Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

Institute for Informatics

Stick Graphs with Length Constraints

Steven Chaplick, Philipp Kindermann, Andre Löffler, Florian Thiele, Alexander Wolff, Alexander Zaft, and **Johannes Zink**

Introduction

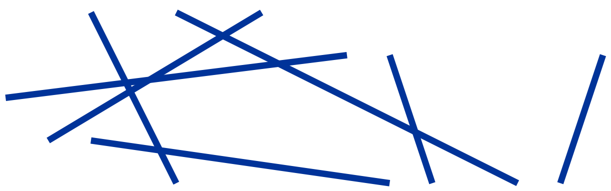
- Given a collection $\mathcal{S}$ of geometric objects

Introduction

- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.

Introduction

- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.



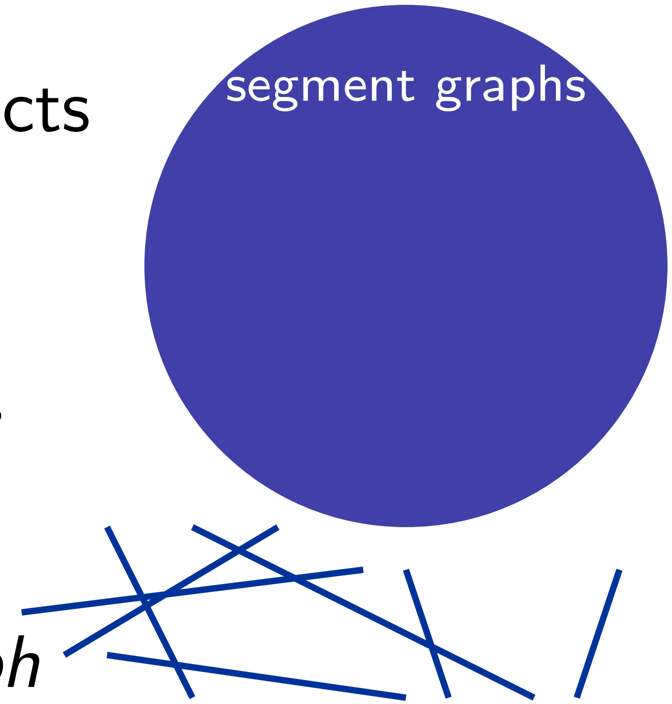
- $\mathcal{S}$: line segments

Introduction

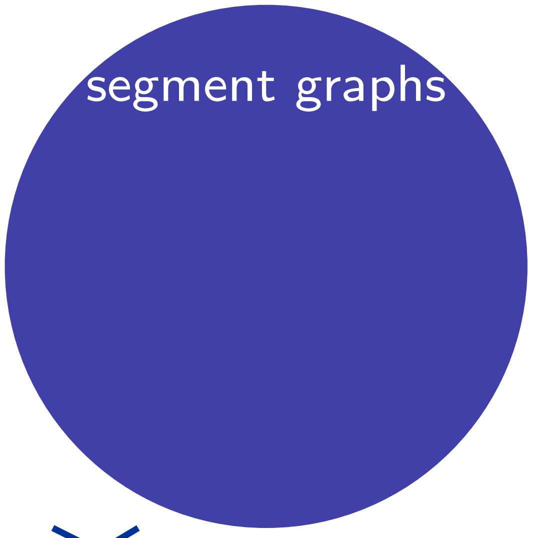
- Given a collection $\mathcal{S}$ of geometric objects

- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.

- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*

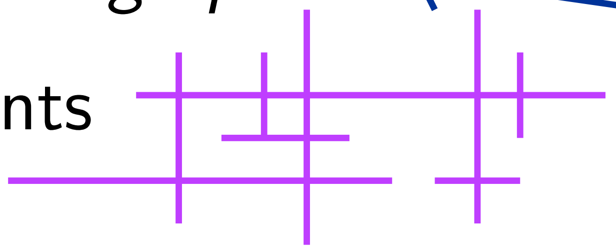


Introduction



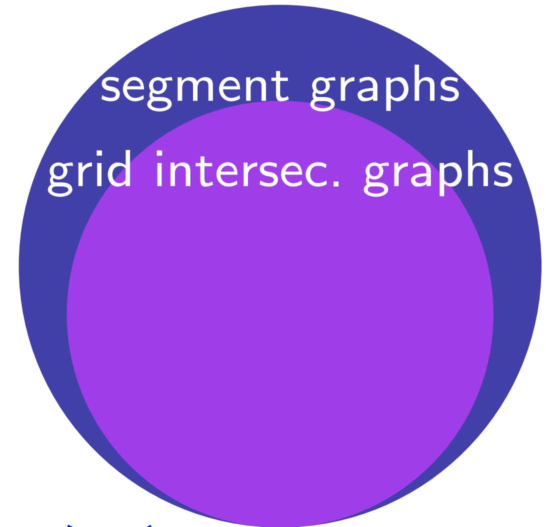
- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.
- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*

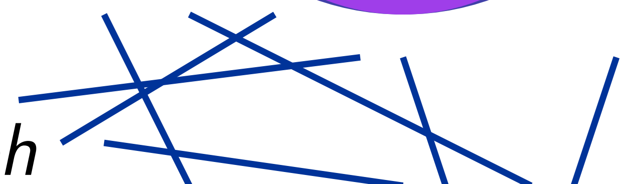
- $\mathcal{S}$: horizontal & vertical segments

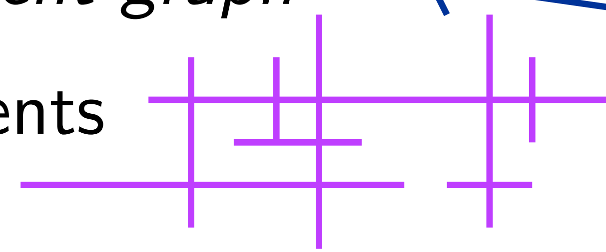


Introduction

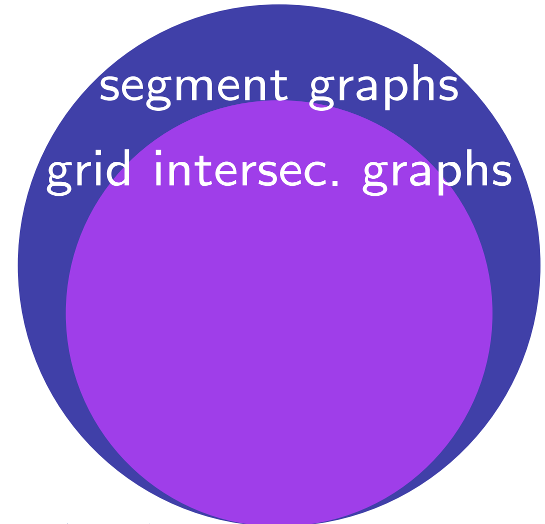
- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.
- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*
- $\mathcal{S}$: horizontal & vertical segments $\Rightarrow$ G: *grid intersection graph*



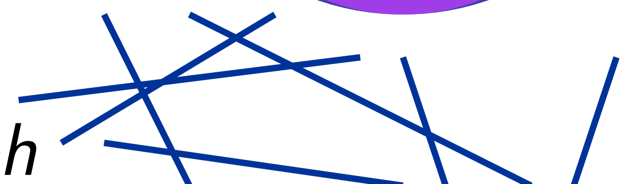




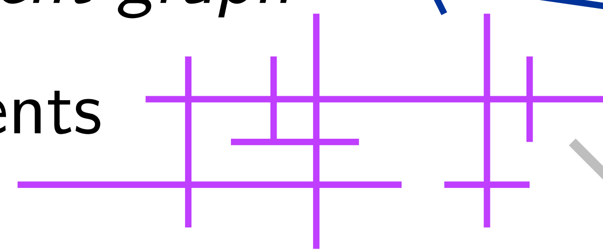
Introduction



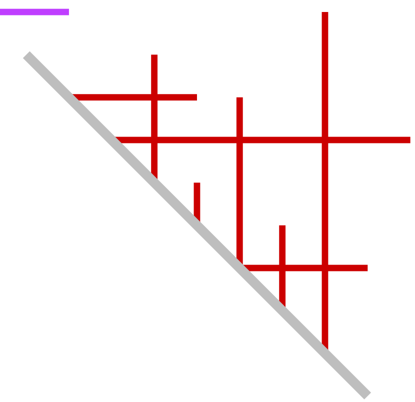
- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.



- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*

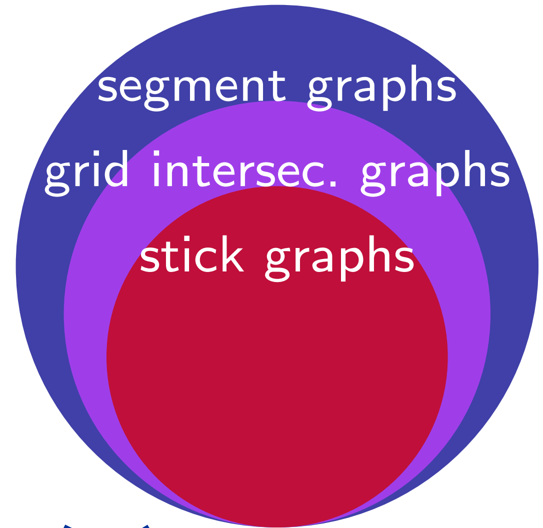


- $\mathcal{S}$: horizontal & vertical segments $\Rightarrow$ G: *grid intersection graph*

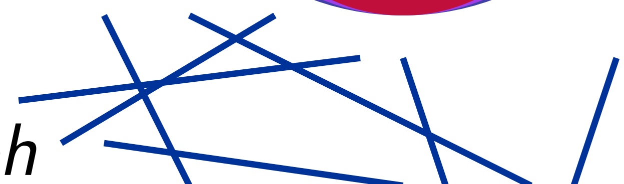


- $\mathcal{S}$: horizontal & vertical segments grounded on a line of slope -1

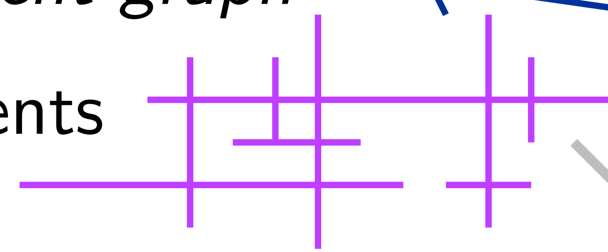
Introduction



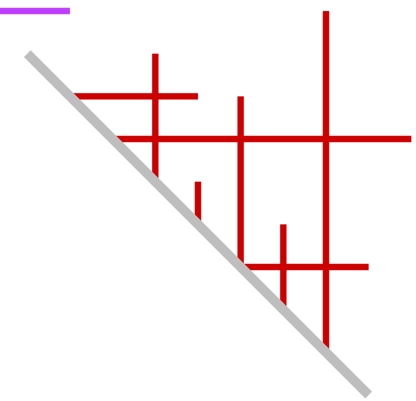
- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.
- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*



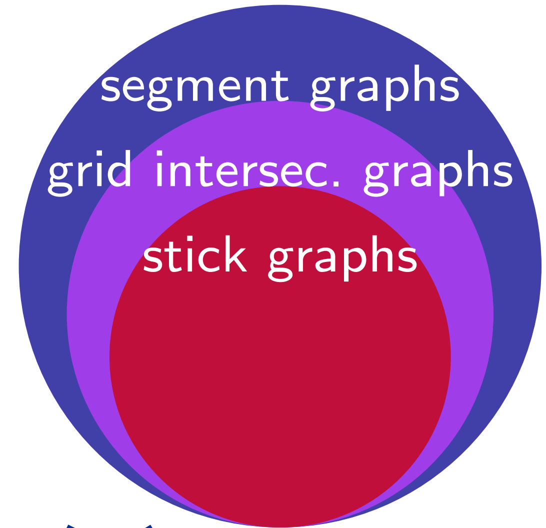
- $\mathcal{S}$: horizontal & vertical segments $\Rightarrow$ G: *grid intersection graph*



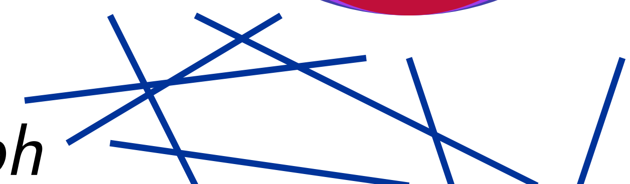
- $\mathcal{S}$: horizontal & vertical segments grounded on a line of slope -1 $\Rightarrow$ G: *stick graph*



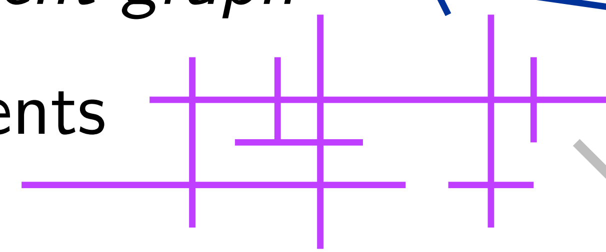
Introduction



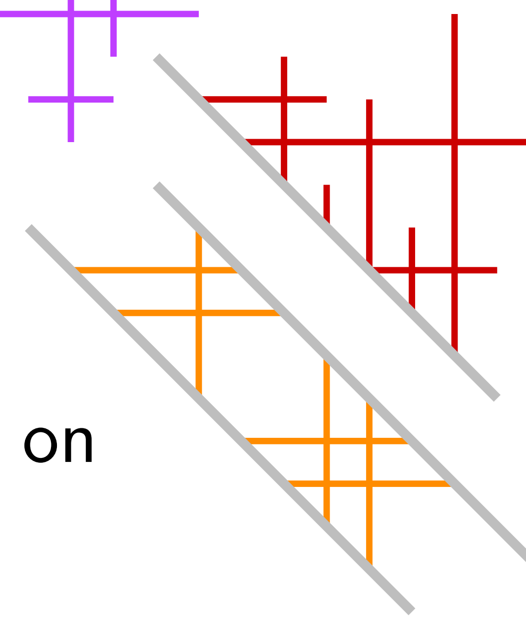
- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.



- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*

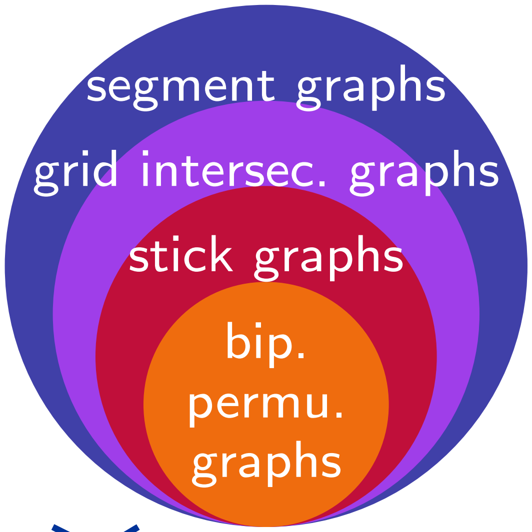


- $\mathcal{S}$: horizontal & vertical segments $\Rightarrow$ G: *grid intersection graph*
- $\mathcal{S}$: horizontal & vertical segments grounded on a line of slope -1 $\Rightarrow$ G: *stick graph*



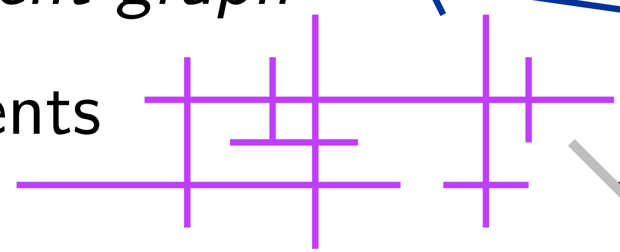
- $\mathcal{S}$: horizontal & vertical segments grounded on two parallel lines

Introduction

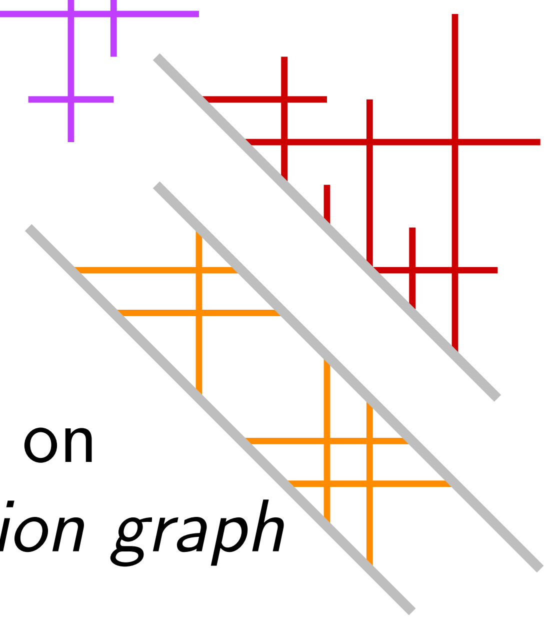


- Given a collection $\mathcal{S}$ of geometric objects
- The *intersection graph* G of $\mathcal{S}$ has
  - $\mathcal{S}$ as its vertex set
  - an edge for each two elements of $\mathcal{S}$ that intersect or touch each other.

- $\mathcal{S}$: line segments $\Rightarrow$ G: *segment graph*



- $\mathcal{S}$: horizontal & vertical segments $\Rightarrow$ G: *grid intersection graph*



- $\mathcal{S}$: horizontal & vertical segments grounded on a line of slope -1 $\Rightarrow$ G: *stick graph*
- $\mathcal{S}$: horizontal & vertical segments grounded on two parallel lines $\Rightarrow$ G: *bipartite permutation graph*

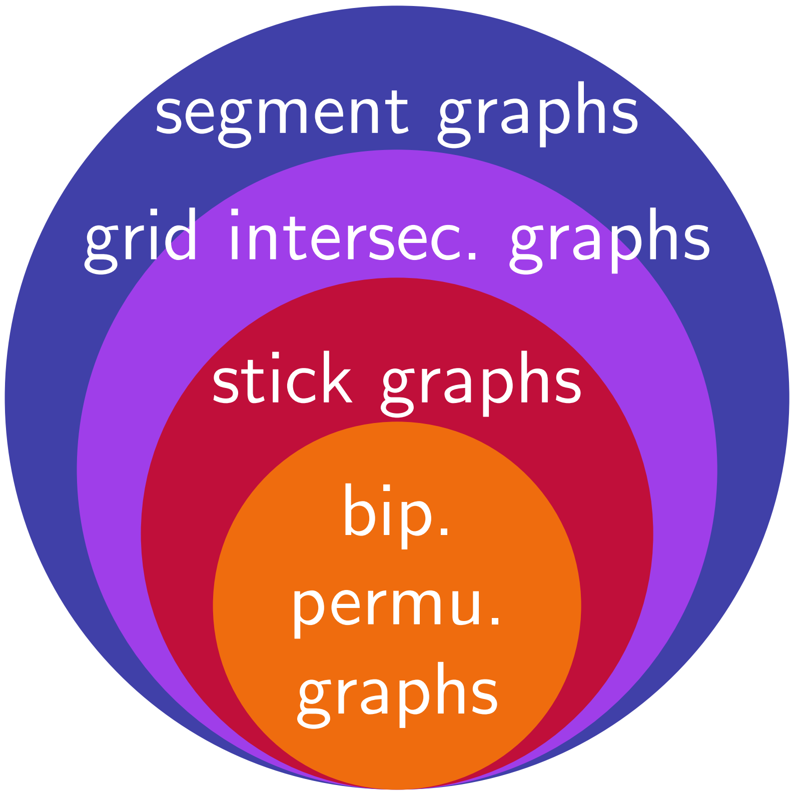
Computational Complexity

Recognition problem:
Decide whether a given graph is an intersection graph.

Computational Complexity

Recognition problem:
Decide whether a given graph is an intersection graph.

How hard for each class?

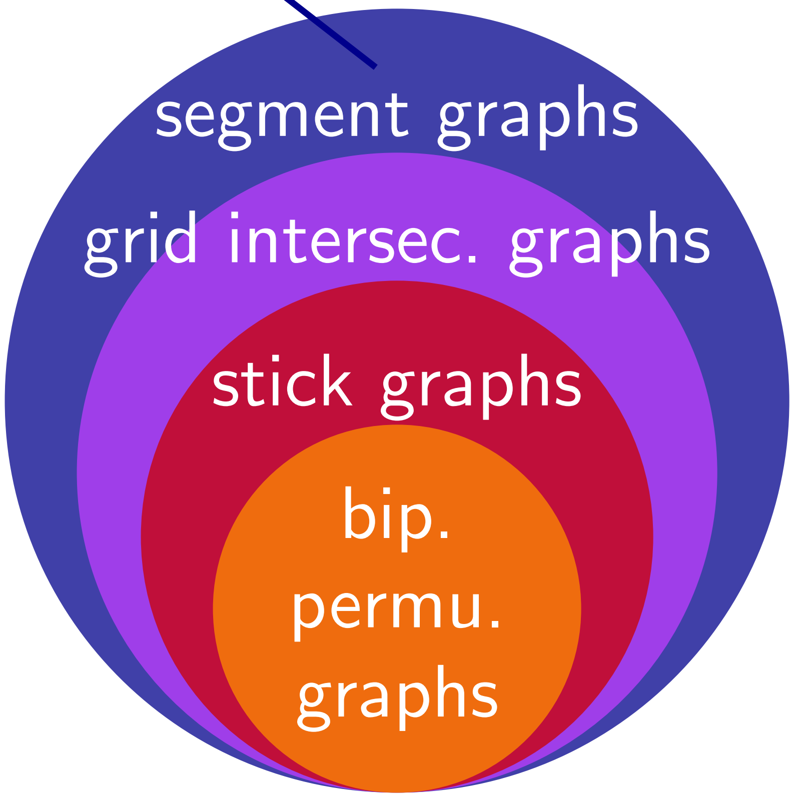


Computational Complexity

Recognition problem:
Decide whether a given graph is an intersection graph.

How hard for each class?

segment graphs: $\exists\mathbb{R}$-complete
[Kratochvíl, Matoušek '94; Matoušek '14]



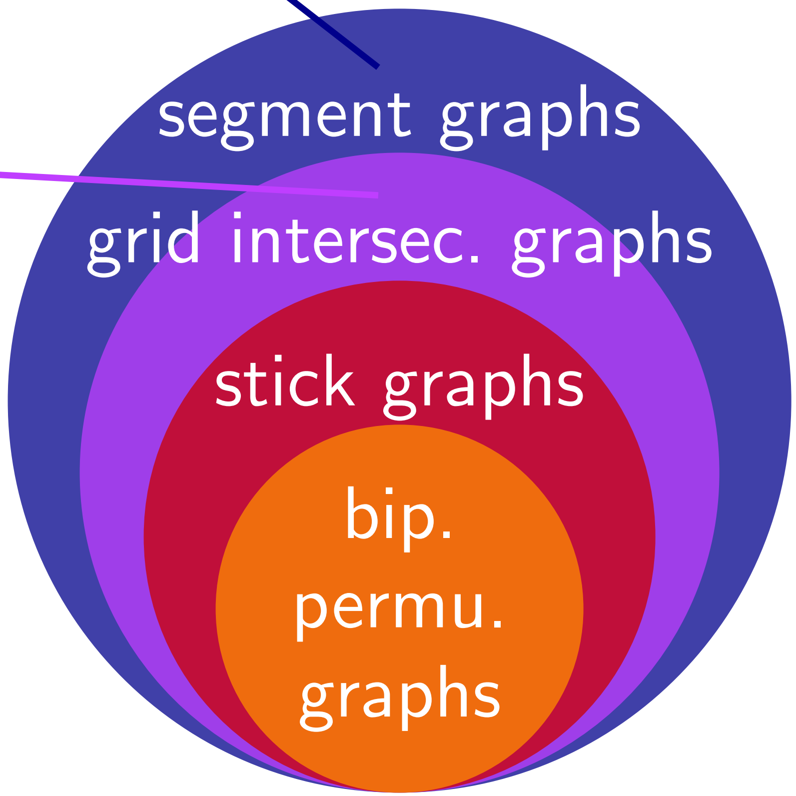
Computational Complexity

Recognition problem:
Decide whether a given graph is an intersection graph.

How hard for each class?

segment graphs: $\exists\mathbb{R}$-complete
[Kratochvíl, Matoušek '94; Matoušek '14]

grid intersection graphs: NP-complete
[Kratochvíl '94]



Computational Complexity

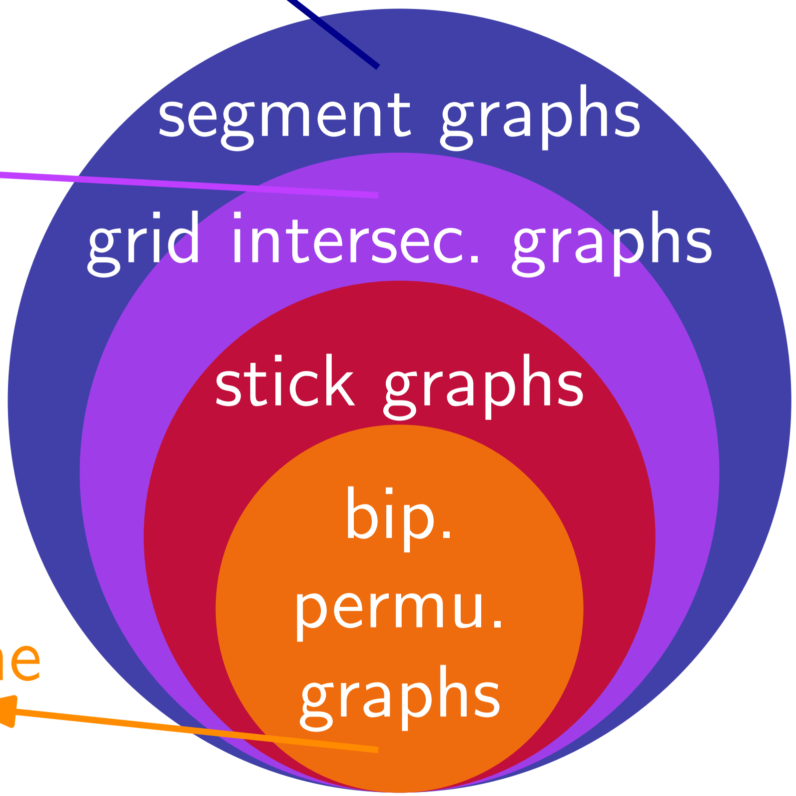
Recognition problem:
Decide whether a given graph is an intersection graph.

How hard for each class?

segment graphs: $\exists\mathbb{R}$-complete
[Kratochvíl, Matoušek '94; Matoušek '14]

grid intersection graphs: NP-complete
[Kratochvíl '94]

(bipart.) permutation graphs: linear time
[Spinrad et al. '87; Kratsch et al. '06]



Computational Complexity

Recognition problem:
Decide whether a given graph is an intersection graph.

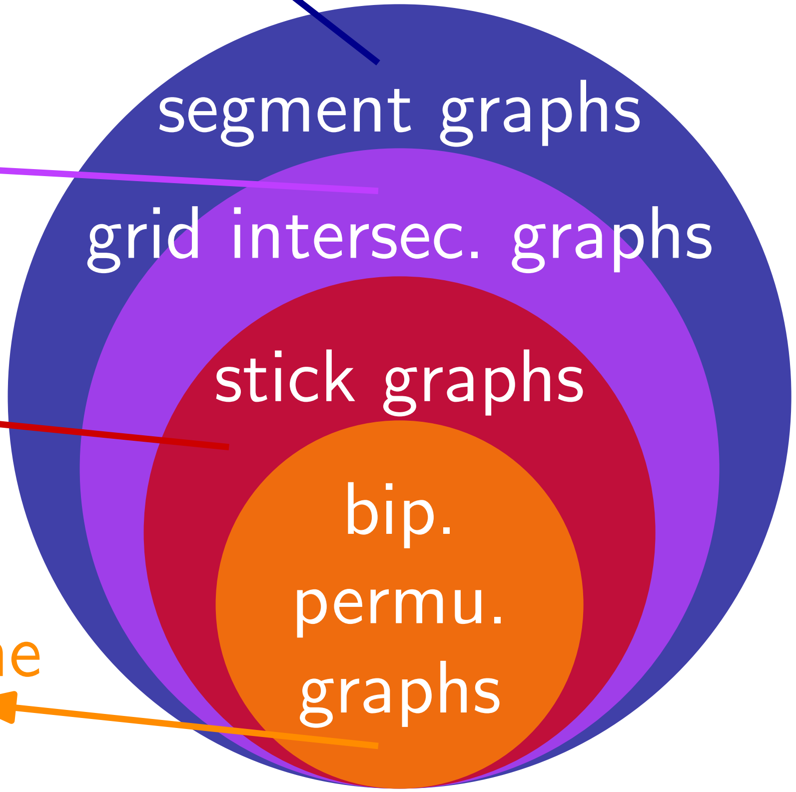
How hard for each class?

segment graphs: $\exists\mathbb{R}$-complete
[Kratochvíl, Matoušek '94; Matoušek '14]

grid intersection graphs: NP-complete
[Kratochvíl '94]

stick graphs:

(bipart.) permutation graphs: linear time
[Spinrad et al. '87; Kratsch et al. '06]



Computational Complexity

Recognition problem:
Decide whether a given graph is an intersection graph.

How hard for each class?

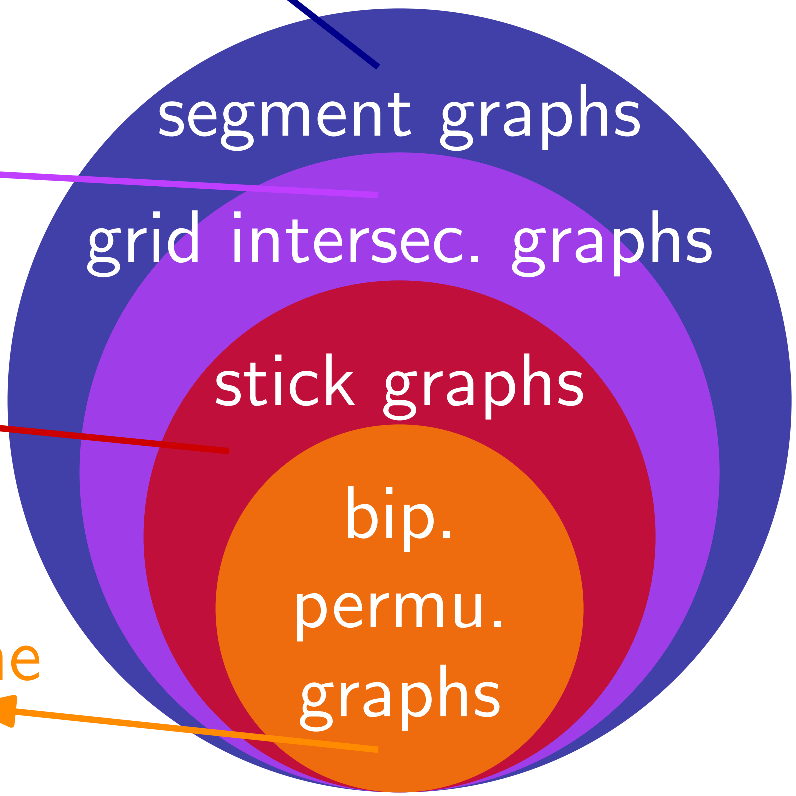
segment graphs: $\exists\mathbb{R}$-complete
[Kratochvíl, Matoušek '94; Matoušek '14]

grid intersection graphs: NP-complete
[Kratochvíl '94]

stick graphs: ???

remains open. . .

(bipart.) permutation graphs: linear time
[Spinrad et al. '87; Kratsch et al. '06]



Versions of Stick Graph Recognition

Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

Versions of Stick Graph Recognition

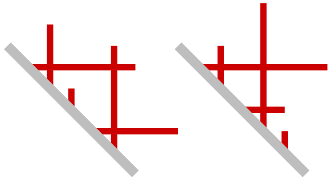
Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

- **STICK:** ...?

Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

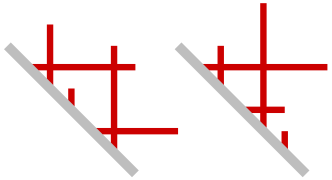
- **STICK:** ...?



Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

- **STICK:** ... ?

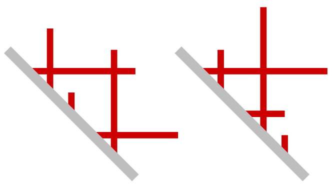


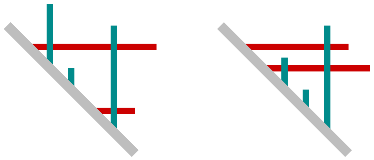
- **STICK$_\mathbf{A}$:**
  ... if a permutation of the vertices in A is given?

Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

- **STICK:** ... ?



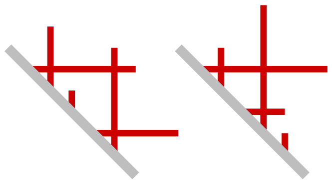


- **STICK$_\mathbf{A}$:**
  ... if a permutation of the vertices in A is given?

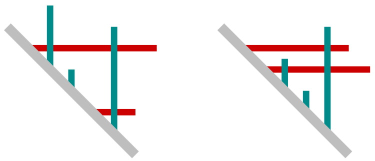
Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation . . .

- **STICK:** . . . ?



- **STICK$_\mathbf{A}$:**
  . . . if a permutation of the vertices in A is given?

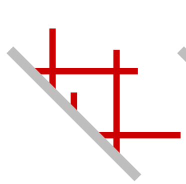


- **STICK$_\mathbf{AB}$:**
  . . . if a permutation of the vertices in A and a permutation of the vertices in B is given?

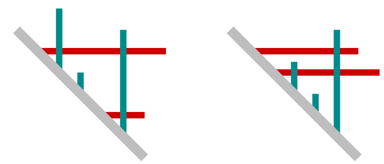
Versions of Stick Graph Recognition

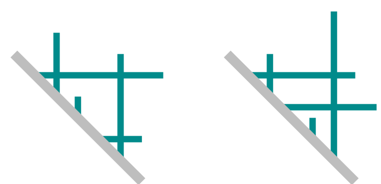
Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

- **STICK:** ...?



- **STICK$_\mathbf{A}$:**
  ...if a permutation of the vertices in A is given?



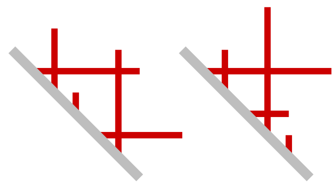


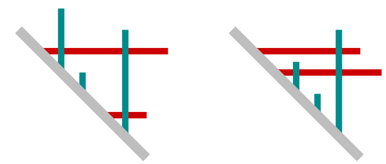
- **STICK$_\mathbf{AB}$:**
  ...if a permutation of the vertices in A and a permutation of the vertices in B is given?

Versions of Stick Graph Recognition

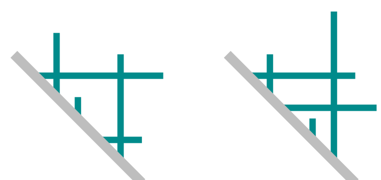
Given a bipartite Graph $G = (A \dot{\cup} B, E)$, does G admit a stick representation ...

- **STICK:** ...?





- **STICK$_{\mathbf{A}}$:**
  ...if a permutation of the vertices in A is given?



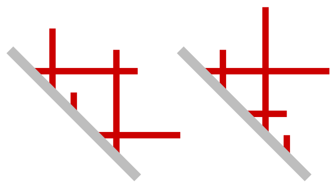
- **STICK$_{\mathbf{AB}}$:**
  ...if a permutation of the vertices in A and a permutation of the vertices in B is given?

- **STICK$^{\mathbf{fix}}_{\star}$:**
  ...if a stick length for each vertex (and possibly permutations of A, or A and B) is given?

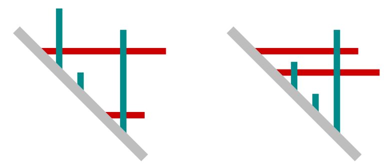
Versions of Stick Graph Recognition

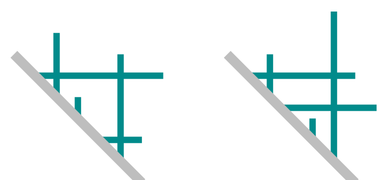
Given a bipartite Graph $G = (A \dot\cup B, E)$, does G admit a stick representation ...

- **STICK:** ...?

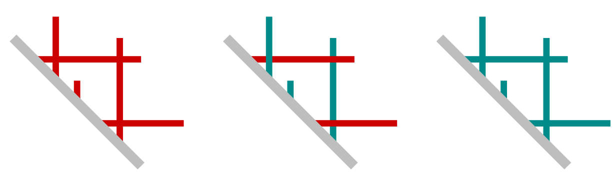


- **$\mathbf{STICK_A}$:**
  ...if a permutation of the vertices in A is given?





- **$\mathbf{STICK_{AB}}$:**
  ...if a permutation of the vertices in A and a permutation of the vertices in B is given?

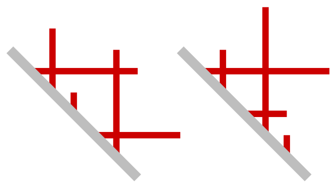


- **$\mathbf{STICK_{\star}^{fix}}$:**
  ...if a stick length for each vertex (and possibly permutations of A, or A and B) is given?

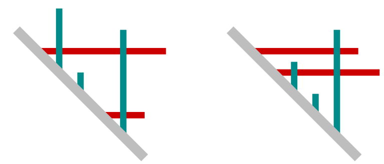
Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot\cup B, E)$, does G admit a stick representation ...

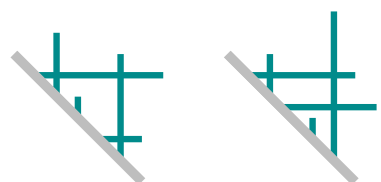
- **STICK:** ... ?



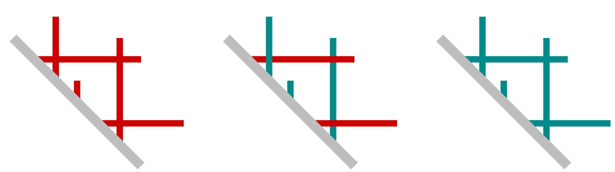
- **STICK$_\mathbf{A}$:**
  ... if a permutation of the vertices in A is given?



- **STICK$_\mathbf{AB}$:**
  ... if a permutation of the vertices in A and a permutation of the vertices in B is given?



- **STICK$^\mathbf{fix}_\star$:**
  ... if a stick length for each vertex (and possibly permutations of A, or A and B) is given?

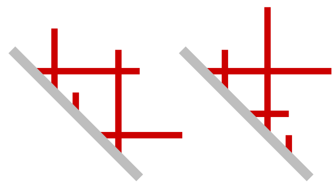


First investigated by De Luca et al. [GD'18]

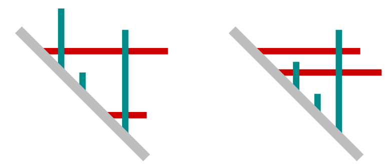
Versions of Stick Graph Recognition

Given a bipartite Graph $G = (A \dot\cup B, E)$, does G admit a stick representation ...

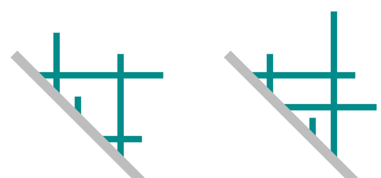
- **STICK:** ...?



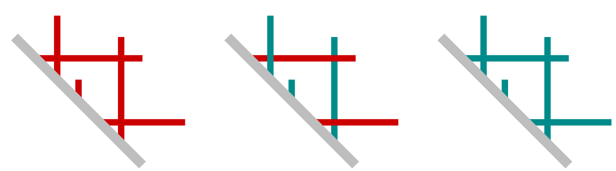
- **STICK$_{\mathbf{A}}$:**
  ...if a permutation of the vertices in A is given?



- **STICK$_{\mathbf{AB}}$:**
  ...if a permutation of the vertices in A and a permutation of the vertices in B is given?



- **STICK$_\star^{\mathbf{fix}}$:**
  ...if a stick length for each vertex (and possibly permutations of A, or A and B) is given?



First investigated by De Luca et al. [GD'18]

new

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

$\star$	$\textbf{STICK}_\star$	$\textbf{STICK}_\star^{\textbf{fix}}$
A		
AB		

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

$\star$	**STICK**$_\star$	**STICK**$_\star^{\mathbf{fix}}$
	?	**?**
A	**?**[1]	**?**
AB	$O(\lvert A\rvert\lvert B\rvert)$ [De Luca et al. GD'18]	**?**

[1] an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	**STICK**$_\star$	**STICK**$_\star^{\text{fix}}$
	?	**?**
A	**?**[1]	**?**
AB	$O(\|A\|\|B\|)$ [De Luca et al. GD'18]	**?**

[1] an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	**STICK**$_\star$	**STICK**$_\star^{\text{fix}}$
	?	**?**
A	**?**[1] $O(\lvert A\rvert \lvert B\rvert)$	**?**
AB	$O(\lvert A\rvert \lvert B\rvert)$ [De Luca et al. GD'18]	**?**

[1] an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	$\textsf{STICK}_\star$	$\textsf{STICK}^{\text{fix}}_\star$
	?	**?**
A	**?**[1] $O(\lvert A\rvert \lvert B\rvert)$	**?**
AB	$O(\lvert A\rvert \lvert B\rvert)$ [De Luca et al. GD'18] $O(\lvert E\rvert)$	**?**

[1]an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	**STICK**$_\star$	**STICK**$_\star^{\text{fix}}$
	?	**?** NP-complete
A	**?**[1] $O(\|A\|\|B\|)$	**?**
AB	$O(\|A\|\|B\|)$ [De Luca et al. GD'18] $O(\|E\|)$	**?**

[1]an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	STICK$_\star$	STICK$_\star^{\text{fix}}$
	?	? NP-complete
A	?[1] $O(\|A\|\|B\|)$	? NP-complete
AB	$O(\|A\|\|B\|)$ [De Luca et al. GD'18] $O(\|E\|)$	?

[1] an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	**STICK**$_\star$	**STICK**$_\star^{\text{fix}}$
	?	**?** NP-complete
A	**?**[1] $O(\|A\|\|B\|)$	**?** NP-complete
AB	$O(\|A\|\|B\|)$ [De Luca et al. GD'18] $O(\|E\|)$	**?** in general: NP-complete

[1] an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	**STICK**$_\star$	**STICK**$^{\text{fix}}_\star$
	?	**?** NP-complete
A	**?**[1] $O(\lvert A\rvert\lvert B\rvert)$	**?** NP-complete
AB	$O(\lvert A\rvert\lvert B\rvert)$ [De Luca et al. GD'18] $O(\lvert E\rvert)$	**?** in general: NP-complete w/o isolated vtc.: $O((\lvert A\rvert + \lvert B\rvert)^2)$

[1]an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	$\text{STICK}_\star$	$\text{STICK}_\star^{\text{fix}}$
	next ?	? NP-complete
A	?[1] $O(\|A\|\|B\|)$	? NP-complete
AB	$O(\|A\|\|B\|)$ [De Luca et al. GD'18] $O(\|E\|)$	? in general: NP-complete w/o isolated vtc.: $O((\|A\|+\|B\|)^2)$

[1] an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Complexity of Recognition

for a bipartite graph $G = (A \cup B, E)$

our results

$\star$	**STICK**$_\star$		**STICK**$^{\text{fix}}_\star$	
	?		**?**	NP-complete
A	**?**[1]	$O(\|A\|\|B\|)$	**?**	NP-complete
AB	$O(\|A\|\|B\|)$ [De Luca et al. GD'18]	$O(\|E\|)$	**?**	in general: NP-complete w/o isolated vtc.: $O((\|A\| + \|B\|)^2)$

next

afterwards

[1]an $O(|A|^3|B|^3)$ time algorithm proposed by De Luca et al. turned out to be wrong

Algorithm for STICK_{A}

$\star$	**STICK**$_\star$	**STICK**$_\star^{\text{fix}}$
	?	NP-complete
A	$O(\lvert A\rvert\lvert B\rvert)$	NP-complete
AB	$O(\lvert E\rvert)$	in general: NP-complete w/o isolated vtc.: $O((\lvert A\rvert+\lvert B\rvert)^2)$

Algorithm for $\mathsf{STICK_A}$

- Sweep-line along the ordered vertical sticks in A: *enter event* (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$

Algorithm for $\mathrm{STICK_A}$

- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* $(i\!\rightarrow)$ for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p

Algorithm for STICK_A

- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:

Algorithm for $\mathrm{STICK_A}$

- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:
  - contains two types of nodes: leaves and non-leaves

Algorithm for STICK$_\mathsf{A}$

- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:
  - contains two types of nodes: leaves and non-leaves
  - each leaf corresponds to a vertex in B^p

Algorithm for STICK$_\mathrm{A}$

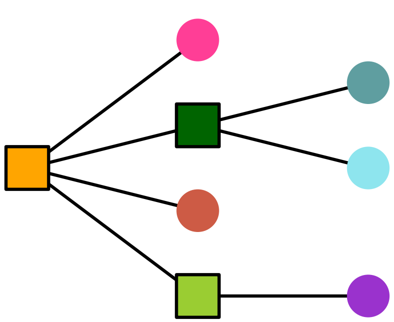
- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i{\scriptstyle\rightarrow}$) for each $a_i \in A$
- Let $p \in \{\ i,\ i{\scriptstyle\rightarrow}\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:
  - contains two types of nodes: leaves and non-leaves
  - each leaf corresponds to a vertex in B^p
  - the order of leaves is free; the order of non-leaves is fixed

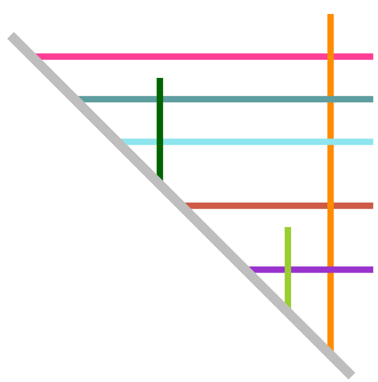
Algorithm for STICK$_\mathsf{A}$

- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:
  - contains two types of nodes: leaves and non-leaves
  - each leaf corresponds to a vertex in B^p
  - the order of leaves is free; the order of non-leaves is fixed
  - encodes all realizable permutations of B^p

Algorithm for STICK$_A$

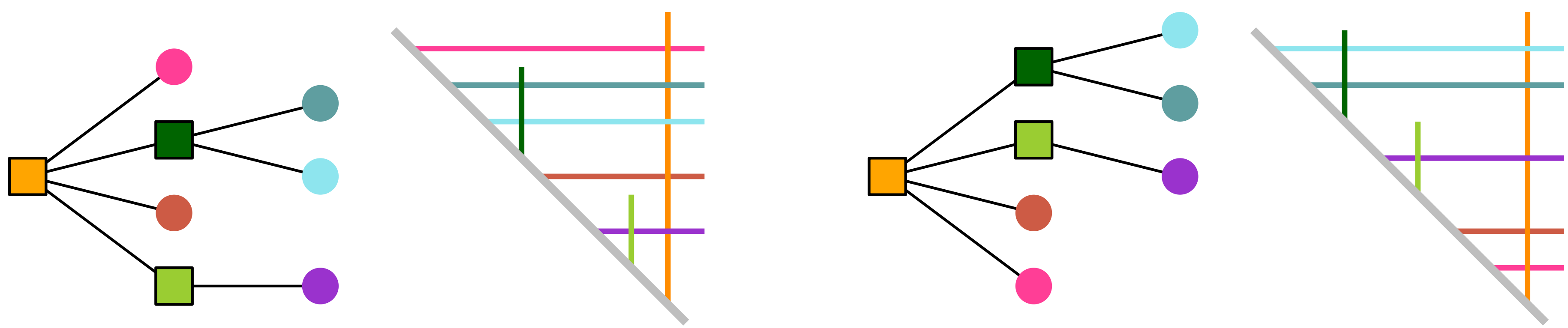
- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:
  - contains two types of nodes: leaves and non-leaves
  - each leaf corresponds to a vertex in B^p
  - the order of leaves is free; the order of non-leaves is fixed
  - encodes all realizable permutations of B^p





Algorithm for STICK$_A$

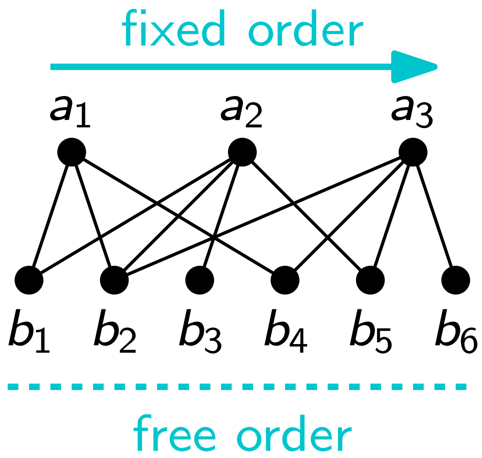
- Sweep-line along the ordered vertical sticks in A:
  enter event (i) and *exit event* ($i\!\rightarrow$) for each $a_i \in A$
- Let $p \in \{\ i,\ i\!\rightarrow\ \}$,
  $G^i :=$ subgraph induced by $a_1, \ldots, a_i$ and their neighbors,
  $B^p :=$ vertices in B that intersect the sweep-line at event p
- (Rooted) tree data structure $\mathcal{T}^p$:
  - contains two types of nodes: leaves and non-leaves
  - each leaf corresponds to a vertex in B^p
  - the order of leaves is free; the order of non-leaves is fixed
  - encodes all realizable permutations of B^p



Example for STICK_A

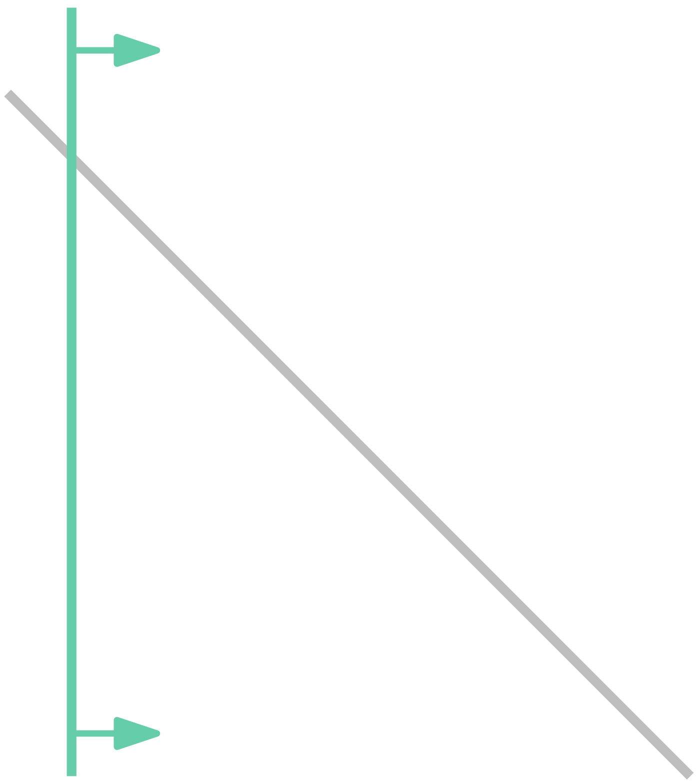
$i = 0$ Event: *Start* $B^0 = \emptyset$

G:



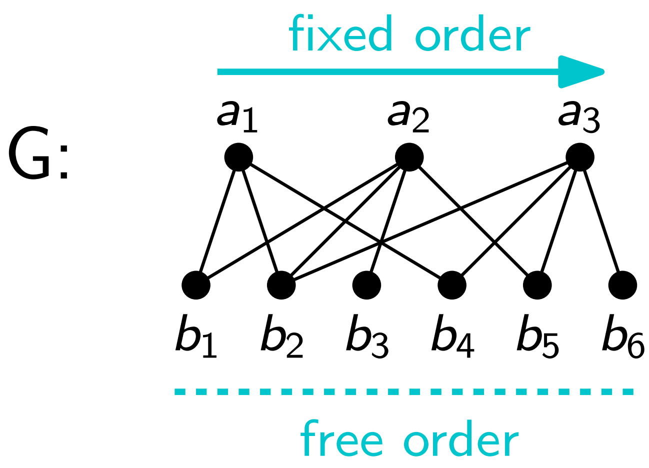
G^0 :

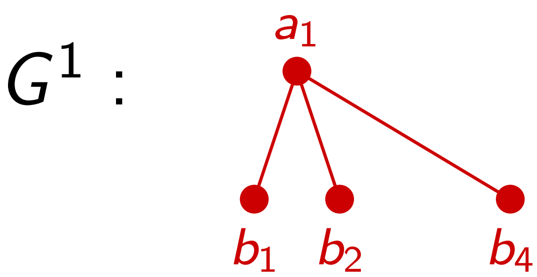
$\mathcal{T}^0$:

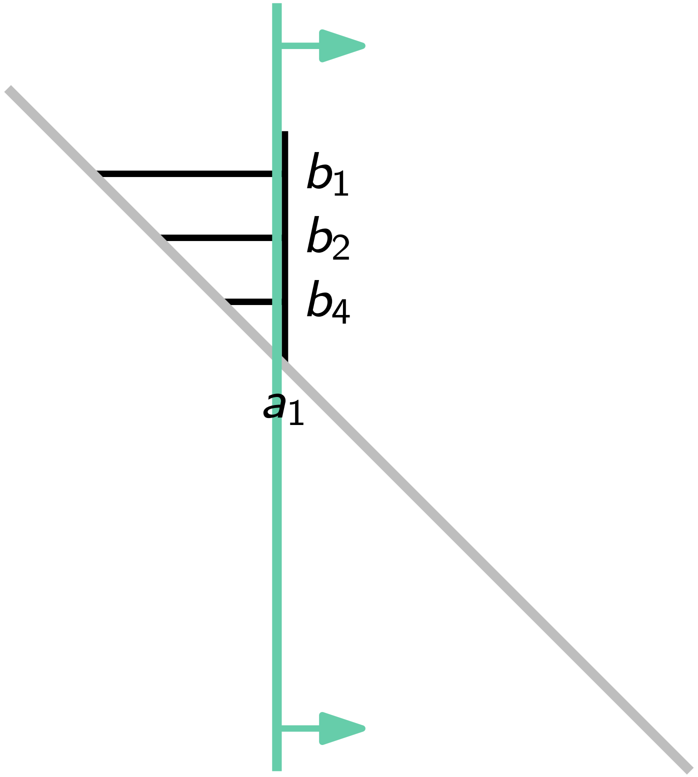


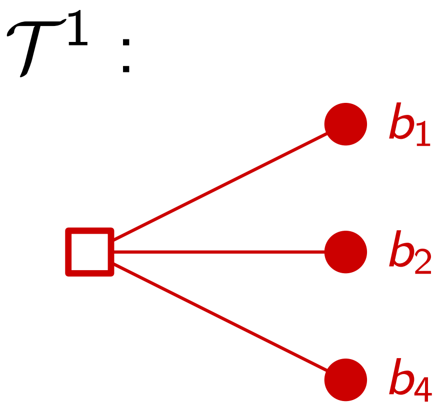
Example for STICK$_A$

$i = 1$ Event: 1 $B^1 = \{b_1, b_2, b_4\}$



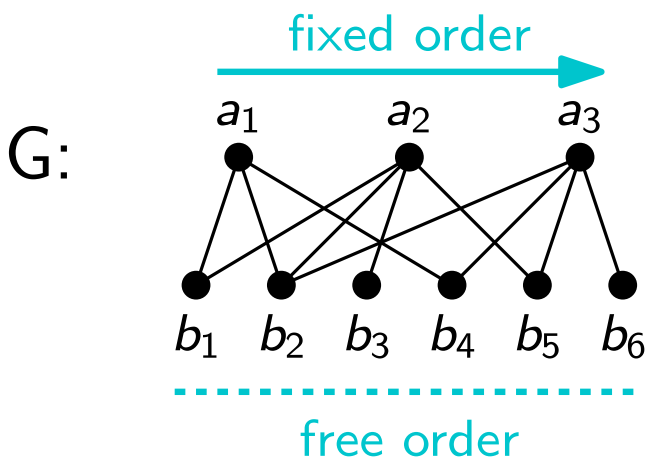


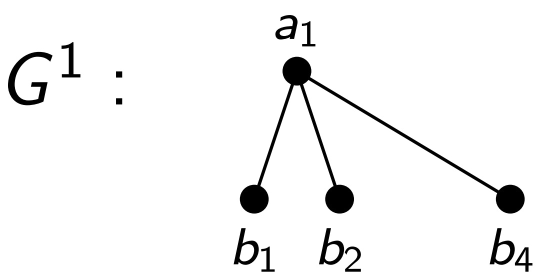


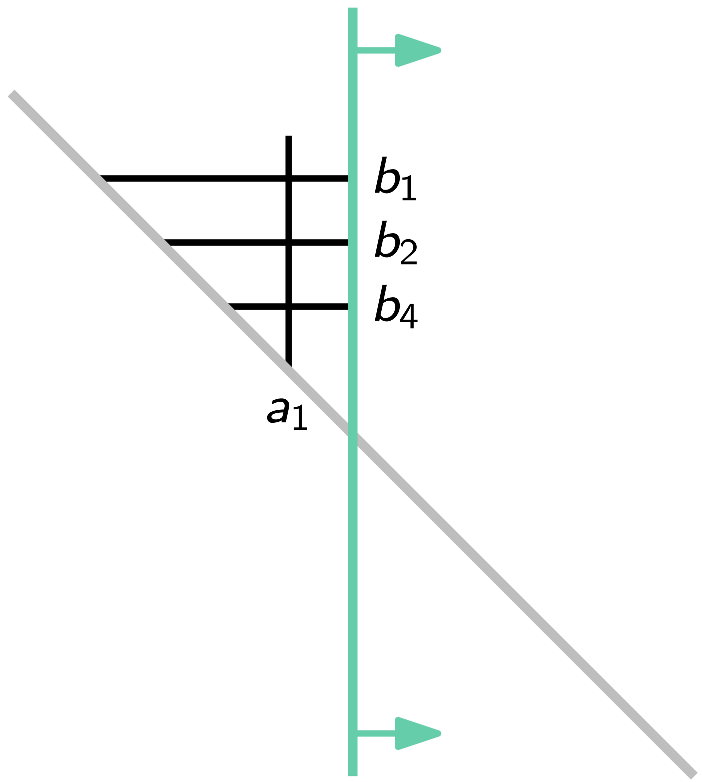


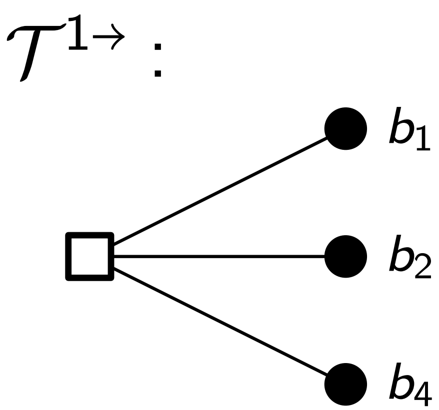
Example for STICK$_A$

$i = 1$ Event: $1{\rightarrow}$ $B^{1\rightarrow} = \{b_1, b_2, b_4\}$









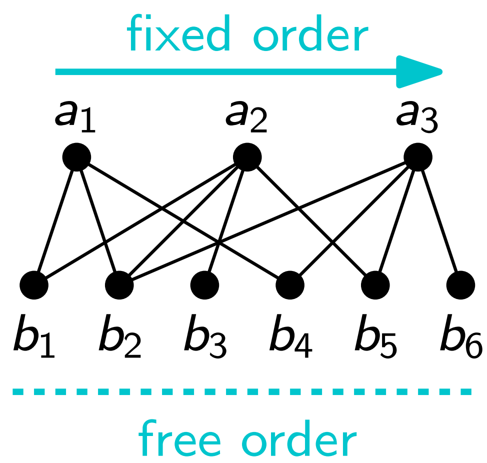
Example for STICK$_A$

$i = 2$

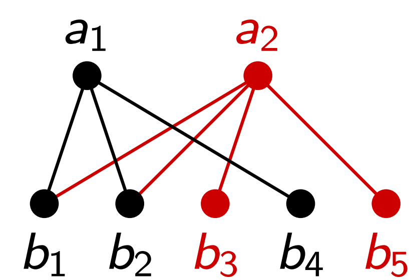
Event: 2

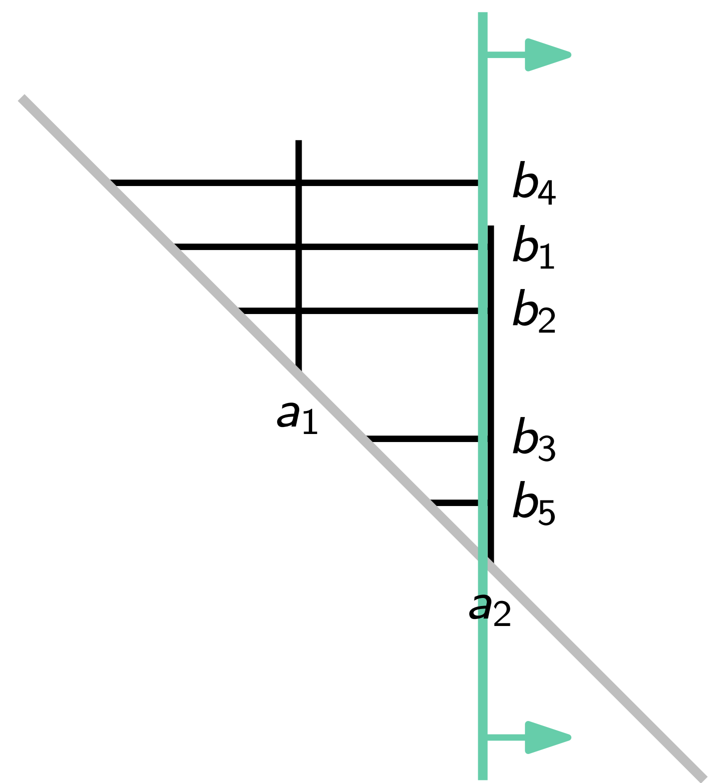
$B^2 = \{b_1, b_2, b_3, b_4, b_5\}$

G:

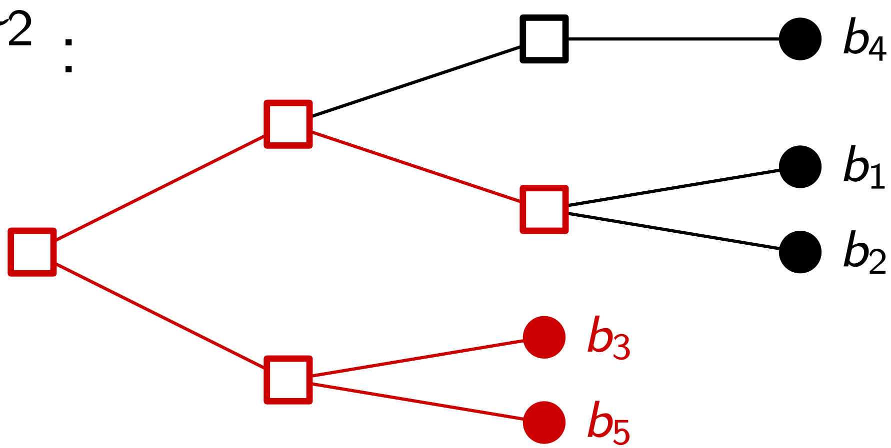


G^2 :





$\mathcal{T}^2$:



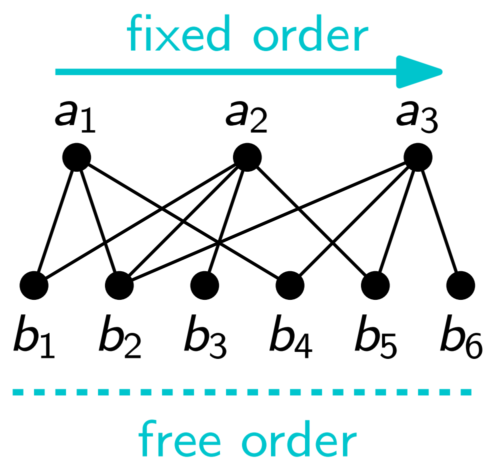
Example for STICK_A

$i = 2$

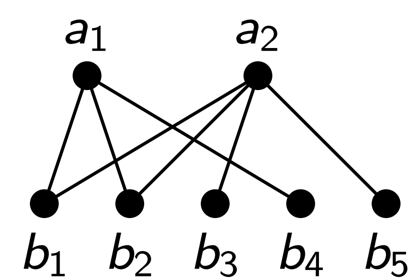
Event: $2{\rightarrow}$

$B^{2\rightarrow} = \{b_1, b_2, b_3, b_4, b_5\}$

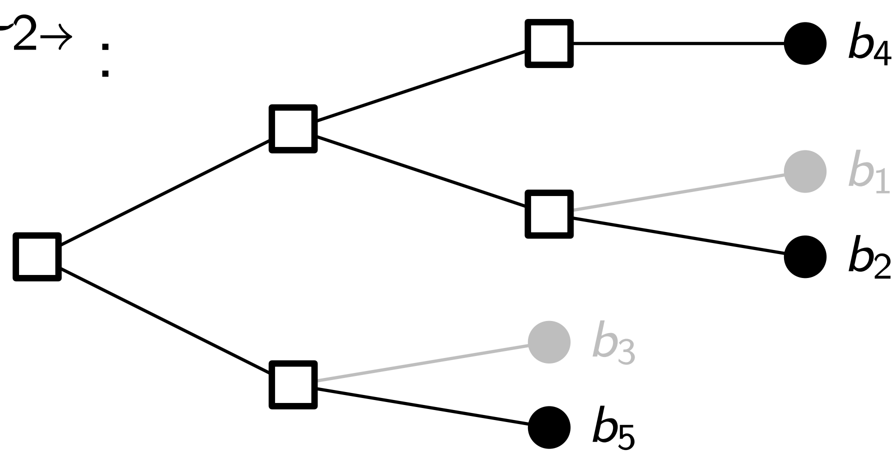
G:

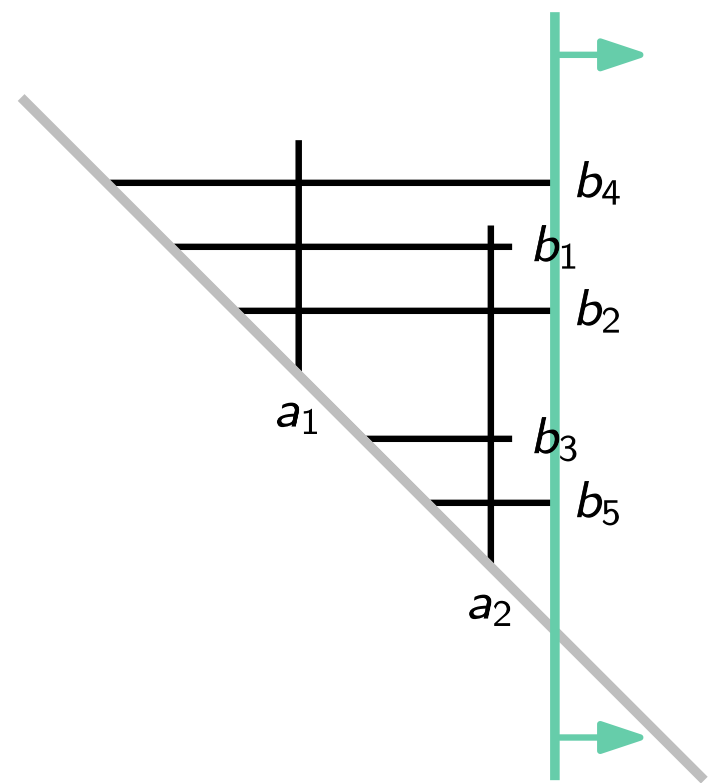


G^2 :



$\mathcal{T}^{2\rightarrow}$:

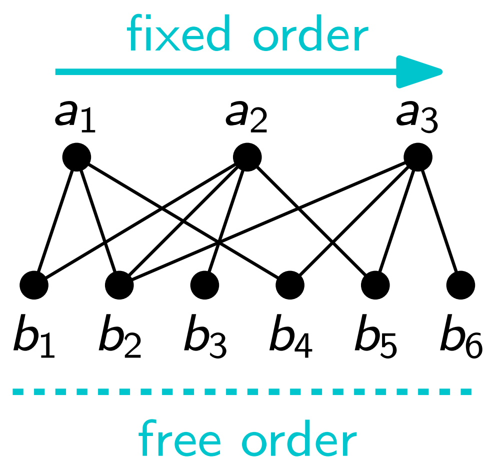




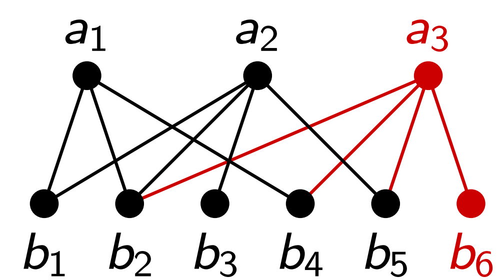
Example for STICK_A

$i = 3$ Event: 3 $B^3 = \{b_2, b_4, b_5, b_6\}$

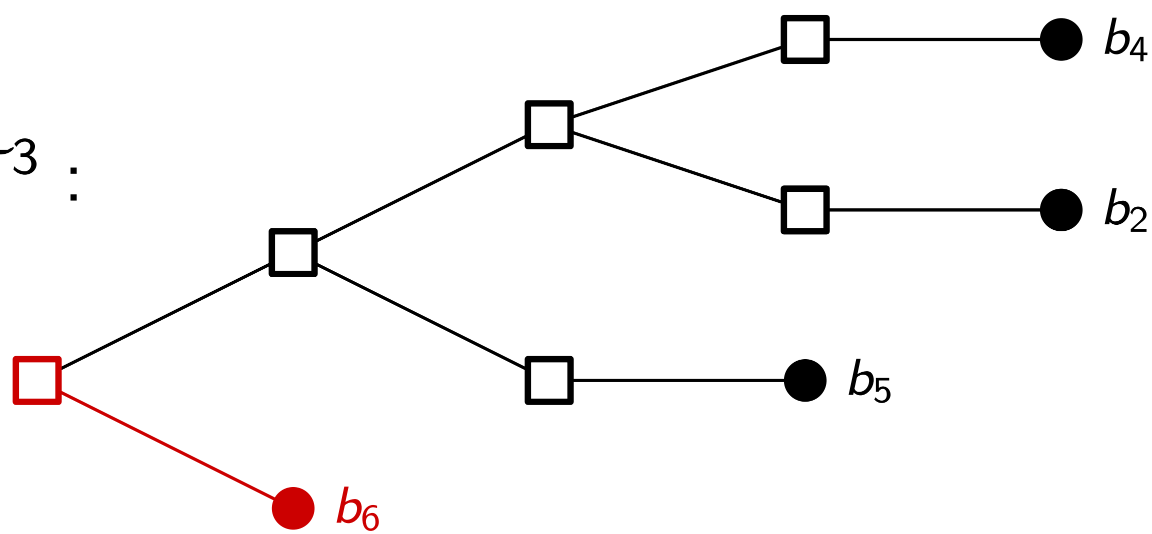
G:

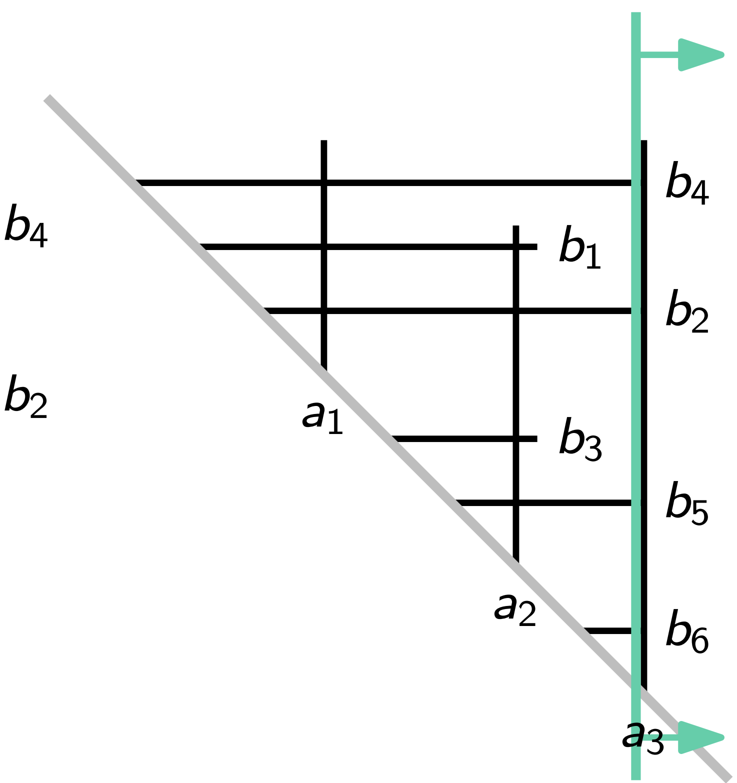


G^3 :



$\mathcal{T}^3$:

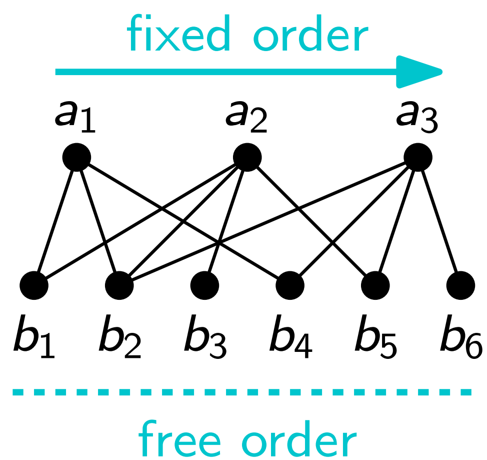




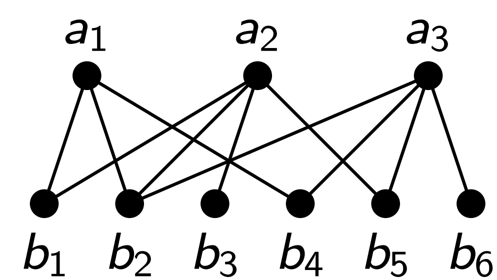
Example for STICK_A

$i = 3$ Event: $3{\rightarrow}$ $B^{3\rightarrow} = \{b_2, b_4, b_5, b_6\}$

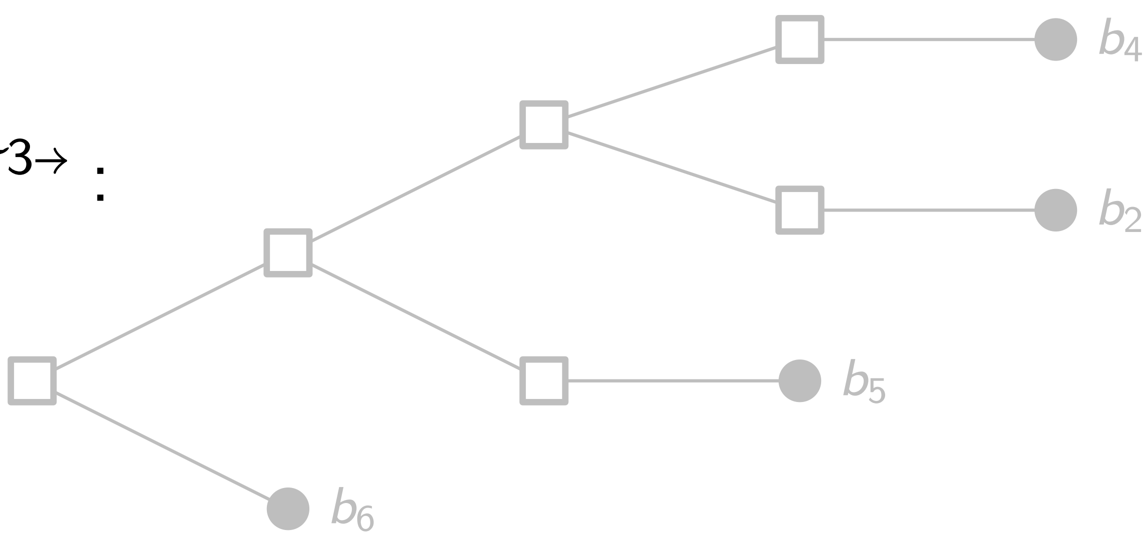
G:

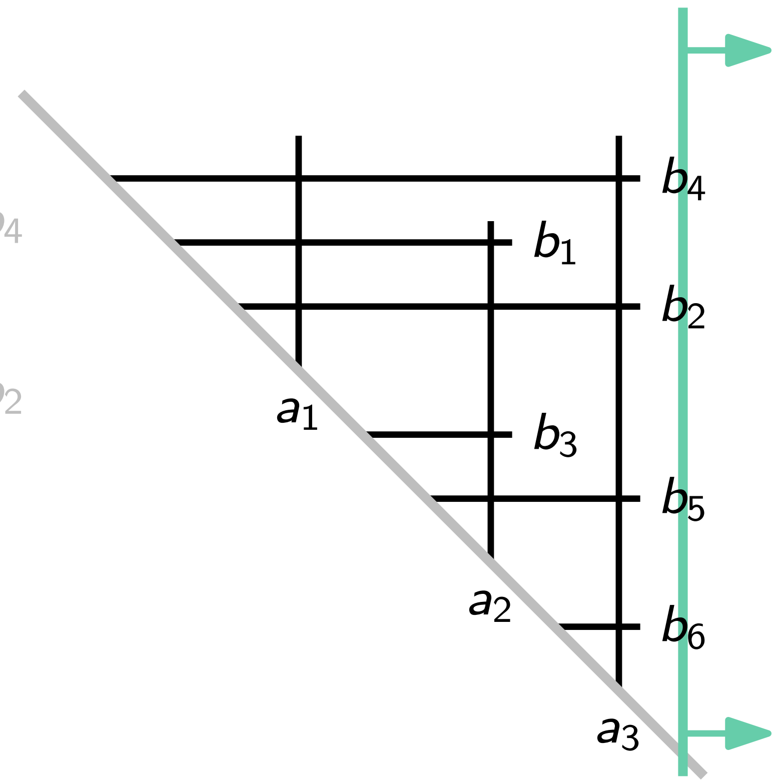


G^3 :



$\mathcal{T}^{3\rightarrow}$:

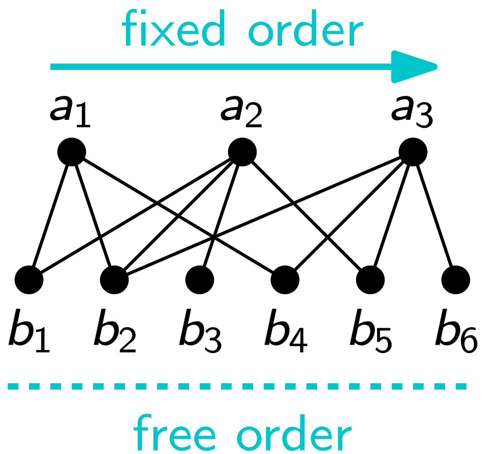


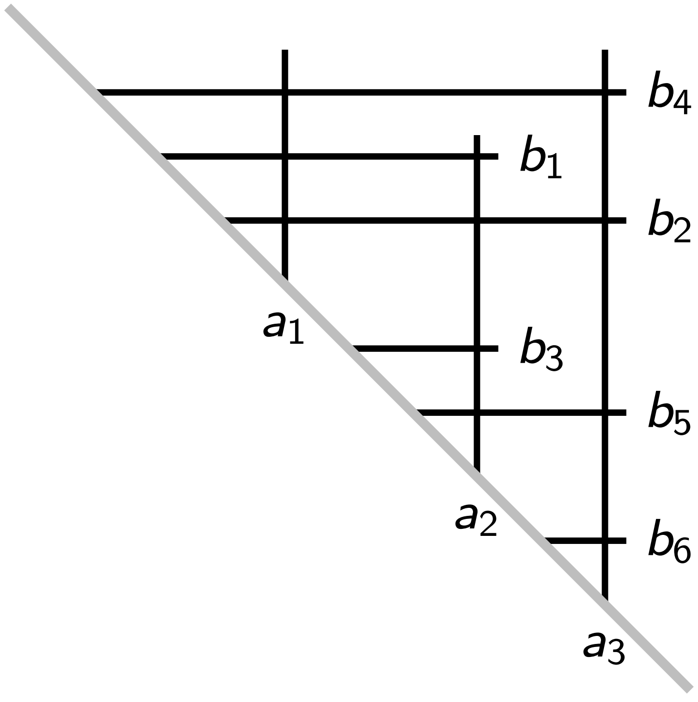


Example for STICK_A

$i = 3$ Event: *End*

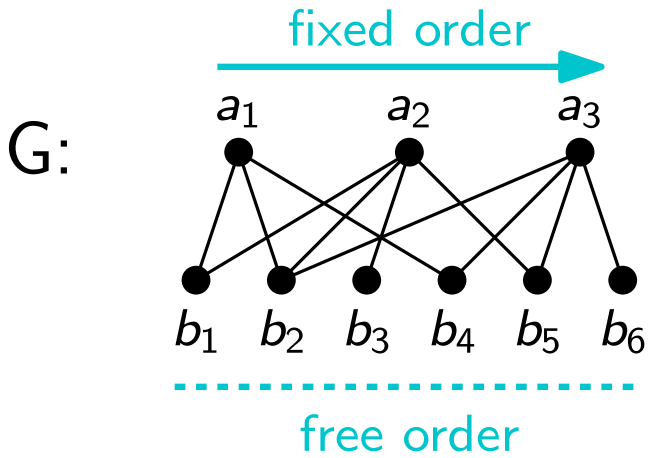
G:



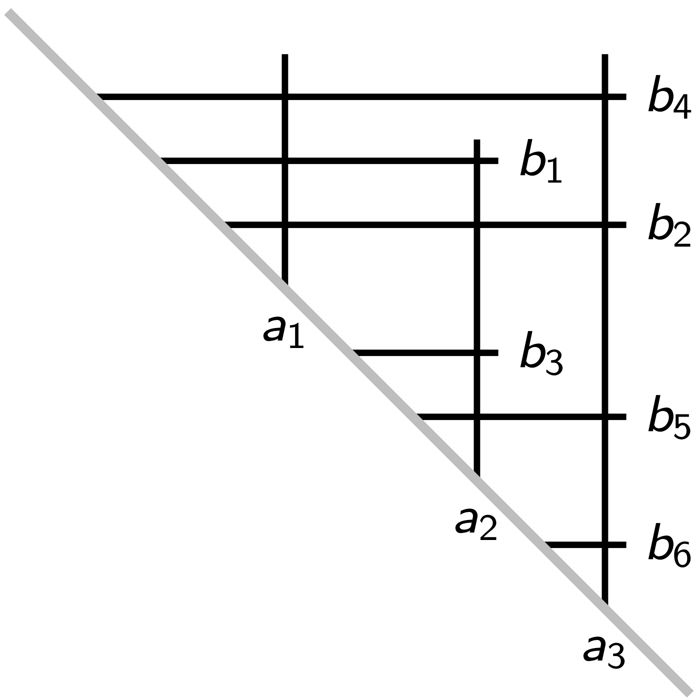


Example for STICK_A

$i = 3$ Event: *End*



Runtime in $O(|A| \cdot |B|)$



$\mathsf{STICK}_{\mathsf{AB}}^{\mathsf{fix}}$ with isolated vertices

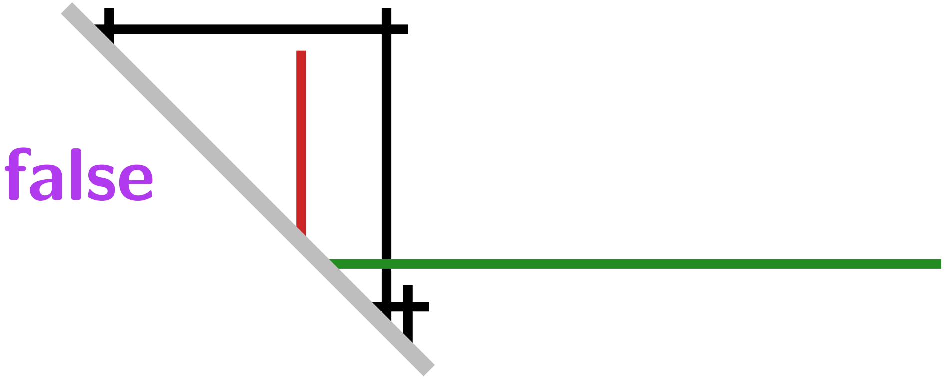
$\star$	$\mathsf{STICK}_\star$	$\mathsf{STICK}_\star^{\mathsf{fix}}$
	?	NP-complete
A	$O(\|A\|\|B\|)$	NP-complete
AB	$O(\|E\|)$	in general: NP-complete w/o isolated vtc.: $O((\|A\|+\|B\|)^2)$

Hardness of STICK$^{\mathsf{fix}}_{\mathsf{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT

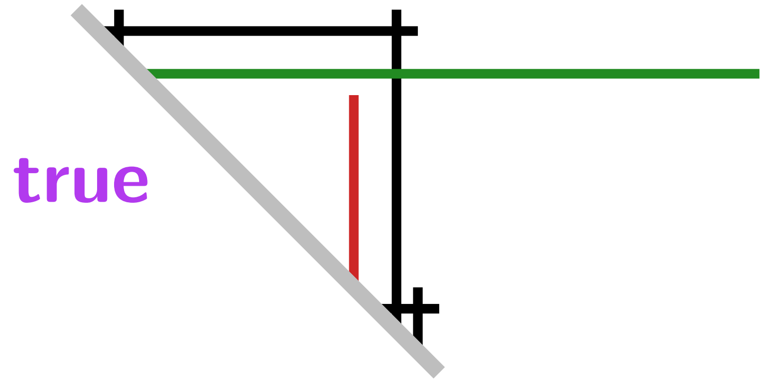
Hardness of $\mathrm{STICK}^{\mathrm{fix}}_{\mathrm{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:



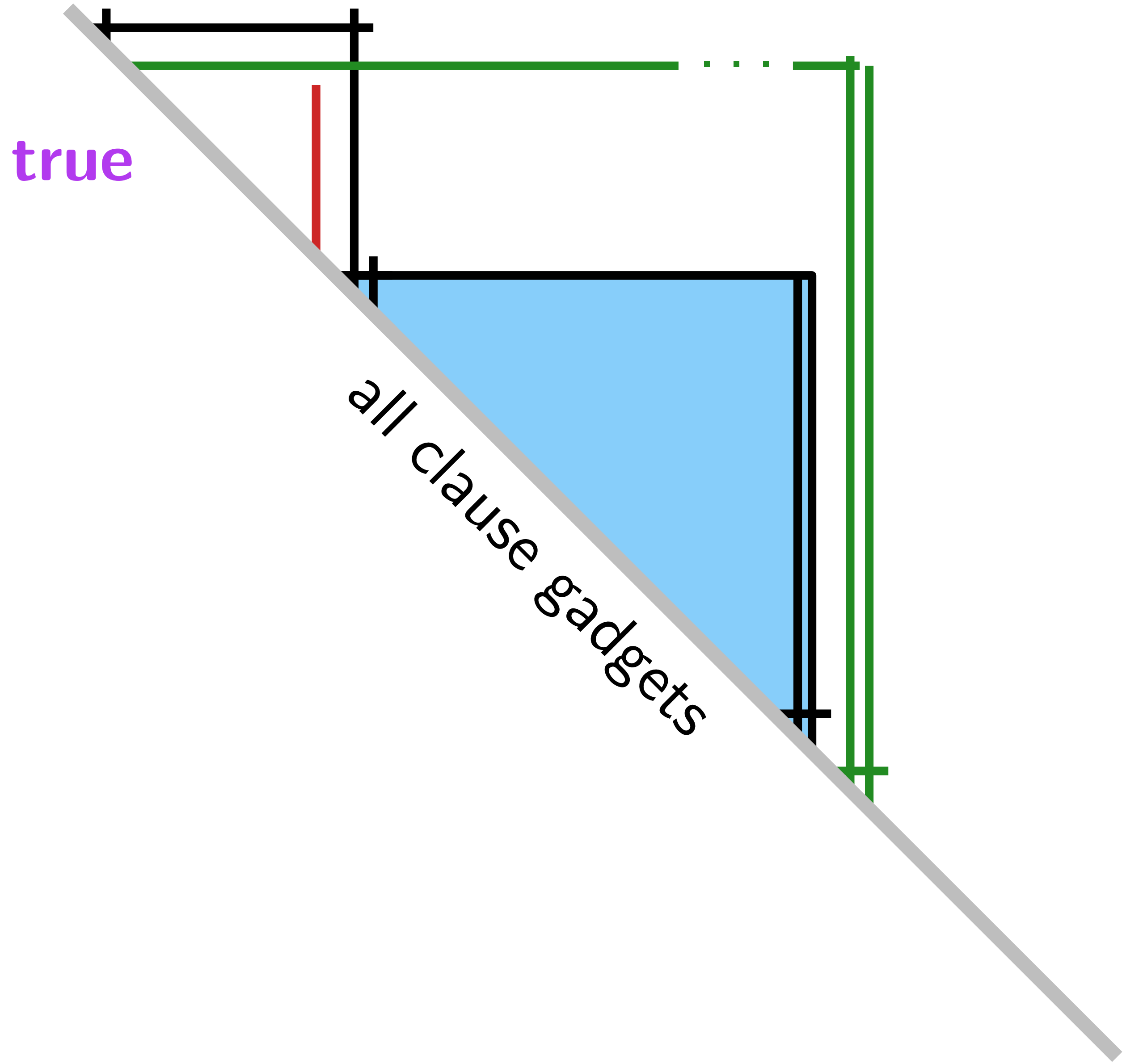
Hardness of STICK$^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:



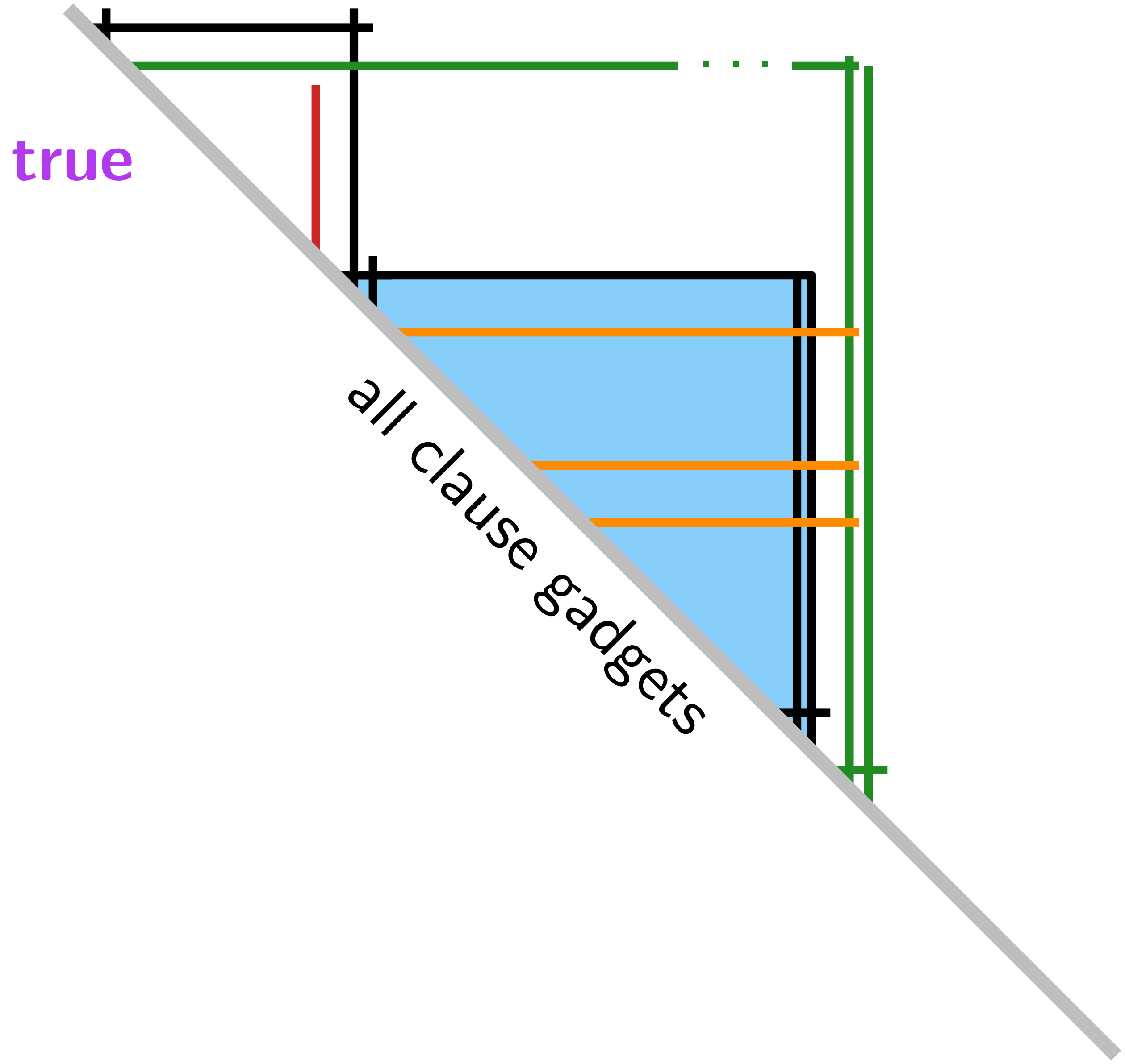
Hardness of STICK$^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:



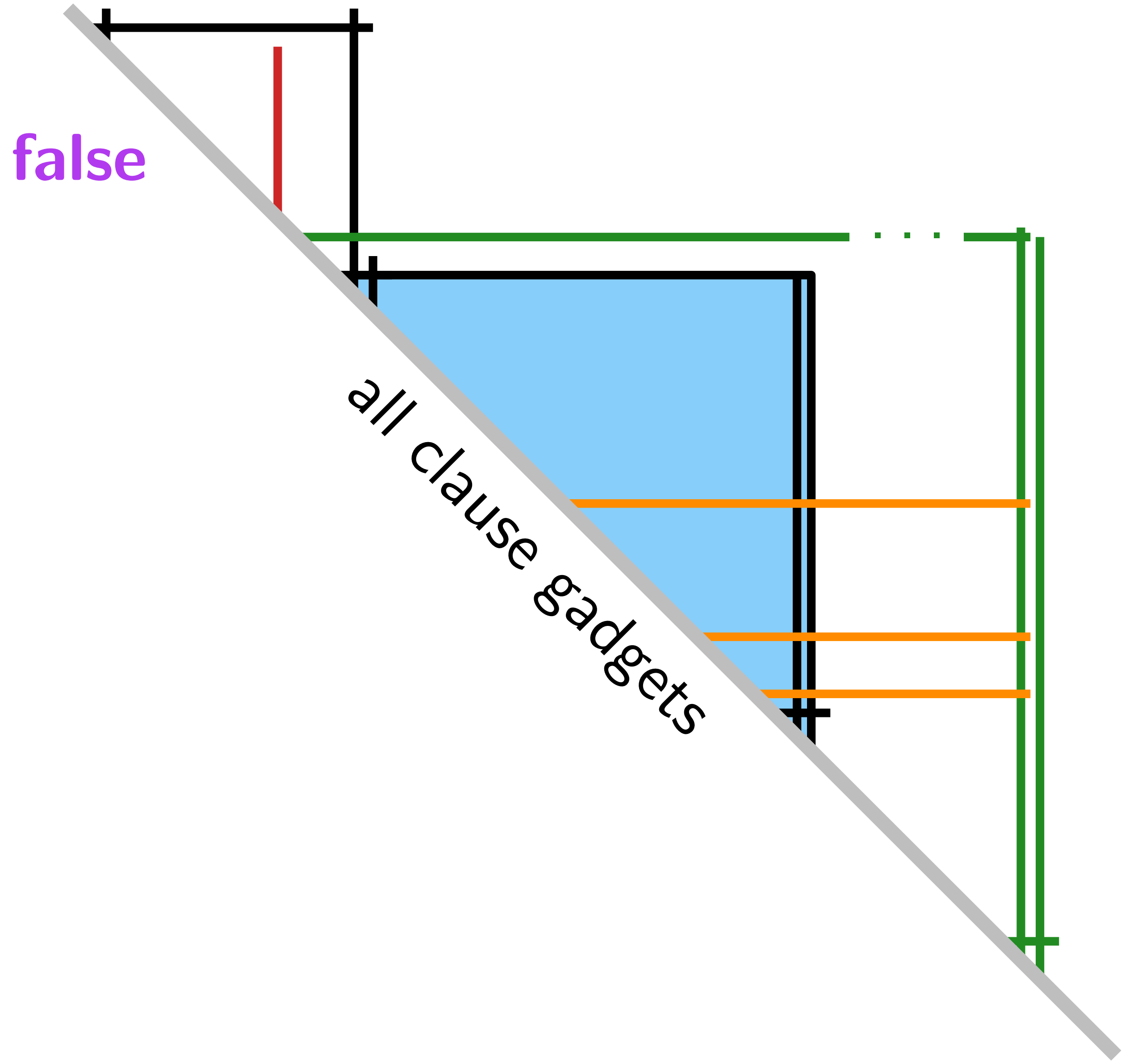
Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:



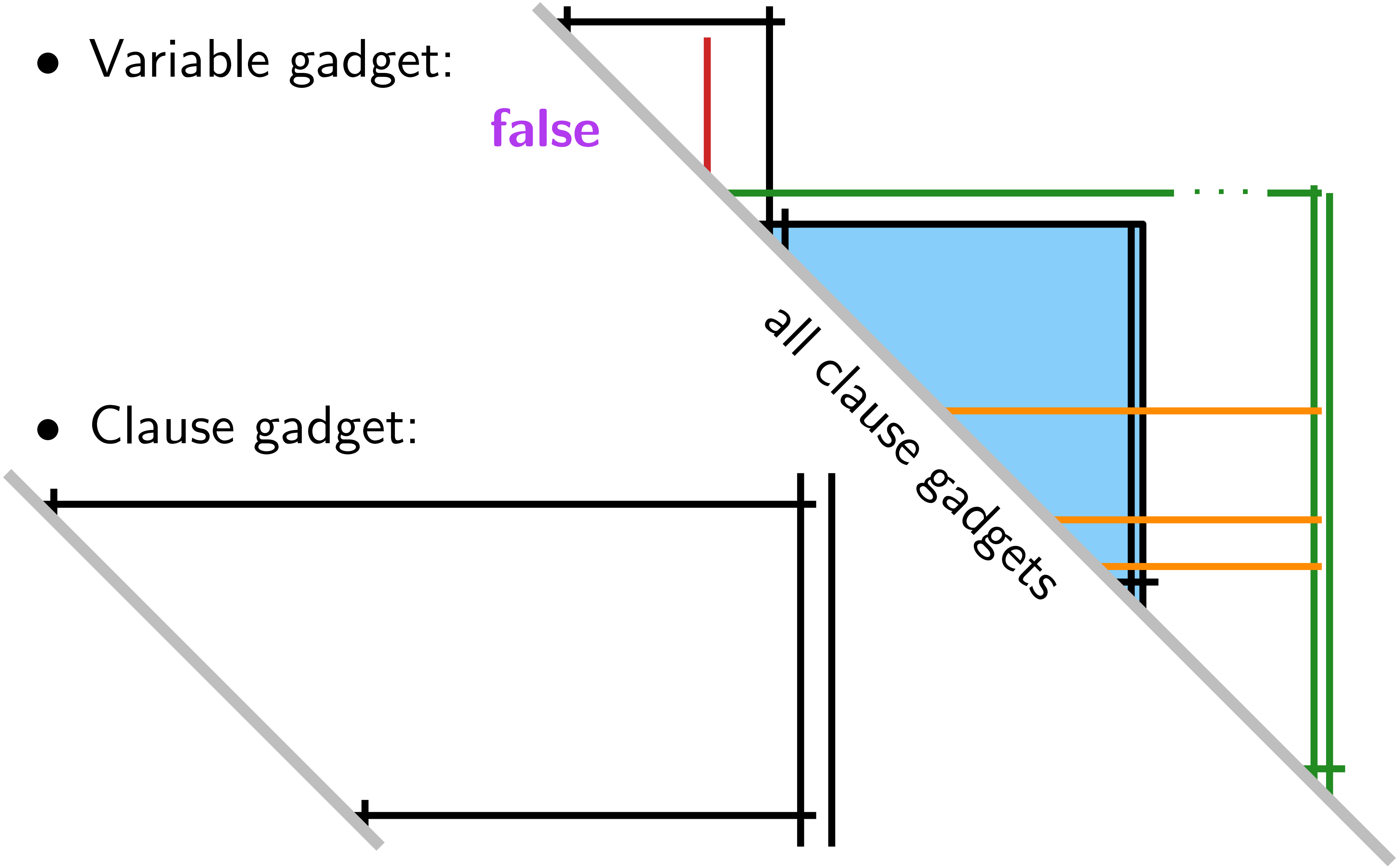
Hardness of STICK$^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:



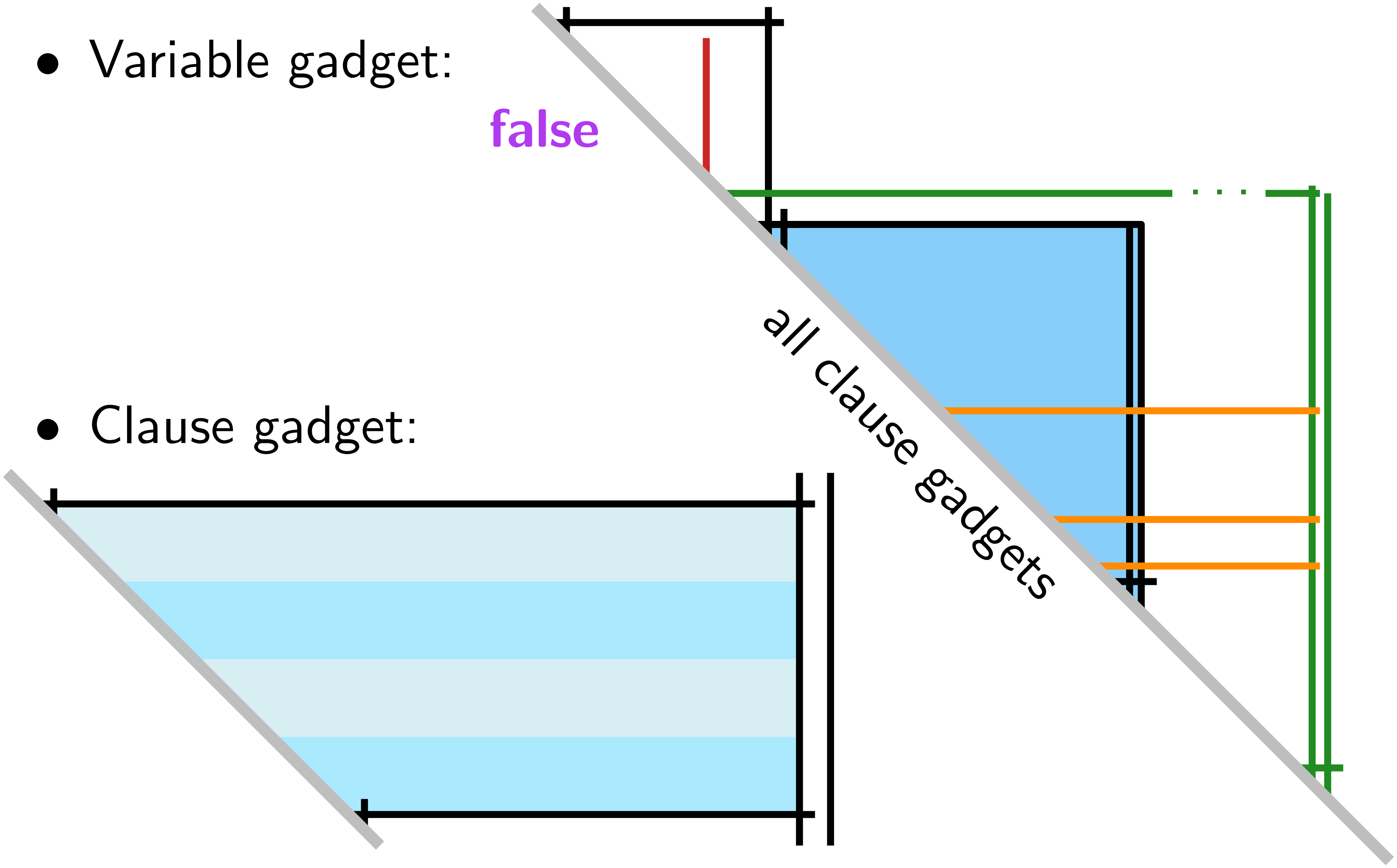
Hardness of $\text{STICK}^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



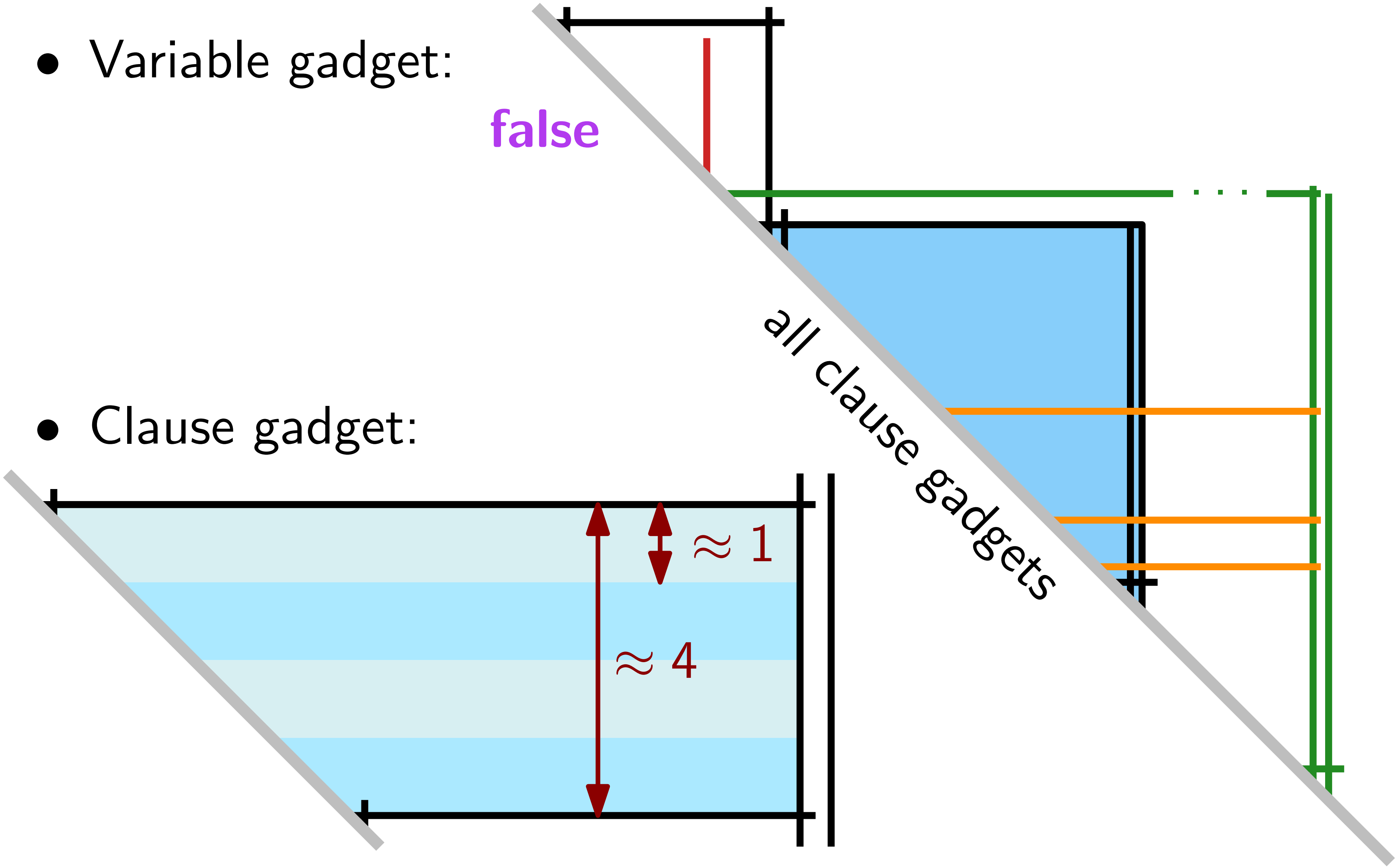
Hardness of $\mathrm{STICK}^{\mathrm{fix}}_{\mathrm{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



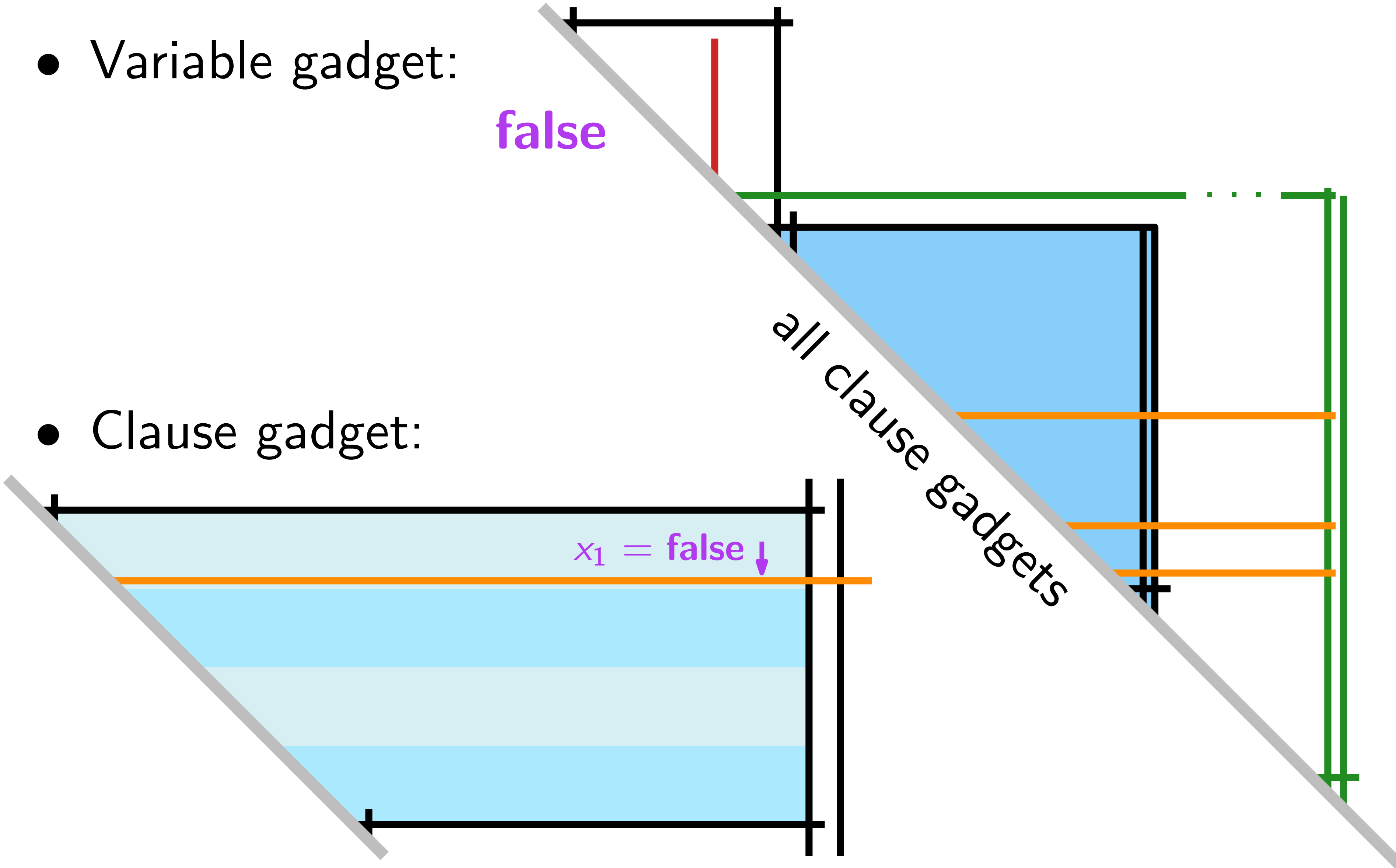
Hardness of $\text{STICK}^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



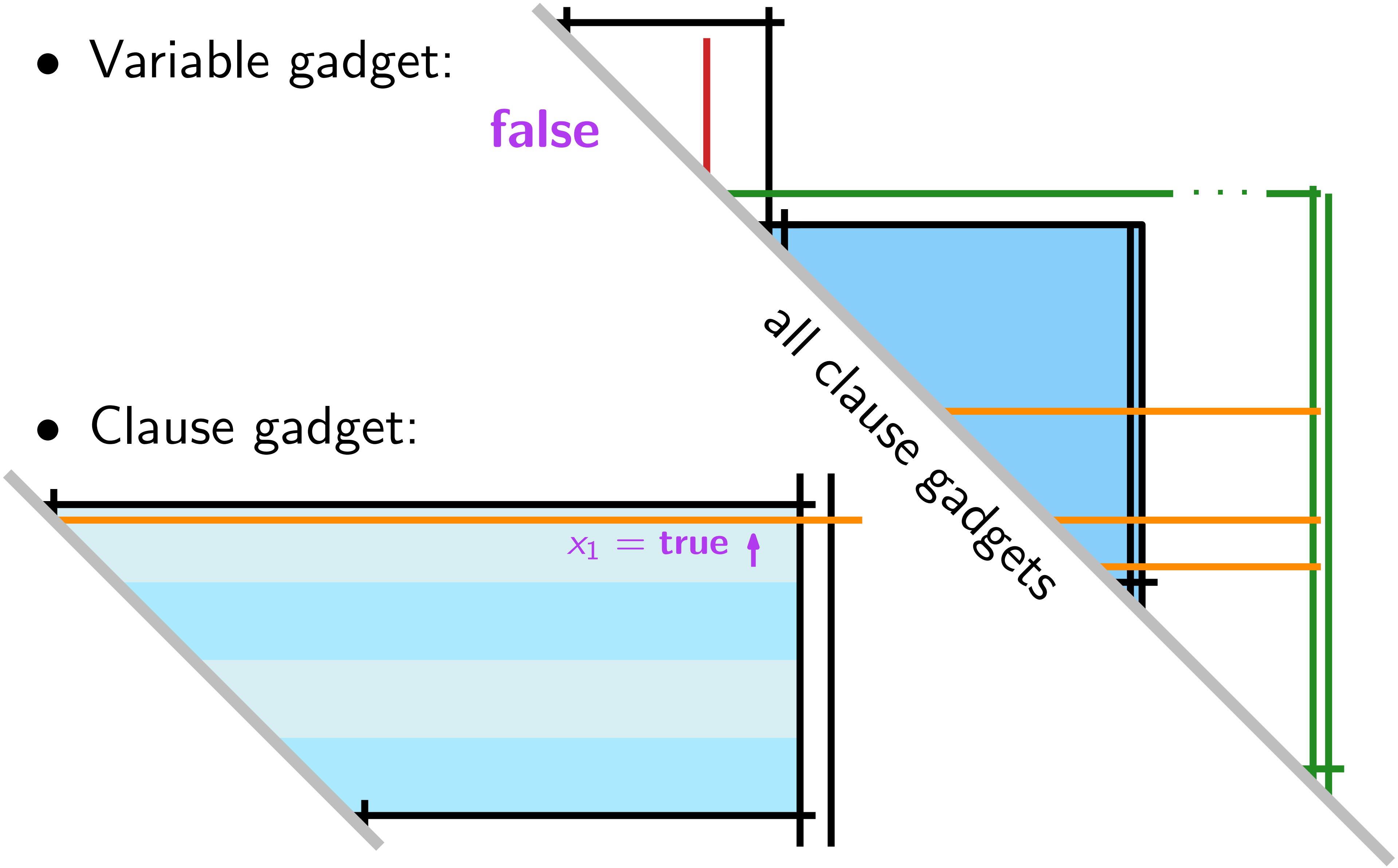
Hardness of STICK$^{\mathrm{fix}}_{\mathrm{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



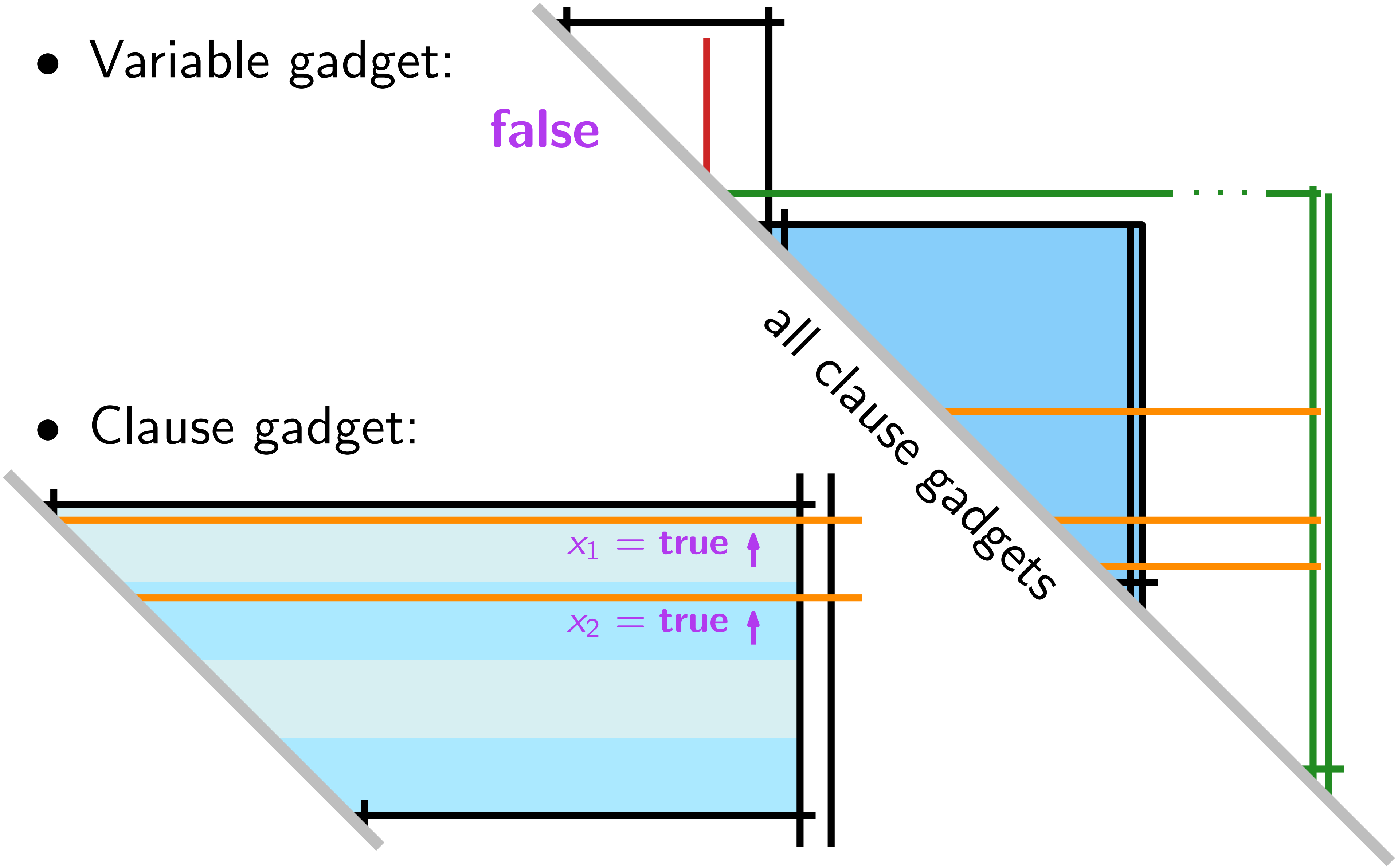
Hardness of $\text{STICK}^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



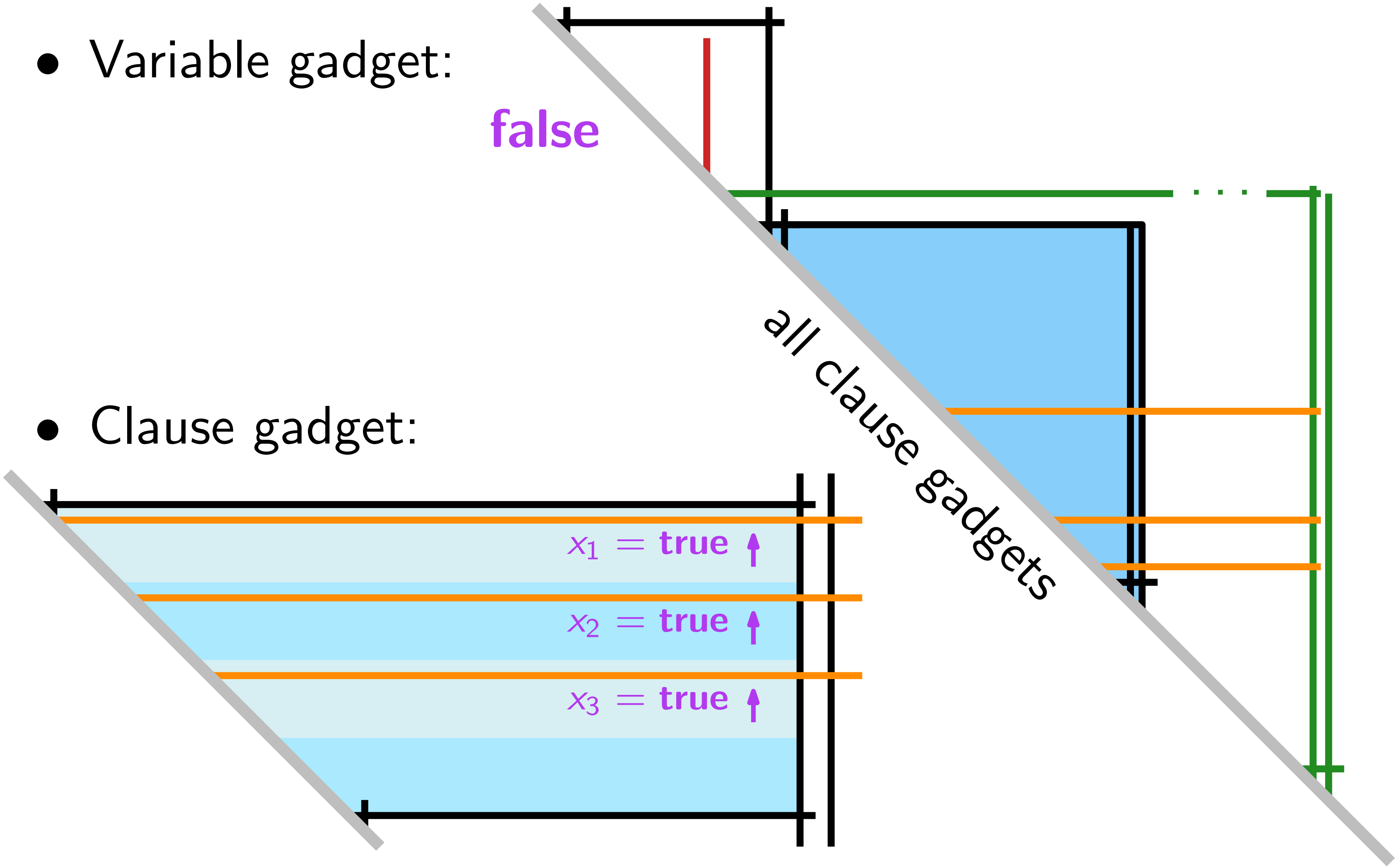
Hardness of $\mathrm{STICK}^{\mathrm{fix}}_{\mathrm{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



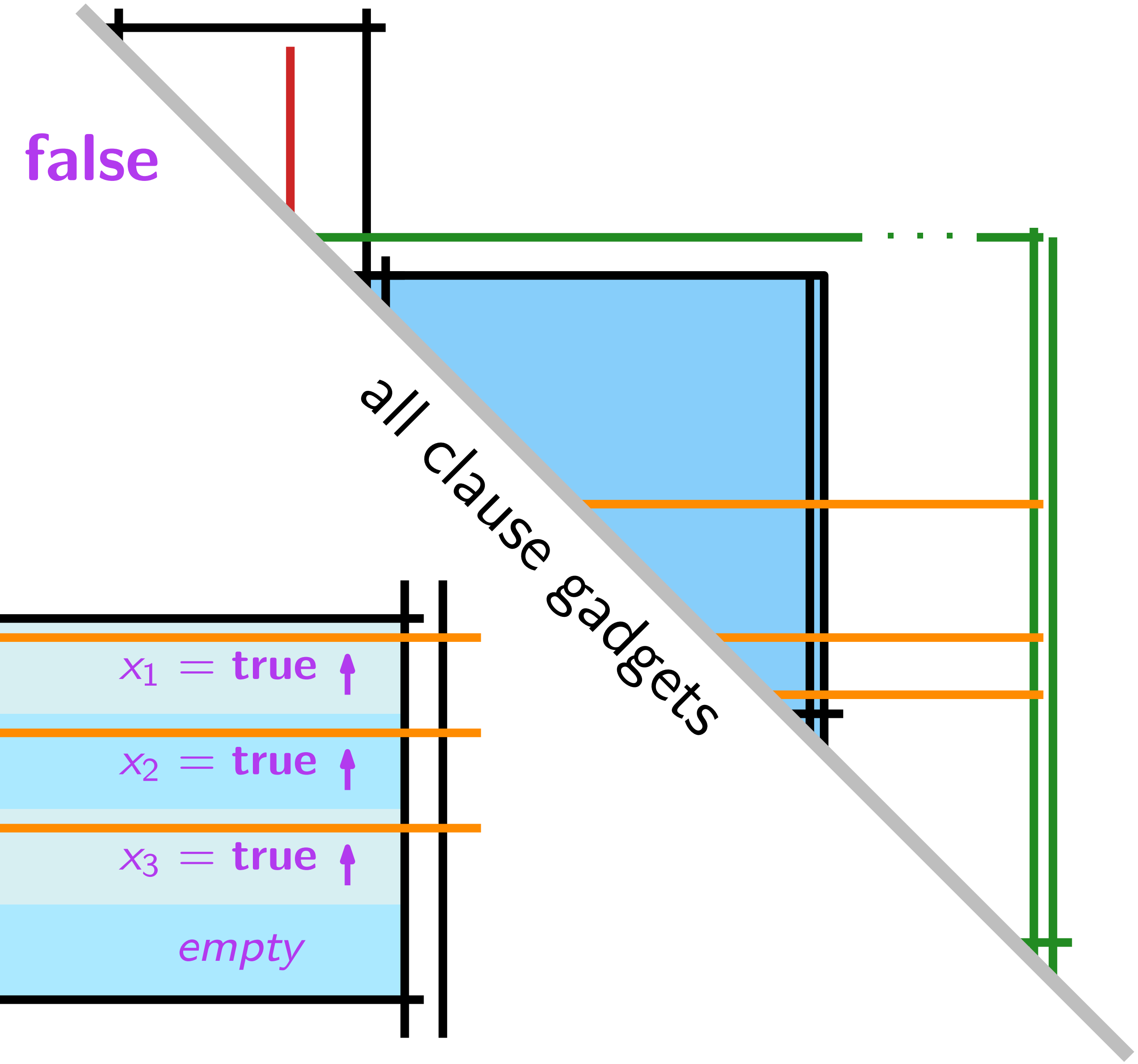
Hardness of $\text{STICK}^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:

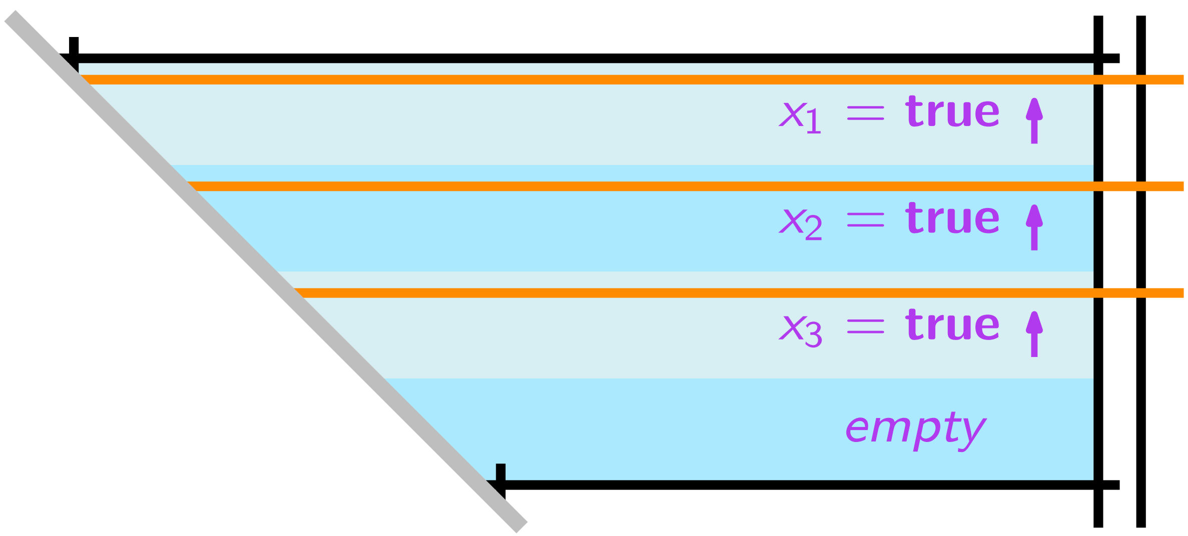


Hardness of $\text{STICK}^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:

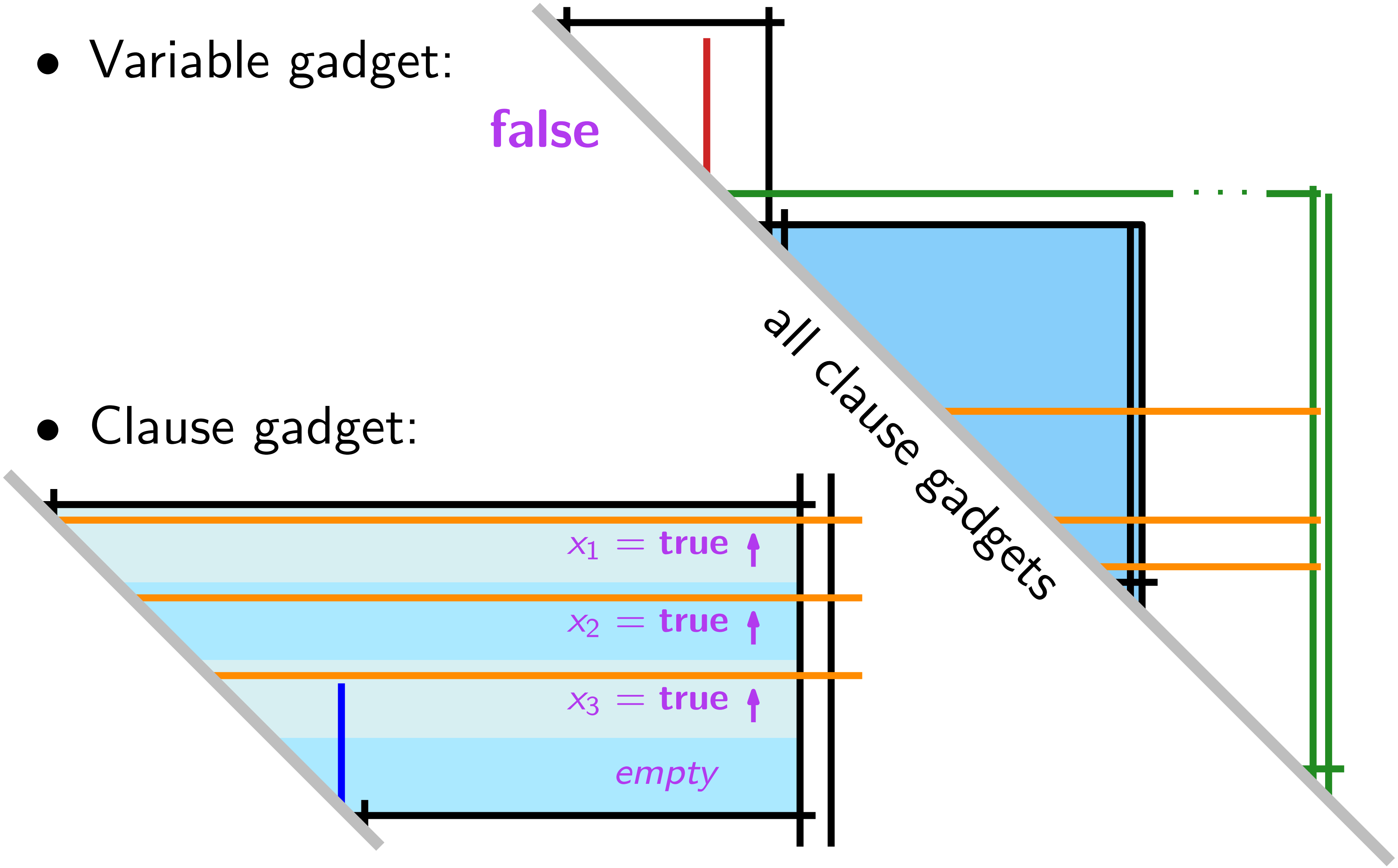


- Clause gadget:



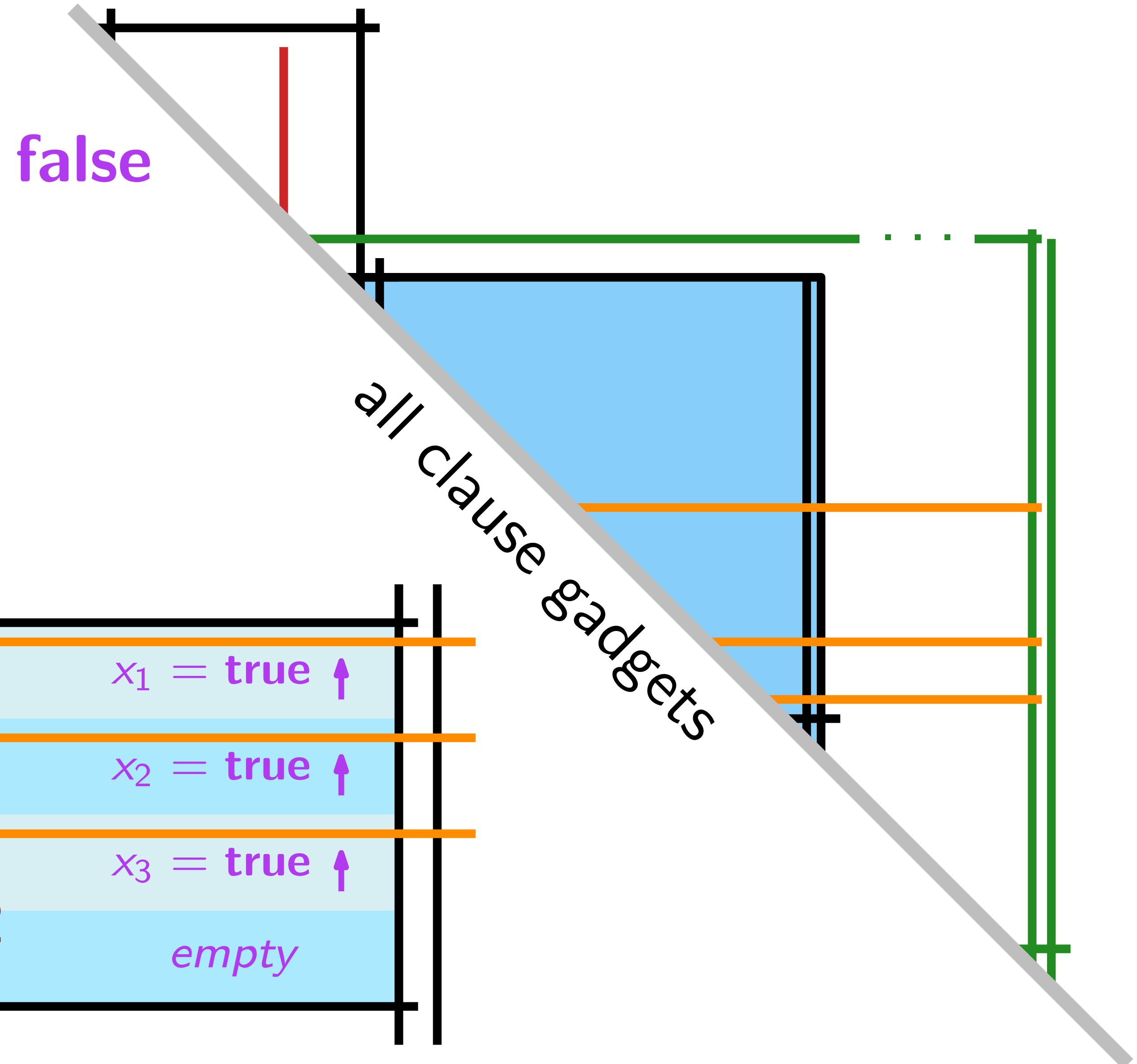
Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:

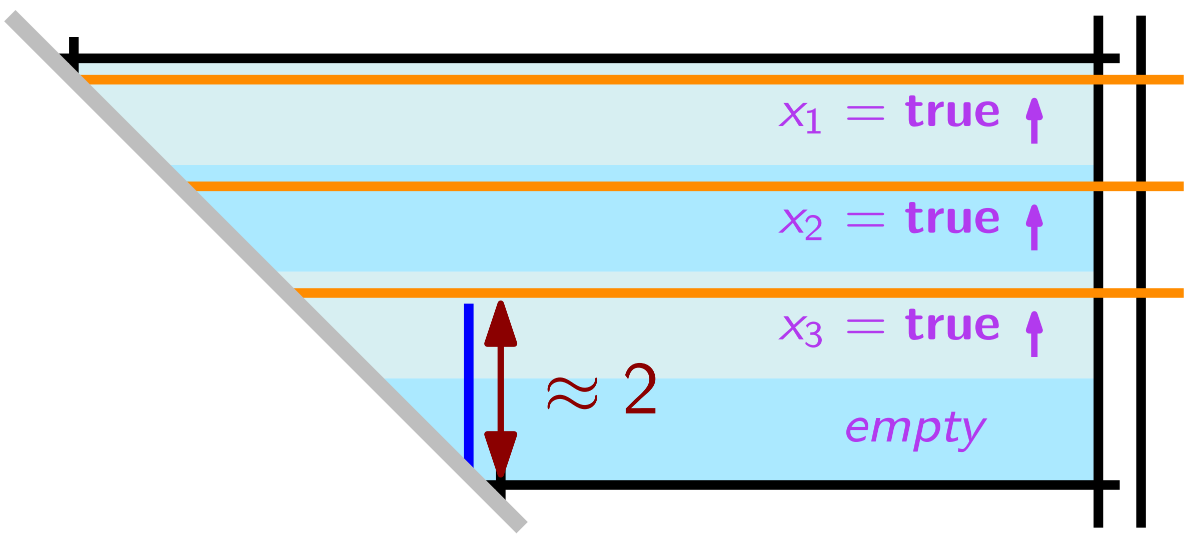


Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:

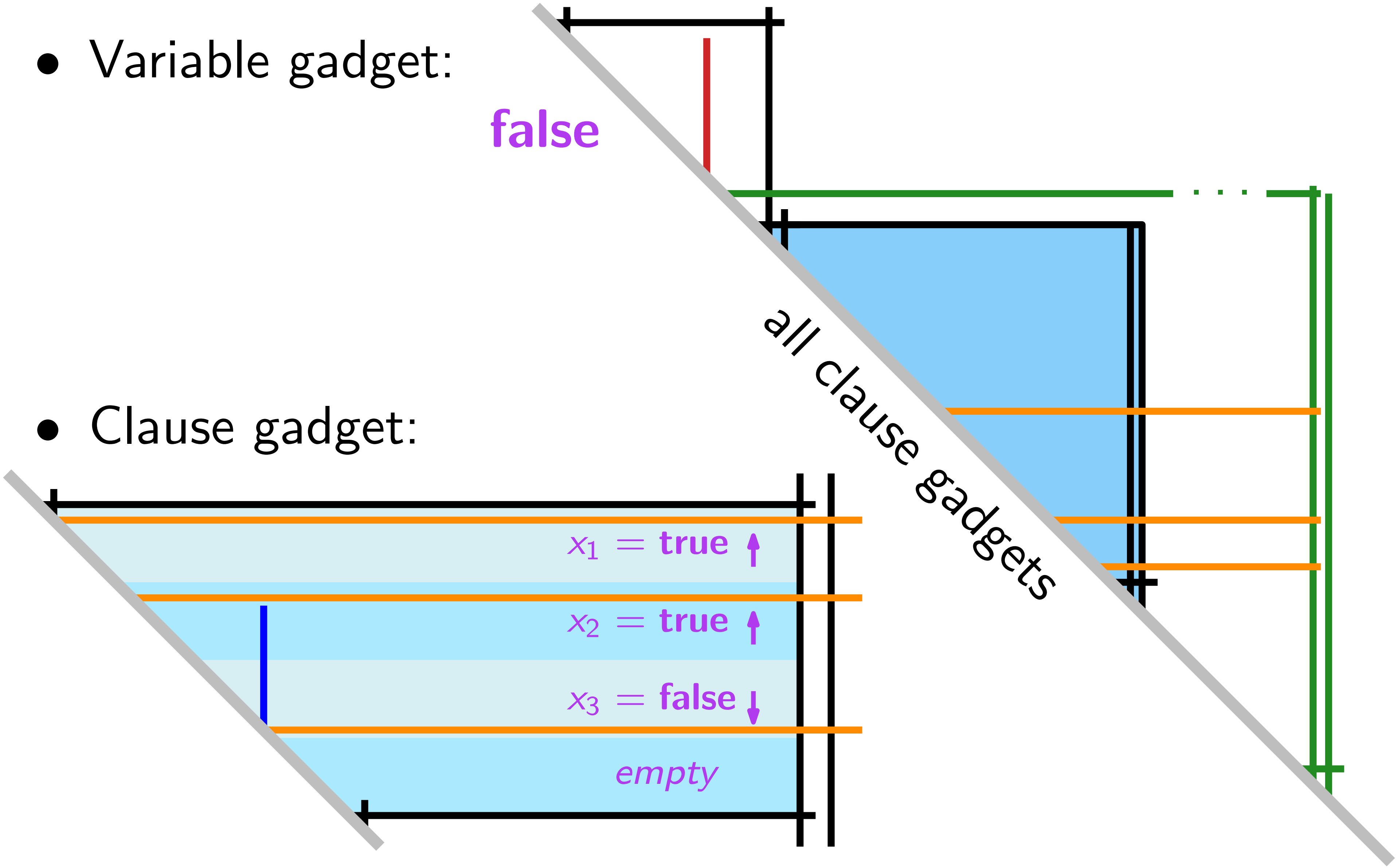


- Clause gadget:



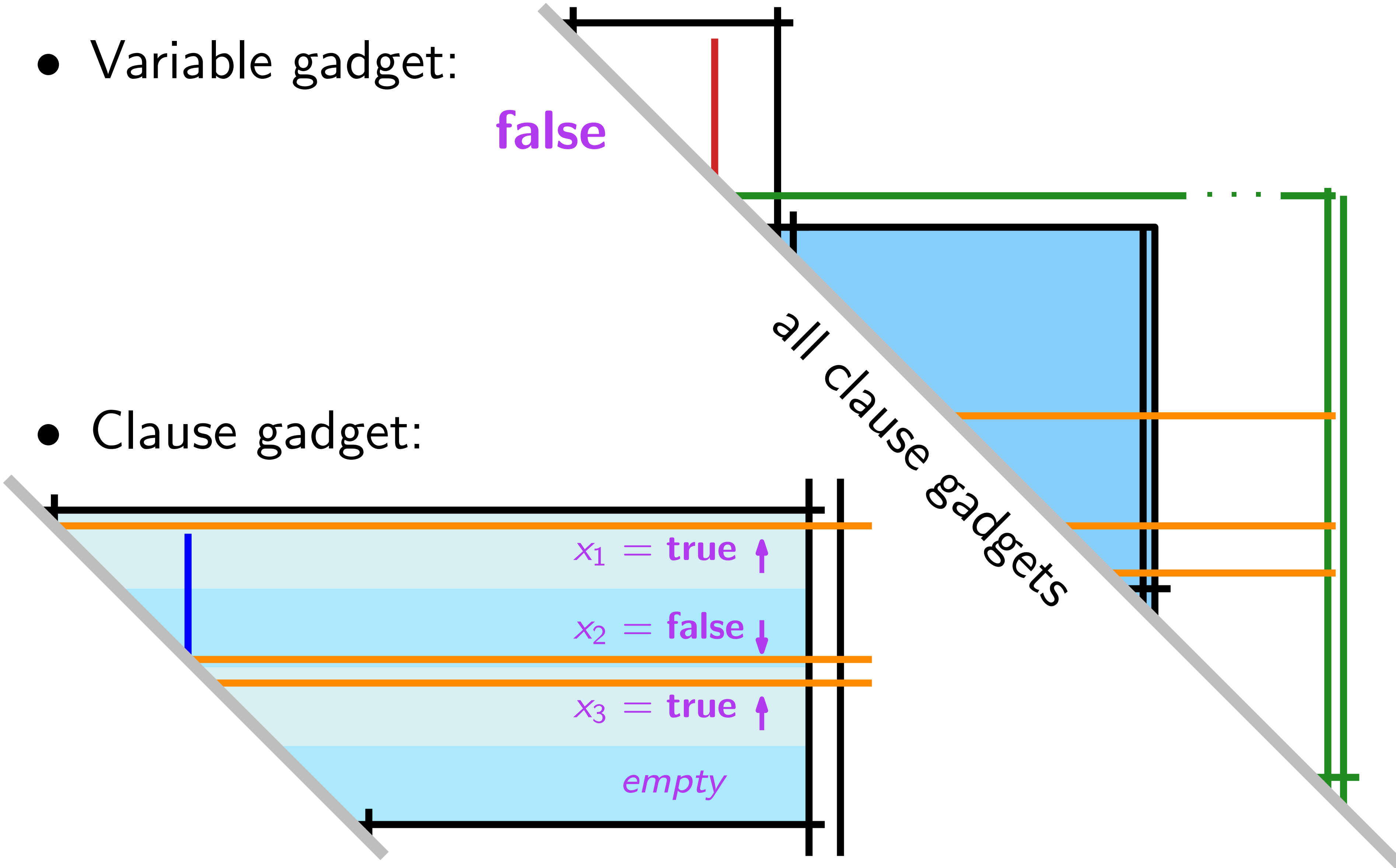
Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



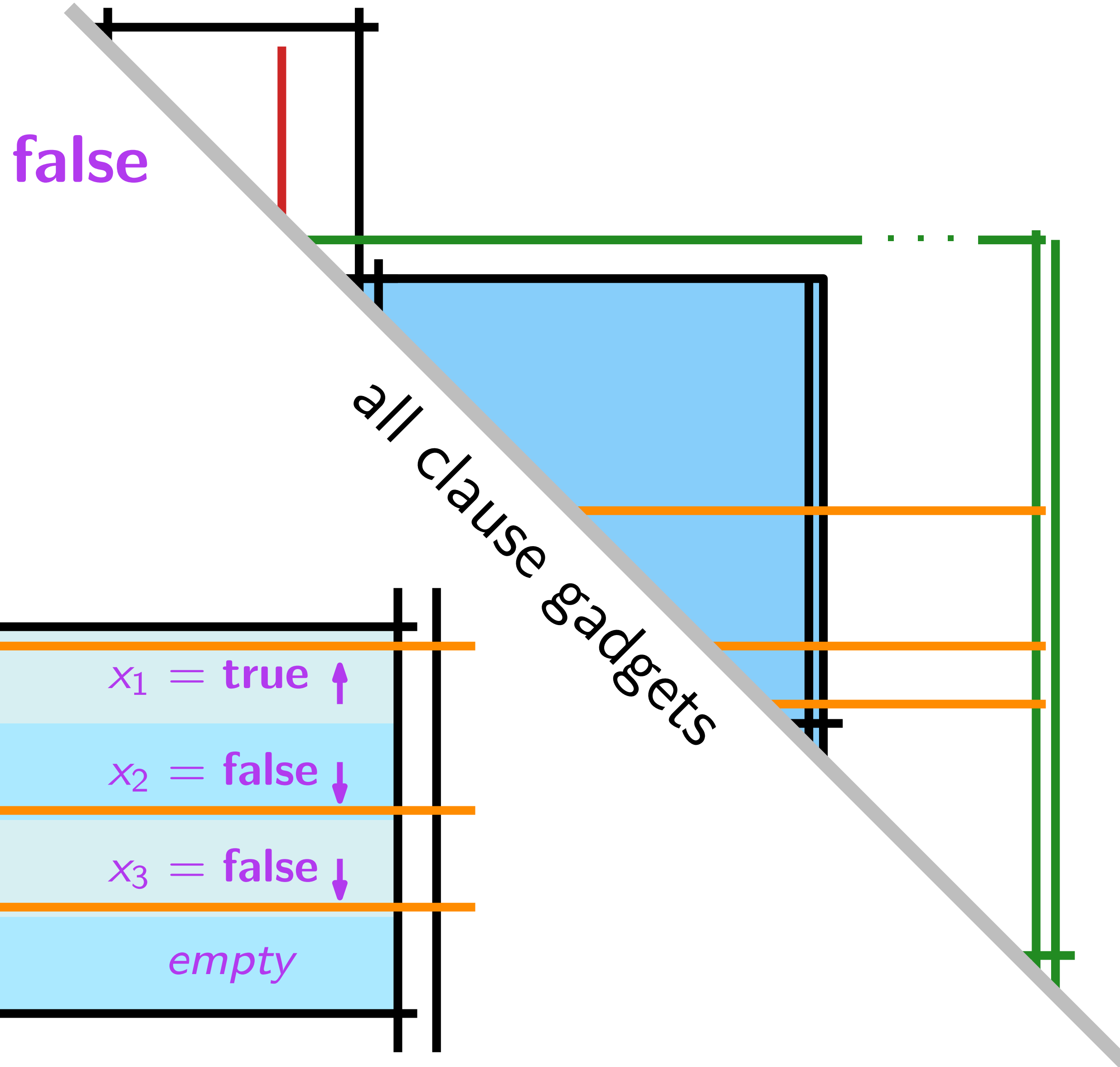
Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:

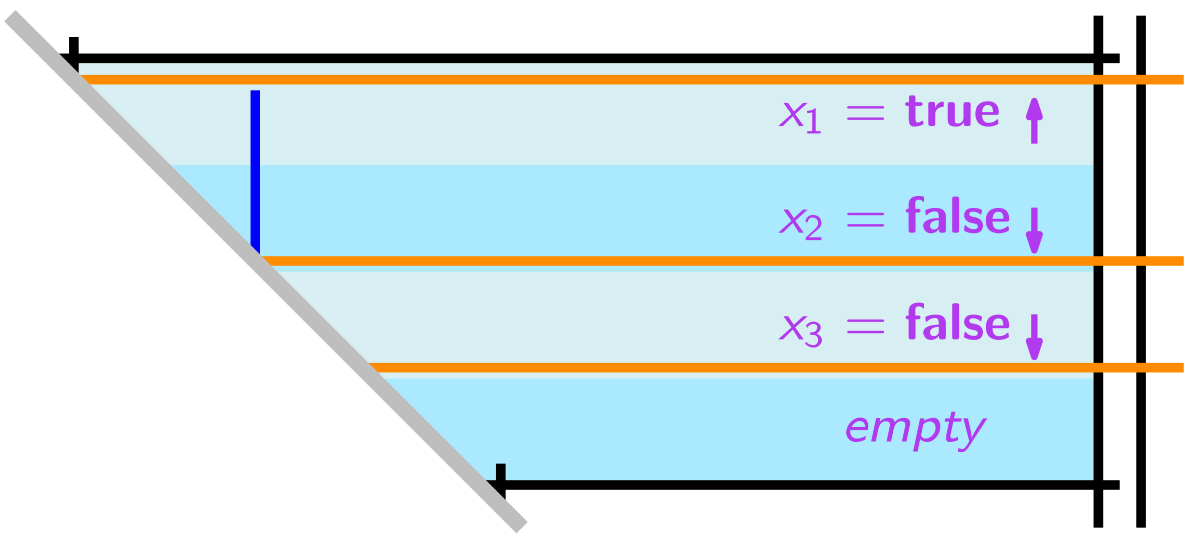


Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:

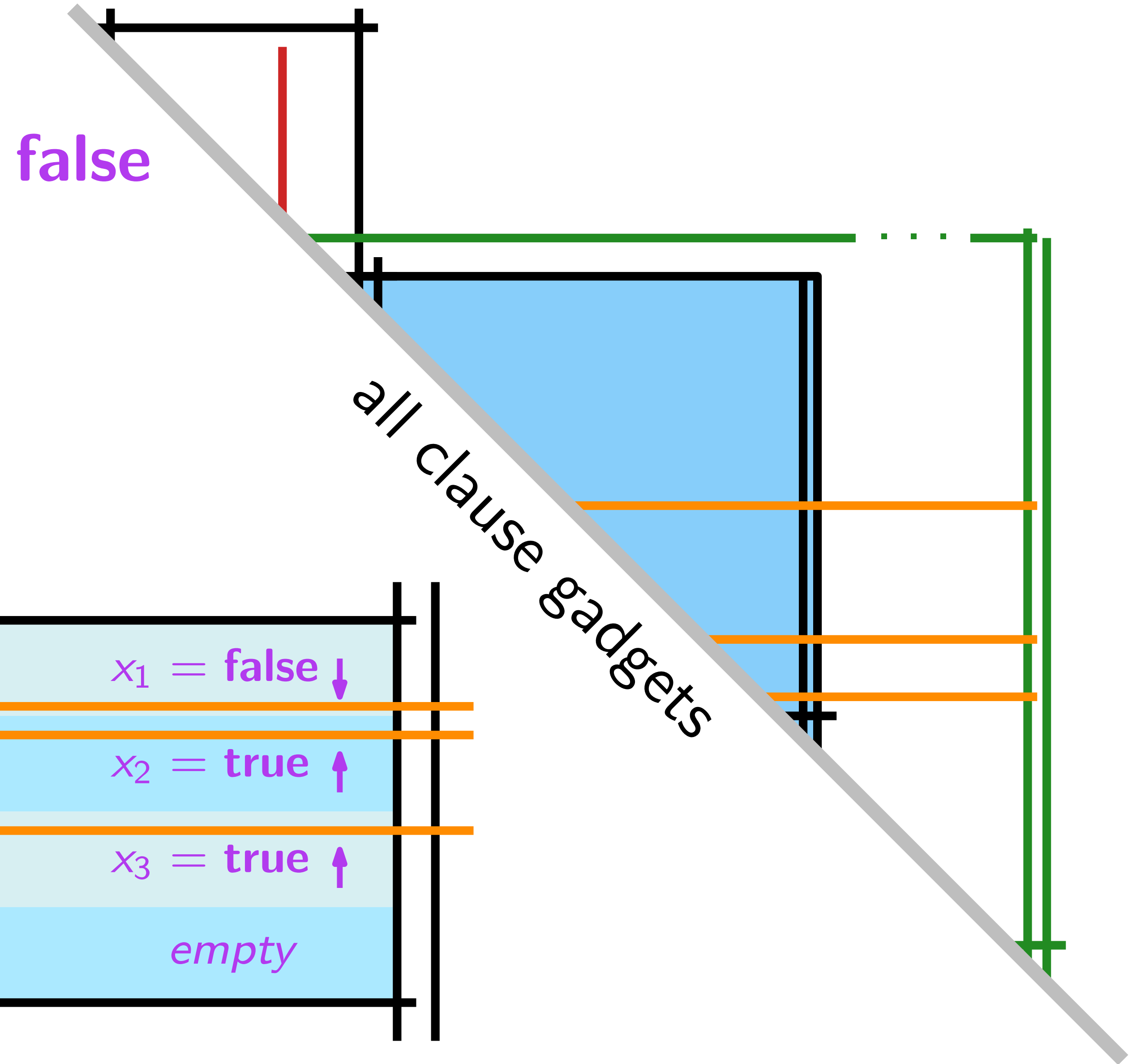


- Clause gadget:

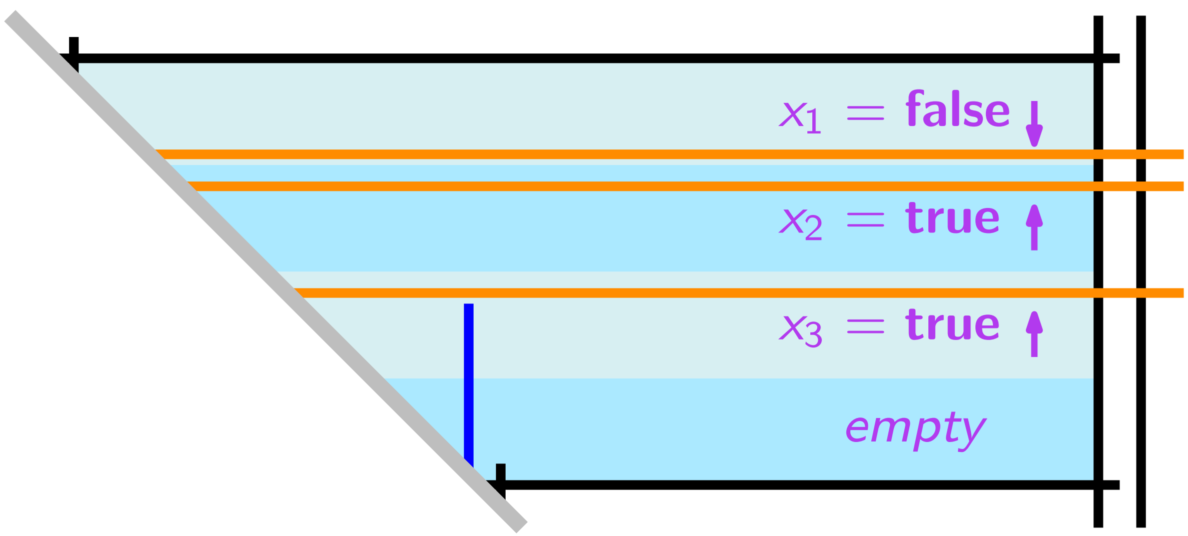


Hardness of $\mathrm{STICK}_{\mathrm{AB}}^{\mathrm{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:

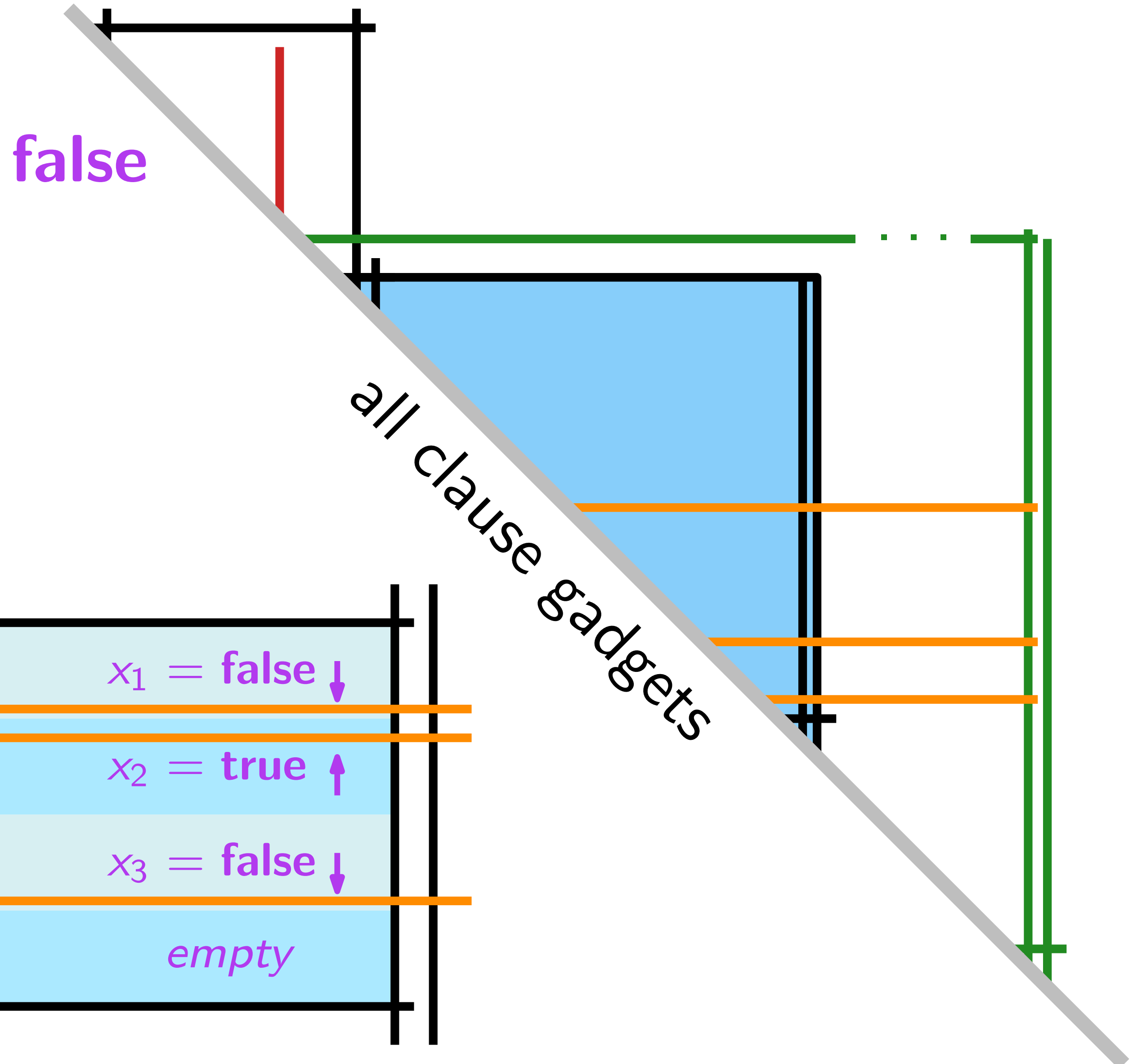


- Clause gadget:

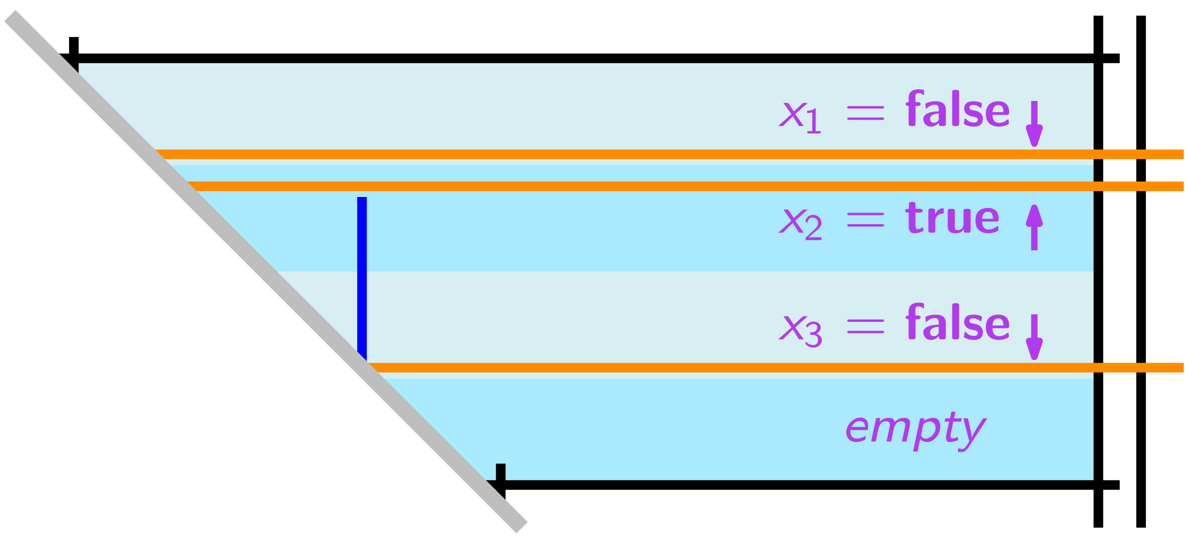


Hardness of STICK$^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:

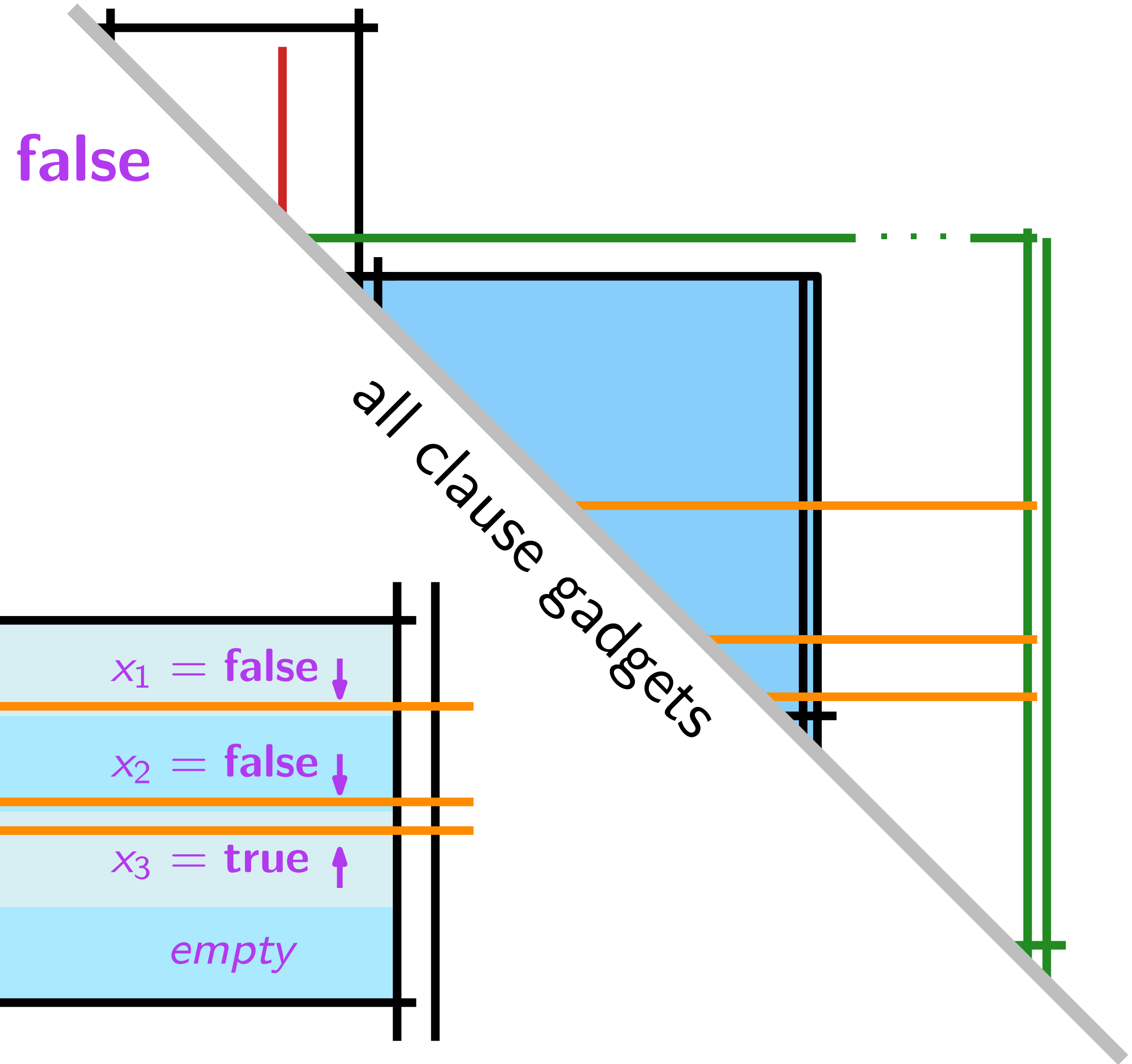


- Clause gadget:

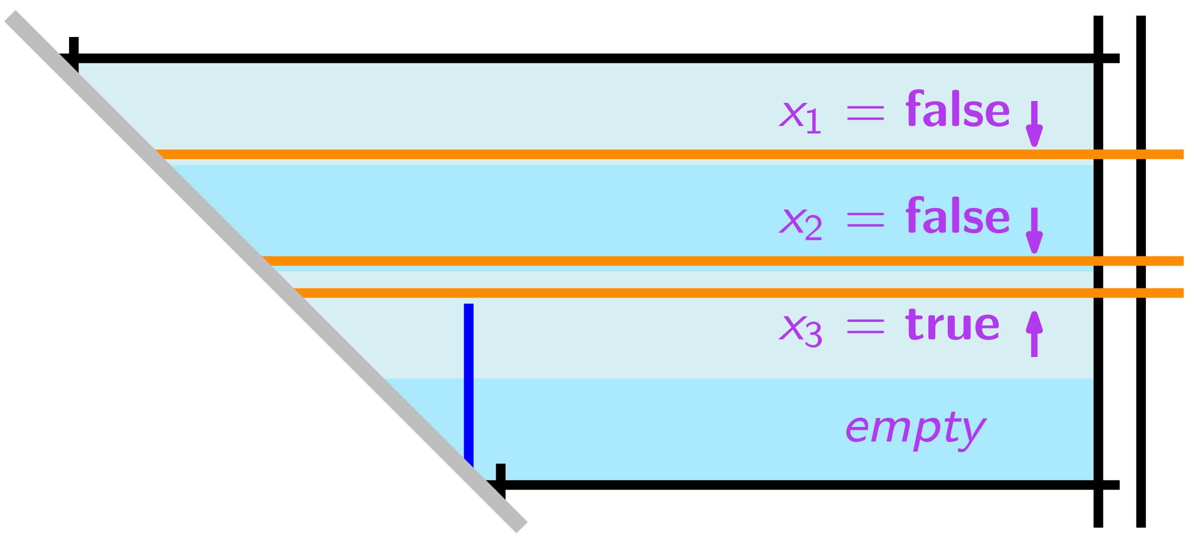


Hardness of $\text{STICK}^{\text{fix}}_{\text{AB}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:

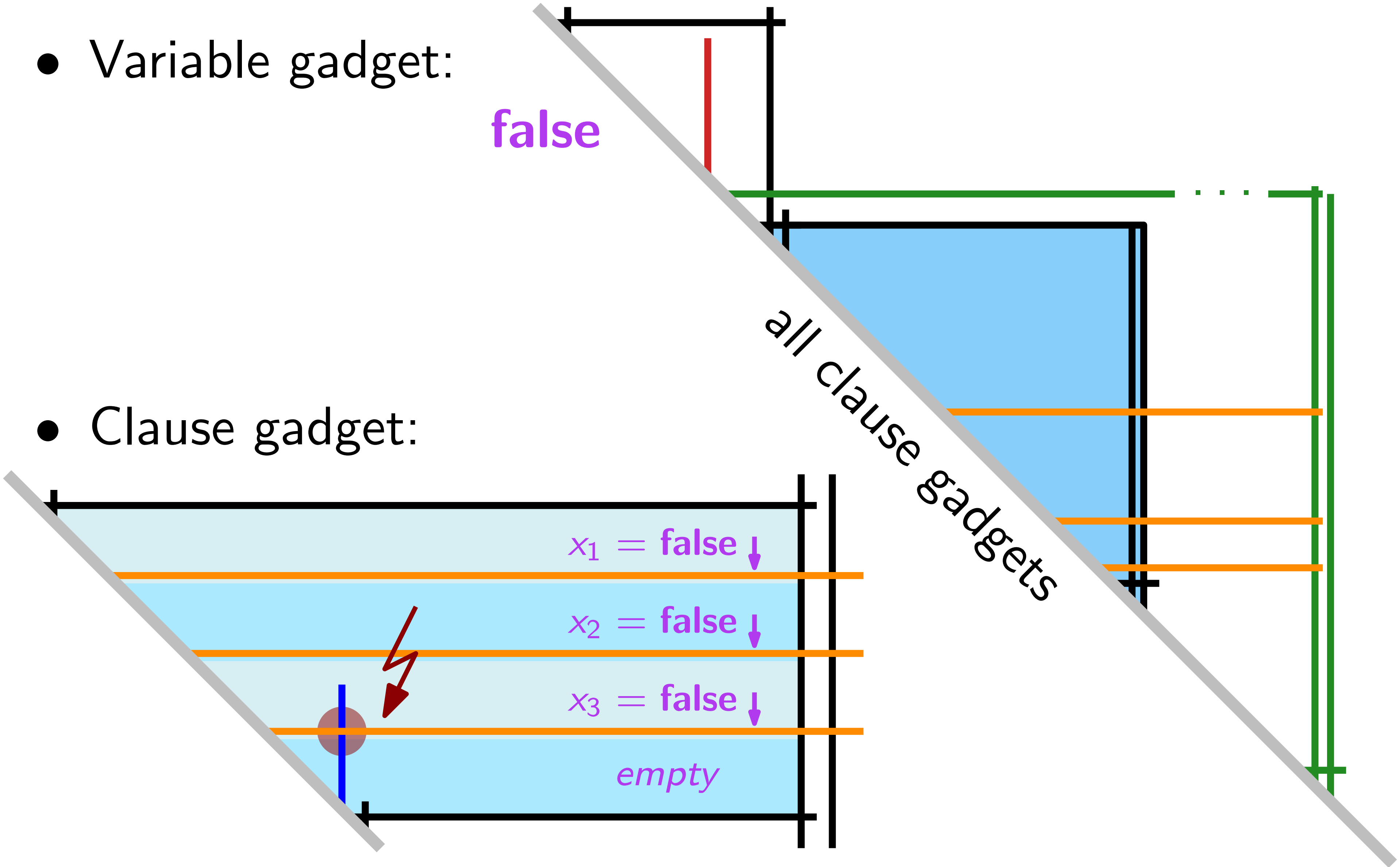


- Clause gadget:



Hardness of $\text{STICK}_{\text{AB}}^{\text{fix}}$

- NP-hardness by reduction from MONOTONE-3-SAT
- Variable gadget:
- Clause gadget:



Example

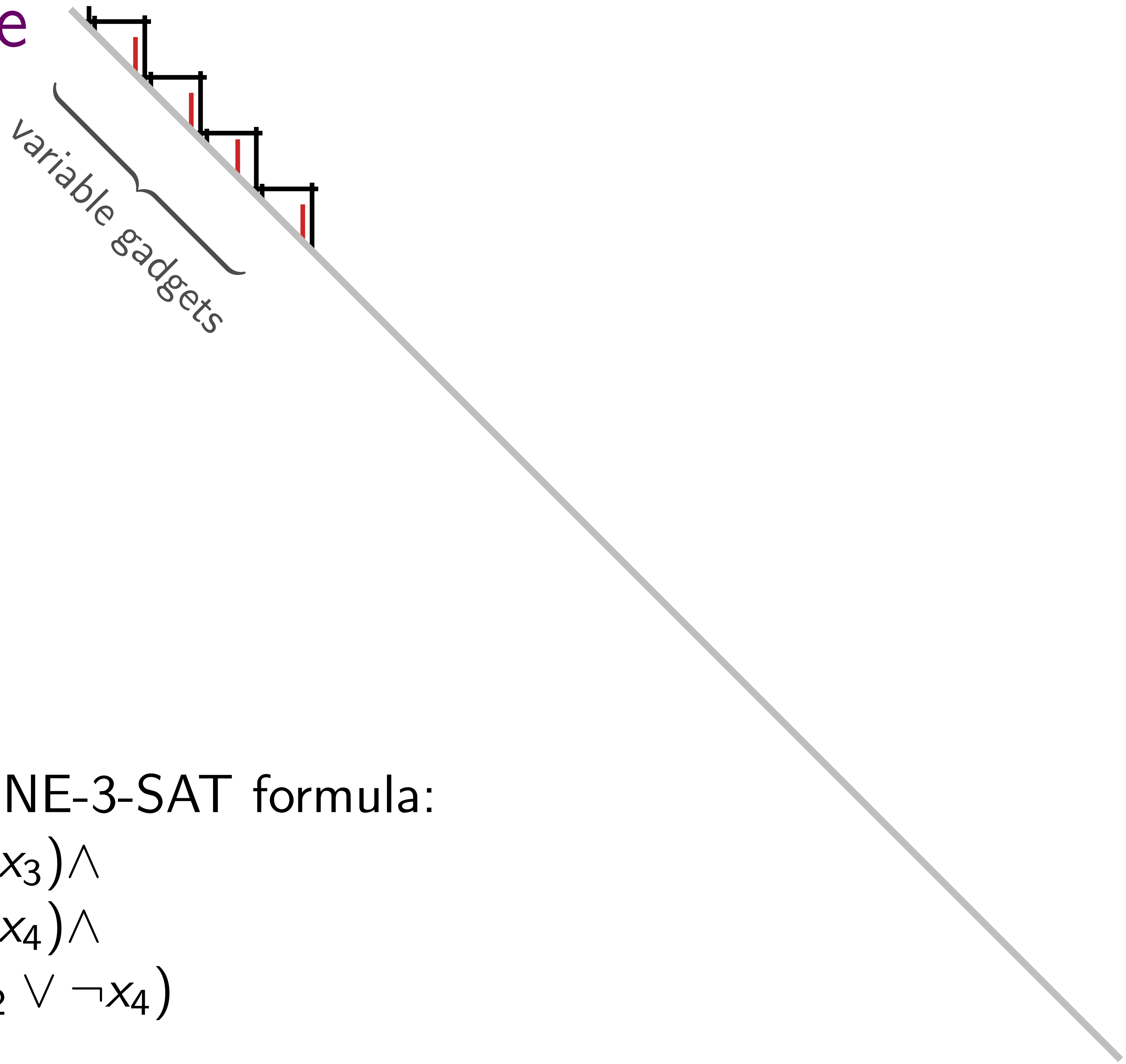
MONOTONE-3-SAT formula:

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

$(x_2 \vee x_3 \vee x_4) \wedge$

$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

Example



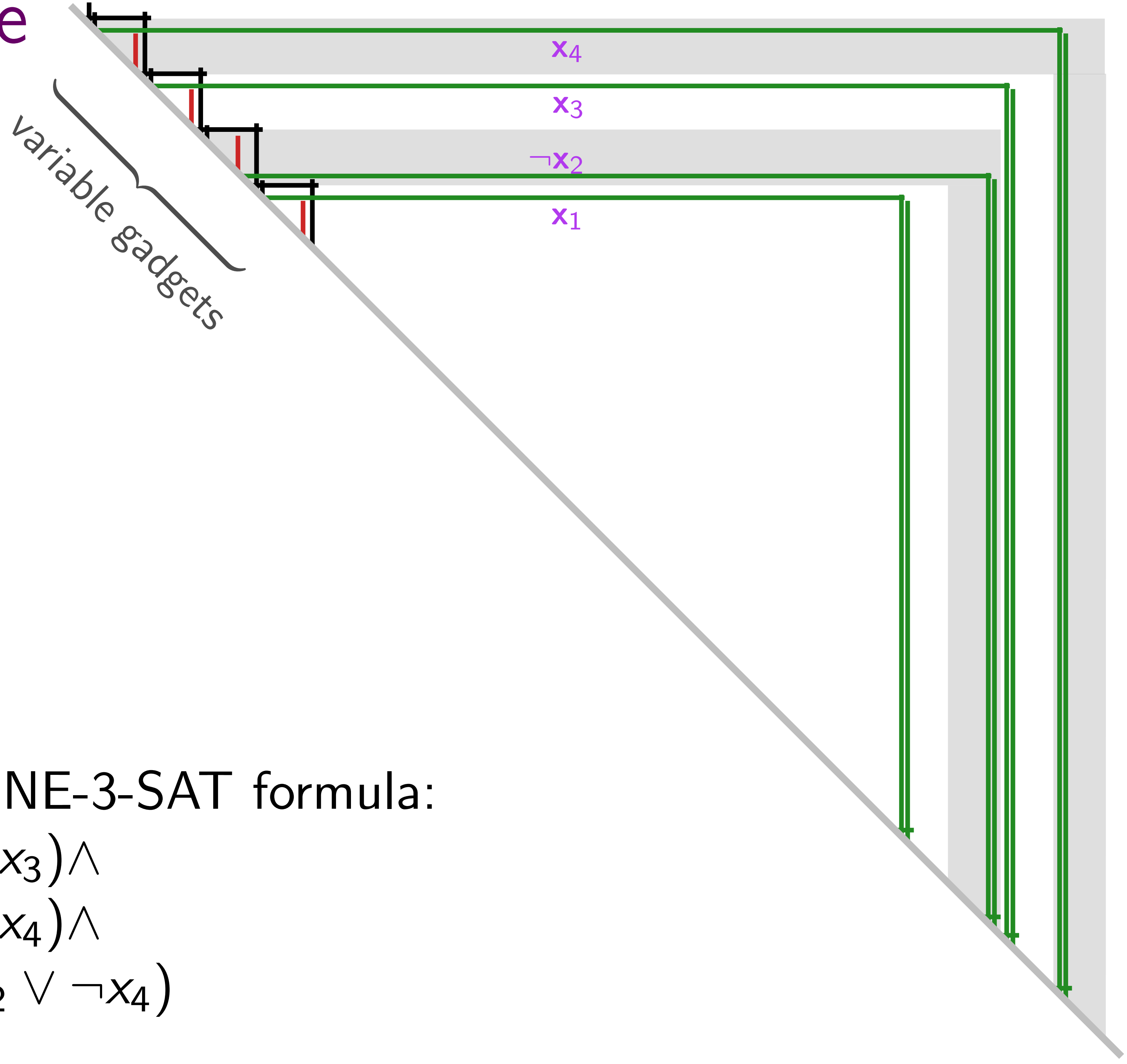
MONOTONE-3-SAT formula:

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

$(x_2 \vee x_3 \vee x_4) \wedge$

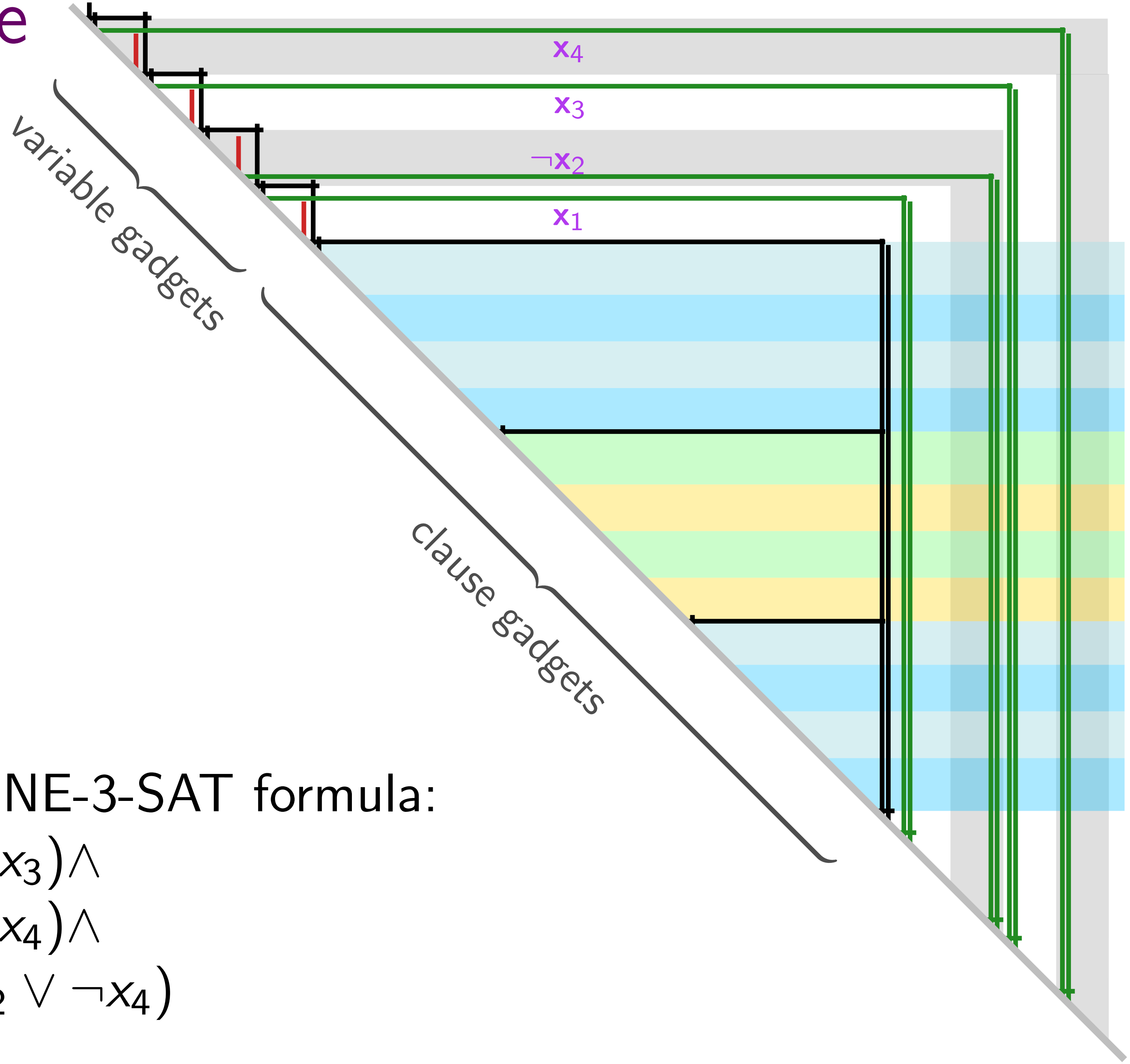
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

Example



MONOTONE-3-SAT formula:
$(x_1 \vee x_2 \vee x_3) \wedge$
$(x_2 \vee x_3 \vee x_4) \wedge$
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

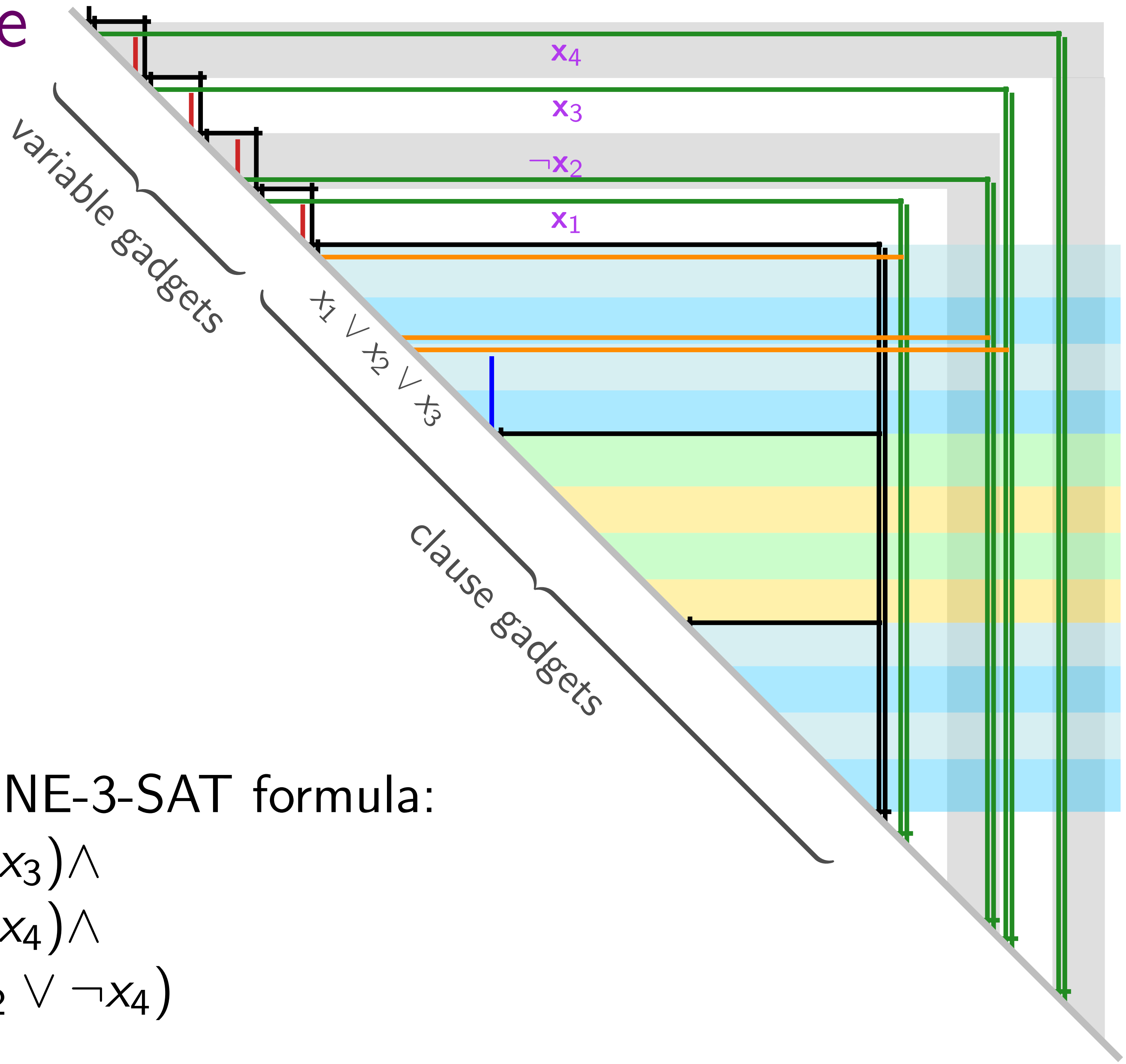
Example



MONOTONE-3-SAT formula:

$(x_1 \vee x_2 \vee x_3) \wedge$
$(x_2 \vee x_3 \vee x_4) \wedge$
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

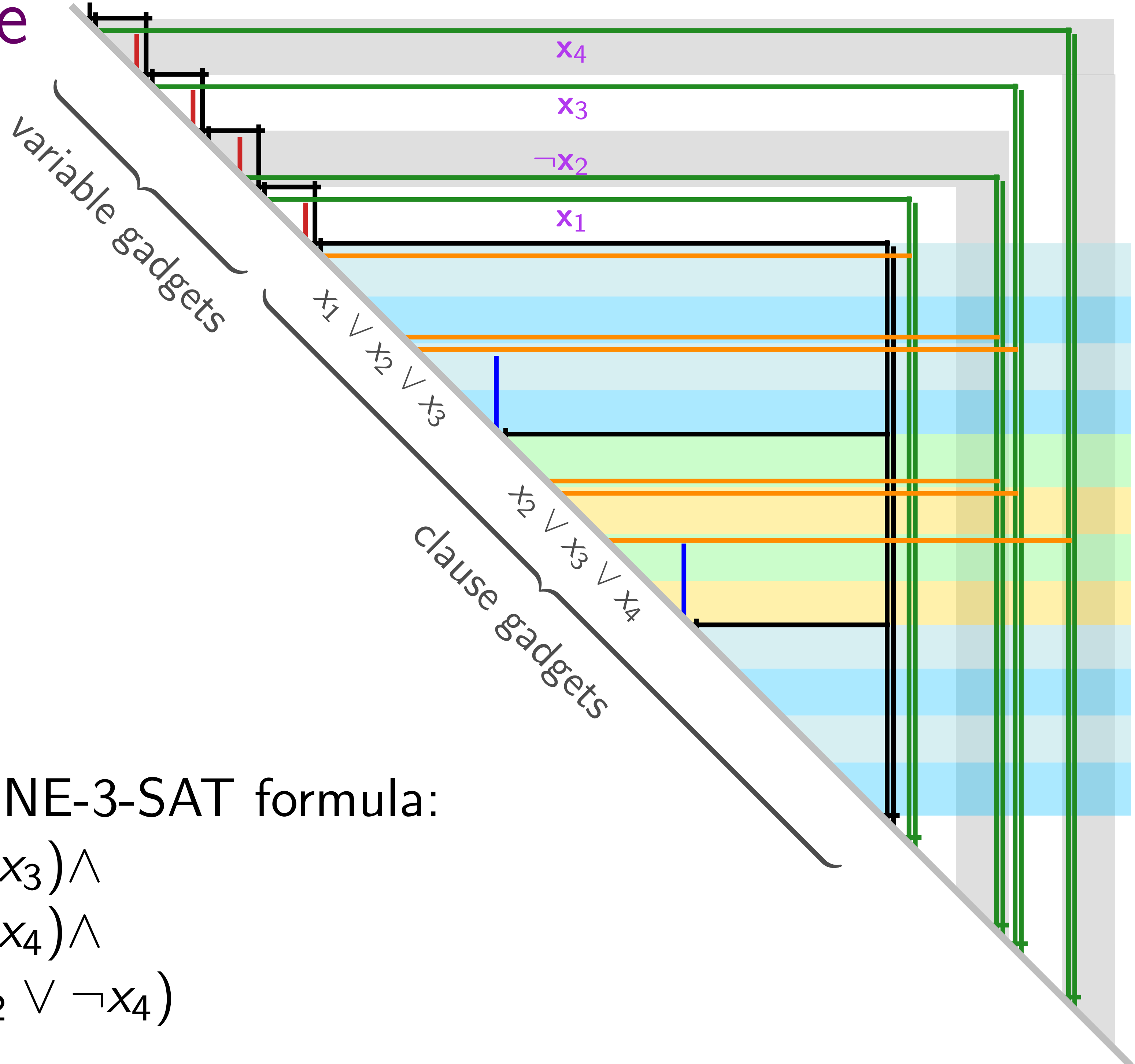
Example



MONOTONE-3-SAT formula:

$(x_1 \vee x_2 \vee x_3) \wedge$
$(x_2 \vee x_3 \vee x_4) \wedge$
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

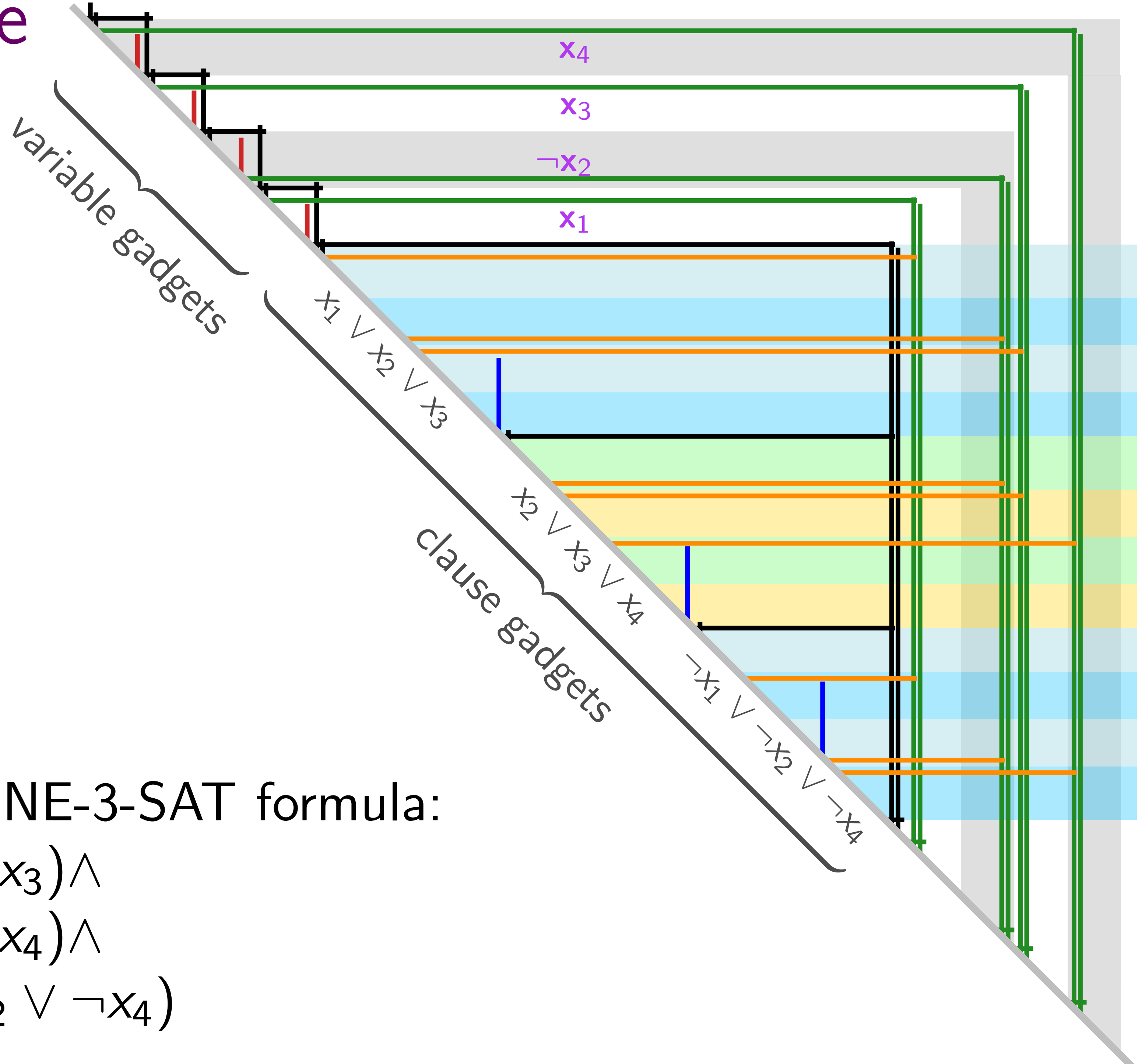
Example



MONOTONE-3-SAT formula:

$(x_1 \vee x_2 \vee x_3) \wedge$
$(x_2 \vee x_3 \vee x_4) \wedge$
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

Example



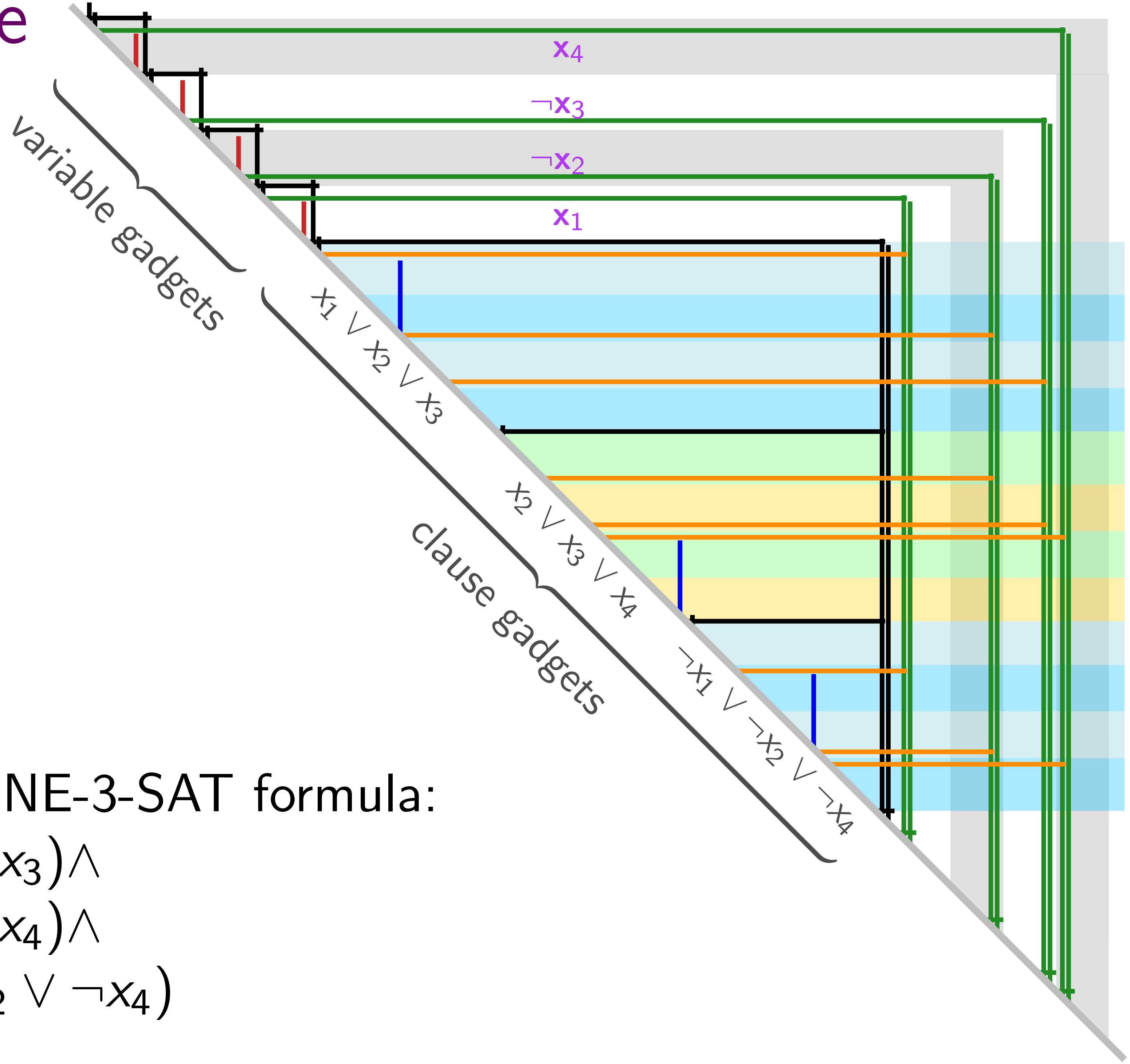
MONOTONE-3-SAT formula:

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

$(x_2 \vee x_3 \vee x_4)\wedge$

$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

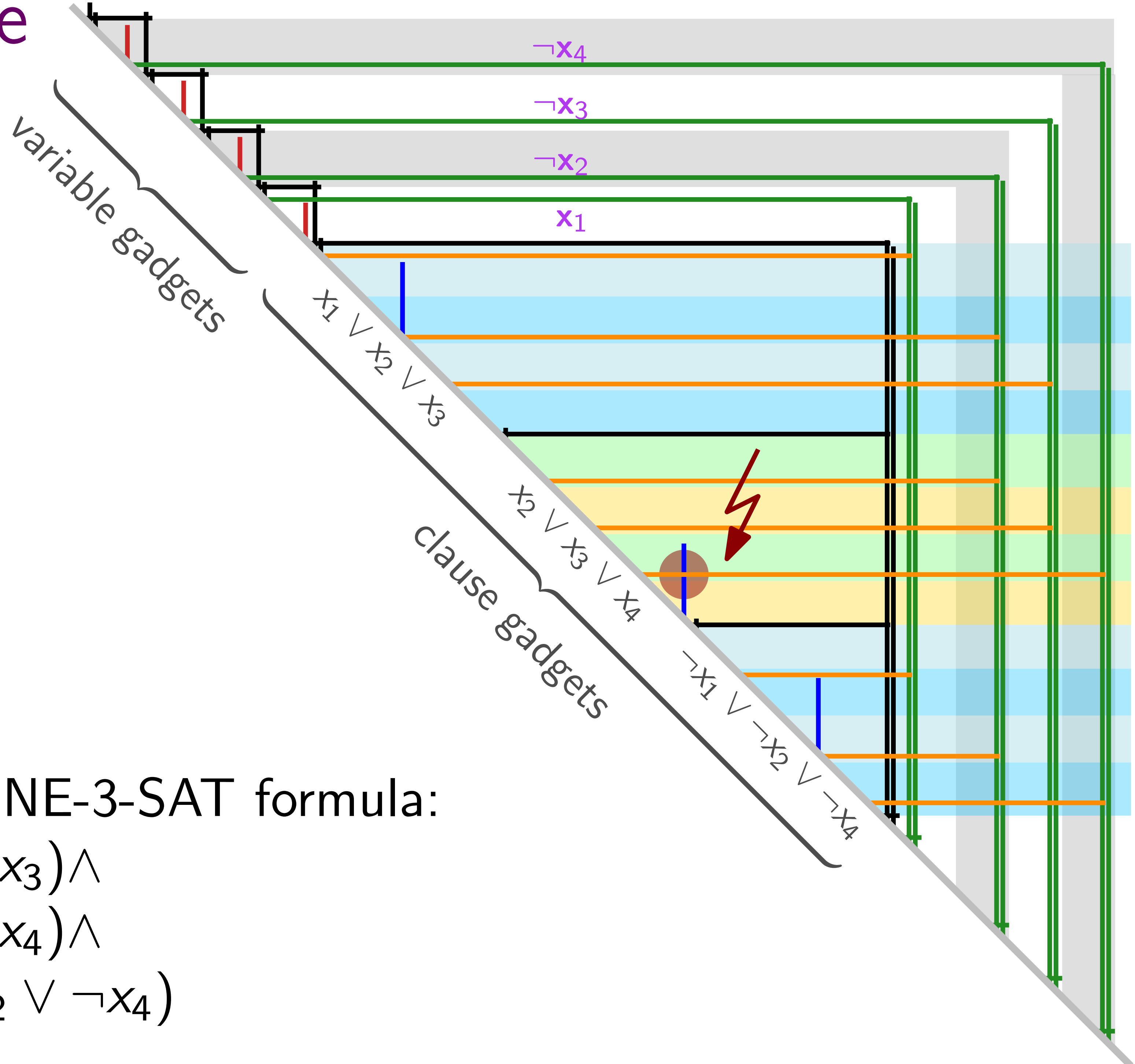
Example



MONOTONE-3-SAT formula:

$(x_1 \vee x_2 \vee x_3) \wedge$
$(x_2 \vee x_3 \vee x_4) \wedge$
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

Example



MONOTONE-3-SAT formula:

$(x_1 \vee x_2 \vee x_3) \wedge$
$(x_2 \vee x_3 \vee x_4) \wedge$
$(\neg x_1 \vee \neg x_2 \vee \neg x_4)$

$\text{STICK}^{\text{fix}}_{\text{AB}}$ without isolated vertices

$\star$	**$\text{STICK}_{\star}$**	**$\text{STICK}^{\text{fix}}_{\star}$**
	?	NP-complete
A	$O(\|A\|\|B\|)$	NP-complete
AB	$O(\|E\|)$	in general: NP-complete w/o isolated vtc.: $O((\|A\|+\|B\|)^2)$

Uniqueness Lemma

Lemma:

Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathsf{STICK}_{\mathsf{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Uniqueness Lemma

Lemma: In all stick representations of an instance of STICK_{AB}, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathrm{STICK}_{\mathrm{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order

Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathrm{STICK}_{\mathrm{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

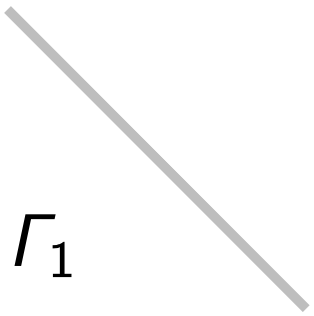
- Assume: stick representations Γ_1, Γ_2 have different order

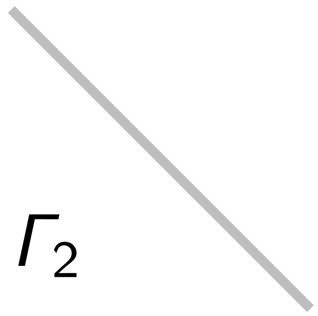
Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathsf{STICK}_{\mathsf{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2





Uniqueness Lemma

Lemma: In all stick representations of an instance of STICK_{AB}, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

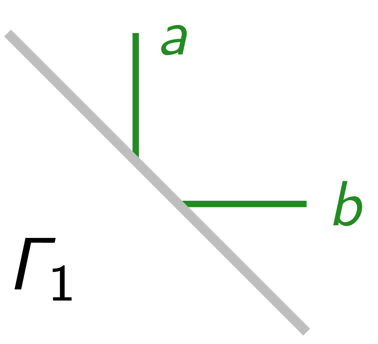
- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\;\Rightarrow ab \notin E$

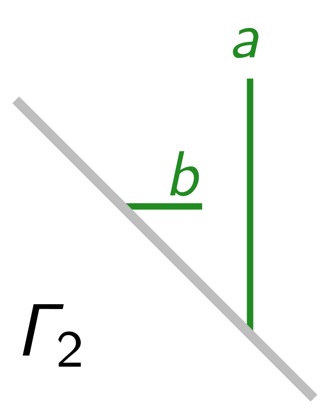
Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathrm{STICK_{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$





Uniqueness Lemma

Lemma: In all stick representations of an instance of STICK_{AB}, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\;\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$

Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathrm{STICK}_{\mathrm{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

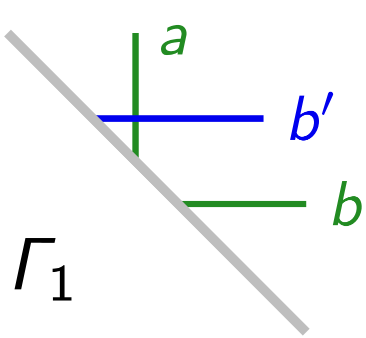
- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$

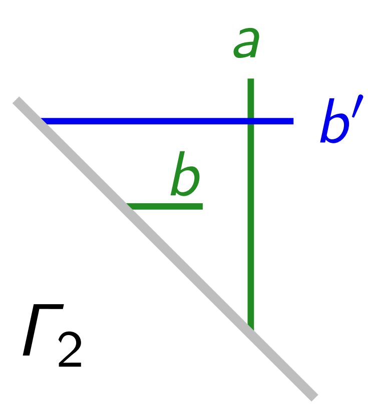
Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathrm{STICK}_{\mathrm{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$



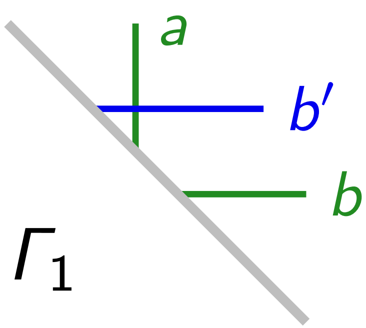


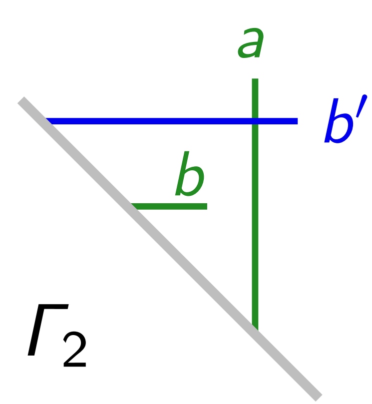
Uniqueness Lemma

Lemma: In all stick representations of an instance of STICK_{AB}, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$
- a' is adjacent to $b \Rightarrow a < a'$



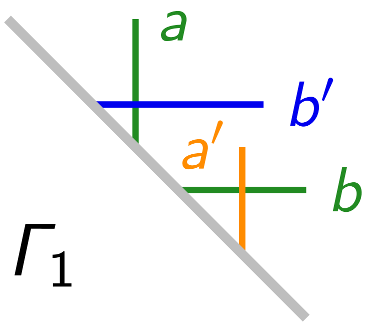


Uniqueness Lemma

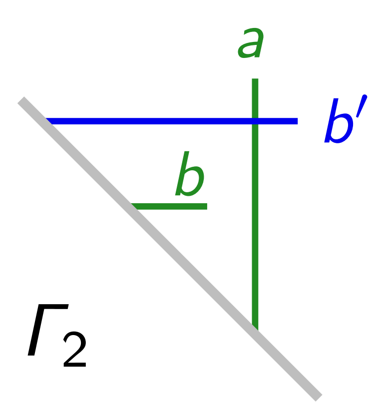
Lemma: In all stick representations of an instance of STICK_{AB}, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$
- a' is adjacent to $b \Rightarrow a < a'$



Γ_1



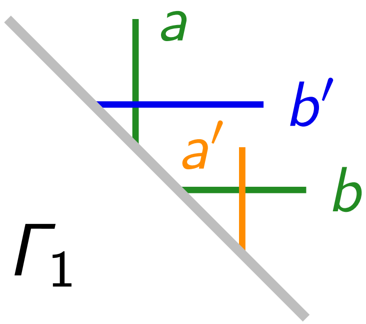
Γ_2

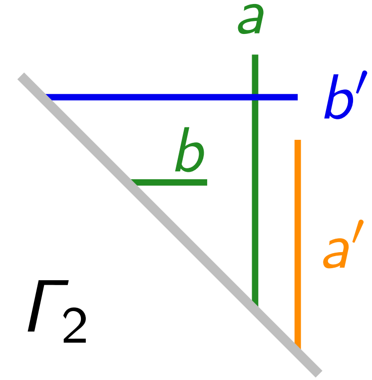
Uniqueness Lemma

Lemma: In all stick representations of an instance of $\mathrm{STICK}_{\mathrm{AB}}$, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$
- a' is adjacent to $b \Rightarrow a < a'$



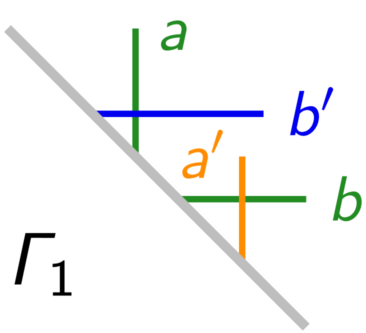


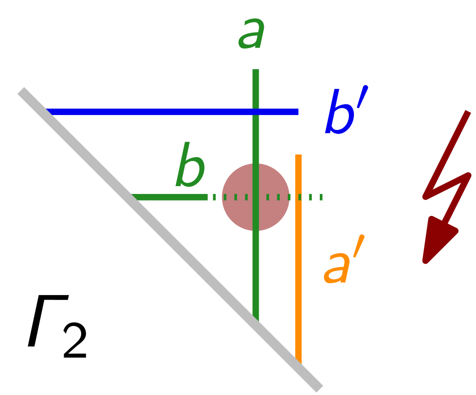
Uniqueness Lemma

Lemma: In all stick representations of an instance of STICK_{AB}, the order of vertices on the ground line is the same after removing all isolated vertices. This order can be found in time $O(|E|)$.

Proof (Sketch):

- Assume: stick representations Γ_1, Γ_2 have different order
- W.l.o.g. $a < b$ in Γ_1 and $b < a$ in Γ_2 $\Rightarrow ab \notin E$
- b' is adjacent to $a \Rightarrow b' < b$
- a' is adjacent to $b \Rightarrow a < a'$





Linear Program for $\text{STICK}_{\text{AB}}^{\text{fix}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$

Linear Program for $\text{STICK}_{\text{AB}}^{\text{fix}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$
- Variable x_i for the x-coordinate of v_i's footpoint

Linear Program for $\text{STICK}^{\text{fix}}_{\text{AB}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$
- Variable x_i for the x-coordinate of v_i's footpoint
- Constraints: $x_1 < \cdots < x_n$

Linear Program for $\text{STICK}_{\text{AB}}^{\text{fix}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$
- Variable x_i for the x-coordinate of v_i's footpoint
- Constraints: $x_1 < \cdots < x_n$
- For each vertex 2 more constraints incorporating the predefined stick-lengths:
  - for intersecting its last neighbor
  - for not intersecting its first non-neighbor

Linear Program for $\text{STICK}_{\text{AB}}^{\text{fix}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$
- Variable x_i for the x-coordinate of v_i's footpoint
- Constraints: $x_1 < \cdots < x_n$
- For each vertex 2 more constraints incorporating the predefined stick-lengths:
  - for intersecting its last neighbor
  - for not intersecting its first non-neighbor
- Is a system of difference constraints $\Rightarrow$ can be modeled as a shortest-path problem in a directed weighted graph

Linear Program for $\text{STICK}_{\text{AB}}^{\text{fix}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$
- Variable x_i for the x-coordinate of v_i's footpoint
- Constraints: $x_1 < \cdots < x_n$
- For each vertex 2 more constraints incorporating the predefined stick-lengths:
  - for intersecting its last neighbor
  - for not intersecting its first non-neighbor
- Is a system of difference constraints $\Rightarrow$ can be modeled as a shortest-path problem in a directed weighted graph
  $\Rightarrow$ Solvable in $O((|A| + |B|)^2)$ time with Bellman–Ford

Linear Program for $\text{STICK}_{\text{AB}}^{\text{fix}}$

- No isolated vertices $\Rightarrow$ Compute ordering $v_1, \ldots, v_n$
- Variable x_i for the x-coordinate of v_i's footpoint
- Constraints: $x_1 < \cdots < x_n$
- For each vertex 2 more constraints incorporating the predefined stick-lengths:
  - for intersecting its last neighbor
  - for not intersecting its first non-neighbor
- Is a system of difference constraints $\Rightarrow$ can be modeled as a shortest-path problem in a directed weighted graph
  $\Rightarrow$ Solvable in $O((|A| + |B|)^2)$ time with Bellman–Ford

$\Rightarrow$ **Isolated vertices make $\text{STICK}_{\text{AB}}^{\text{fix}}$ NP-hard**

Summary

$\star$	**STICK**$_\star$	**STICK**$_\star^{\textbf{fix}}$
	still open	NP-complete
A	$O(\lvert A\rvert\lvert B\rvert)$	NP-complete
AB	$O(\lvert A\rvert\lvert B\rvert)$ $O(\lvert E\rvert)$ [De Luca et al. GD'18]	in general: NP-complete w/o isolated vtc.: $O((\lvert A\rvert+\lvert B\rvert)^2)$

Summary

$\star$	**STICK**$_\star$	**STICK**$_\star^{\mathbf{fix}}$
	still open	NP-complete (by reduction from 3-PARTITION)
A	$O(\lvert A\rvert\lvert B\rvert)$	NP-complete
AB	$O(\lvert A\rvert\lvert B\rvert)$ [De Luca et al. GD'18] $O(\lvert E\rvert)$	in general: NP-complete w/o isolated vtc.: $O((\lvert A\rvert+\lvert B\rvert)^2)$

Summary

$\star$	**STICK**$_\star$	**STICK**$_\star^{\mathbf{fix}}$
	still open	NP-complete (by reduction from 3-PARTITION)
A	$O(\lvert A\rvert\lvert B\rvert)$	NP-complete (by reduction from MONO-3-SAT)
AB	$O(\lvert A\rvert\lvert B\rvert)$ [De Luca et al. GD'18] $O(\lvert E\rvert)$	in general: NP-complete w/o isolated vtc.: $O((\lvert A\rvert + \lvert B\rvert)^2)$