Recognizing read-once functions from depth-3 formulas

Alexander Kozachinskiy

National Research University Higher School of Economics

CSR 2018, June 8, 2018

Main Problem

Problem: given a monotone Boolean formula Φ, is it possible to come up with a formula equivalent to Φ, in which every variable appears at most once?

Main Problem

Problem: given a monotone Boolean formula Φ, is it possible to come up with a formula equivalent to Φ, in which every variable appears at most once?

Examples

Main Problem

Problem: given a monotone Boolean formula Φ, is it possible to come up with a formula equivalent to Φ, in which every variable appears at most once?

Examples

- $\Phi = (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_4)$

Main Problem

Problem: given a monotone Boolean formula Φ, is it possible to come up with a formula equivalent to Φ, in which every variable appears at most once?

Examples

- $\Phi = (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_4)$

  YES: $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$.

Main Problem

Problem: given a monotone Boolean formula Φ, is it possible to come up with a formula equivalent to Φ, in which every variable appears at most once?

Examples

- $\Phi = (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_4)$

  YES: $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$.

- $\Phi = x_2 x_3 x_4 \vee x_1 x_3 x_4 \vee x_1 x_2 x_4 \vee x_2 x_3 x_4$.

Main Problem

Problem: given a monotone Boolean formula Φ, is it possible to come up with a formula equivalent to Φ, in which every variable appears at most once?

Examples

- $\Phi = (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_4)$

  YES: $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$.

- $\Phi = x_2x_3x_4 \vee x_1x_3x_4 \vee x_1x_2x_4 \vee x_2x_3x_4$.

  NO: check all formulas.

Definitions

Definition

A monotone Boolean formula is called *read-k*, if every variable appears in it at most k times

Special case: read-once (read-1) formulas.

Definition

A monotone Boolean function is called *read-once*, if there is a monotone read-once formula, computing it.

Definitions

Definition

A monotone Boolean formula is called *read-k*, if every variable appears in it at most k times

Special case: read-once (read-1) formulas.

Definition

A monotone Boolean function is called *read-once*, if there is a monotone read-once formula, computing it.

Read-once recognition

Definition

ROR is the problem of deciding whether a given monotone Boolean formula computes read-once function.

Special cases of ROR:

- depth-2 ROR (the input is a monotone DNF or CNF);
- read-2 ROR (the input is read-2 monotone formula);
- depth-3 read-2 ROR.

Read-once recognition

Definition

ROR is the problem of deciding whether a given monotone Boolean formula computes read-once function.

Special cases of ROR:

- depth-2 ROR (the input is a monotone DNF or CNF);
- read-2 ROR (the input is read-2 monotone formula);
- depth-3 read-2 ROR.

Read-once recognition

Definition

ROR is the problem of deciding whether a given monotone Boolean formula computes read-once function.

Special cases of ROR:

- depth-2 ROR (the input is a monotone DNF or CNF);
- read-2 ROR (the input is read-2 monotone formula);
- depth-3 read-2 ROR.

Read-once recognition

Definition

ROR is the problem of deciding whether a given monotone Boolean formula computes read-once function.

Special cases of ROR:

- depth-2 ROR (the input is a monotone DNF or CNF);
- read-2 ROR (the input is read-2 monotone formula);
- depth-3 read-2 ROR.

Read-once recognition

Definition

ROR is the problem of deciding whether a given monotone Boolean formula computes read-once function.

Special cases of ROR:

- depth-2 ROR (the input is a monotone DNF or CNF);
- read-2 ROR (the input is read-2 monotone formula);
- depth-3 read-2 ROR.

Complexity of ROR

ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Complexity of ROR

ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Complexity of ROR

ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Complexity of ROR

ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Complexity of ROR

ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Complexity of ROR

ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Complexity of ROR

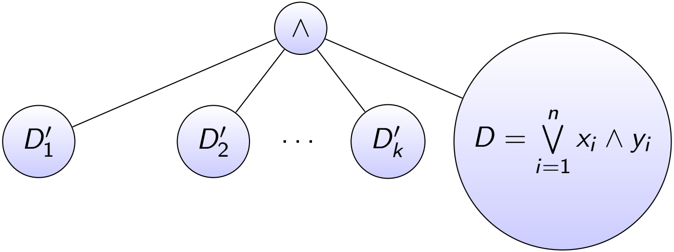
ROR $\in \Sigma_2^P$ (observation).

ROR $\in$ coNP (follows from [Gur77]).

More detalied complexity of ROR:

- depth-2 ROR $\in$ P, $O(nl)$-time algorithm [GMR08].
- depth-4 read-2 ROR is coNP-complete [EMR11]
- depth-3 read-3 ROR is coNP-complete [Gur10].
- depth-3 read-2 ROR is coNP-complete [this work].

Structure of hard depth-3 read-2 formulas

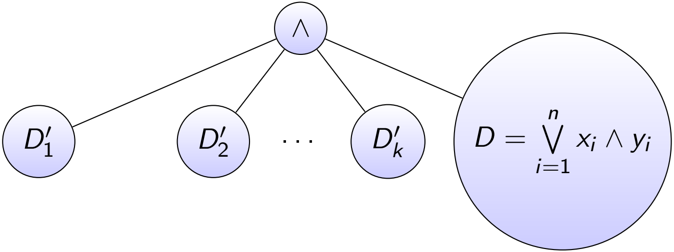


Here $D'_1, \ldots, D'_k$ are read-once disjoint DNF's over $x_1, y_1, \ldots, x_n, y_n$.

Corollary

ROR *is coNP-complete even for inputs of the form* $A \wedge D$*, where* A *is a monotone read-once* $\bigwedge - \bigvee - \bigwedge$ *formula and* $D = \bigvee_{i=1}^{n} x_i \wedge y_i$.

Structure of hard depth-3 read-2 formulas

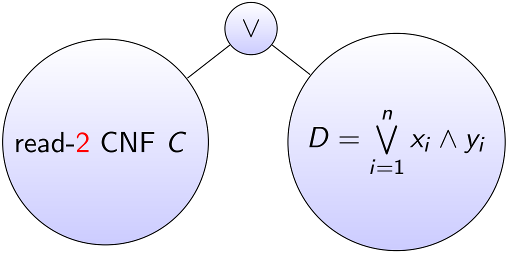


Here $D'_1, \ldots, D'_k$ are read-once disjoint DNF's over $x_1, y_1, \ldots, x_n, y_n$.

Corollary

ROR *is coNP-complete even for inputs of the form* $A \wedge D$, *where* A *is a monotone read-once* $\bigwedge - \bigvee - \bigwedge$ *formula and* $D = \bigvee_{i=1}^{n} x_i \wedge y_i$.

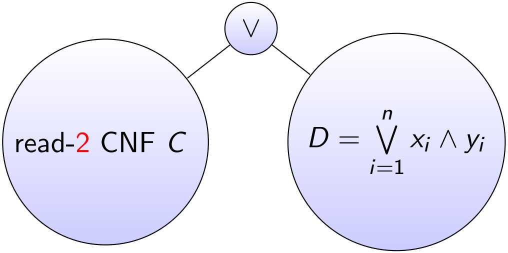
Gurvich's hardness result in more detail.



Theorem (Gur10)

ROR *is coNP-complete for inputs of the form* $C \vee D$*, where* C *is a monotone read-2 CNF and* $D = \bigvee_{i=1}^{n} x_i \wedge y_i$*.*

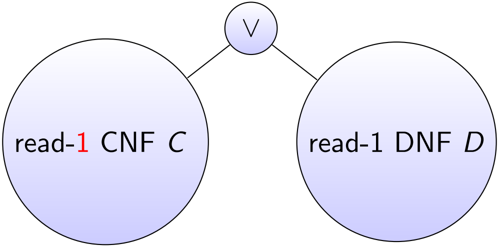
Gurvich's hardness result in more detail.



Theorem (Gur10)

ROR *is coNP-complete for inputs of the form* $C \vee D$*, where* C *is a monotone read-2 CNF and* $D = \bigvee_{i=1}^{n} x_i \wedge y_i$.

Tractable subclass of depth-3 formulas



Theorem (this work)

ROR *is solvable in polynomial time for inputs of the form* $C \vee D$*, where* C *is a monotone read-1 CNF,* D *is a monotone read-1 DNF and every variable of* C *occurs also in* D*.*

Reduction

Reduction from
CO-CLIQUE$= \{(G, k)$: there is no clique of size k in $G\}$.

$$(G = (V, E), k) \impliedby \text{(Alice} \wedge \text{Bob) vs Merlin game}$$
$$\impliedby \text{depth-3 read-2 } \Phi.$$

Reduction

Reduction from
CO-CLIQUE$= \{(G, k)$: there is no clique of size k in $G\}$.

$$\begin{aligned}(G = (V, E), k) &\Longleftarrow \text{(Alice} \wedge \text{Bob) vs Merlin game} \\ &\Longleftarrow \text{depth-3 read-2 } \Phi.\end{aligned}$$

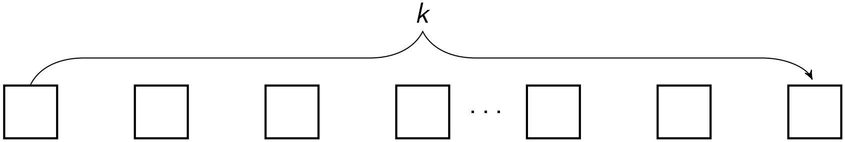
Reduction

Reduction from
CO-CLIQUE$= \{(G, k) :$ there is no clique of size k in $G\}$.

$$\begin{aligned}(G = (V, E), k) &\Longleftarrow \text{(Alice} \wedge \text{Bob) vs Merlin game} \\ &\Longleftarrow \text{depth-3 read-2 } \Phi.\end{aligned}$$

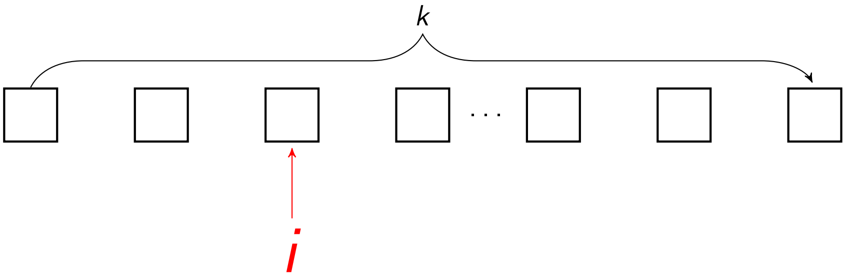
Game

$(G = (V, E), k) \Longleftarrow$ (Alice and Bob) vs Merlin game



Game

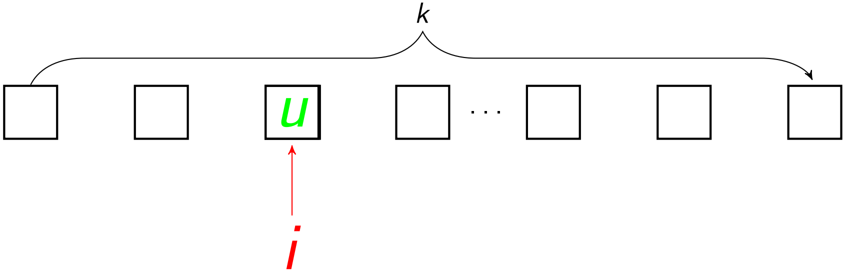
$(G = (V, E), k) \Longleftarrow$ (Alice and Bob) vs Merlin game



1. Merlin chooses $i \in [k]$;

Game

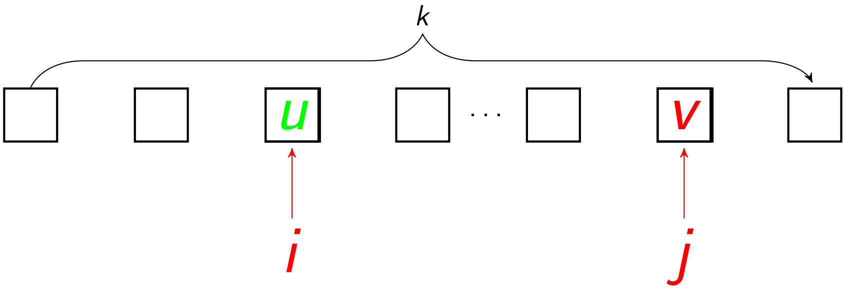
$(G = (V, E), k) \Longleftarrow$ (Alice and Bob) vs Merlin game



1. Merlin chooses $i \in [k]$;
2. Alice chooses $u \in V$;

Game

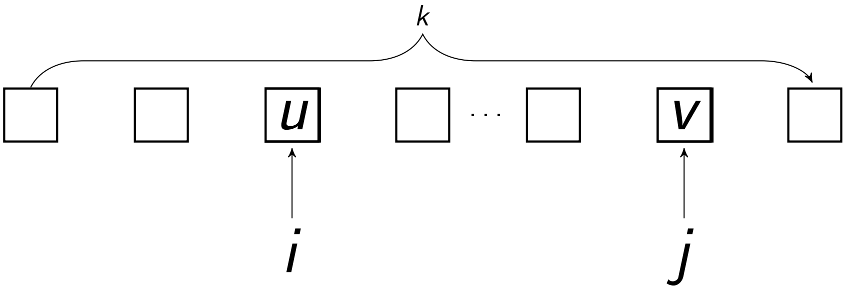
$(G = (V, E), k) \Longleftarrow$ (Alice and Bob) vs Merlin game



1. Merlin chooses $i \in [k]$;
2. Alice chooses $u \in V$;
3. Merlin chooses $j \in [k], v \in V$ s.t $j \neq i$ and $\{u, v\} \notin E$.

Game

$(G = (V, E), k) \Longleftarrow$ (Alice and Bob) vs Merlin game

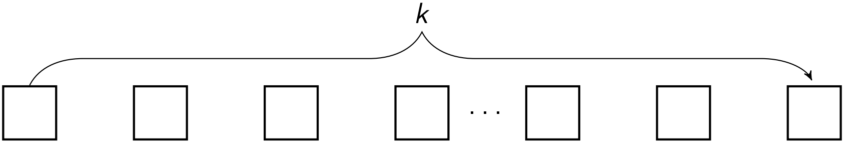


1. Merlin chooses $i \in [k]$;
2. Alice chooses $u \in V$;
3. Merlin chooses $j \in [k], v \in V$ s.t $j \neq i$ and $\{u, v\} \notin E$.
4. Bob comes in and *tries to guess a cell which was touched first.*

There is a k-clique in $G \iff$ Alice and Bob have a winning strategy

The first direction

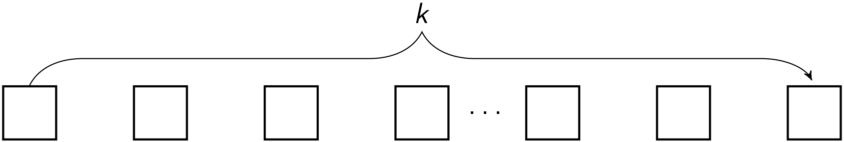
There is a k-clique in $G \implies$ Alice and Bob have a winning strategy



The first direction

There is a k-clique in $G \implies$ Alice and Bob have a winning strategy

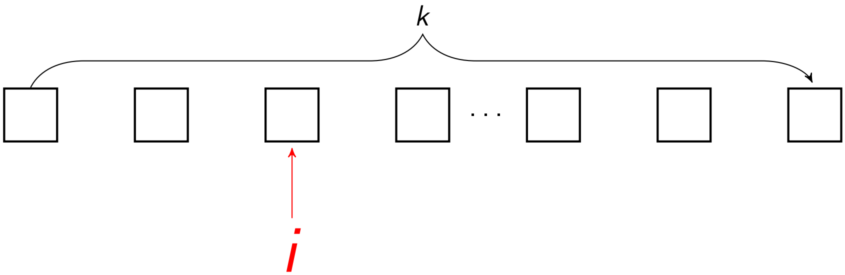
Assume $w_1, \ldots, w_k$ form a clique.



The first direction

There is a k-clique in $G \implies$ Alice and Bob have a winning strategy

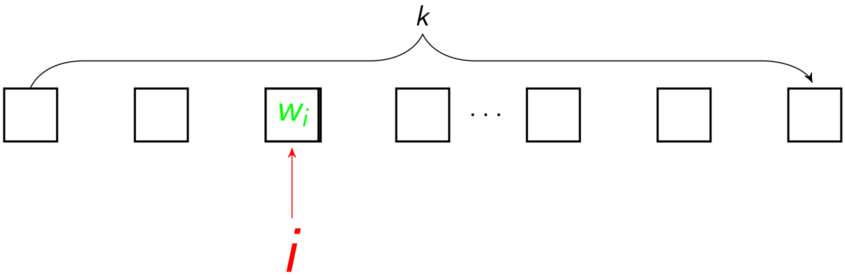
Assume $w_1, \ldots, w_k$ form a clique.



The first direction

There is a k-clique in G $\implies$ Alice and Bob have a winning strategy

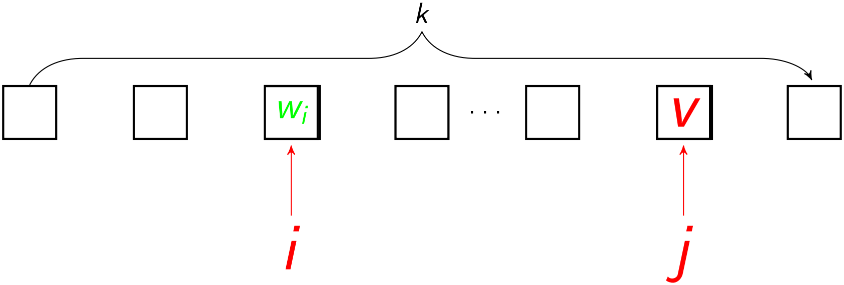
Assume $w_1, \ldots, w_k$ form a clique.



The first direction

There is a k-clique in $G \implies$ Alice and Bob have a winning strategy

Assume $w_1, \ldots, w_k$ form a clique.



Bob outputs a cell with a corresponding vertex of a clique in it.

The second direction

Alice and Bob have a winning strategy $\implies$ **there is a** k**-clique in** G.

The second direction

Alice and Bob have a winning strategy $\implies$ there is a k-clique in G.

Assume that v_i is the vertex which Alice puts in the i^{th} cell at Merlin request.

The second direction

Alice and Bob have a winning strategy $\implies$ there is a k-clique in G.

Assume that v_i is the vertex which Alice puts in the i^{th} cell at Merlin request.

Observe that $v_1, \dots, v_k$ form a clique.

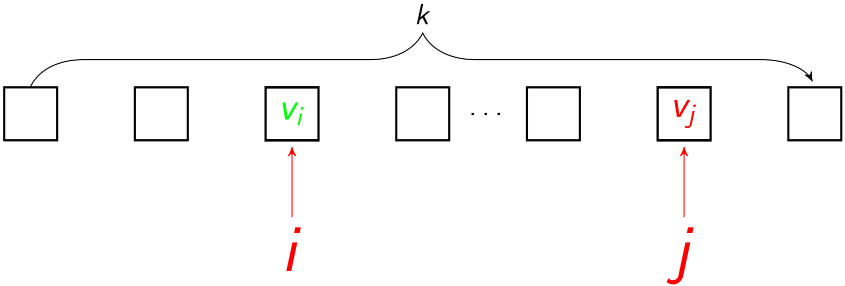
The second direction

Alice and Bob have a winning strategy $\implies$ there is a k-clique in G.

Assume that v_i is the vertex which Alice puts in the i^{th} cell at Merlin request.

Observe that $v_1, \ldots, v_k$ form a clique.

Indeed, if v_i and v_j are not connected by an edge, then:



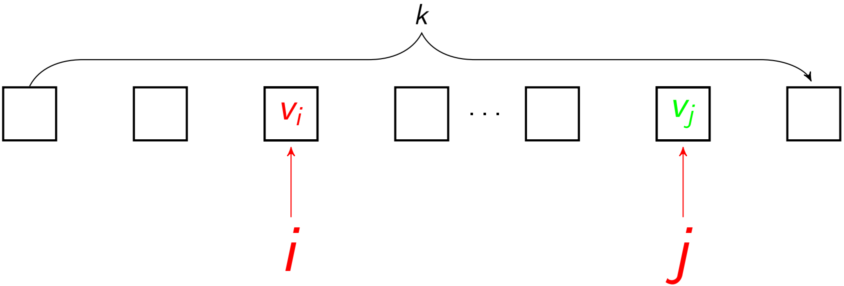
The second direction

Alice and Bob have a winning strategy $\implies$ there is a k**-clique in** G.

Assume that v_i is the vertex which Alice puts in the i^{th} cell at Merlin request.

Observe that $v_1, \ldots, v_k$ form a clique.

Indeed, if v_i and v_j are not connected by an edge, then:



From game to a formula

Constructing a formula:

- Variables correspond to possible outcomes of Alice-Merlin interaction:

$$x_{j,v}^{i,u} \qquad \text{over all } i \neq j,\ \{u,v\} \notin E.$$

- Assignments of variables encode Bob's possible strategies.

$$x_{j,v}^{i,u} = 1 \iff \text{Bob picks } i^{th} \text{ cell when he sees } \{(i,u),(j,v)\}.$$

- Bob's strategy is correct iff

$$B = \bigvee (x_{j,v}^{i,u} \wedge x_{i,u}^{j,v}) = 0.$$

- Alice can always reach a variable set to 1 iff

$$A = \bigwedge_{i} \bigvee_{u} \bigwedge_{(j,v)} x_{j,v}^{i,u} = 1.$$

From game to a formula

Constructing a formula:

- Variables correspond to possible outcomes of Alice-Merlin interaction:

$$x_{j,v}^{i,u} \qquad \text{over all } i \neq j,\ \{u,v\} \notin E.$$

- Assignments of variables encode Bob's possible strategies.

$$x_{j,v}^{i,u} = 1 \iff \text{Bob picks } i^{th} \text{ cell when he sees } \{(i,u),(j,v)\}.$$

- Bob's strategy is correct iff

$$B = \bigvee (x_{j,v}^{i,u} \wedge x_{i,u}^{j,v}) = 0.$$

- Alice can always reach a variable set to 1 iff

$$A = \bigwedge_{i} \bigvee_{u} \bigwedge_{(j,v)} x_{j,v}^{i,u} = 1.$$

From game to a formula

Constructing a formula:

- Variables correspond to possible outcomes of Alice-Merlin interaction:

$$x^{i,u}_{j,v} \qquad \text{over all } i \neq j,\ \{u, v\} \notin E.$$

- Assignments of variables encode Bob's possible strategies.

$x^{i,u}_{j,v} = 1 \iff$ Bob picks i^{th} cell when he sees $\{(i, u), (j, v)\}$.

- Bob's strategy is correct iff

$$B = \bigvee (x^{i,u}_{j,v} \wedge x^{j,v}_{i,u}) = 0.$$

- Alice can always reach a variable set to 1 iff

$$A = \bigwedge_{i} \bigvee_{u} \bigwedge_{(j,v)} x^{i,u}_{j,v} = 1.$$

From game to a formula

Constructing a formula:

- Variables correspond to possible outcomes of Alice-Merlin interaction:

$$x_{j,v}^{i,u} \qquad \text{over all } i \neq j,\ \{u,v\} \notin E.$$

- Assignments of variables encode Bob's possible strategies.

$$x_{j,v}^{i,u} = 1 \iff \text{Bob picks } i^{th} \text{ cell when he sees } \{(i,u),(j,v)\}.$$

- Bob's strategy is correct iff

$$B = \bigvee (x_{j,v}^{i,u} \wedge x_{i,u}^{j,v}) = 0.$$

- Alice can always reach a variable set to 1 iff

$$A = \bigwedge_{i} \bigvee_{u} \bigwedge_{(j,v)} x_{j,v}^{i,u} = 1.$$

From game to a formula

Constructing a formula:

- Variables correspond to possible outcomes of Alice-Merlin interaction:

$$x_{j,v}^{i,u} \qquad \text{over all } i \neq j,\ \{u, v\} \notin E.$$

- Assignments of variables encode Bob's possible strategies.

$$x_{j,v}^{i,u} = 1 \iff \text{Bob picks } i^{th} \text{ cell when he sees } \{(i, u), (j, v)\}.$$

- Bob's strategy is correct iff

$$B = \bigvee (x_{j,v}^{i,u} \wedge x_{i,u}^{j,v}) = 0.$$

- Alice can always reach a variable set to 1 iff

$$A = \bigwedge_{i} \bigvee_{u} \bigwedge_{(j,v)} x_{j,v}^{i,u} = 1.$$

Final remarks

$$
\begin{aligned}
G \text{ has } k\text{-clique} &\iff \text{Alice and Bob have a winning strategy} \\
&\iff A \to B \text{ is not a tautology} \\
&\iff A \wedge (w_1 w_3 \vee w_2 w_4) \wedge (B \vee w_1 w_2 \vee w_3 w_4) \notin \mathrm{ROR}
\end{aligned}
$$

Thank you! Any questions?