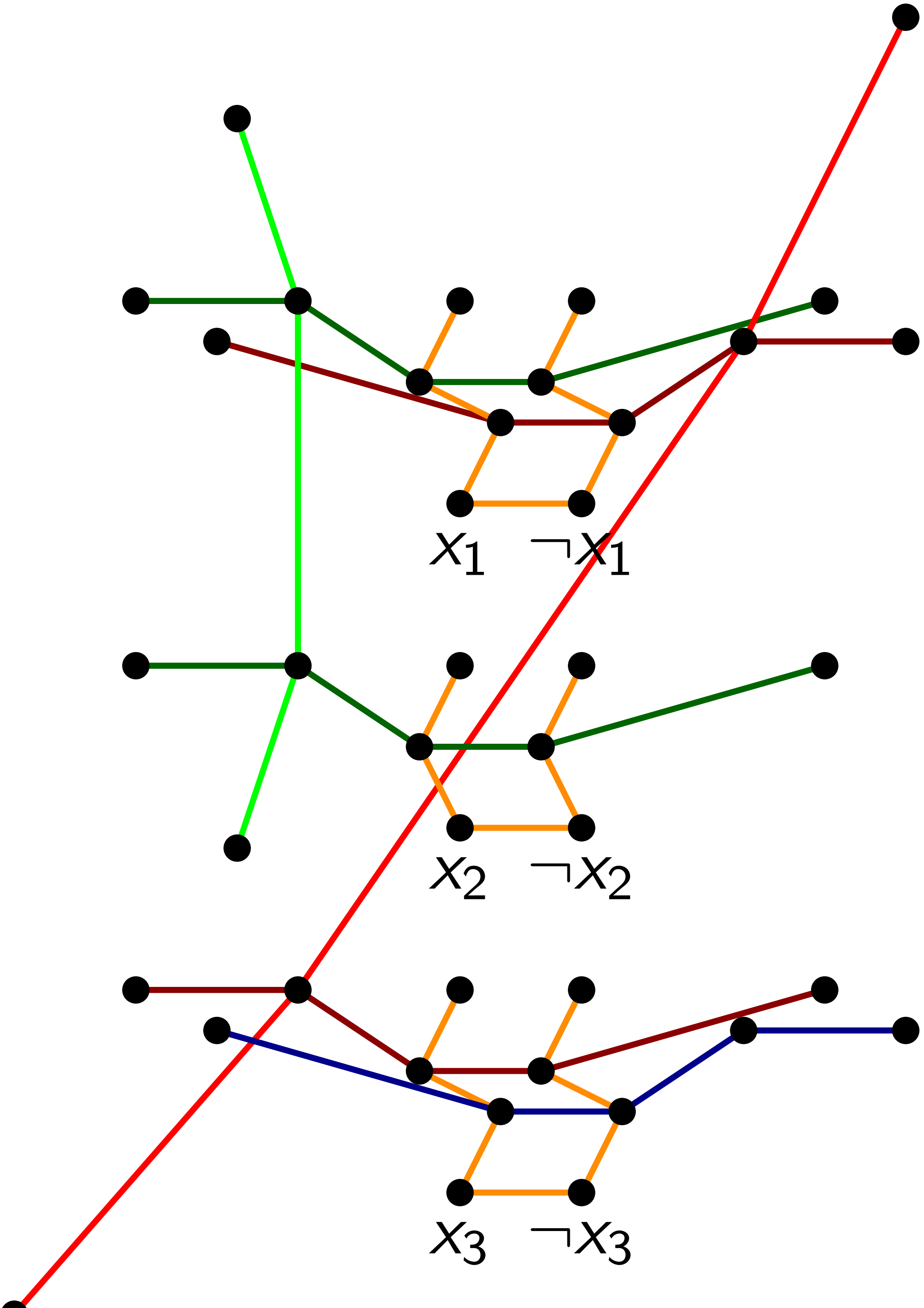
$\boxed{(x_1 \vee x_2)}\wedge$
$(\neg x_1 \vee x_3)\wedge$
$(\neg x_3)$



Full Example

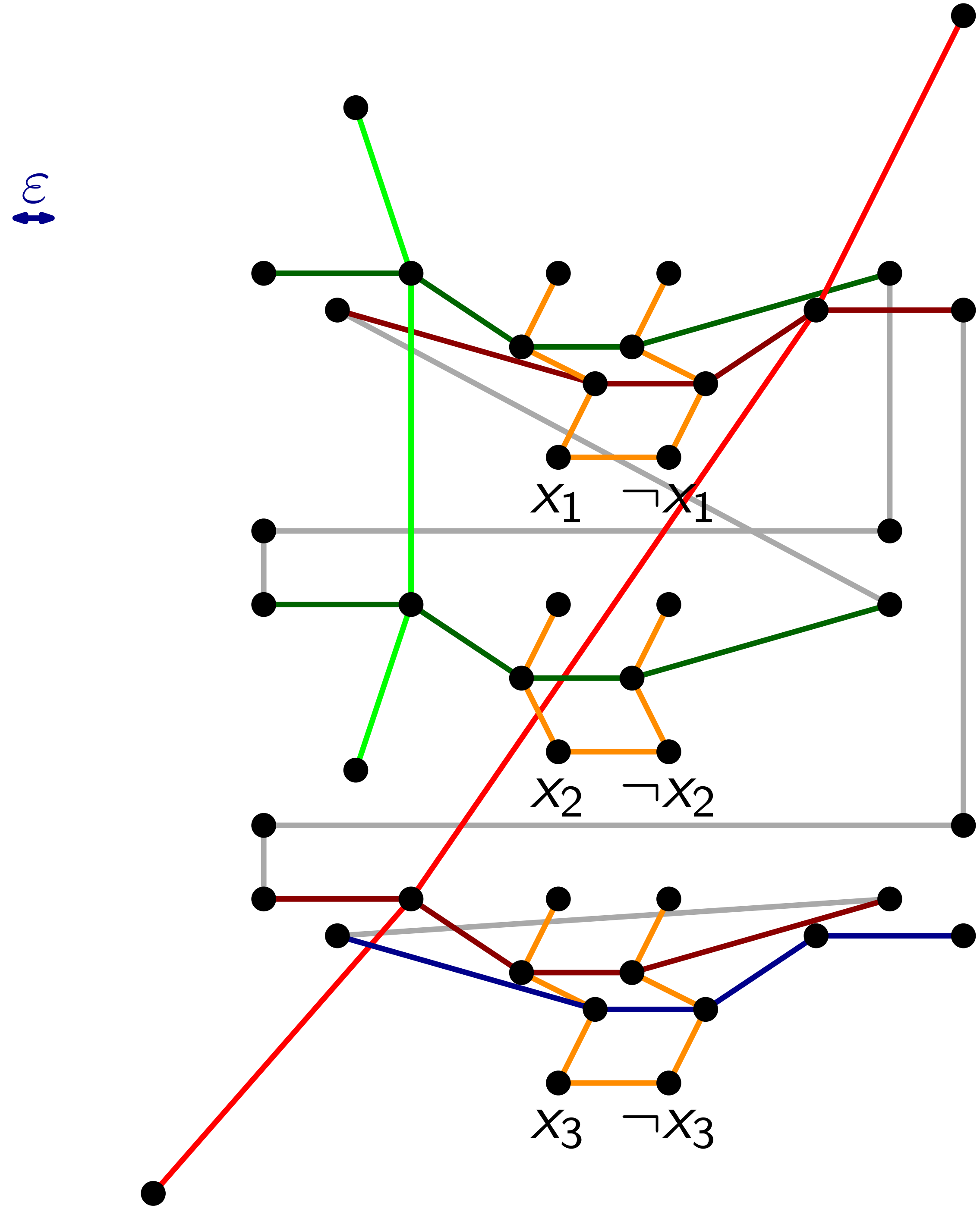
$(x_1 \lor x_2) \land$
$\boxed{(\neg x_1 \lor x_3)} \land$
$(\neg x_3)$

ε



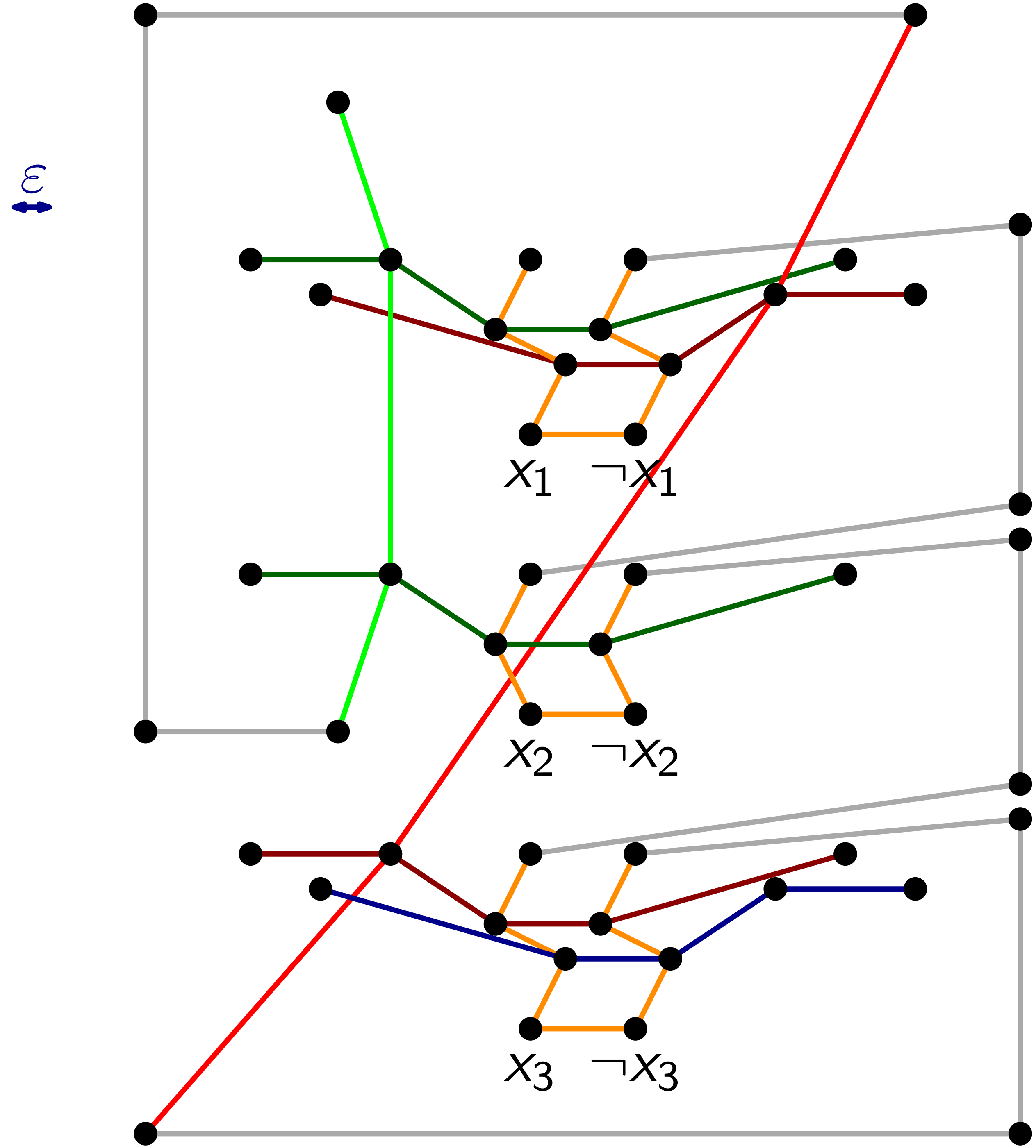
Full Example

$\big(x_1 \lor x_2\big) \land$
$\big(\neg x_1 \lor x_3\big) \land$
$\big(\neg x_3\big)$



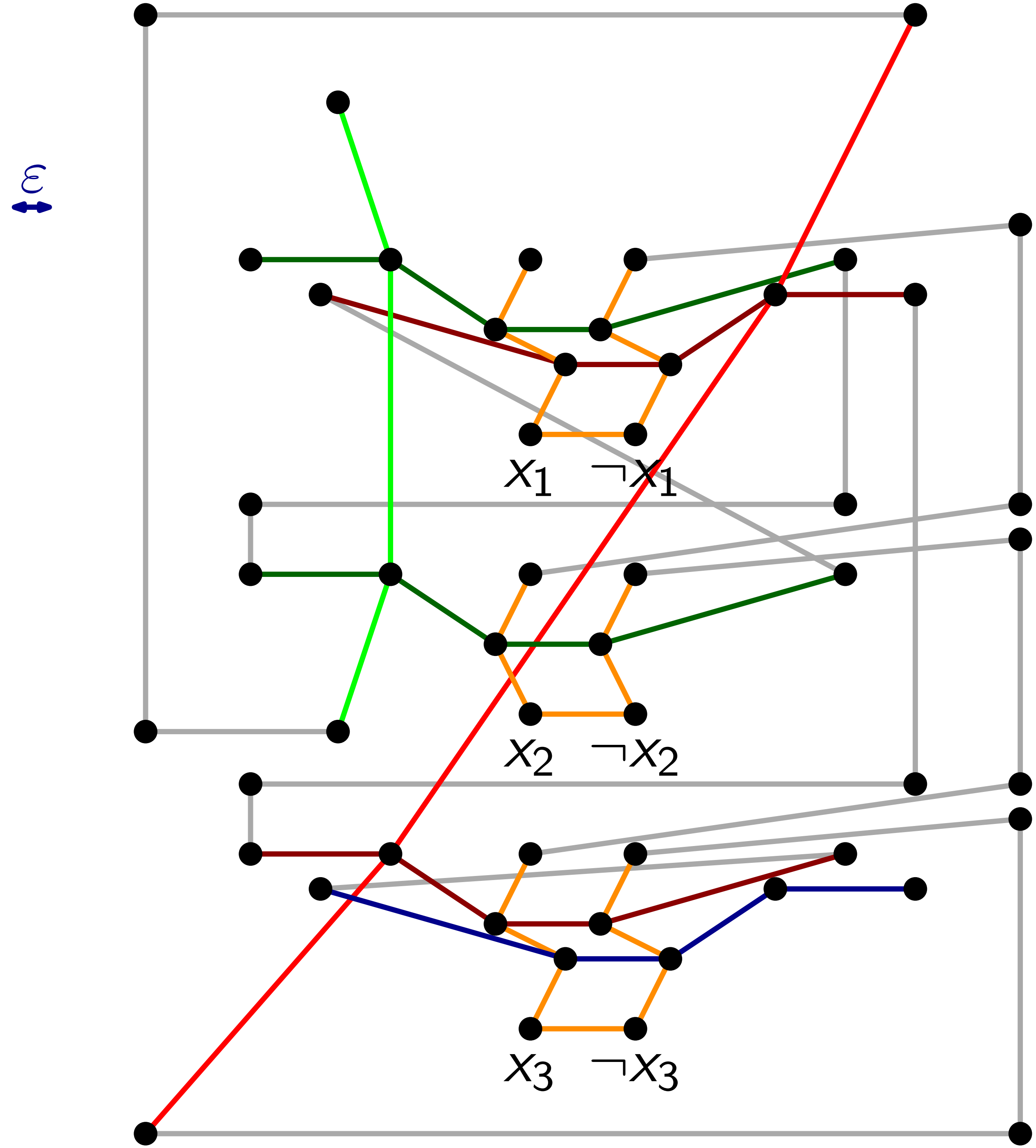
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



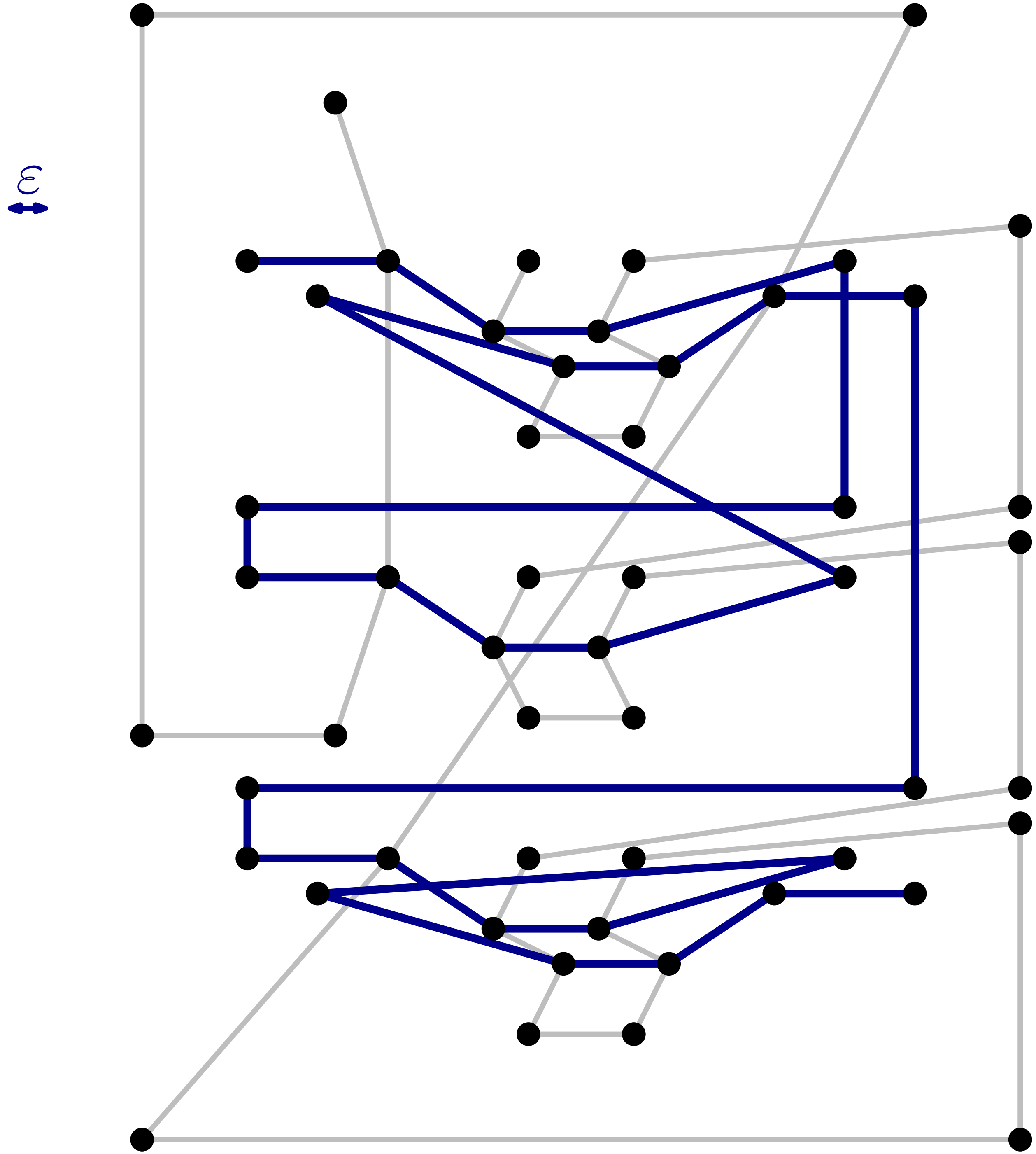
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



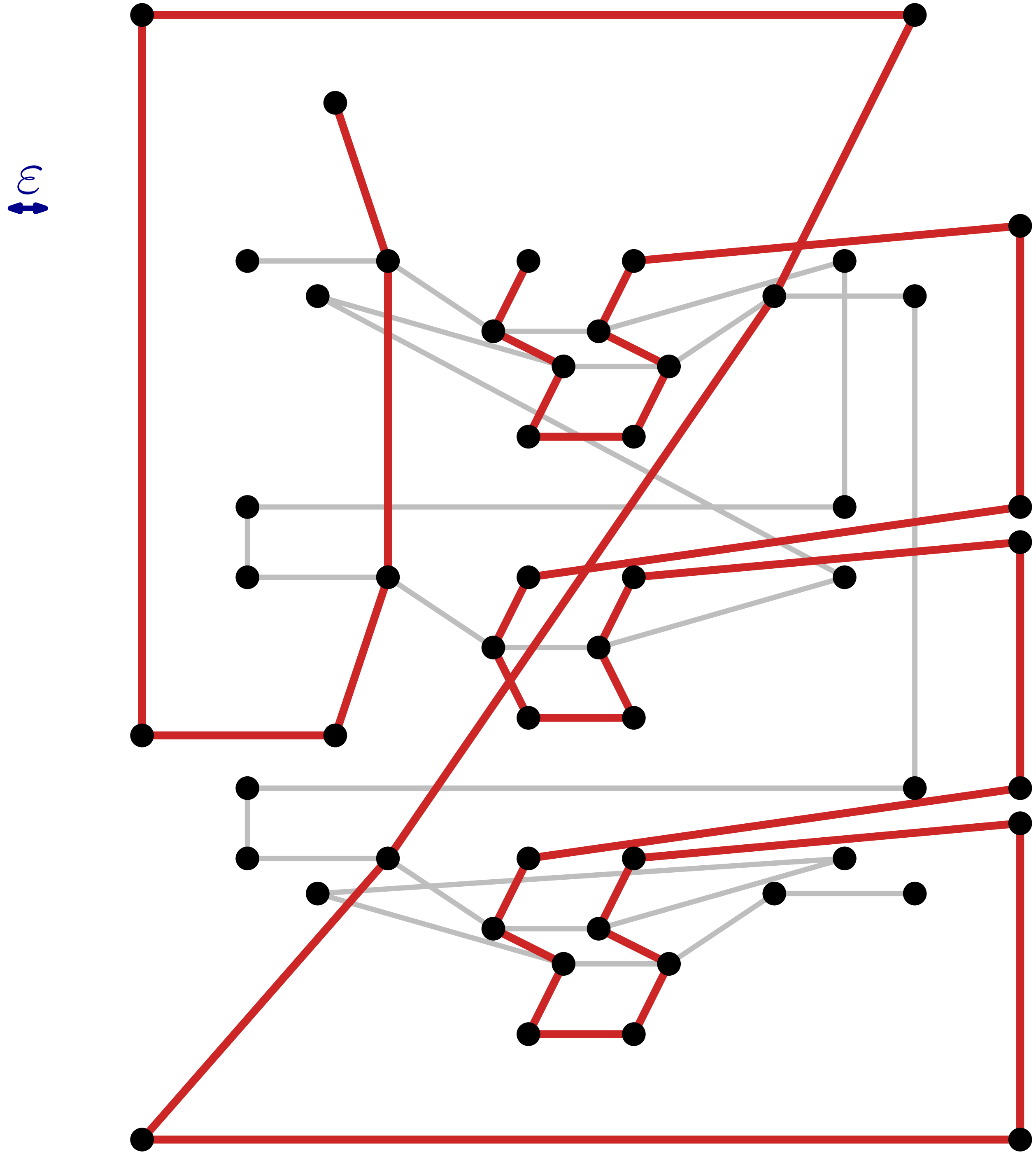
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



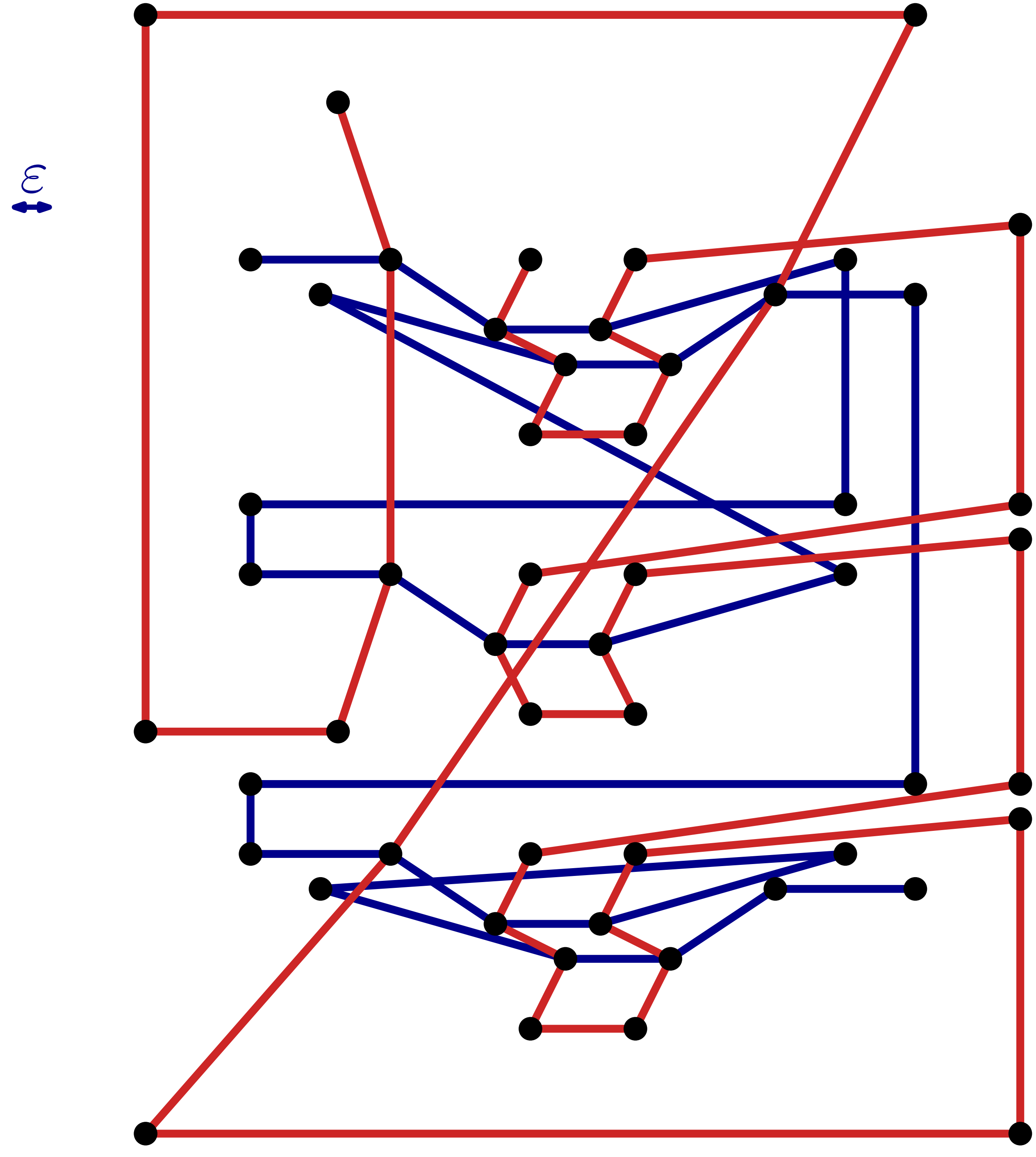
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



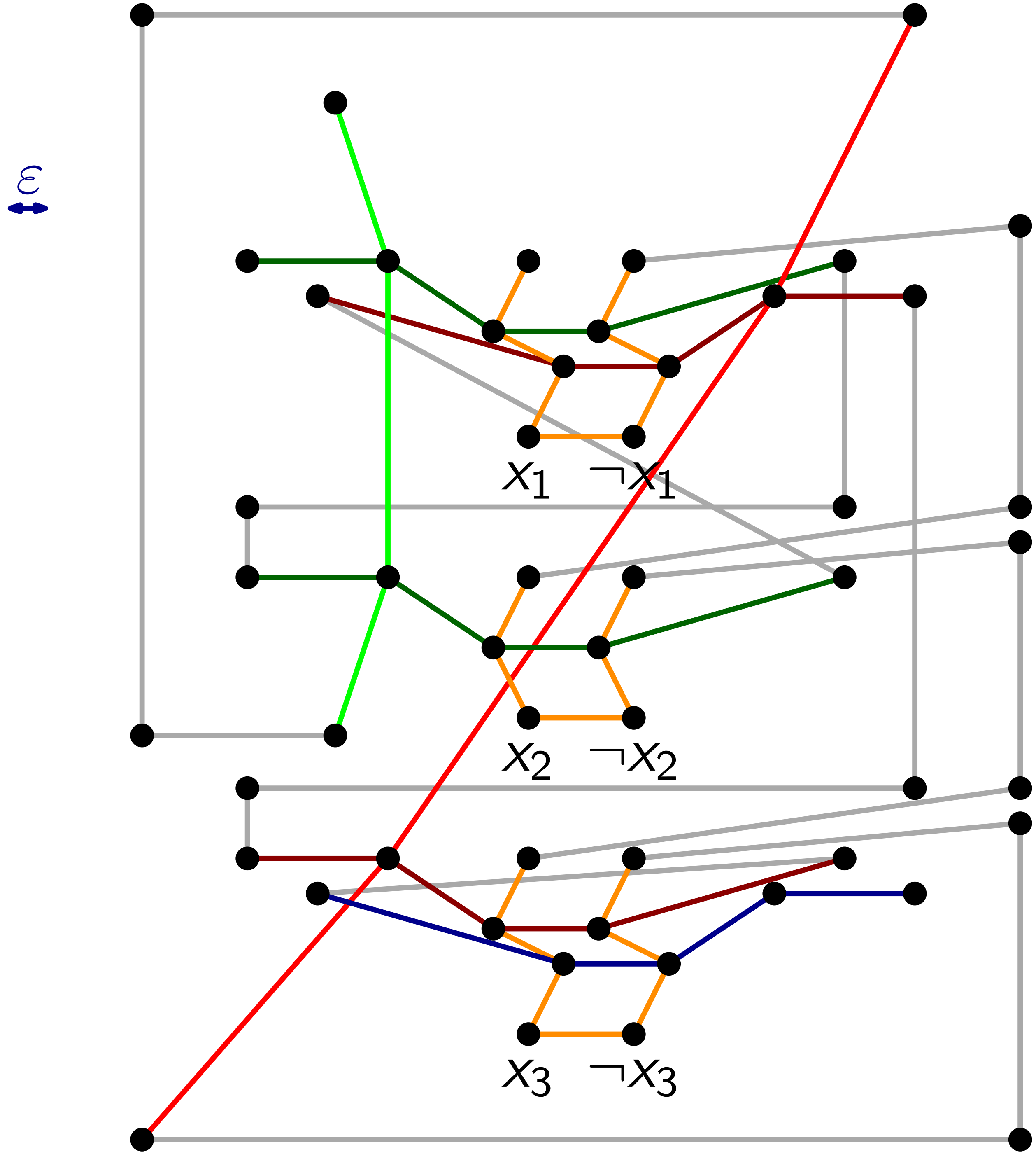
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



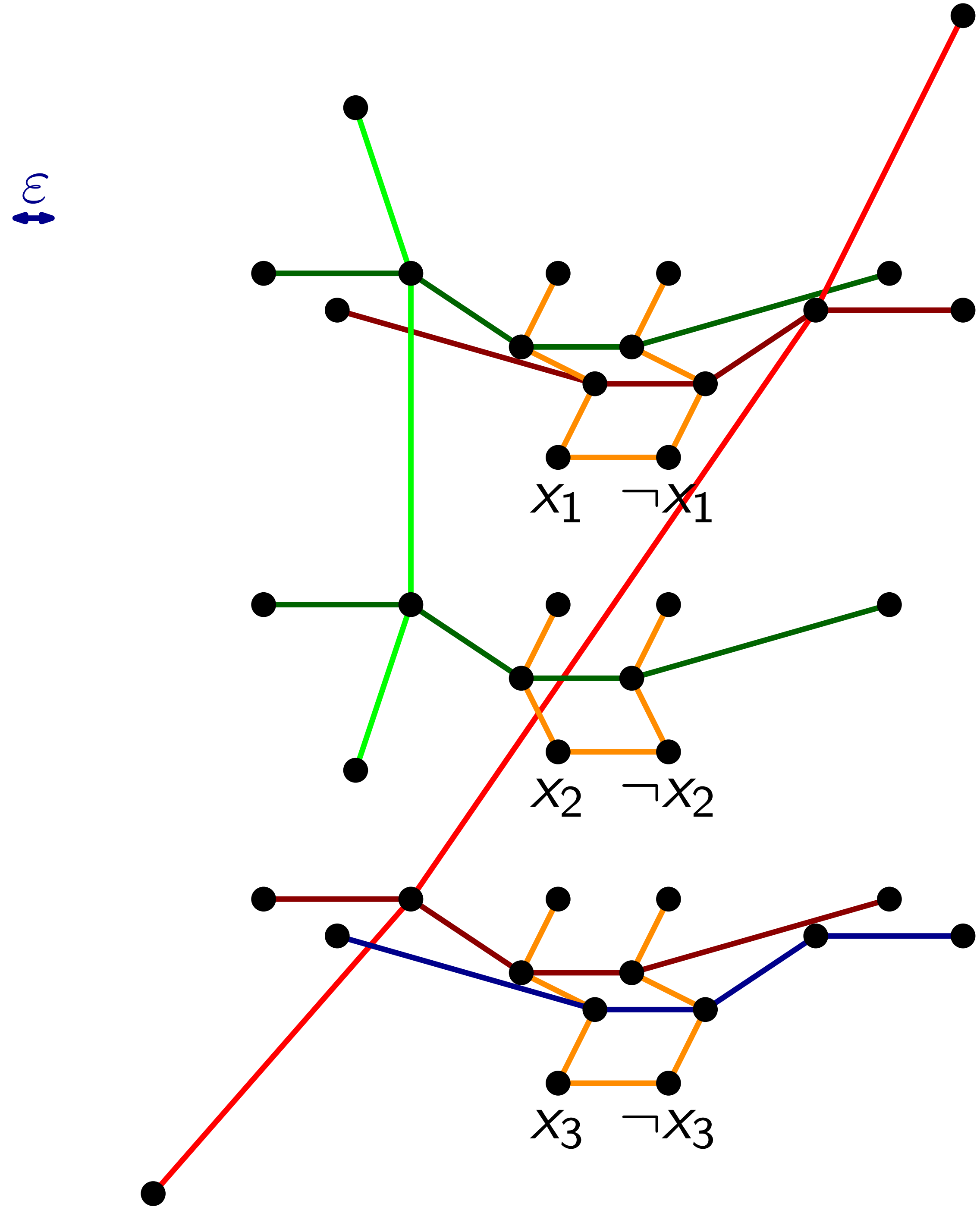
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



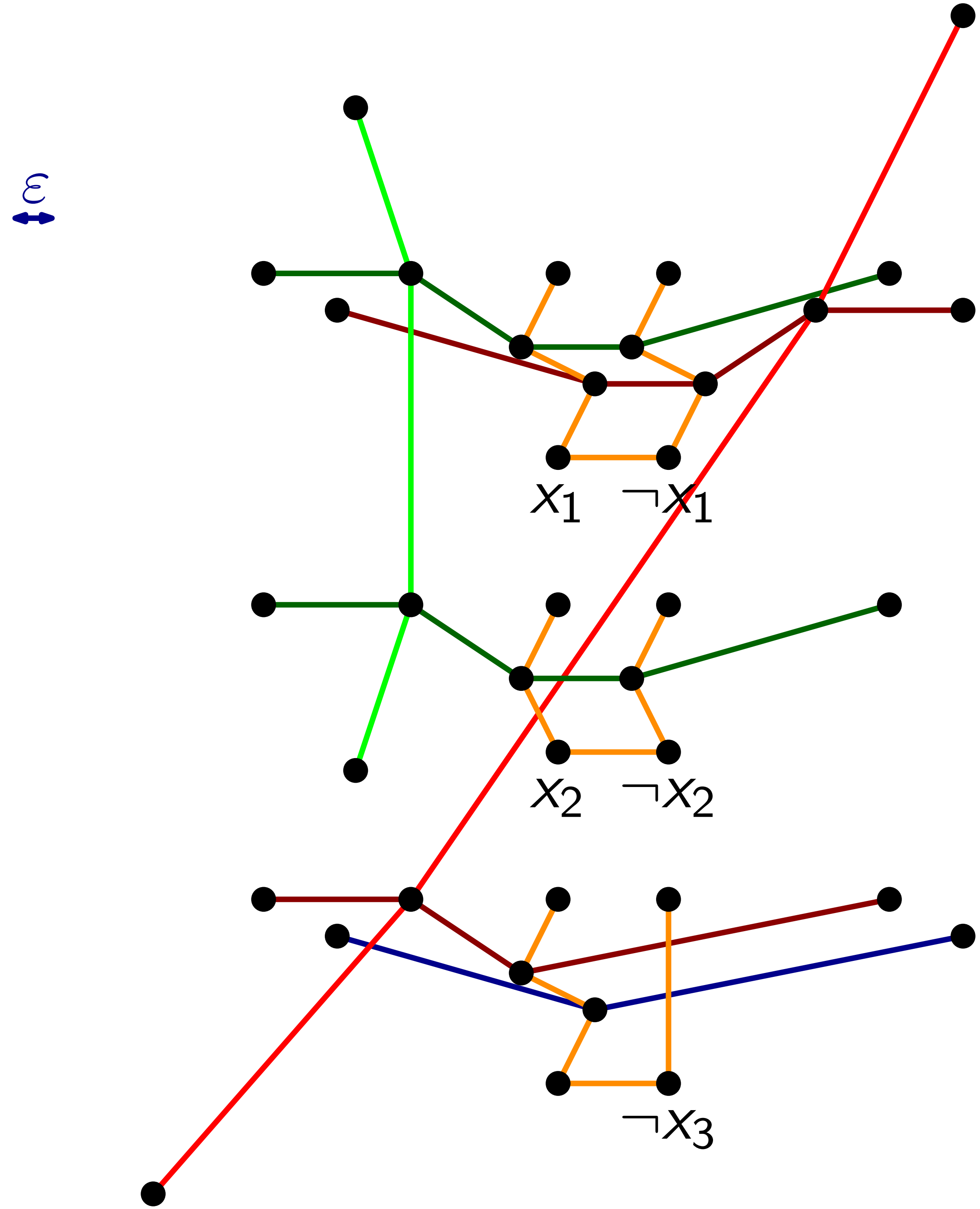
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



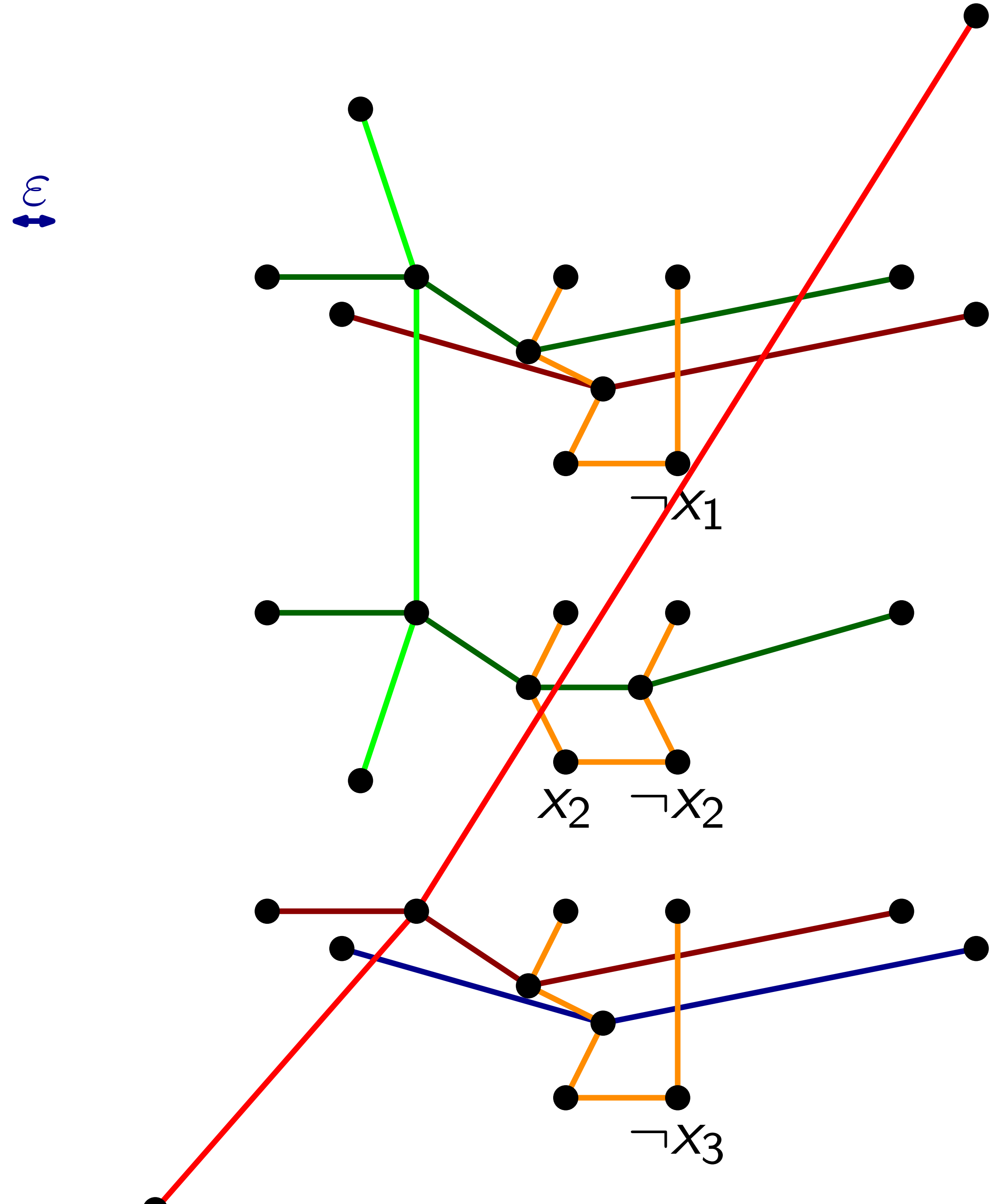
Full Example

$(x_1 \vee x_2) \wedge$
$(\neg x_1 \vee x_3) \wedge$
$(\neg x_3)$



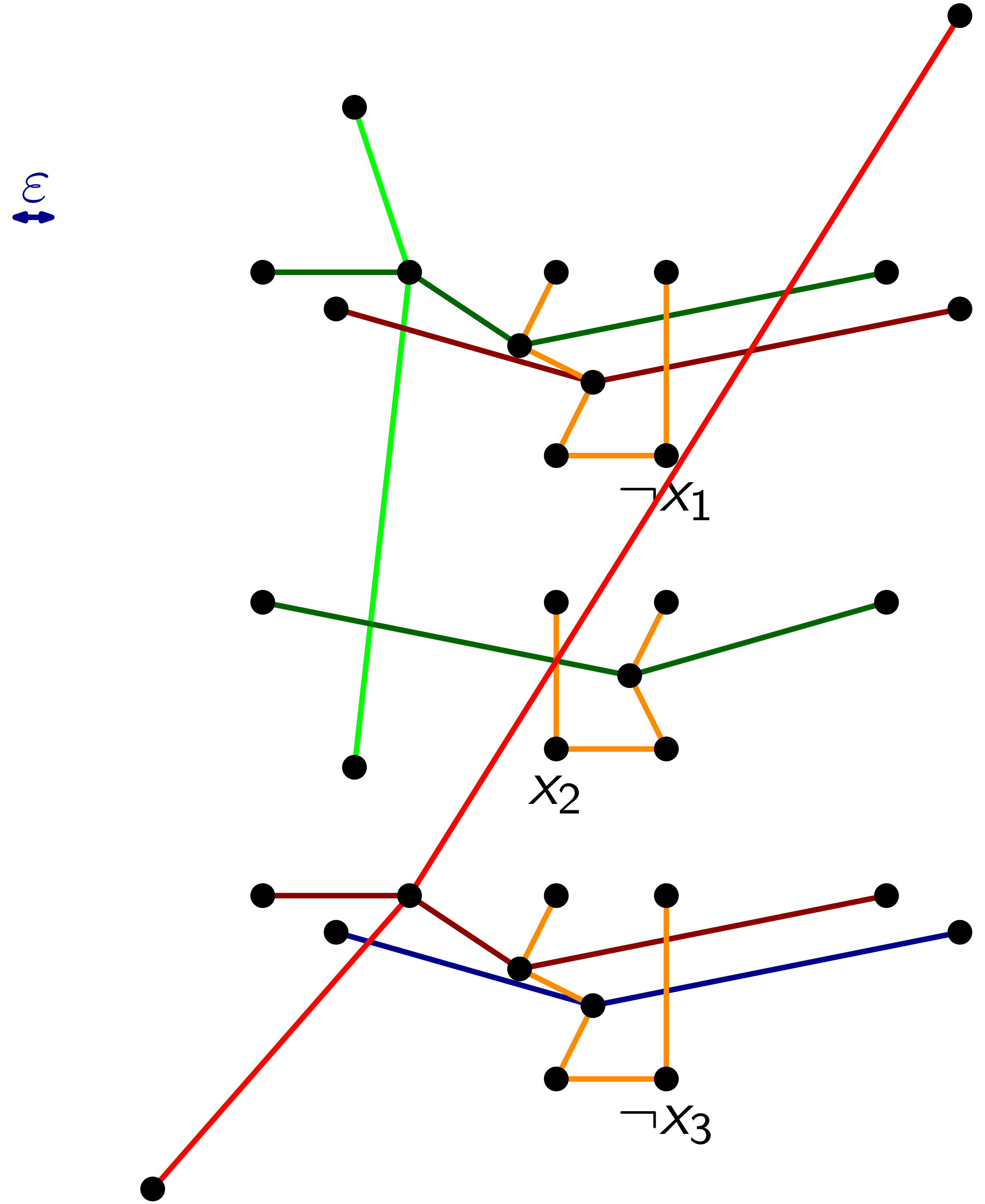
Full Example

$(x_1 \vee x_2)\wedge$
$(\neg x_1 \vee x_3)\wedge$
$(\neg x_3)$



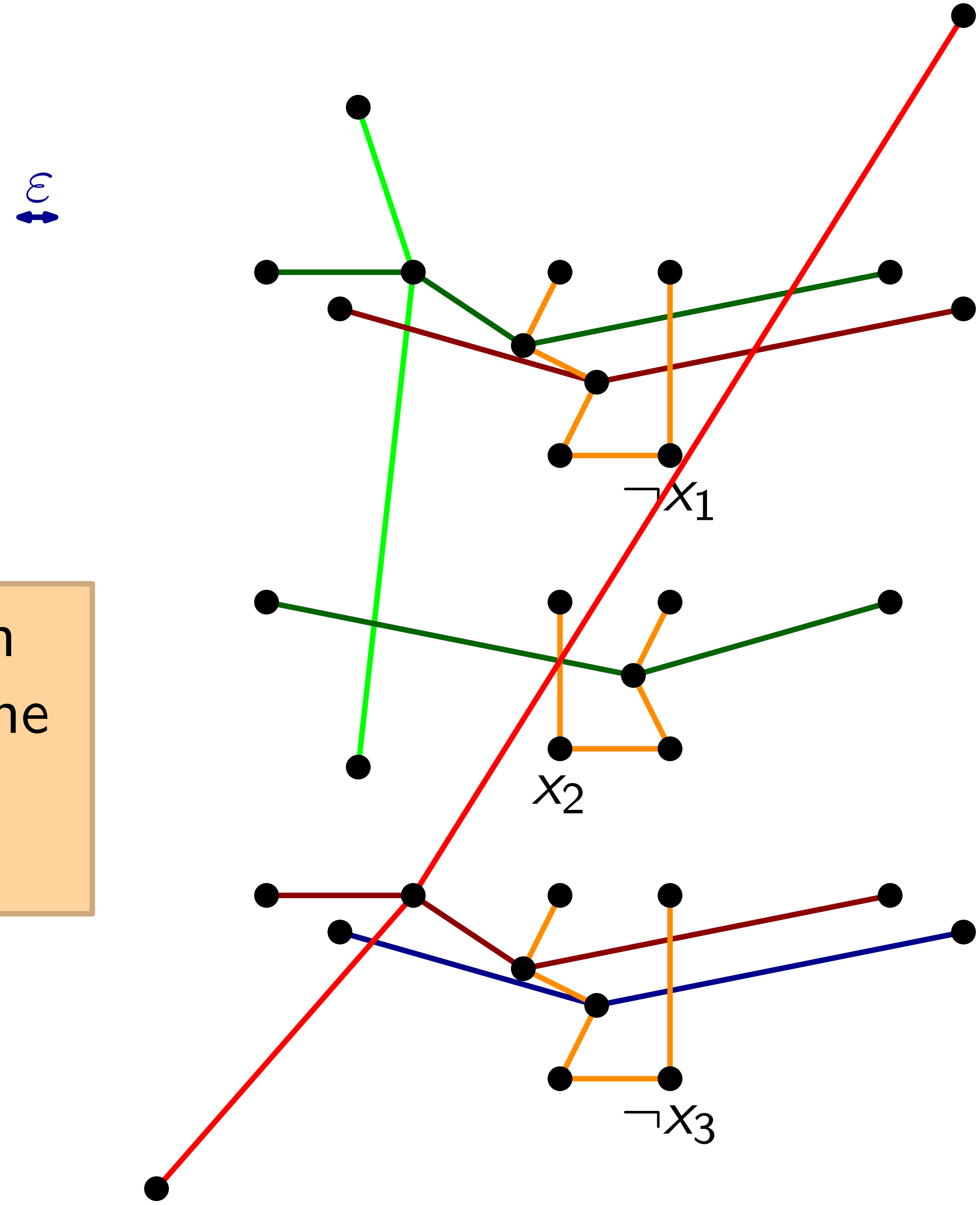
Full Example

$\big(x_1 \vee x_2\big) \wedge$
$\big(\neg x_1 \vee x_3\big) \wedge$
$\big(\neg x_3\big)$



Full Example

$(x_1 \vee x_2)\wedge$
$(\neg x_1 \vee x_3)\wedge$
$(\neg x_3)$



We can even obtain APX-hardness by the reduction from MAX-2-SAT

Fixed-Parameter Tractability

Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$.

Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals Min-Vertices and Min-Edges.

Proof Sketch:

Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$.

Proof Sketch:

- Assume for each subset V' of the shared vertices V_{shared} that V' is in the optimal solution and $V_{\mathsf{shared}} \setminus V'$ is not.

Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$.

Proof Sketch:

- Assume for each subset V' of the shared vertices V_{shared} that V' is in the optimal solution and $V_{\text{shared}} \setminus V'$ is not.
- Compute the simplification of the remaining (simple-polyline) sections in the classic way, e.g., with the algorithm by Chan and Chin.

Fixed-Parameter Tractability

Theorem 2:
Simplifying a bundle of polylines is fixed-parameter tractable in the number of shared vertices for the goals $\textsc{Min-Vertices}$ and $\textsc{Min-Edges}$.

Proof Sketch:

- Assume for each subset V' of the shared vertices V_{shared} that V' is in the optimal solution and $V_{\text{shared}} \setminus V'$ is not.
- Compute the simplification of the remaining (simple-polyline) sections in the classic way, e.g., with the algorithm by Chan and Chin.
- Running time in $O(2^k \cdot \ell n^2)$

$k := |V_{\text{shared}}|, \quad \ell := \#$ polylines, $\quad n := \#$ vertices

Summary

Summary

Problem:
Simplify a set of polylines sharing some vertices and edges
Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

Summary

Problem:
Simplify a set of polylines sharing some vertices and edges
Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline

Summary

Problem:
Simplify a set of polylines sharing some vertices and edges
Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline
- Becomes NP-hard and APX-hard

Summary

Problem:
Simplify a set of polylines sharing some vertices and edges
Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline
- Becomes NP-hard and APX-hard
- FPT in the number of shared vertices

Summary

Problem:
Simplify a set of polylines sharing some vertices and edges
Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline
- Becomes NP-hard and APX-hard
- FPT in the number of shared vertices
- Not FPT in the number of polylines

Summary

Problem:
Simplify a set of polylines sharing some vertices and edges
Goal 1: Minimize the number of vertices
Goal 2: Minimize the number of edges

- Generalizes the well-known problem of simplifying a single polyline
- Becomes NP-hard and APX-hard
- FPT in the number of shared vertices
- Not FPT in the number of polylines

- Since there is no PTAS, is there a constant factor approximation algorithm?