Number of Variables for Graph Differentiation and the Resolution of Graph Isomorphism Formulas

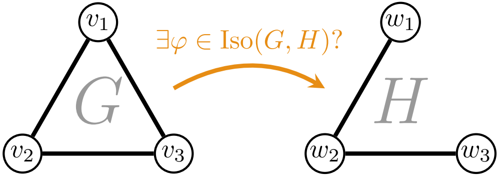
Jacobo Torán & Florian Wörz

Universität Ulm

GI Meeting Deduction and Logic
Friedrich-Alexander-Universität Erlangen-Nürnberg
April 8, 2022

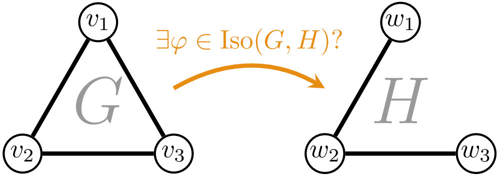
Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem…



Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$$x_{i,j} = 1 \quad :\Longleftrightarrow \quad v_i \text{ is mapped to } w_j$$

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

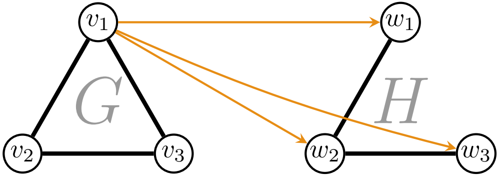
- **Type 3 clauses:** adjacency relation

$\forall i < j$ and $k \neq \ell$ with

$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$

Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \quad :\Longleftrightarrow \quad v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

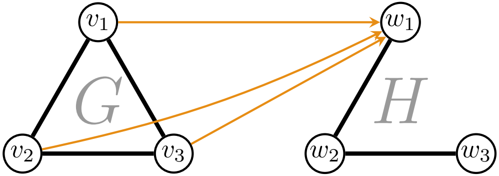
- **Type 3 clauses:** adjacency relation

$\forall i < j$ and $k \neq \ell$ with

$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$

Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \quad :\Longleftrightarrow \quad v_i$ is mapped to w_j

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$

$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$

$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

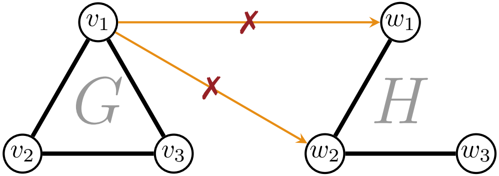
- **Type 3 clauses:** adjacency relation

$\forall i < j$ and $k \neq \ell$ with

$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$

Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem…



$$x_{i,j} = 1 \quad :\Longleftrightarrow \quad v_i \text{ is mapped to } w_j$$

…and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

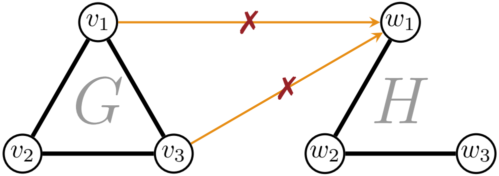
- **Type 3 clauses:** adjacency relation

$\forall i < j$ and $k \neq \ell$ with

$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$$x_{i,j} = 1 \quad :\Longleftrightarrow \quad v_i \text{ is mapped to } w_j$$

...and encode it as the formula $\mathrm{ISO}(G,H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

$$\forall i,j,k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i,j,k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

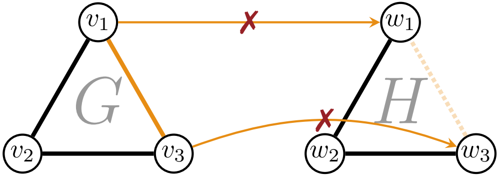
- **Type 3 clauses:** adjacency relation

$\forall i < j$ and $k \neq \ell$ with

$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$

Topic of this Talk: Graph Isomorphism Formulas

Take the Graph Isomorphism Problem...



$x_{i,j} = 1 \quad :\Longleftrightarrow \quad v_i \text{ is mapped to } w_j$

...and encode it as the formula $\mathrm{ISO}(G, H)$:

- **Type 1 clauses:** consider all vertices

$$\forall i \in [n] : (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n})$$
$$\forall j \in [n] : (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j})$$

- **Type 2 clauses:** function + injective

$$\forall i, j, k \in [n] \text{ with } j \neq k : (\overline{x_{i,j}} \vee \overline{x_{i,k}})$$
$$\forall i, j, k \in [n] \text{ with } i \neq j : (\overline{x_{i,k}} \vee \overline{x_{j,k}})$$

- **Type 3 clauses:** adjacency relation

$\forall i < j$ and $k \neq \ell$ with

$$\{v_i, v_j\} \in E_G \Leftrightarrow \{v_k, v_\ell\} \notin E_H : (\overline{x_{i,k}} \vee \overline{x_{j,\ell}})$$

World 1:
The Resolution Proof System

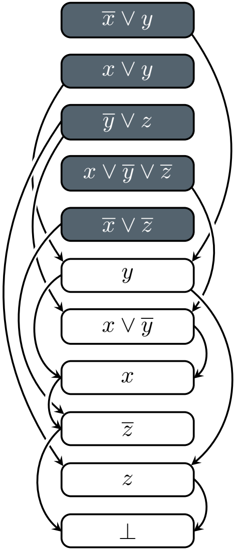
The Proof System Resolution

Resolution Rule:

$$\frac{A \lor x \qquad B \lor \overline{x}}{A \lor B}$$

Distinction by Cases: [Galesi & Thapen]

$$\frac{A_1 \lor \overline{x_1} \qquad \dots \qquad A_m \lor \overline{x_m}}{B \lor A_1 \lor \dots \lor A_m} \quad \text{if} \quad (B \lor x_1 \lor \dots \lor x_m) \in F$$



Complexity Measures for Resolution

Size

$\#$ clauses

Width

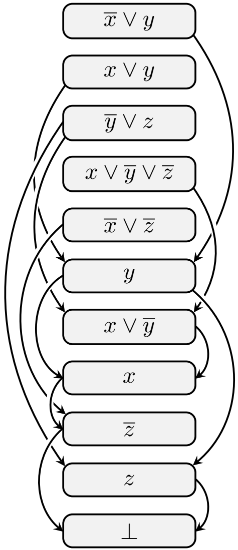
$\#$ literals in largest clause

Narrow Width

exclude all axioms

Space

max $\#$ clauses in memory



Complexity Measures for Resolution

Size

  # clauses (here: 11)

Width

  # literals in largest clause

Narrow Width

  exclude all axioms

Space

  max # clauses in memory

1. $\overline{x} \lor y$
2. $x \lor y$
3. $\overline{y} \lor z$
4. $x \lor \overline{y} \lor \overline{z}$
5. $\overline{x} \lor \overline{z}$
6. y
7. $x \lor \overline{y}$
8. x
9. $\overline{z}$
10. z
11. $\perp$

Complexity Measures for Resolution

Size

$\#$ clauses (here: 11)

Width

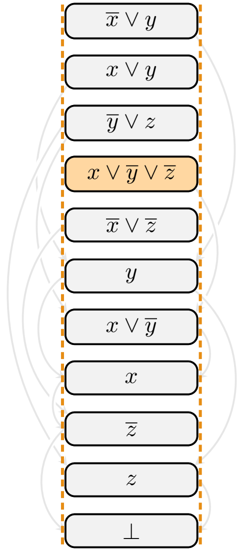
$\#$ literals in largest clause (here: 3)

Narrow Width

exclude all axioms

Space

max $\#$ clauses in memory



Complexity Measures for Resolution

Size

$\#$ clauses (here: 11)

Width

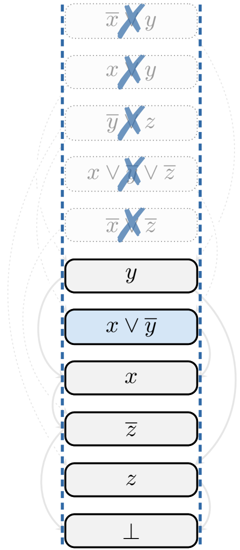
$\#$ literals in largest clause (here: 3)

Narrow Width

exclude all axioms (here: 2)

Space

max $\#$ clauses in memory



Complexity Measures for Resolution

Size

$\#$ clauses (here: 11)

Width

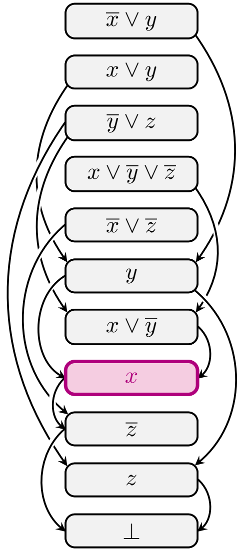
$\#$ literals in largest clause (here: 3)

Narrow Width

exclude all axioms (here: 2)

Space

max $\#$ clauses in memory



Complexity Measures for Resolution

Size

$\#$ clauses (here: 11)

Width

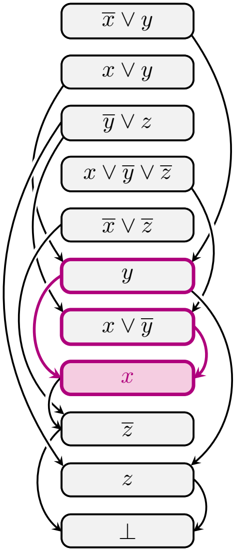
$\#$ literals in largest clause (here: 3)

Narrow Width

exclude all axioms (here: 2)

Space

max $\#$ clauses in memory



Complexity Measures for Resolution

Size

$\#$ clauses (here: 11)

Width

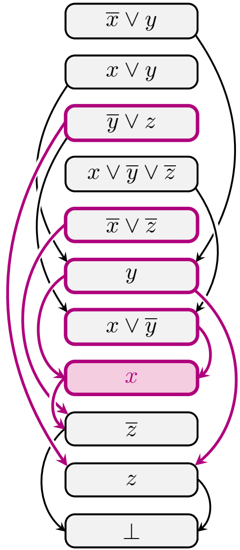
$\#$ literals in largest clause (here: 3)

Narrow Width

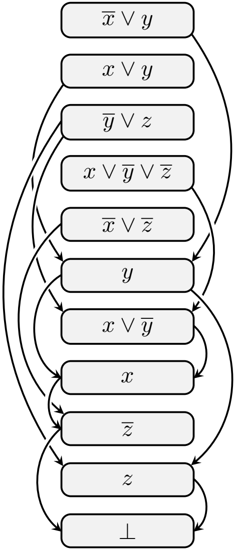
exclude all axioms (here: 2)

Space

max $\#$ clauses in memory (here: 5 at time 8)



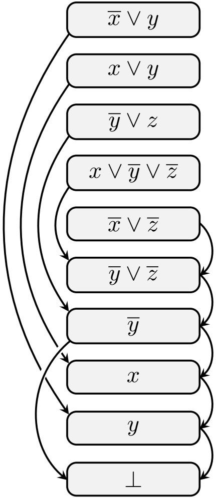
Complexity Measures for Resolution—What we really care about



For each complexity measure $\mathscr{C}$:

Take minimum over all refutations π

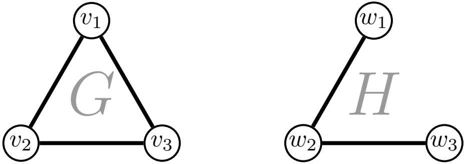
$$\mathscr{C}(F \vdash \perp) := \min_{\pi : F \vdash \perp} \mathscr{C}(\pi)$$

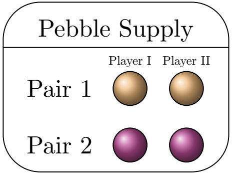


World 2:
Descriptive Complexity / First-order logic

Immerman's k-pebble game: Player I wants to show $G \ncong H$

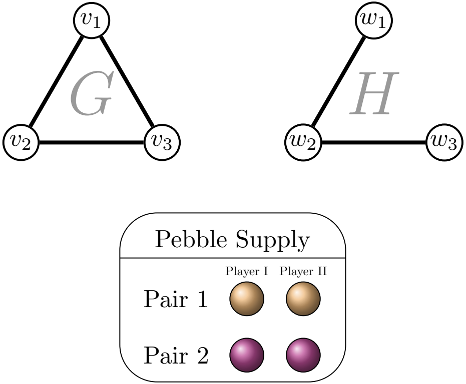
- Player I and Player II have k pebble pairs





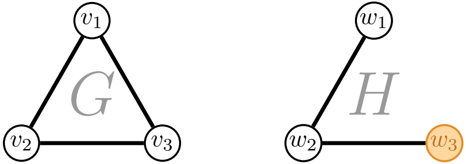
Immerman's k-pebble game: Player I wants to show $G \not\cong H$

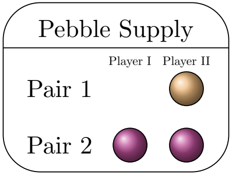
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph



Immerman's k-pebble game: Player I wants to show $G \not\cong H$

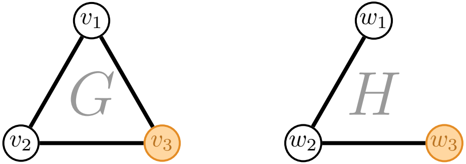
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph

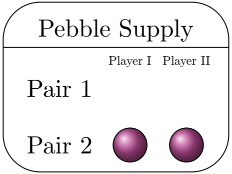




Immerman's k-pebble game: Player I wants to show $G \not\cong H$

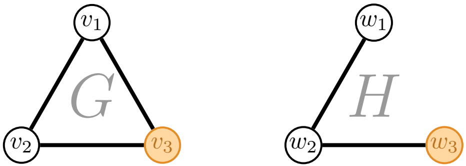
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**

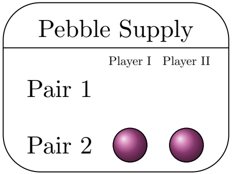




Immerman's k-pebble game: Player I wants to show $G \not\cong H$

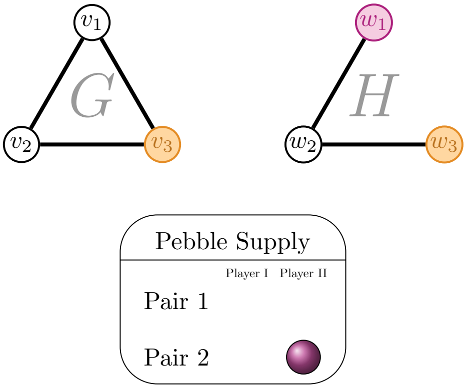
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic





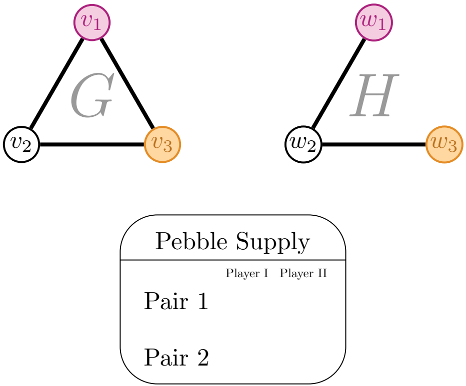
Immerman's k-pebble game: Player I wants to show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic



Immerman's k-pebble game: Player I wants to show $G \not\cong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic ✗ Player I won!

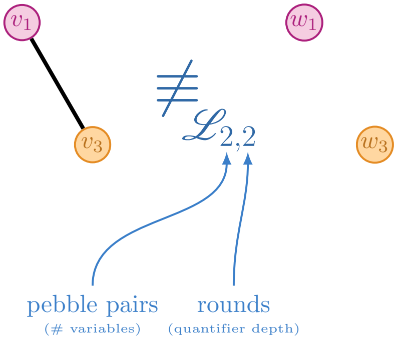


Immerman's k-pebble game: Player I wants to show $G \ncong H$

- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic ✗ Player I won!

Immerman's k-pebble game: Player I wants to show $G \not\cong H$

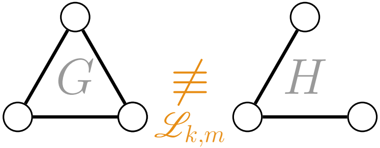
- Player I and Player II have k pebble pairs
- In each round:
  - **Player I chooses:**
    - put a pebble pair back into the box, OR
    - place a new pebble of a pair on any graph
  - **Player II simply reacts.**
- Player II survives if pebbled subgraphs are isomorphic ✗ Player I won!



Our Main Result:
Combining both worlds

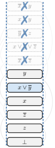
Main Result: Connection between FO and PC

Immerman's Pebble Game
on G and H



$\Longleftrightarrow$

Narrow Width Refutation
of $\mathrm{ISO}(G, H)$



$\text{N-Width}(\pi) \leq k-1$
$\text{PosDepth}(\pi) \leq m$

Implications

For every pair of non-isomorphic graphs (G, H) with n vertices each and for every $k \in \mathbb{N}$:

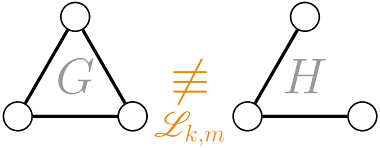
1 $G \not\equiv_{\mathscr{L}_k} H \implies \mathrm{Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \leq n^{\mathrm{O}(k)}$

2 $G \equiv_{\mathscr{L}_k} H \implies \begin{cases} \mathrm{Tree\text{-}Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \geq 2^k \\ \mathrm{Space}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \geq k+1 \end{cases}$

3 $(G,\lambda) \equiv_{\mathscr{L}_k} (H,\mu) \implies \mathrm{Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \geq \exp\Big(\Omega\big(\frac{k^2}{\text{sum of color class sizes}}\big)\Big)$

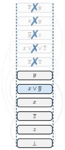
Proof Idea: Use k-witnessing game

Immerman's Pebble Game
on G and H



$\Longleftrightarrow$

Narrow Width Refutation
of $\mathrm{ISO}(G, H)$



$$\text{N-Width}(\pi) \leq k - 1$$
$$\text{PosDepth}(\pi) \leq m$$

$\Updownarrow$

k-witnessing game

Spoiler wins on $\mathrm{ISO}(G, H)$

Spoiler vs. Duplicator: Spoiler Wants to Prove Unsatisfiability

They compete in the ***k*-witnessing game** on the formula $\mathrm{ISO}(G, H)$

- Game state is a partial assignment, initially $\alpha_0 = \varepsilon$
- In each round i

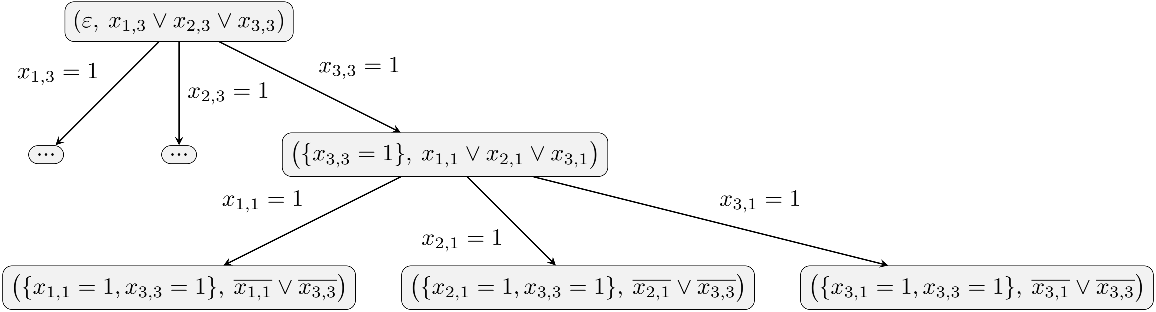
  Spoiler: Chooses a subset $\alpha' \subseteq \alpha_{i-1}$ of **size at most $\boldsymbol{k-1}$**
  Chooses a Type 1 clause C in $\mathrm{ISO}(G, H)$

  Duplicator: Extends $\alpha_i := \alpha' \cup \{\ell = 1\}$ for some literal $\ell \in C$

- Game ends when Duplicator cannot extend such that
  - α_i satisfies C and
  - does not falsify any other clause in $\mathrm{ISO}(G, H)$

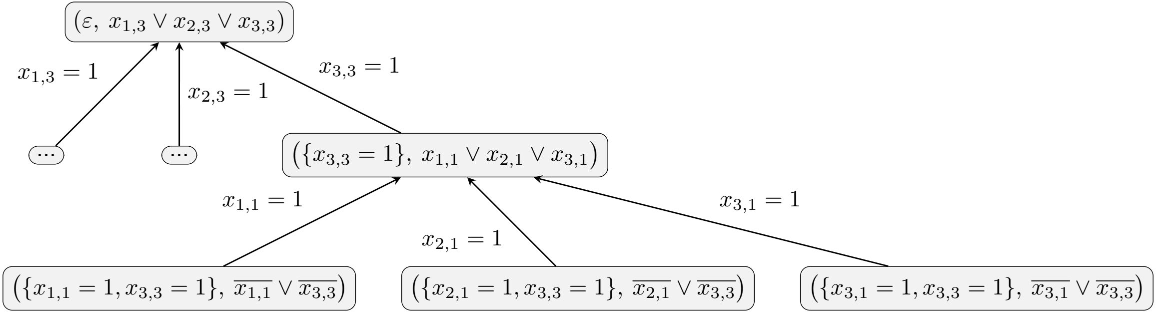
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation



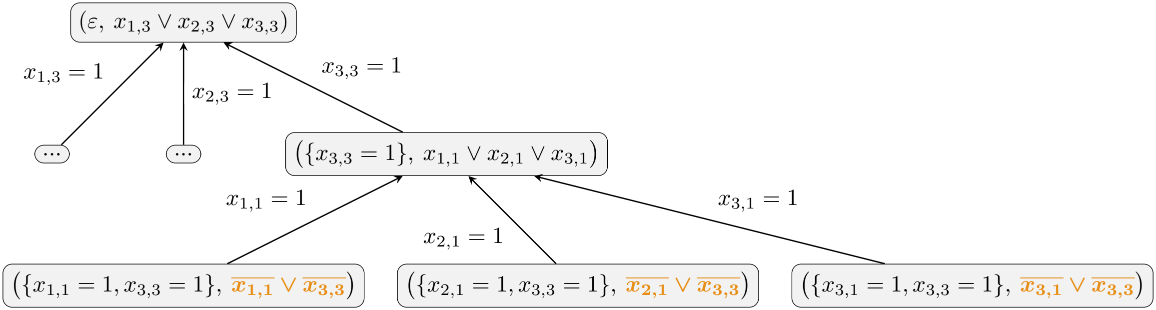
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation



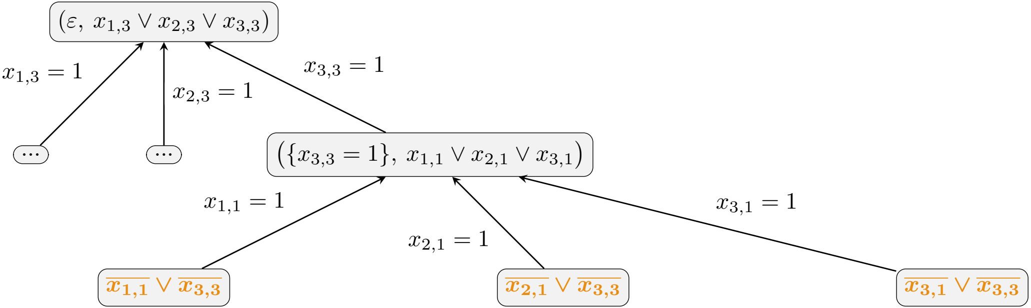
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation



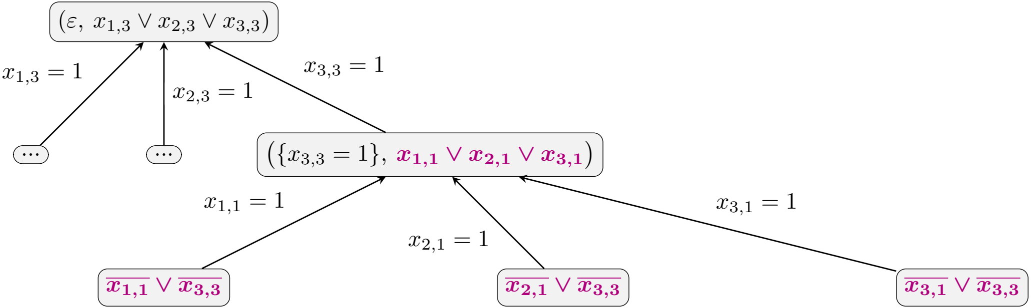
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation



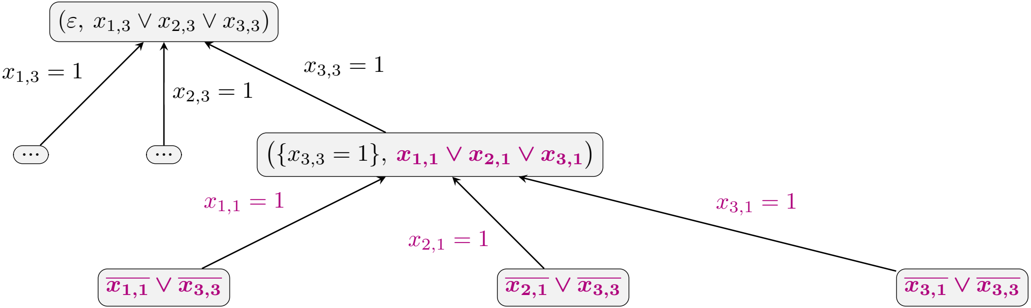
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation



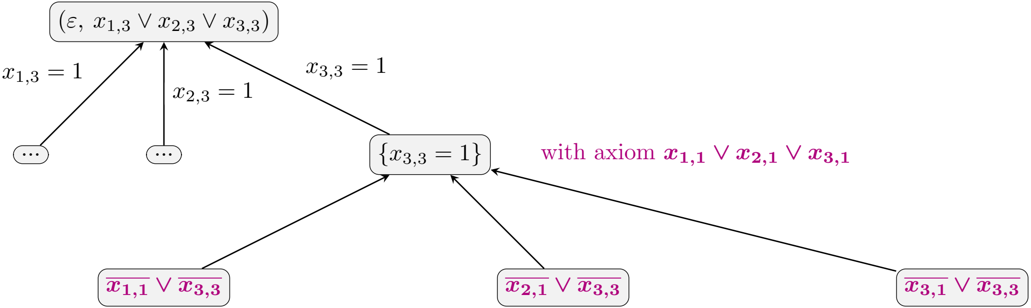
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation



Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

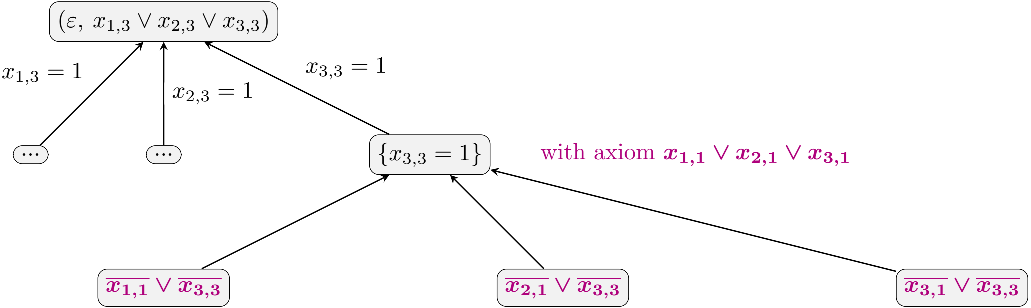
Convert Strategy Graph of Spoiler into Narrow Width Refutation



Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation:

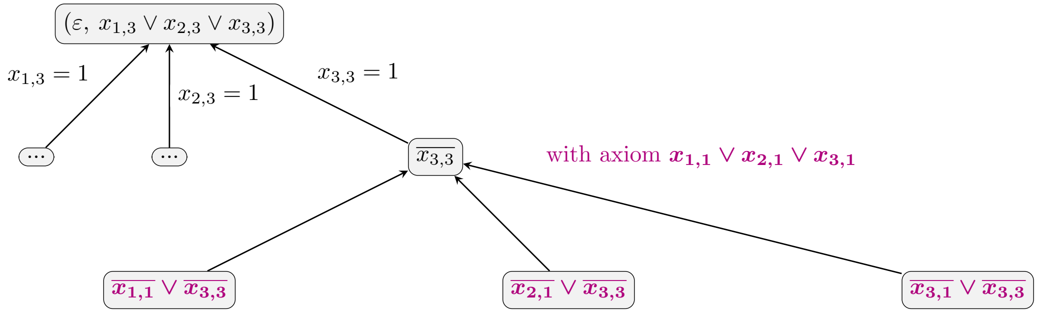
$(\alpha, C) \rightsquigarrow C_\alpha$ (set of literals falsified by α)



Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

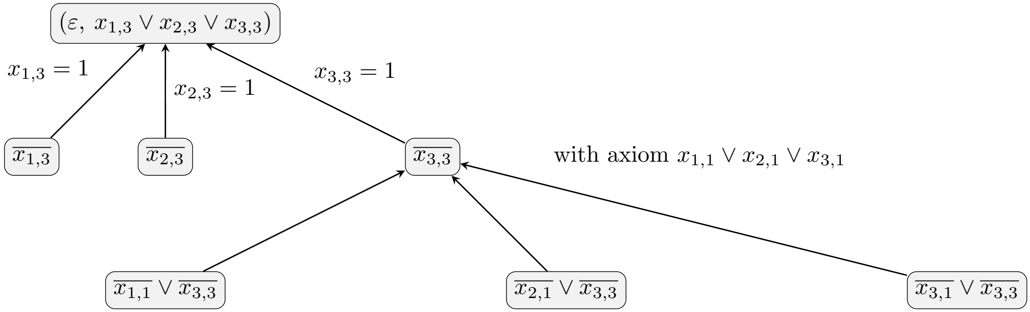
Convert Strategy Graph of Spoiler into Narrow Width Refutation:

$(\alpha, C) \rightsquigarrow C_\alpha$ (set of literals falsified by α)



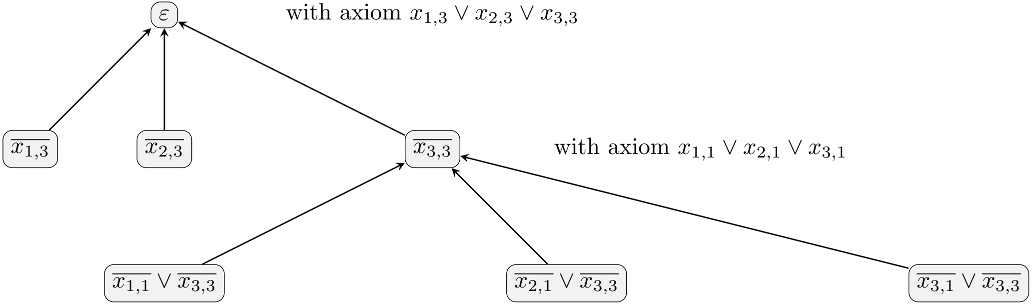
Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\text{ISO}(G, H) \vdash \bot\big) \leq k - 1$

Convert Strategy Graph of Spoiler into Narrow Width Refutation:
$(\alpha, C) \rightsquigarrow C_\alpha$ (set of literals falsified by α)



Proof: $G \not\equiv_{\mathscr{L}_k} H \implies \text{N-Width}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \leq k-1$

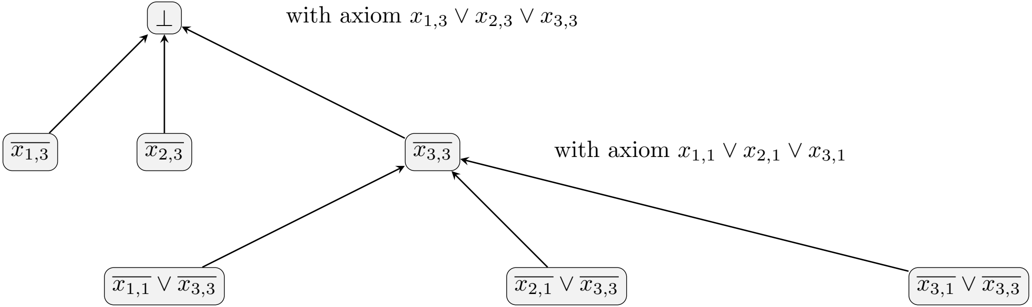
Convert Strategy Graph of Spoiler into Narrow Width Refutation:
$(\alpha, C) \rightsquigarrow C_\alpha$ (set of literals falsified by α)



Proof: $G \not\equiv_{\mathscr{L}_k} H \Longrightarrow \text{N-Width}\big(\text{ISO}(G,H) \vdash \bot\big) \leq k-1$

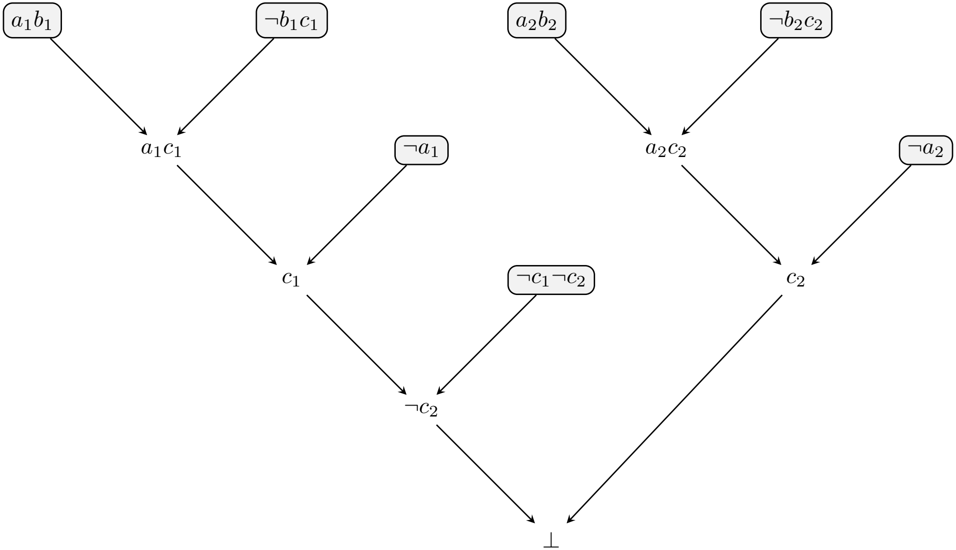
Convert Strategy Graph of Spoiler into Narrow Width Refutation:

$(\alpha, C) \rightsquigarrow C_\alpha$ (set of literals falsified by α)

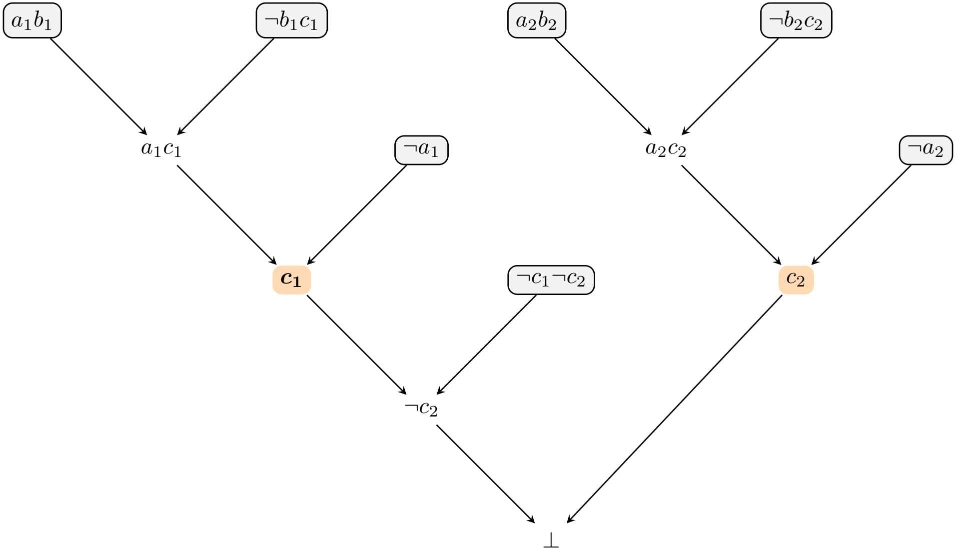


Symmetric Resolution
An Exponential GI Lower Bound for SRC-1

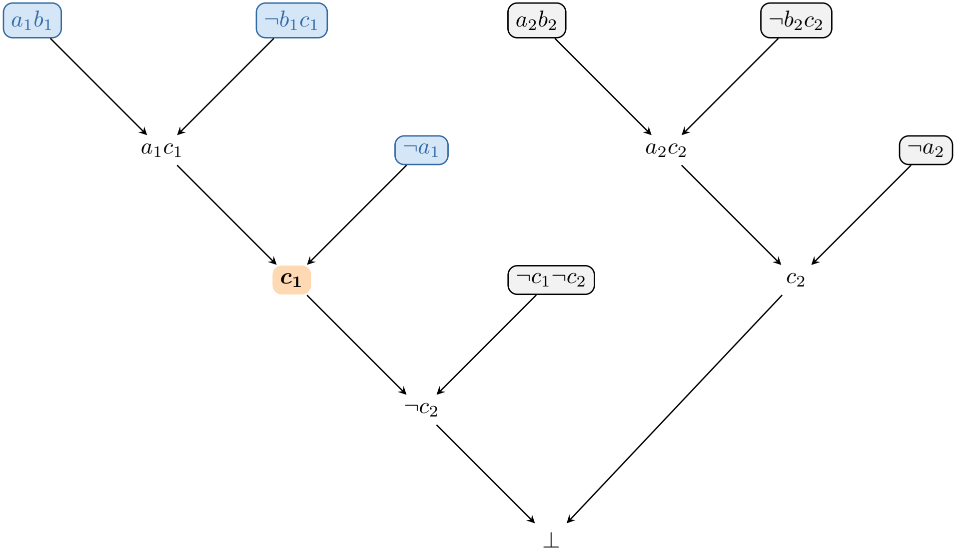
Krishnamurthy's Symmetry Rules



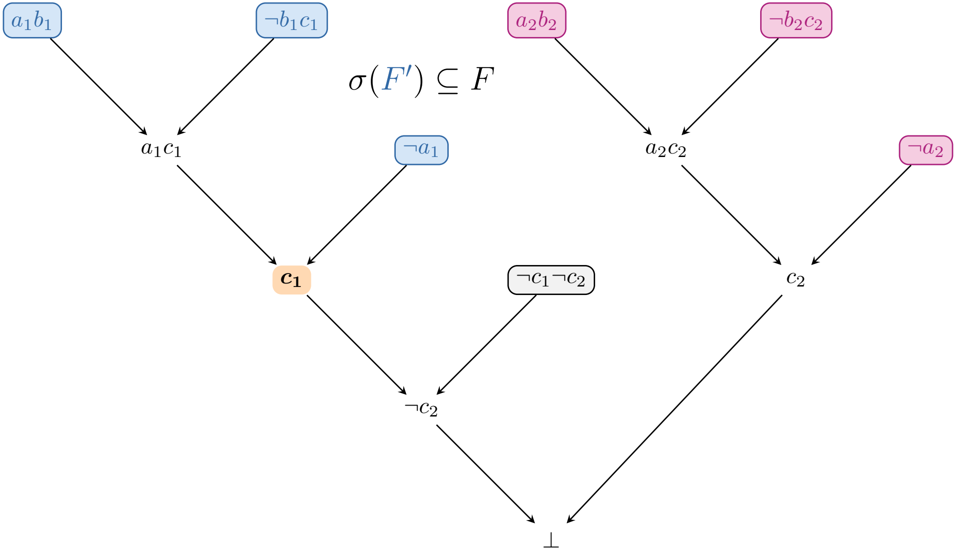
Krishnamurthy's Symmetry Rules



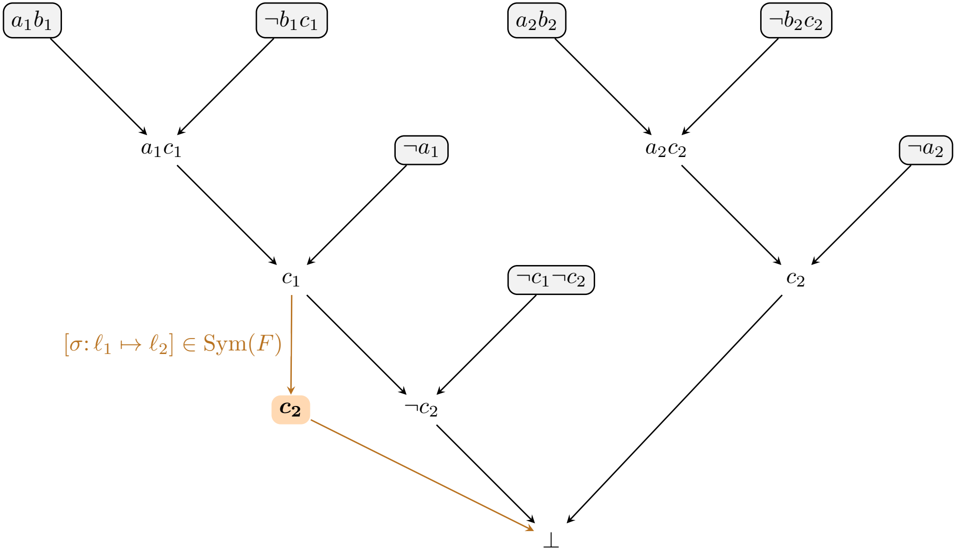
Krishnamurthy's Symmetry Rules



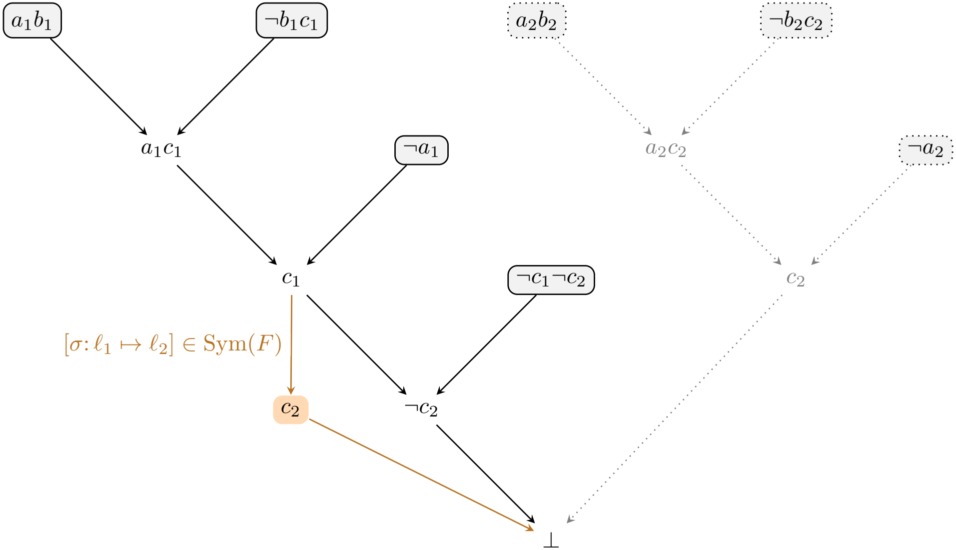
Krishnamurthy's Symmetry Rules



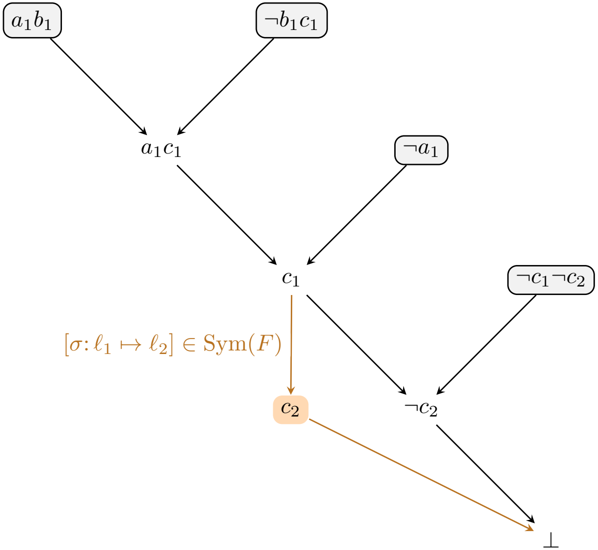
Krishnamurthy's Symmetry Rules



Krishnamurthy's Symmetry Rules



Krishnamurthy's Symmetry Rules



The SRC Proof Systems

Have a derivation $\pi : F' \vdash C$ from a subformula $F' \subseteq F$.
To derive $\sigma(C)$ from C in one step we need a renaming σ with

SRC-1 (Global Symmetries)

$$\sigma(F) = F$$

SRC-2 (Local Symmetries)

$$\sigma(F') \subseteq F$$

SRC-3 (Dynamic Symmetries)

also allow symmetries in resolvents

$$\text{Resolution} \rightarrow \text{SRC-1} \rightarrow \text{SRC-2} \rightarrow \text{SRC-3}$$

Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph G: $\mathrm{Aut}(G) = \{\mathrm{id}\}$

Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph G:	$\mathrm{Aut}(G) = \{\mathrm{id}\}$
Lemma:	Asymmetric graphs $\implies$ Asymmetric ISO-formula

Battle SRC-1 With Asymmetric Graphs

Asymmetric Graph G:	$\mathrm{Aut}(G) = \{\mathrm{id}\}$
Lemma:	Asymmetric graphs $\implies$ Asymmetric ISO-formula
Lemma:	Asymmetric formula $\implies$ Res-Size $=$ SRC-1-Size [Szeider]

Asymmetric Graphs of Large WL-Dimension

[Dawar and Khan] showed: There are pairs of non-isomorphic graphs that are

- asymmetric (unlike CFI-graphs)
- have small size $\mathrm{O}(k)$
- with large WL-dim k
- and color classes of size 4

Without looking at ISO-formula:

$$(G,\lambda) \equiv_{\mathscr{L}_k} (H,\mu) \implies \mathrm{Size}\big(\mathrm{ISO}(G,H) \vdash \bot\big) \geq \exp\left(\Omega\left(\frac{k^2}{\text{sum of color class sizes}}\right)\right)$$

An Exponential GI Lower Bound for SRC-1

Our Result:

There is a family of non-isomorphic graph pairs (G_n, H_n)

- with $\mathrm{O}(n)$ vertices each,
- such that any SRC-1 refutation of $\mathrm{ISO}(G_n, H_n)$ requires

size $\exp\big(\Omega(n)\big)$.

CFI graphs (used for Resolution GI lower bound) don't work! [Schweitzer & Seebach]

Summary and Open Problems

- Number of Variables for Graph Differentiation = Narrow Resolution Width
- Upper and lower bounds for refuting GI in Resolution
- Exponential Size Lower Bound for GI in SRC-1

Q. *How does one show "true" exponential lower bounds (for a symmetric formula) in the systems SRC-2 or SRC-3?*