Causal limits of distributed computation

Francesco d'Amore

Based on joint work with the Xavier Coiteux-Roy, Rishikesh Gajjala, Fabian Kuhn, François Le Gall, Henrik Lievonen, Augusto Modanese, Marc-Olivier Renou, Gustav Schmid, Jukka Suomela

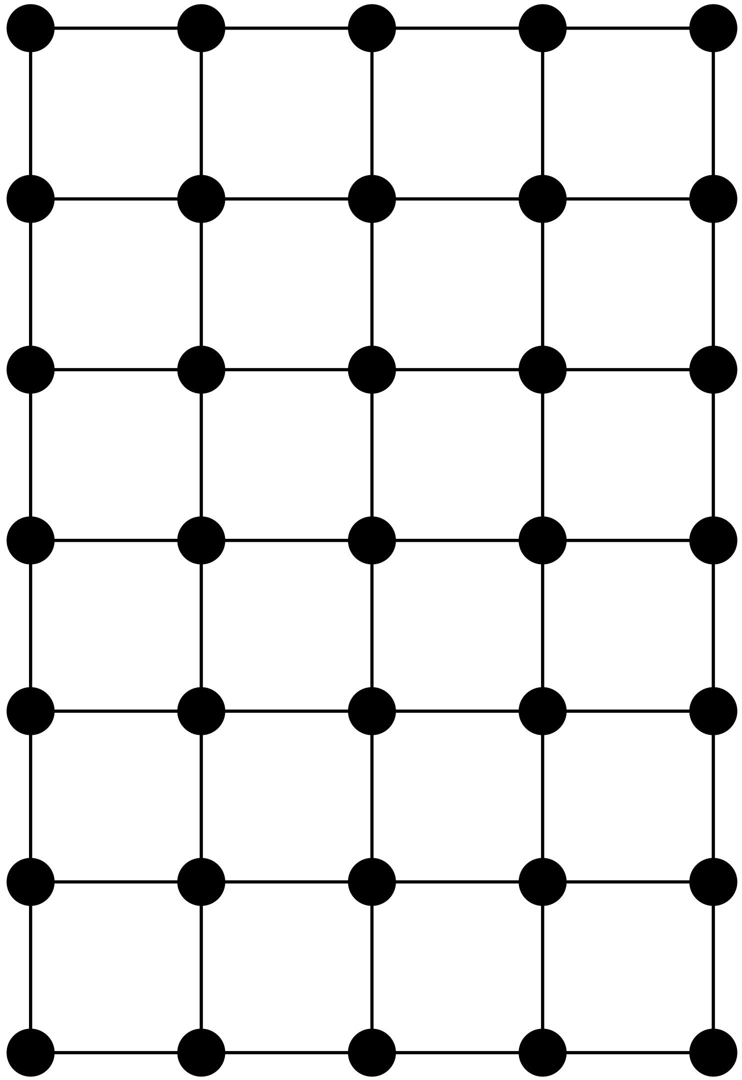
Bocconi University

05 December 2023

The LOCAL model

Introduced by [Linial, FOCS '87]

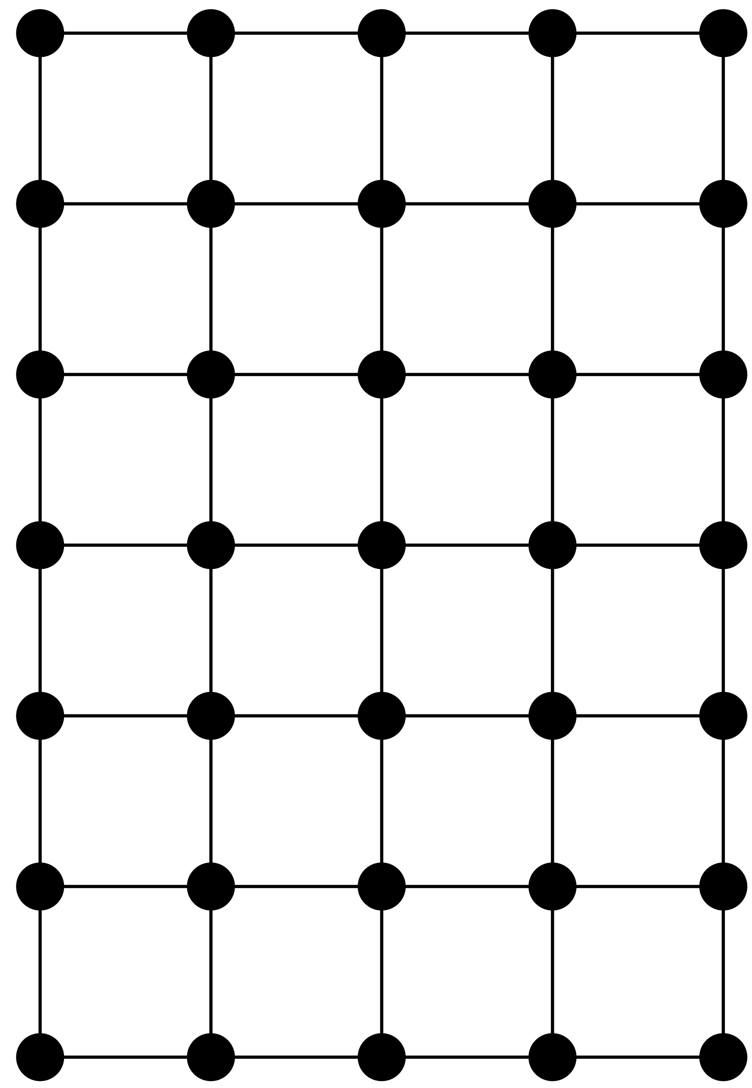
- **distributed systems** with n nodes/processors
- synchronous rounds
- infinite computational power
- unbounded bandwidth



The LOCAL model

Introduced by [Linial, FOCS '87]

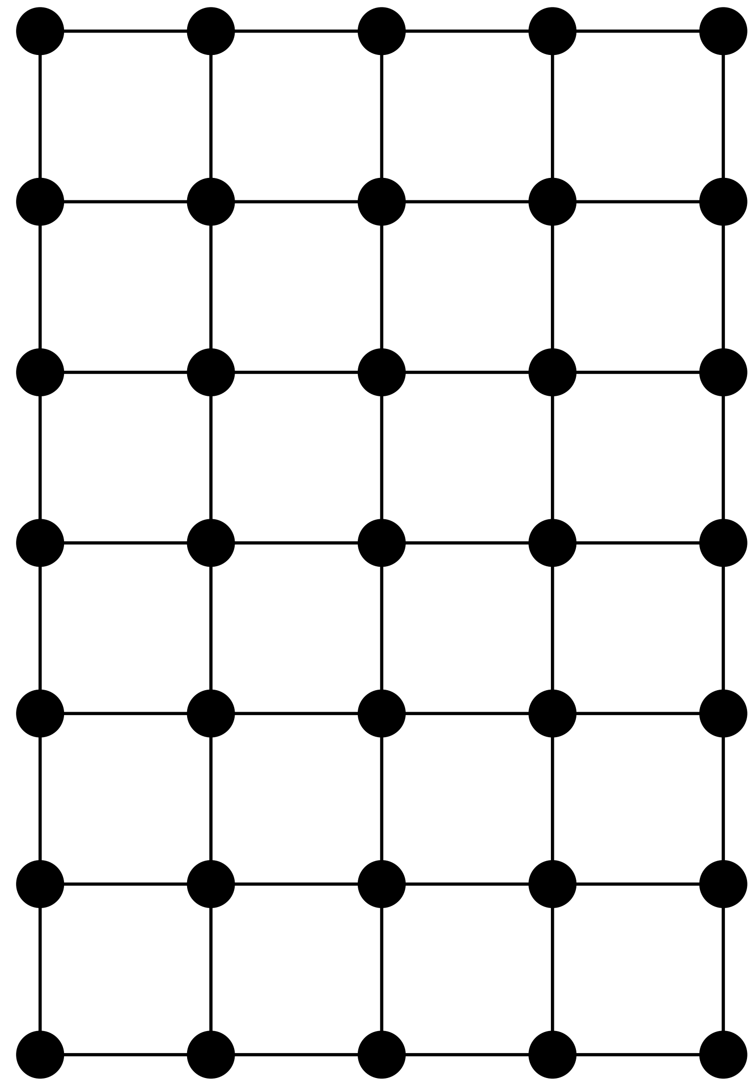
- **distributed systems** with n nodes/processors
- synchronous rounds
- infinite computational power
- unbounded bandwidth
- unique IDs in $\{1, \ldots, n^{\Theta(1)}\}$



The LOCAL model

Introduced by [Linial, FOCS '87]

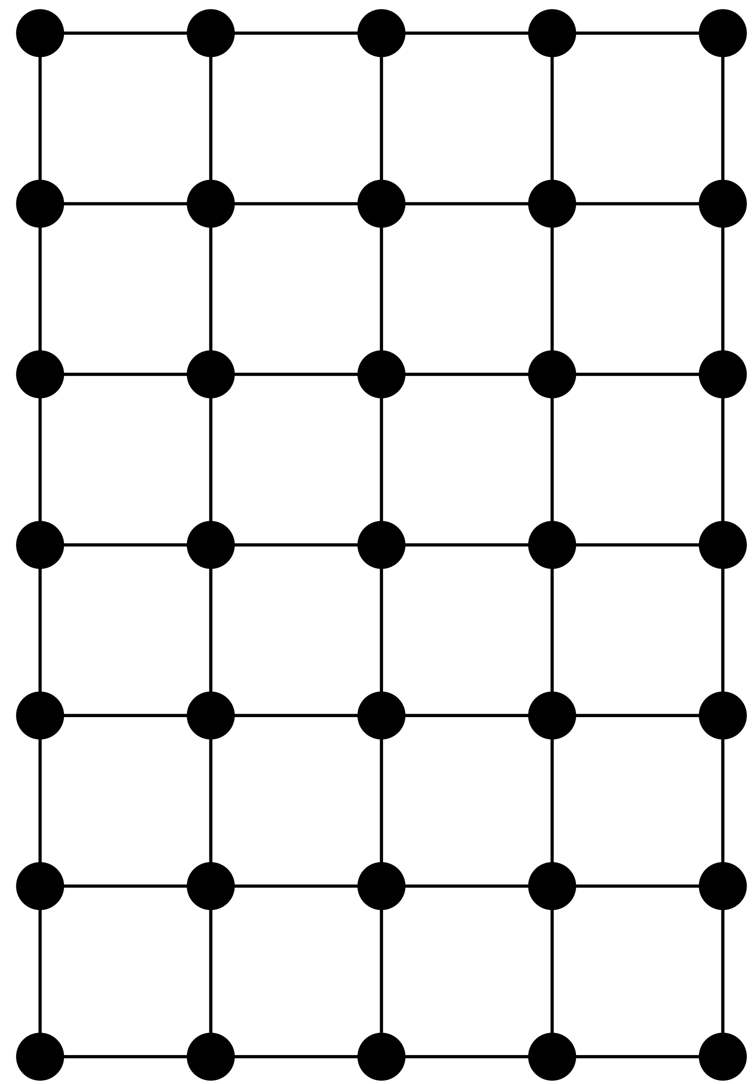
- **distributed systems** with n nodes/processors
  - synchronous rounds
  - infinite computational power
  - unbounded bandwidth
  - unique IDs in $\{1, \ldots, n^{\Theta(1)}\}$
  - knowledge of n



The LOCAL model

Introduced by [Linial, FOCS '87]

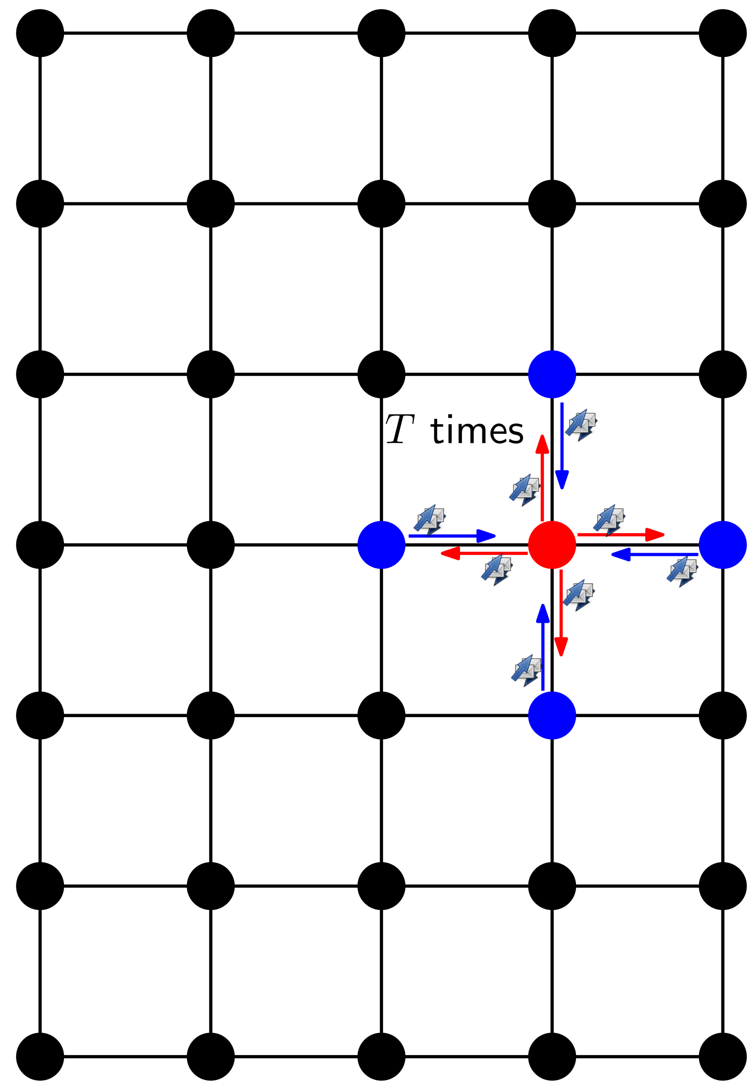
- **distributed systems** with n nodes/processors
  - synchronous rounds
  - infinite computational power
  - unbounded bandwidth
  - unique IDs in $\{1, \ldots, n^{\Theta(1)}\}$
  - knowledge of n
- **complexity** measure: $\#$ of rounds



The LOCAL model

Introduced by [Linial, FOCS '87]

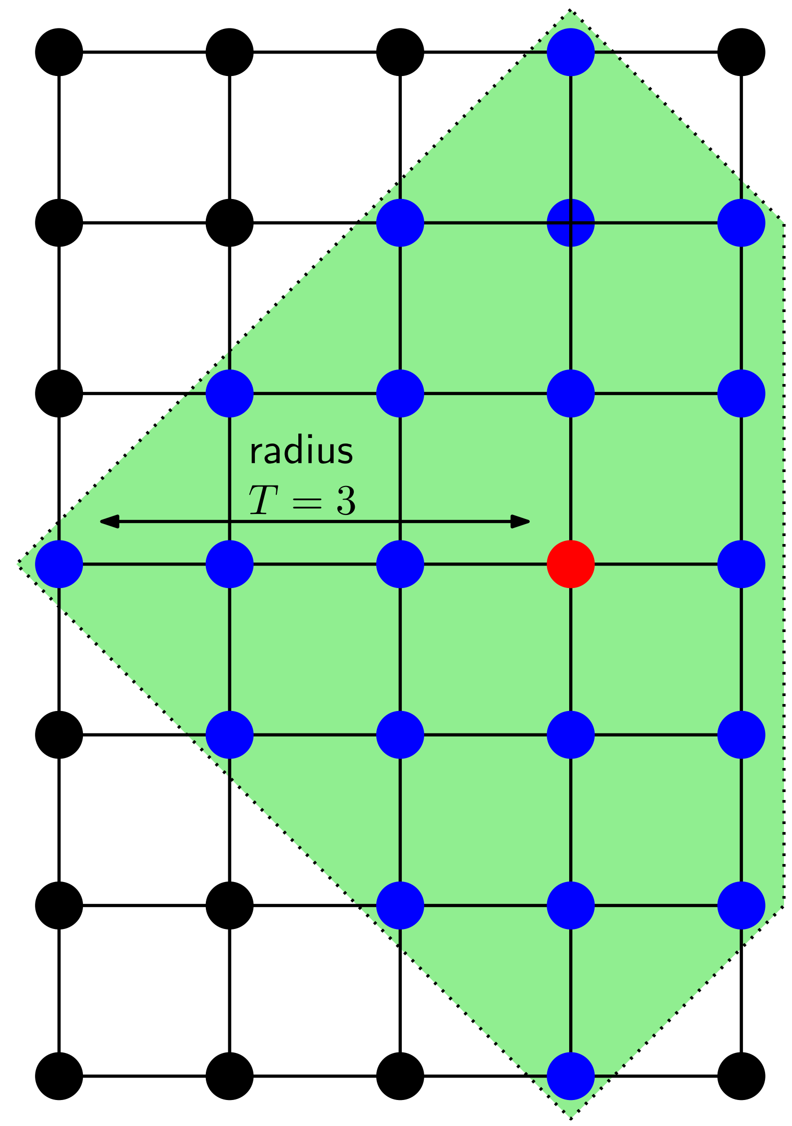
- **distributed systems** with n nodes/processors
  - synchronous rounds
  - infinite computational power
  - unbounded bandwidth
  - unique IDs in $\{1, \ldots, n^{\Theta(1)}\}$
  - knowledge of n
- **complexity** measure: # of rounds
  - E.g., T rounds of computation



The LOCAL model

Introduced by [Linial, FOCS '87]

- **distributed systems** with n nodes/processors
  - synchronous rounds
  - infinite computational power
  - unbounded bandwidth
  - unique IDs in $\{1, \ldots, n^{\Theta(1)}\}$
  - knowledge of n

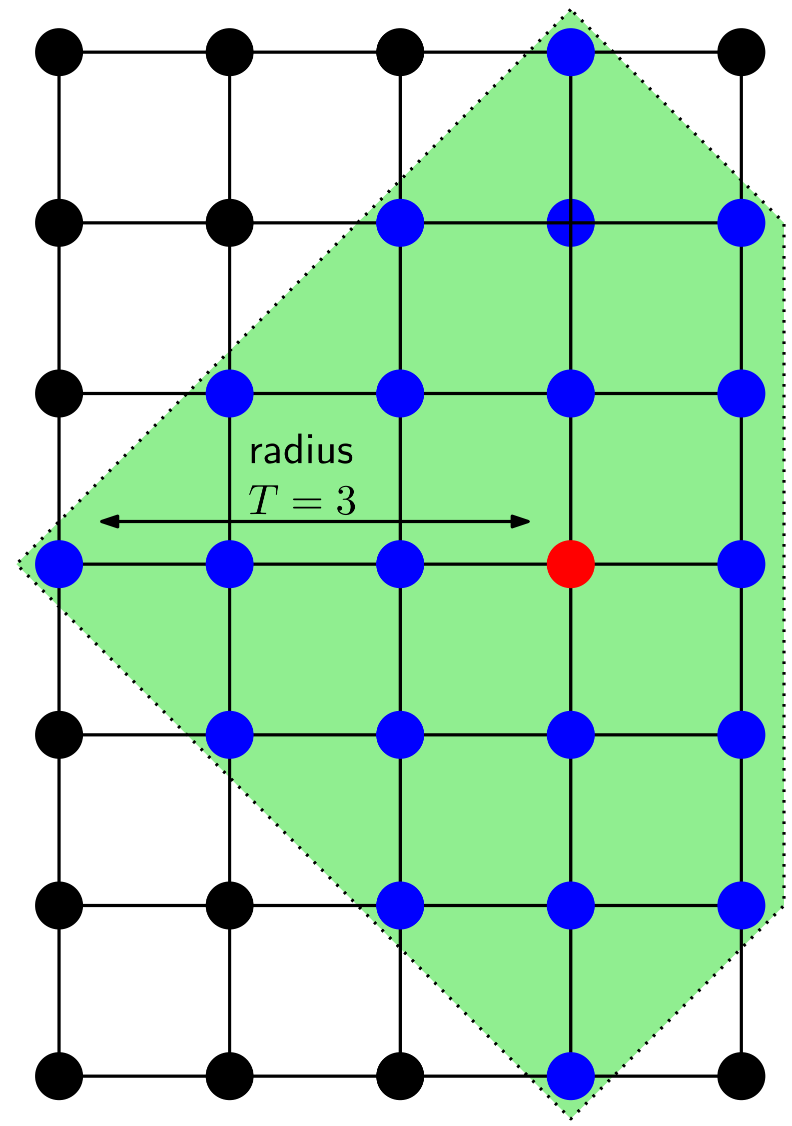


- **complexity** measure: # of rounds
  - E.g., T rounds of computation
  - also called locality

The LOCAL model

Introduced by [Linial, FOCS '87]

- **distributed systems** with n nodes/processors
  - synchronous rounds
  - infinite computational power
  - unbounded bandwidth
  - unique IDs in $\{1, \ldots, n^{\Theta(1)}\}$
  - knowledge of n
- **complexity** measure: # of rounds
  - E.g., T rounds of computation
  - also called locality
- **randomized**-LOCAL
  - infinite i.i.d. random bit strings
  - error probability $\leq 1/n$



The LOCAL model

- strongest **lower bounds** for synchronous distributed computation

- upper bounds $\implies$ lower bounds cannot be moved forward

The LOCAL model

- strongest **lower bounds** for synchronous distributed computation
- upper bounds $\implies$ lower bounds cannot be moved forward
- modelling cost(message exchange) $\gg$ cost(computation)

The LOCAL model

- strongest **lower bounds** for synchronous distributed computation
- upper bounds $\implies$ lower bounds cannot be moved forward
- modelling cost(message exchange) $\gg$ cost(computation)

Sample of **previous results**

- [Linial, SIAM J. of C. '92]
  - 3-coloring cycles: $\Omega(\log^\star n)$
  - $\mathcal{O}(\sqrt{d})$-coloring d-regular trees: Θ(diameter)

The LOCAL model

- strongest **lower bounds** for synchronous distributed computation
- upper bounds $\implies$ lower bounds cannot be moved forward
- modelling cost(message exchange) $\gg$ cost(computation)

Sample of **previous results**

- [Linial, SIAM J. of C. '92]
  - 3-coloring cycles: $\Omega(\log^\star n)$
  - $\mathcal{O}(\sqrt{d})$-coloring d-regular trees: Θ(diameter)
- [Kuhn and Wattenhofer, PODC '06]
  - $(\Delta + 1)$-coloring: $\mathcal{O}(\Delta \log \Delta + \log^\star n)$

The LOCAL model

- strongest **lower bounds** for synchronous distributed computation
- upper bounds $\implies$ lower bounds cannot be moved forward
- modelling cost(message exchange) $\gg$ cost(computation)

Sample of **previous results**

- [Linial, SIAM J. of C. '92]
  - 3-coloring cycles: $\Omega(\log^\star n)$
  - $\mathcal{O}(\sqrt{d})$-coloring d-regular trees: Θ(diameter)
- [Kuhn and Wattenhofer, PODC '06]
  - $(\Delta + 1)$-coloring: $\mathcal{O}(\Delta \log \Delta + \log^\star n)$
- [Chang and Pettie, FOCS '16]
  - $T(n)$-round rand-algoritm $\implies$ $T(2^{2^n})$-round det-algorithm

The LOCAL model

- strongest **lower bounds** for synchronous distributed computation
- upper bounds $\implies$ lower bounds cannot be moved forward
- modelling cost(message exchange) $\gg$ cost(computation)

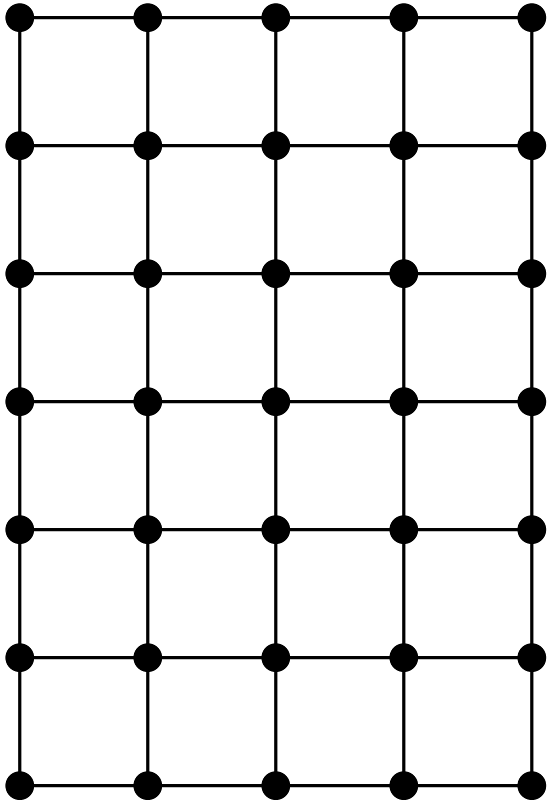
Sample of **previous results**

- [Linial, SIAM J. of C. '92]
  - 3-coloring cycles: $\Omega(\log^\star n)$
  - $\mathcal{O}(\sqrt{d})$-coloring d-regular trees: Θ(diameter)
- [Kuhn and Wattenhofer, PODC '06]
  - $(\Delta+1)$-coloring: $\mathcal{O}(\Delta \log \Delta + \log^\star n)$
- [Chang and Pettie, FOCS '16]
  - $T(n)$-round rand-algoritm $\implies$ $T(2^{2^n})$-round det-algorithm
- [Balliu et al., FOCS '19]
  - MM and MIS cannot be solved in $o(\Delta) + \mathcal{O}(\log^\star n)$

Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

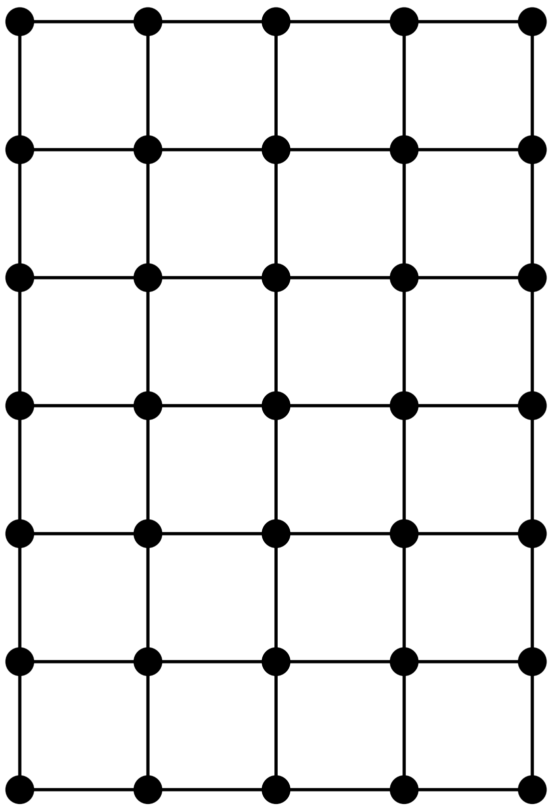
- problems that can be locally checked



Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets

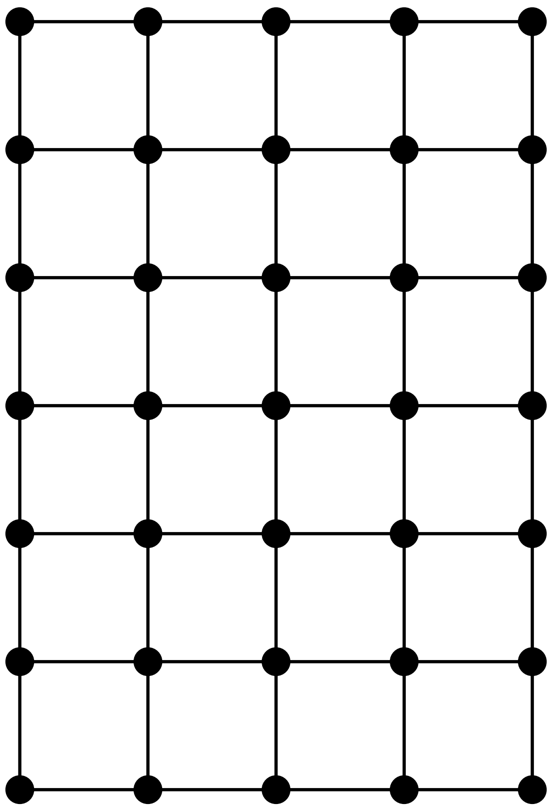


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked

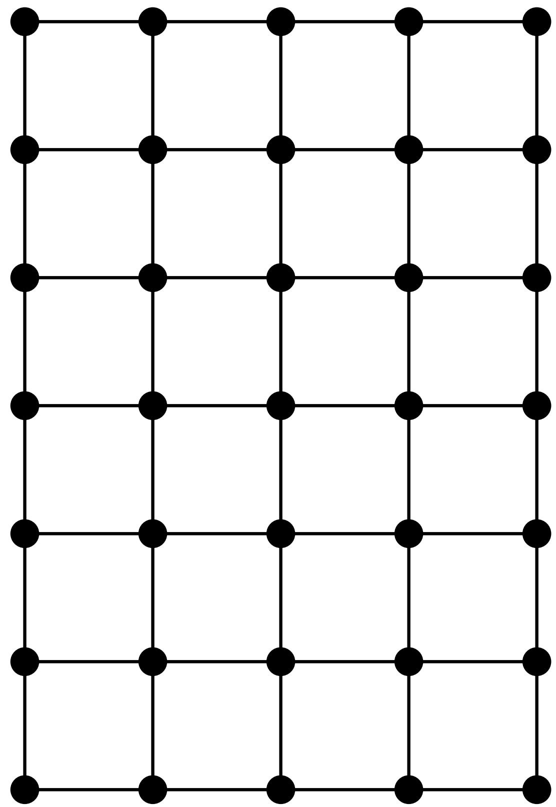
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$



Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

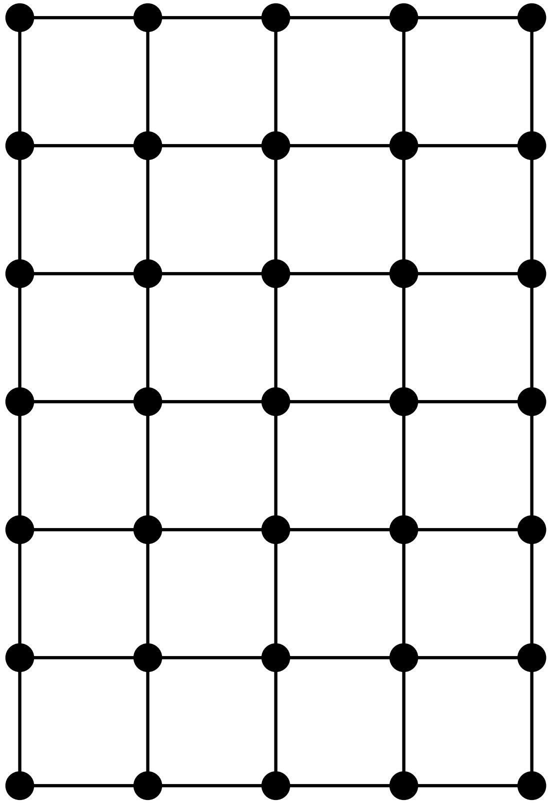
- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ



Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

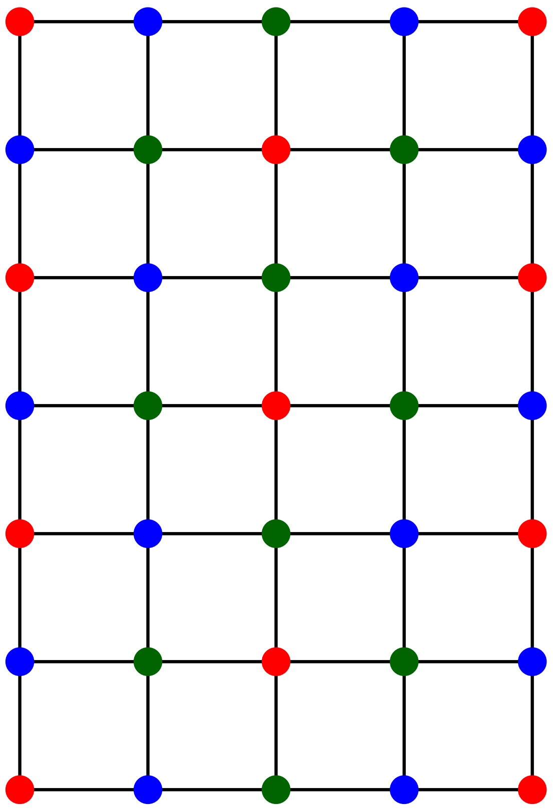


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$

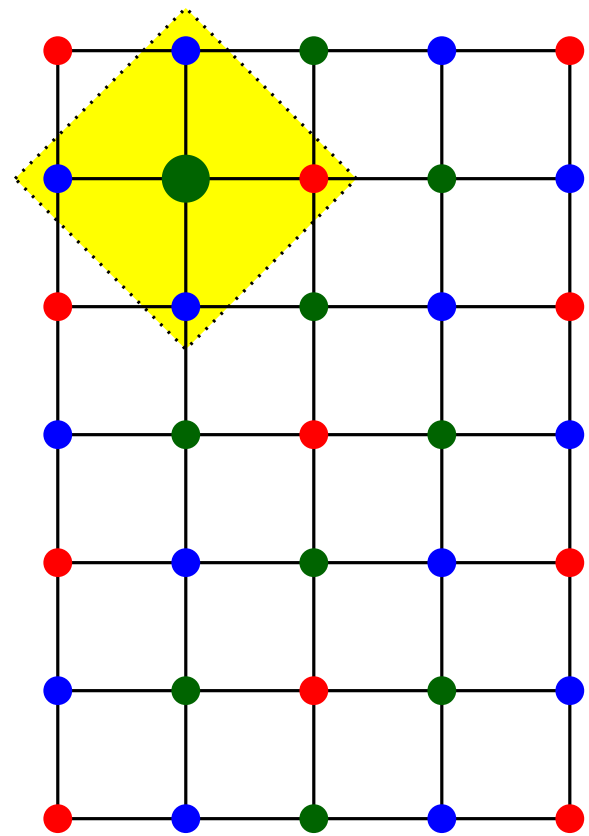


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$

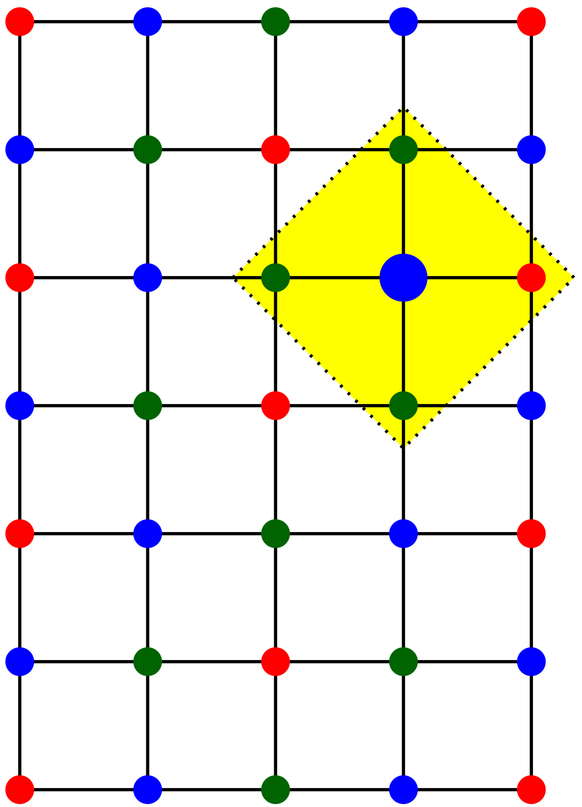


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$

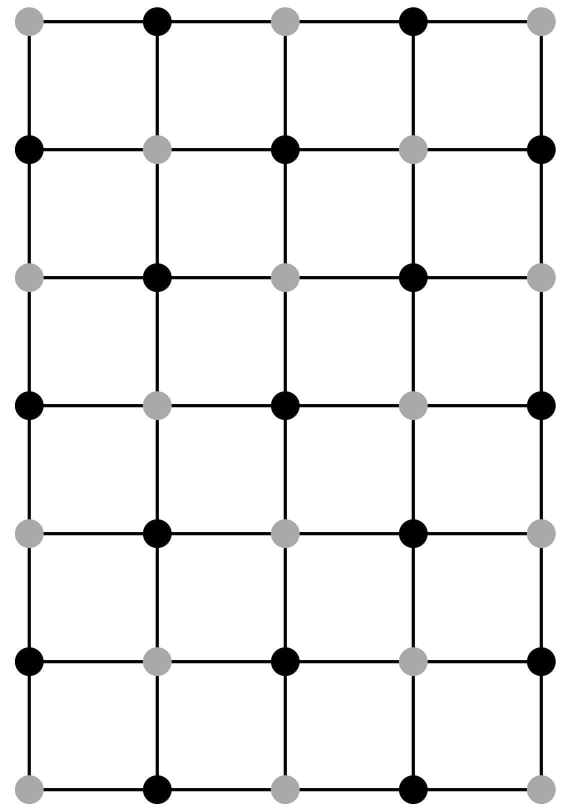


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
  - Σ_{in}, Σ_{out}: finite input and output label sets
  - family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
  - input: labelled graph of degree Δ
  - output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$
- maximal independent set: $r = 1$

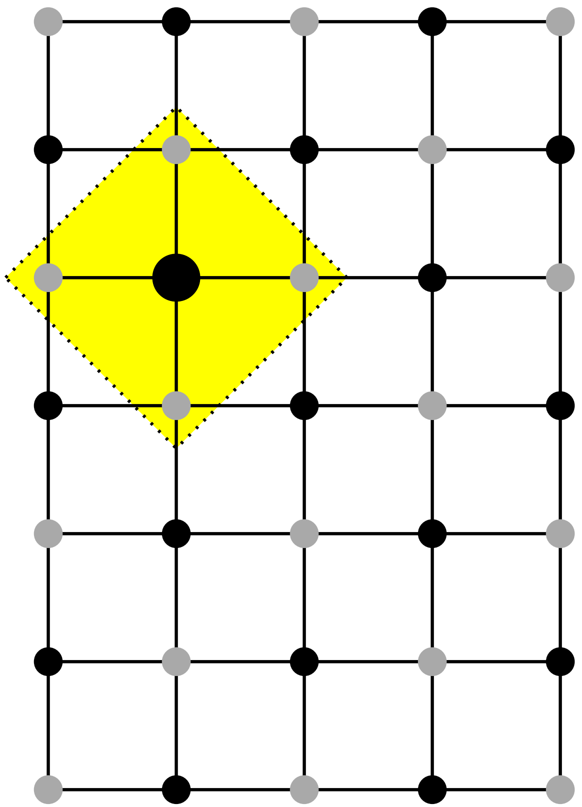


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$
- maximal independent set: $r = 1$

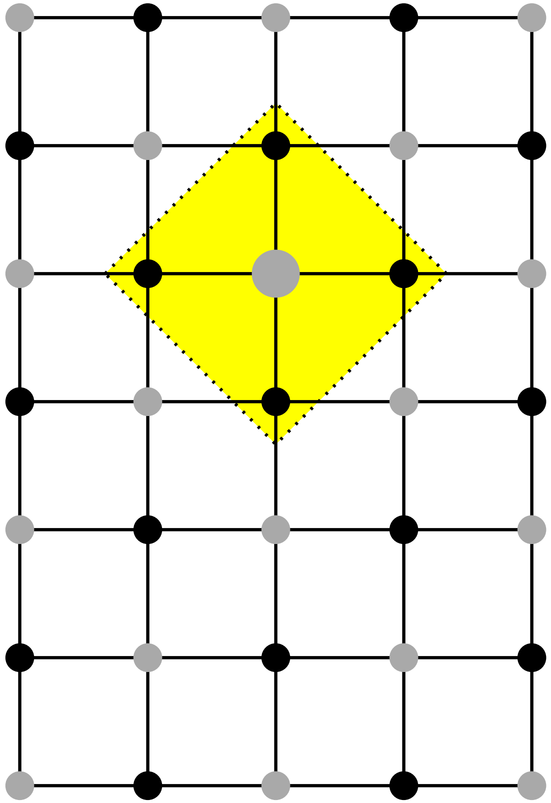


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
  - Σ_{in}, Σ_{out}: finite input and output label sets
  - family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
  - input: labelled graph of degree Δ
  - output: labelled graph that is valid iff each node verifies positively

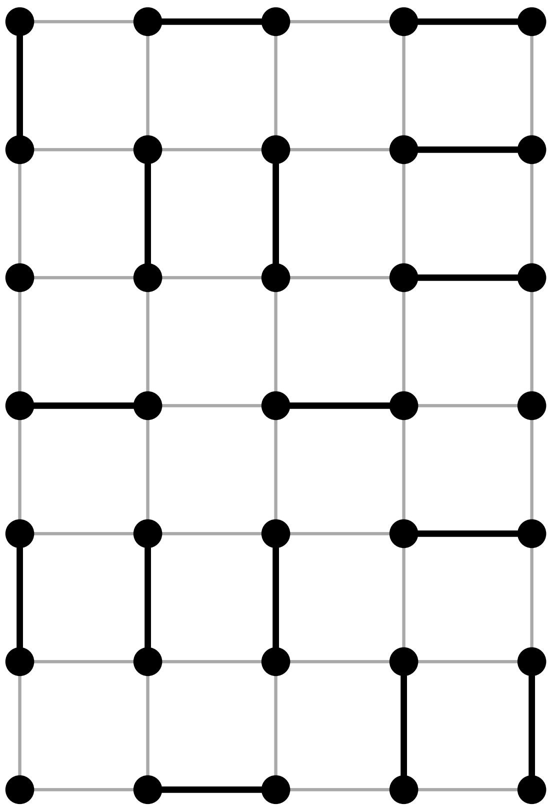
- 3-coloring: $r = 1$
- maximal independent set: $r = 1$



Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively



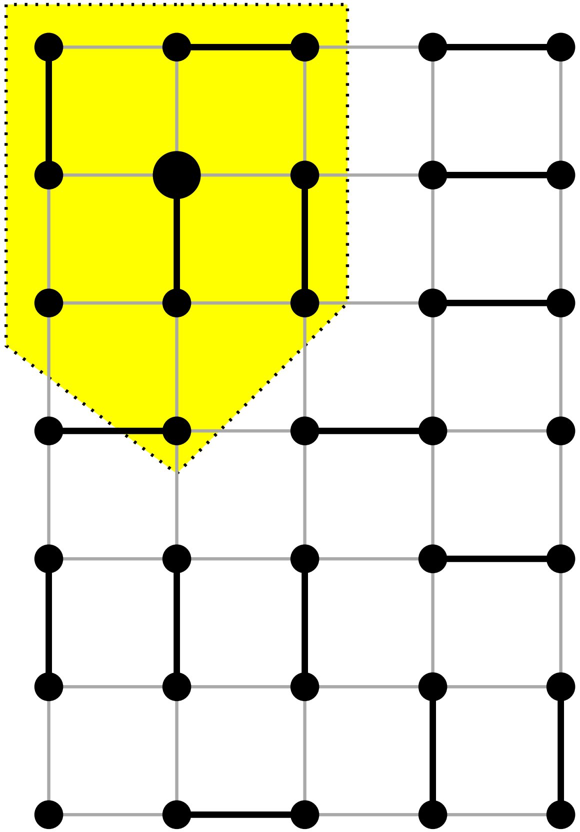
- 3-coloring: $r = 1$
- maximal independent set: $r = 1$
- maximal matching: $r = 2$

Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

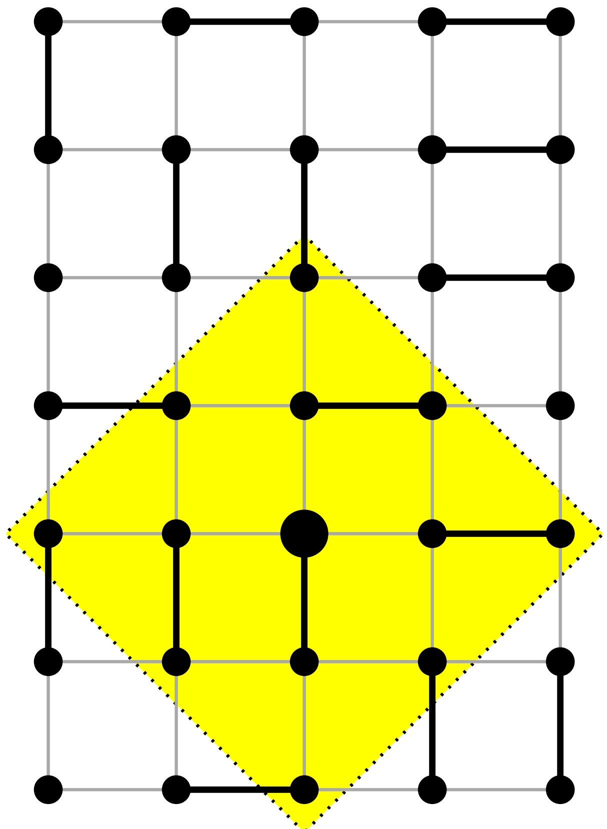
- 3-coloring: $r = 1$
- maximal independent set: $r = 1$
- maximal matching: $r = 2$



Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively



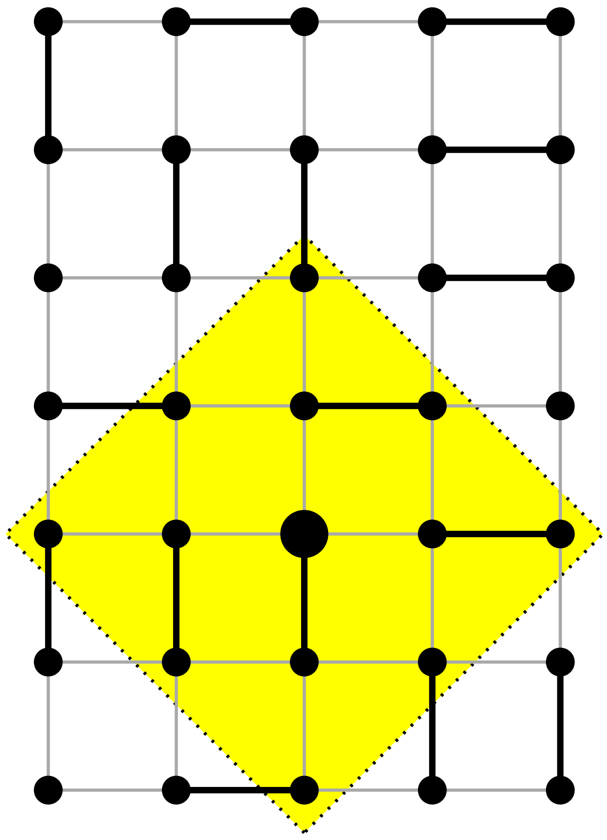
- 3-coloring: $r = 1$
- maximal independent set: $r = 1$
- maximal matching: $r = 2$

Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
  - Σ_{in}, Σ_{out}: finite input and output label sets
  - family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
  - input: labelled graph of degree Δ
  - output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$
- maximal independent set: $r = 1$
- maximal matching: $r = 2$
- sinkless orientation: $r = 1$

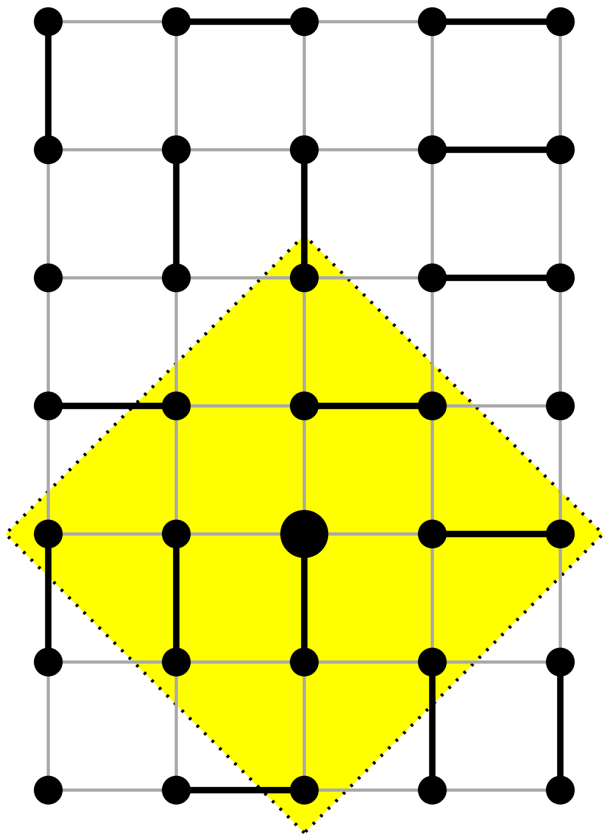


Locally checkable labelling (LCL) problems

[Naor and Stockmeyer, STOC '93]

- problems that can be locally checked
- Σ_{in}, Σ_{out}: finite input and output label sets
- family $\mathcal{F}$ of **admissible output neighborhoods** of radius $r = \Theta(1)$
- input: labelled graph of degree Δ
- output: labelled graph that is valid iff each node verifies positively

- 3-coloring: $r = 1$
- maximal independent set: $r = 1$
- maximal matching: $r = 2$
- sinkless orientation: $r = 1$
- minimal dominating set: $r = 1$
- many others...



Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families
- paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families
  - paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

  - balanced d-dimensional toroidal grids

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families

- paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

- balanced d-dimensional toroidal grids

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$

- bounded-degree trees

<table>
<tr><td>det-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td colspan="2">$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
<tr><td>rand-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td>$\Theta(\log\log n)$</td><td>$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
</table>

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families
- paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

- balanced d-dimensional toroidal grids

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$

- bounded-degree trees

<table>
<tr><td>det-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td colspan="2">$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
<tr><td>rand-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td>$\Theta(\log \log n)$</td><td>$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
</table>

- general graphs: much more complex

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families
- paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

- balanced d-dimensional toroidal grids

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$

- bounded-degree trees

<table>
<tr><td>det-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td colspan="2">$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
<tr><td>rand-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td>$\Theta(\log\log n)$</td><td>$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
</table>

- general graphs: much more complex
  - randomness does not help up to $\Theta(\log^\star n)$ and for $\Theta(n)$

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families
- paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

- balanced d-dimensional toroidal grids

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$

- bounded-degree trees

<table>
<tr><td>det-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td colspan="2">$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
<tr><td>rand-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td>$\Theta(\log\log n)$</td><td>$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
</table>

- general graphs: much more complex
  - randomness does not help up to $\Theta(\log^\star n)$ and for $\Theta(n)$
  - gap $\Theta(\log^\star n)$-$\Theta(\log n)$ ($\Theta(\log\log n)$) for det- (rand-) LOCAL

Complexity of LCLs

- LCLs' complexity has been **characterized** in many graph families
- paths & cycles

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n)$

- balanced d-dimensional toroidal grids

det-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$
rand-LOCAL	$\mathcal{O}(1)$	$\Theta(\log^\star n)$	$\Theta(n^{1/d})$

- bounded-degree trees

<table>
<tr><td>det-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td colspan="2">$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
<tr><td>rand-LOCAL</td><td>$\mathcal{O}(1)$</td><td>$\Theta(\log^\star n)$</td><td>$\Theta(\log\log n)$</td><td>$\Theta(\log n)$</td><td>$\Theta(n^{1/k})$</td><td>$\Theta(n)$</td></tr>
</table>

- general graphs: much more complex
  - randomness does not help up to $\Theta(\log^\star n)$ and for $\Theta(n)$
  - gap $\Theta(\log^\star n)$-$\Theta(\log n)$ ($\Theta(\log\log n)$) for det- (rand-) LOCAL
  - in general, randomness helps both exponentially and polynomially

Some lower bound techniques

- **Round elimination**

- input: LCL problem of complexity T
- output 1: LCL problem of complexity $T - 1$

Some lower bound techniques

- **Round elimination**

- input: LCL problem of complexity T
- output 1: LCL problem of complexity $T - 1$

$\vdots$

- output T: LCL problem of complexity 0

Some lower bound techniques

- **Round elimination**

- input: LCL problem of complexity T
- output 1: LCL problem of complexity $T - 1$

  $\vdots$

- output T: LCL problem of complexity 0

If output T is not 0-round solvable, then lower bound T for the input

Some lower bound techniques

- **Round elimination**

- input: LCL problem of complexity T
- output 1: LCL problem of complexity $T - 1$

$\vdots$

- output T: LCL problem of complexity 0

If output T is not 0-round solvable, then lower bound T for the input

- **Indistinguishability argument**

- local algorithms cannot recognize "global" modifications
- "fool" the algorithm

Some lower bound techniques

- **Round elimination**

- input: LCL problem of complexity T
- output 1: LCL problem of complexity $T - 1$

$\vdots$

- output T: LCL problem of complexity 0

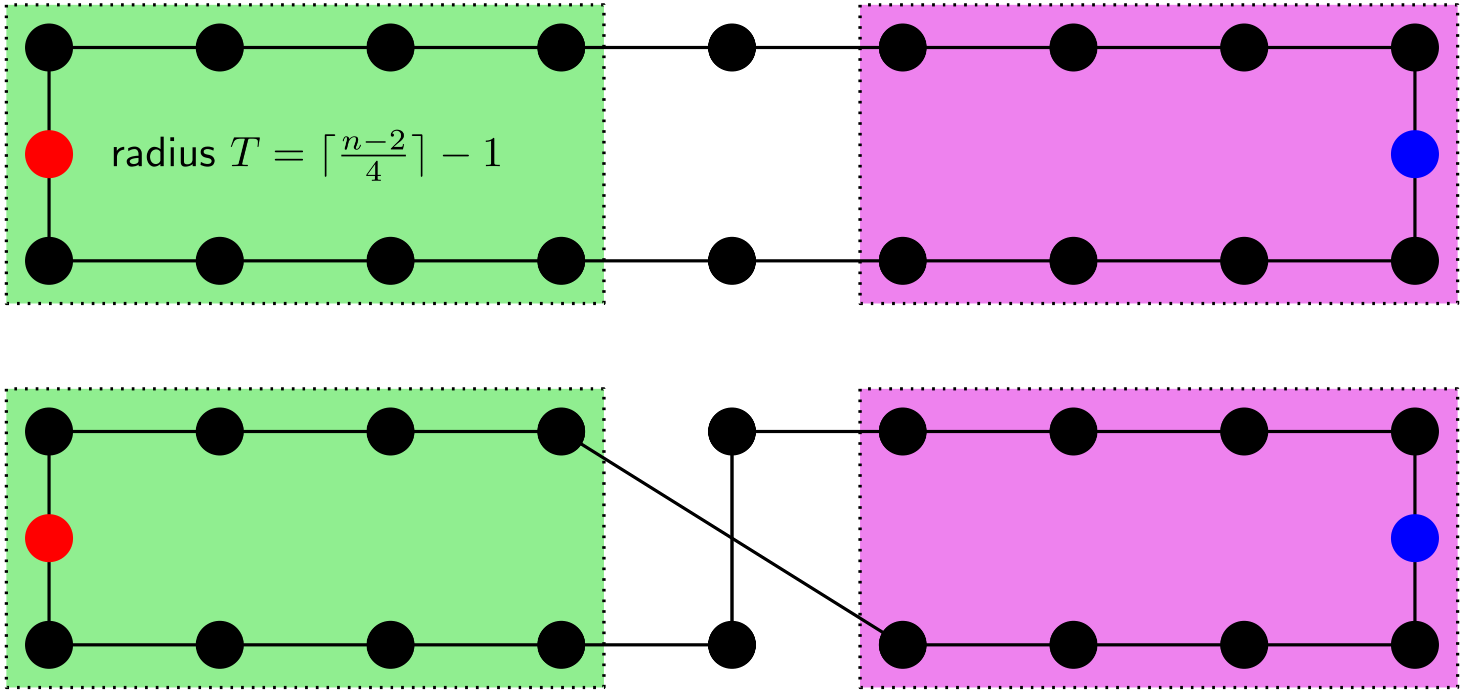
If output T is not 0-round solvable, then lower bound T for the input

- **Indistinguishability argument**

- local algorithms cannot recognize "global" modifications
- "fool" the algorithm

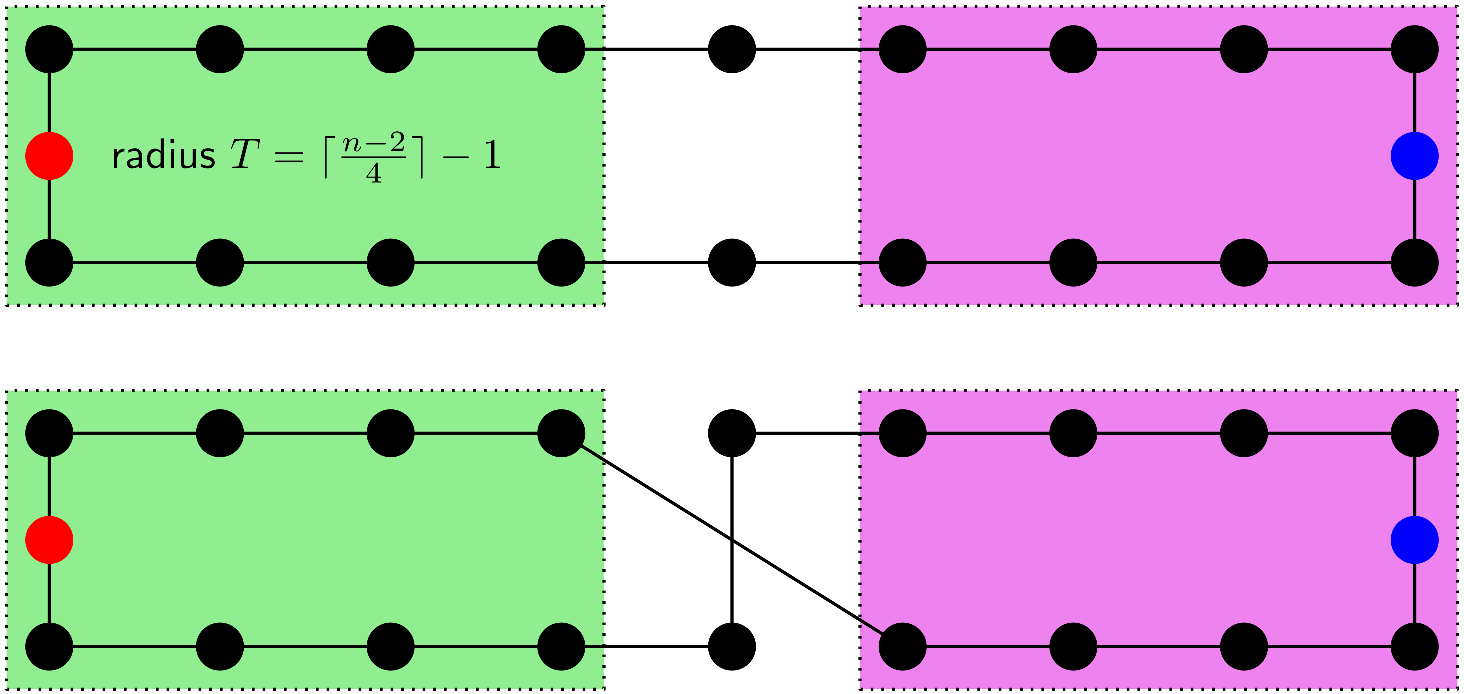
Indistinguishability argument

- Example: 2-coloring cycles requires $\Theta(n)$



Indistinguishability argument

- Example: 2-coloring cycles requires $\Theta(n)$



- red nodes must output the same
- blue nodes must output the same
- more complex arguments show $T \geq \lfloor \frac{n}{2} \rfloor - 1$

Quantum advantage

- Natural questions

- can **quantum communication** bring advantage?

Quantum advantage

- Natural questions
- can **quantum communication** bring advantage?
- can **pre-shared quantum states** bring advantage?

Quantum advantage

- Natural questions
- can **quantum communication** bring advantage?
- can **pre-shared quantum states** bring advantage?

- For non-LCLs, **yes**
- complexity $\Theta(n)$ in LOCAL with shared randomness and $\Theta(1)$ in q-LOCAL with pre-shared quantum state [Gavoille et al., DISC '09]

Quantum advantage

- Natural questions
- can **quantum communication** bring advantage?
- can **pre-shared quantum states** bring advantage?

- For non-LCLs, **yes**
- complexity $\Theta(n)$ in LOCAL with shared randomness and $\Theta(1)$ in q-LOCAL with pre-shared quantum state [Gavoille et al., DISC '09]
- complexity $\Theta(n)$ in rand-LOCAL and $\Theta(1)$ in q-LOCAL [Le Gall et al., STACS '19]

Quantum advantage

- Natural questions
- can **quantum communication** bring advantage?
- can **pre-shared quantum states** bring advantage?

- For non-LCLs, **yes**
- complexity $\Theta(n)$ in LOCAL with shared randomness and $\Theta(1)$ in q-LOCAL with pre-shared quantum state [Gavoille et al., DISC '09]
- complexity $\Theta(n)$ in rand-LOCAL and $\Theta(1)$ in q-LOCAL [Le Gall et al., STACS '19]

- For LCLs, **unknown**
- how to prove lower bounds for q-LOCAL?

Quantum advantage

- Natural questions
- can **quantum communication** bring advantage?
- can **pre-shared quantum states** bring advantage?

- For non-LCLs, **yes**
- complexity $\Theta(n)$ in LOCAL with shared randomness and $\Theta(1)$ in q-LOCAL with pre-shared quantum state [Gavoille et al., DISC '09]
- complexity $\Theta(n)$ in rand-LOCAL and $\Theta(1)$ in q-LOCAL [Le Gall et al., STACS '19]

- For LCLs, **unknown**
- how to prove lower bounds for q-LOCAL?
- some indistinguishability arguments works! [Gavoille et al., DISC '09]

The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*
- In distributed systems?

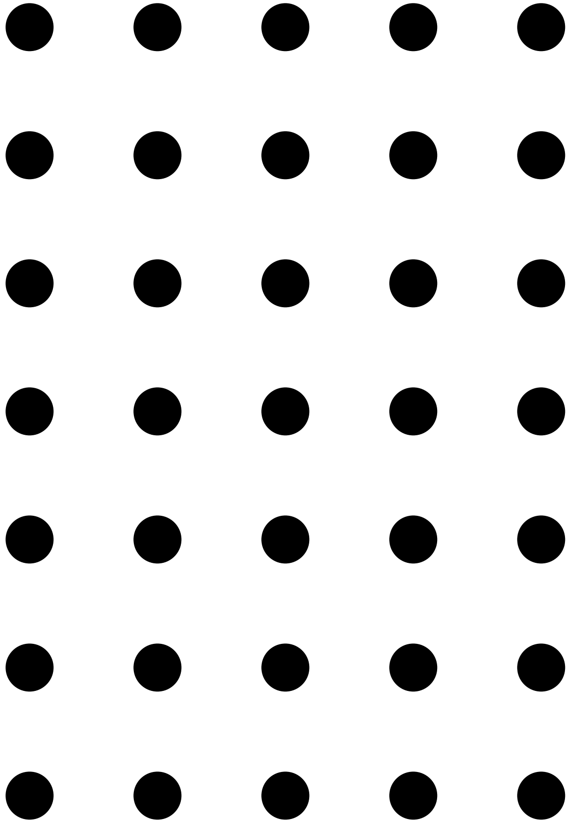
The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

- In distributed systems?

Some preliminary remarks ...

- Fix a set V of n processors



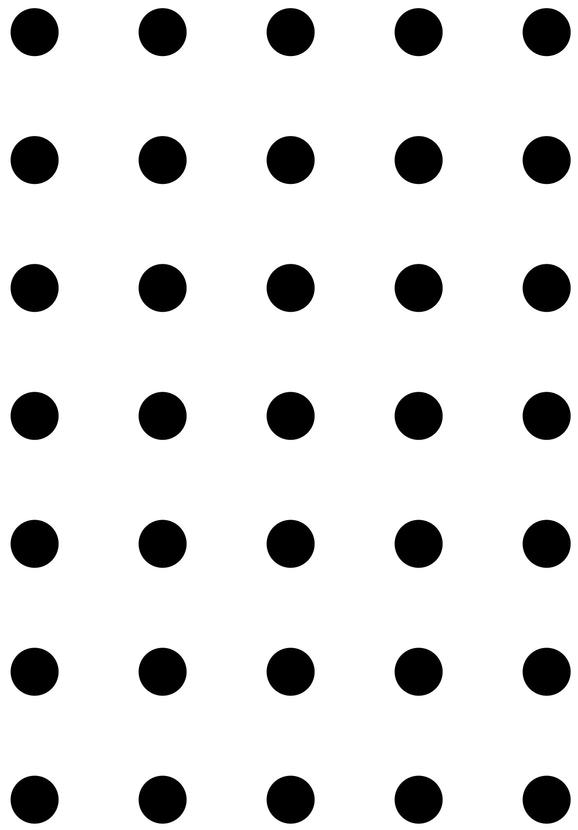
The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

- In distributed systems?

Some preliminary remarks . . .

- Fix a set V of n processors
- fix *any* algorithm $\mathcal{A}$



The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

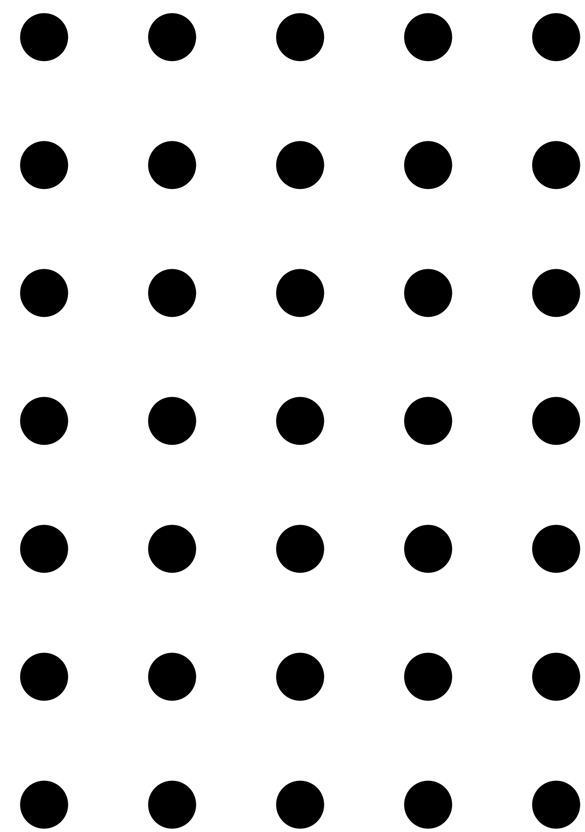
- In distributed systems?

Some preliminary remarks . . .

- Fix a set V of n processors

- fix *any* algorithm $\mathcal{A}$

- freedom on how to set links (the *graph*)



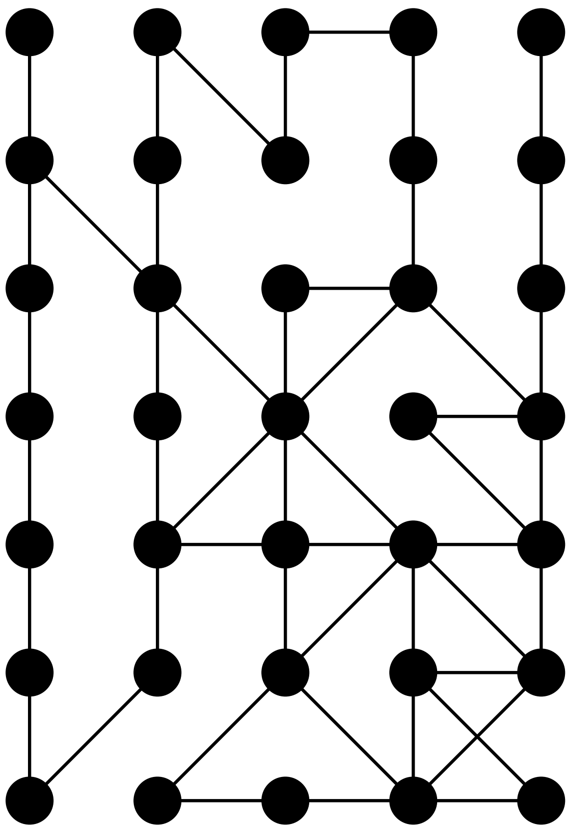
The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

- In distributed systems?

Some preliminary remarks . . .

- Fix a set V of n processors
- fix *any* algorithm $\mathcal{A}$
- freedom on how to set links (the *graph*)



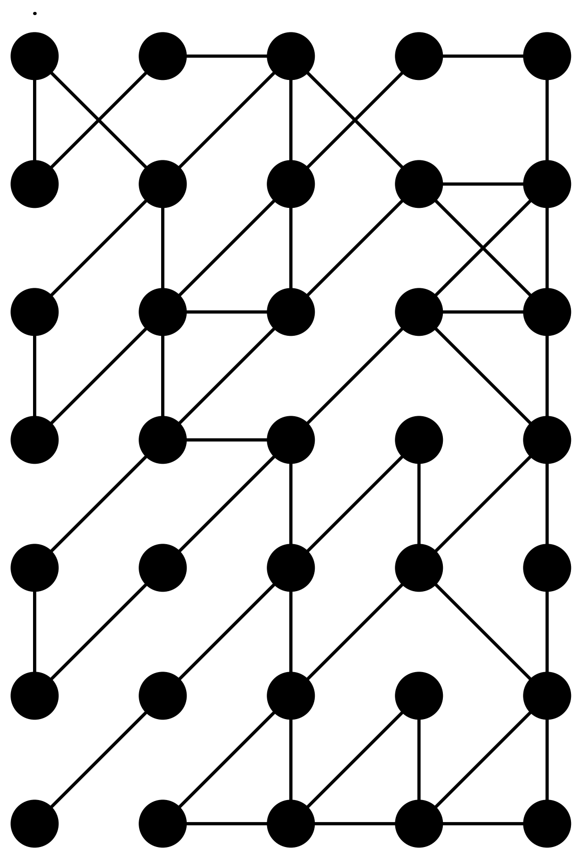
The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

- In distributed systems?

Some preliminary remarks . . .

- Fix a set V of n processors
- fix *any* algorithm $\mathcal{A}$
- freedom on how to set links (the *graph*)



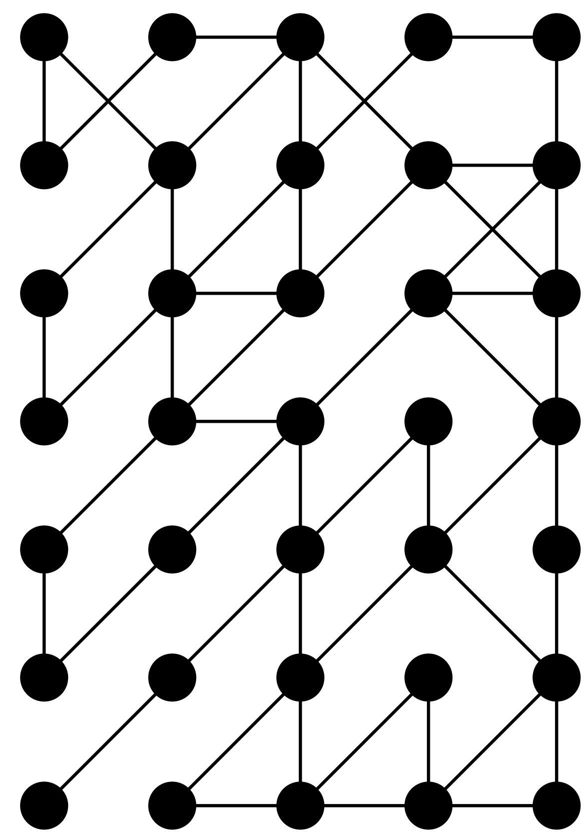
The no-signaling principle

- Principle (informal): *no signal can be sent from the future to the past*

- In distributed systems?

Some preliminary remarks . . .

- Fix a set V of n processors
- fix *any* algorithm $\mathcal{A}$
- freedom on how to set links (the *graph*)
- the algorithm always outputs something



The no-signaling principle

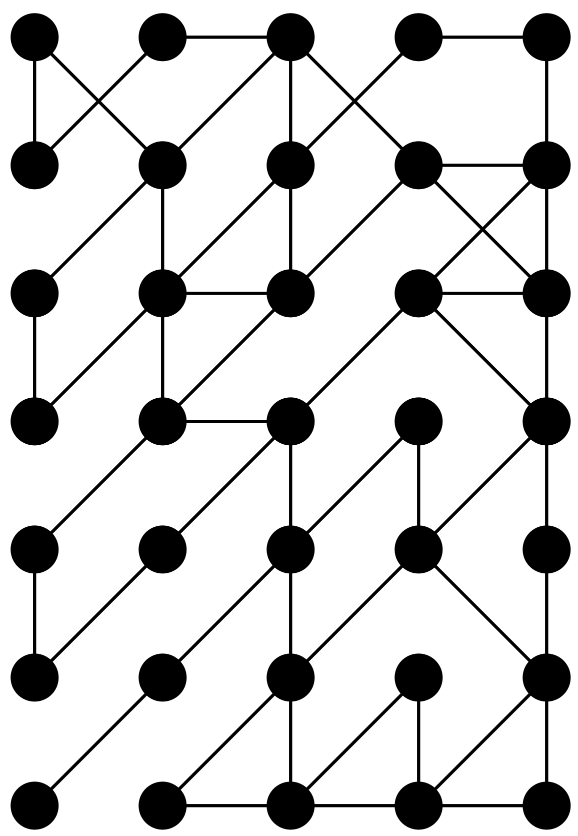
- Principle (informal): *no signal can be sent from the future to the past*

- In distributed systems?

Some preliminary remarks . . .

- Fix a set V of n processors
- fix *any* algorithm $\mathcal{A}$
- freedom on how to set links (the *graph*)
- the algorithm always outputs something

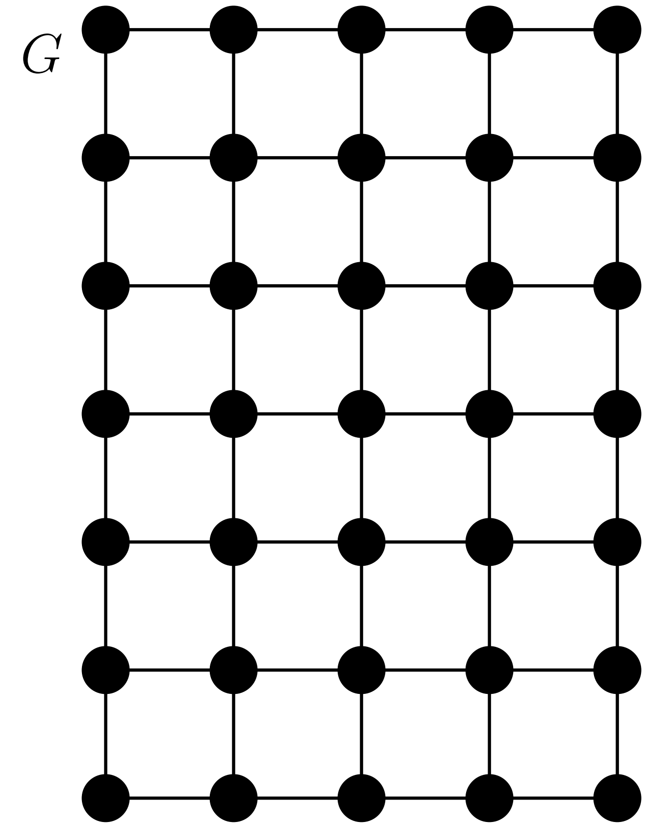
- Wlog, we assume algorithms are defined over **all graphs** of n nodes



The no-signaling principle

- In distributed systems?

- Fix a **distributed system** with $|V| = n$ processors

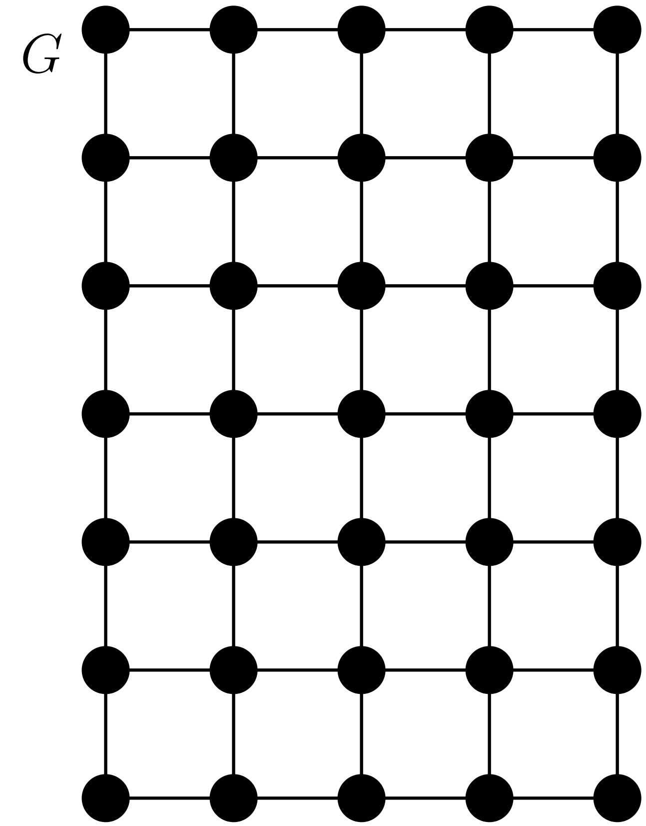


The no-signaling principle

- In distributed systems?

- Fix a **distributed system** with $|V| = n$ processors

- run algorithm $\mathcal{A}$ for T rounds

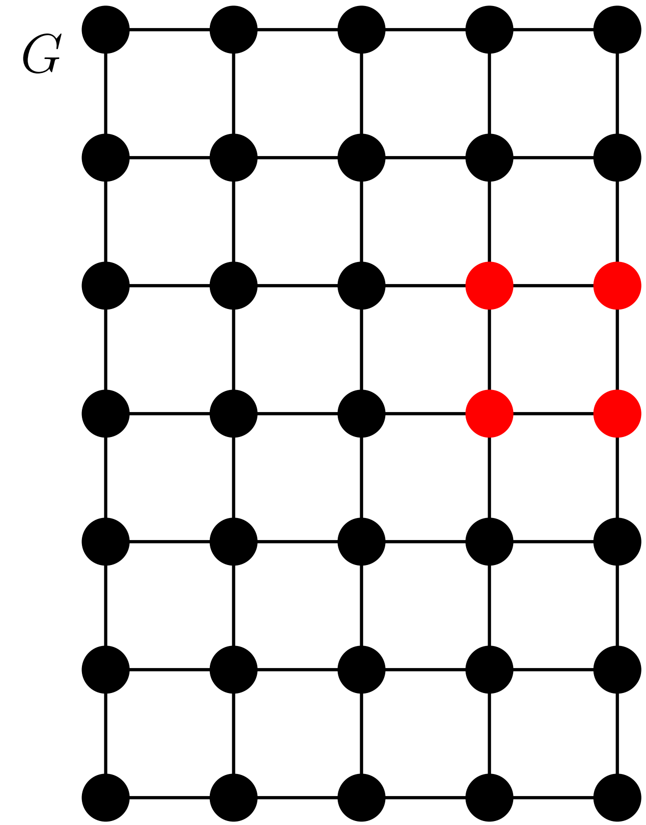


The no-signaling principle

- In distributed systems?

- Fix a **distributed system** with $|V| = n$ processors

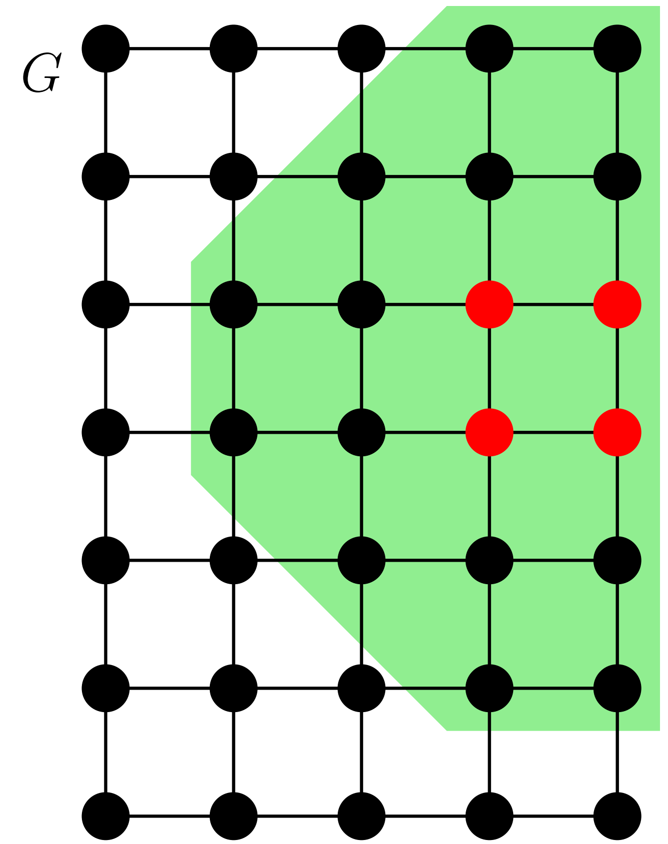
- run algorithm $\mathcal{A}$ for T rounds
- fix $S \subseteq V$



The no-signaling principle

- In distributed systems?

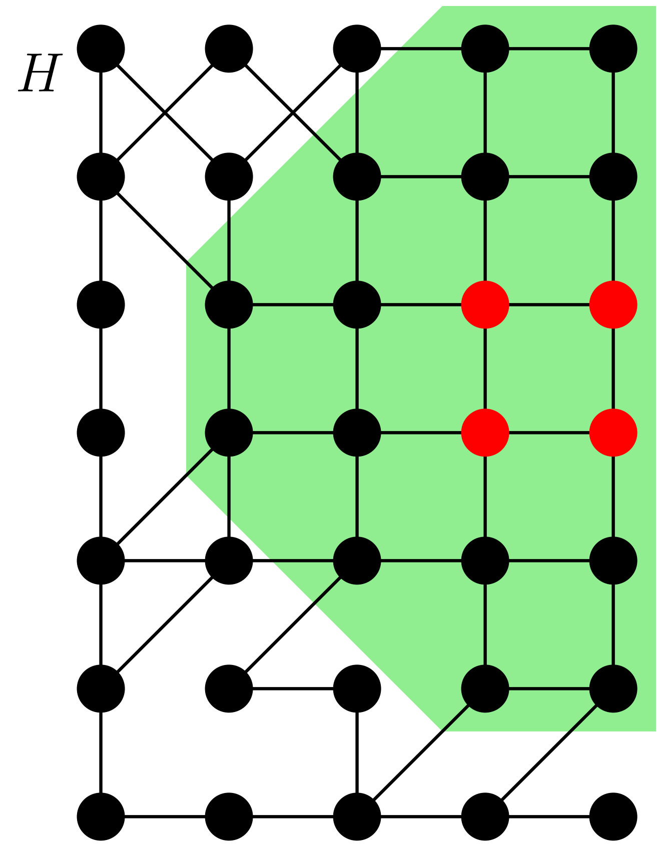
- Fix a **distributed system** with $|V| = n$ processors
- run algorithm $\mathcal{A}$ for T rounds
- fix $S \subseteq V$
- consider $G[\mathcal{N}_T(S)]$ $(T = 2)$



The no-signaling principle

- In distributed systems?

- Fix a **distributed system** with $|V| = n$ processors
- run algorithm $\mathcal{A}$ for T rounds
- fix $S \subseteq V$
- consider $G[\mathcal{N}_T(S)]$ $(T = 2)$
- modify G outside $G[\mathcal{N}_T(S)]$

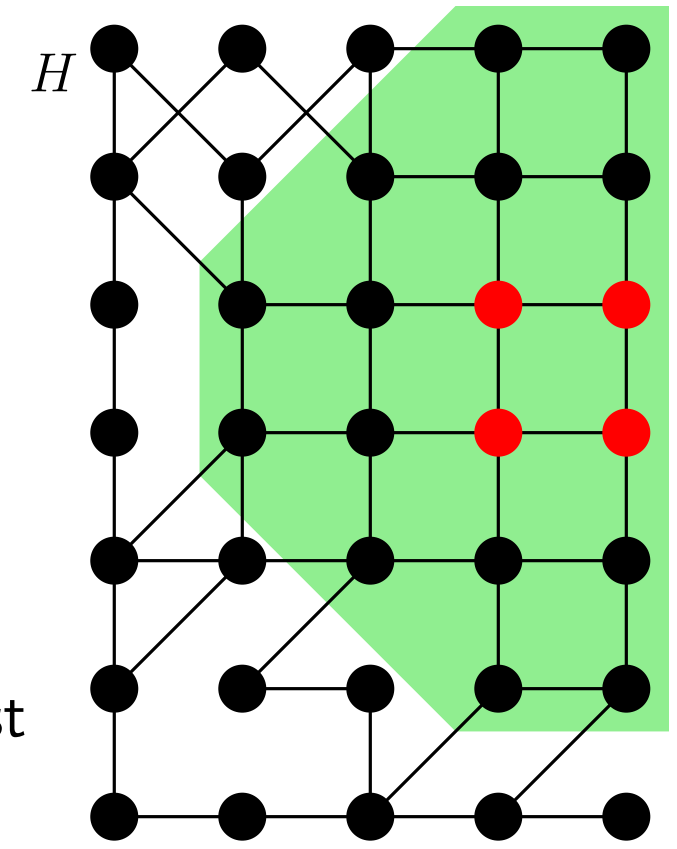


The no-signaling principle

- In distributed systems?

- Fix a **distributed system** with $|V| = n$ processors

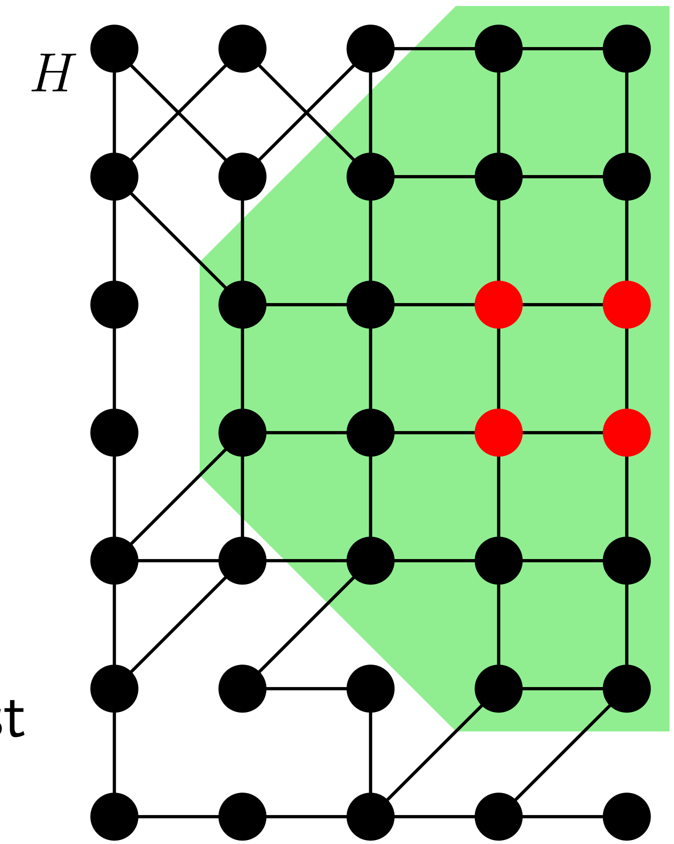
- run algorithm $\mathcal{A}$ for T rounds
- fix $S \subseteq V$
- consider $G[\mathcal{N}_T(S)]$ $(T = 2)$
- modify G outside $G[\mathcal{N}_T(S)]$
- the output distribution over $G[\mathcal{N}_T(S)]$ must be the same **no matter the structure outside**



The no-signaling principle

- In distributed systems?

- Fix a **distributed system** with $|V| = n$ processors
- run algorithm $\mathcal{A}$ for T rounds
- fix $S \subseteq V$
- consider $G[\mathcal{N}_T(S)]$ ($T = 2$)
- modify G outside $G[\mathcal{N}_T(S)]$
- the output distribution over $G[\mathcal{N}_T(S)]$ must be the same **no matter the structure outside**



Outcome: "generalization" of algorithm

- An outcome assigns to inputs (G, x) a distribution over outputs $\{(y_i, p_i)\}_{i \in I}$, $y_i : V \to \Sigma_{\text{out}}$

Non-signaling outcome

- An outcome is **non-signaling beyond distance** T if for all $S \subseteq V$, for all inputs $(G, x), (H, y)$ on V with $G[\mathcal{N}_T(S)] \sim H[\mathcal{N}_T(S)]$, the distributions restricted to $G[S]$ and $H[S]$ are the same

Non-signaling outcome

- An outcome is **non-signaling beyond distance** T if for all $S \subseteq V$, for all inputs $(G, x), (H, y)$ on V with $G[\mathcal{N}_T(S)] \sim H[\mathcal{N}_T(S)]$, the distributions restricted to $G[S]$ and $H[S]$ are the same

Example: 2-coloring cycles

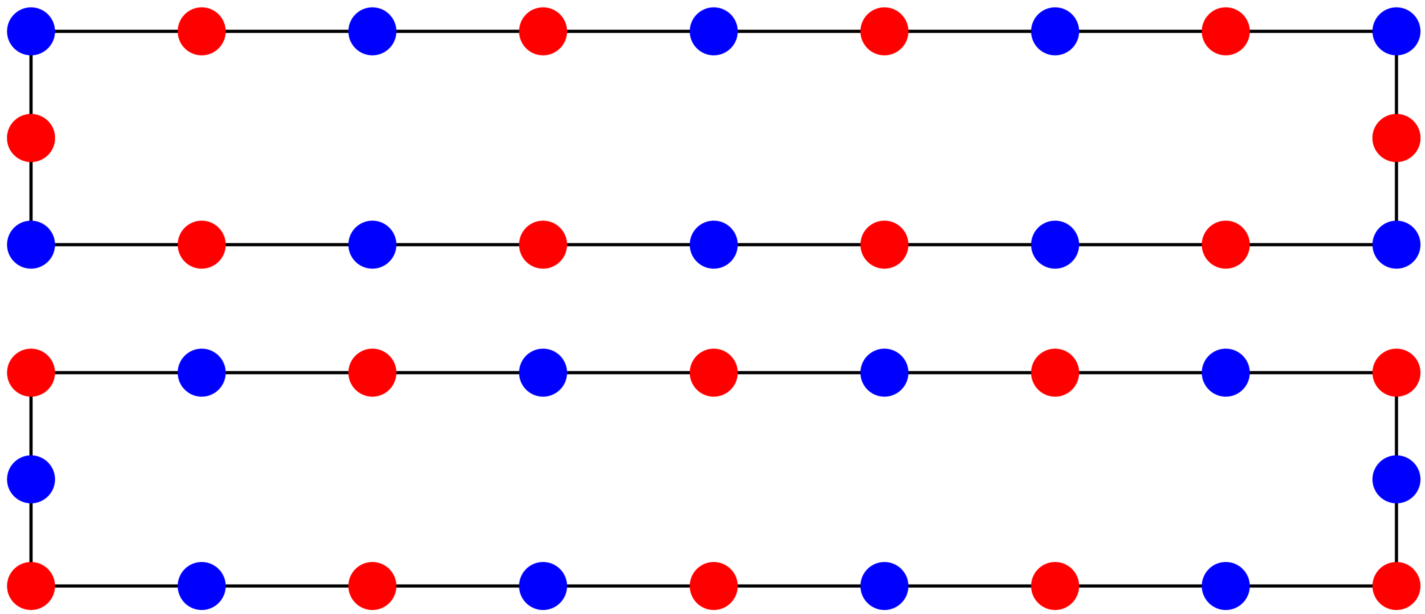
- Outcome: sample uniformly at random a consistent 2-coloring

Non-signaling outcome

- An outcome is **non-signaling beyond distance** T if for all $S \subseteq V$, for all inputs $(G, x), (H, y)$ on V with $G[\mathcal{N}_T(S)] \sim H[\mathcal{N}_T(S)]$, the distributions restricted to $G[S]$ and $H[S]$ are the same

Example: 2-coloring cycles

- Outcome: sample uniformly at random a consistent 2-coloring

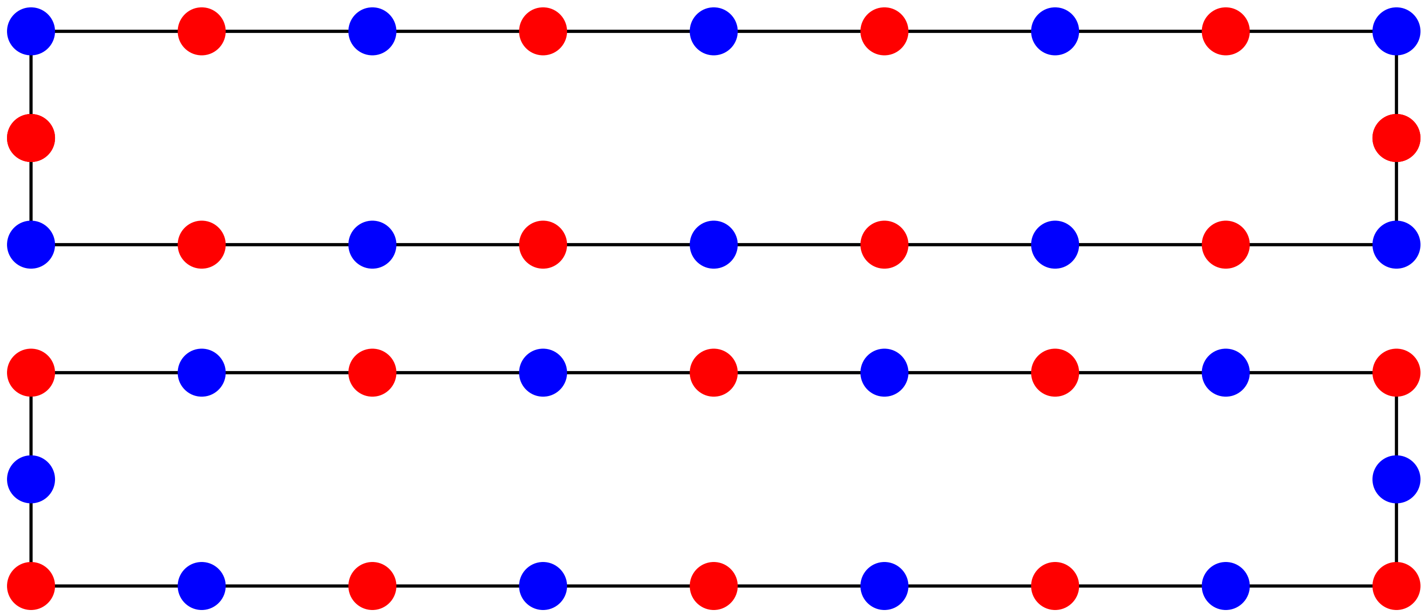


Non-signaling outcome

- An outcome is **non-signaling beyond distance** T if for all $S \subseteq V$, for all inputs $(G, x), (H, y)$ on V with $G[\mathcal{N}_T(S)] \sim H[\mathcal{N}_T(S)]$, the distributions restricted to $G[S]$ and $H[S]$ are the same

Example: 2-coloring cycles

- Outcome: sample uniformly at random a consistent 2-coloring



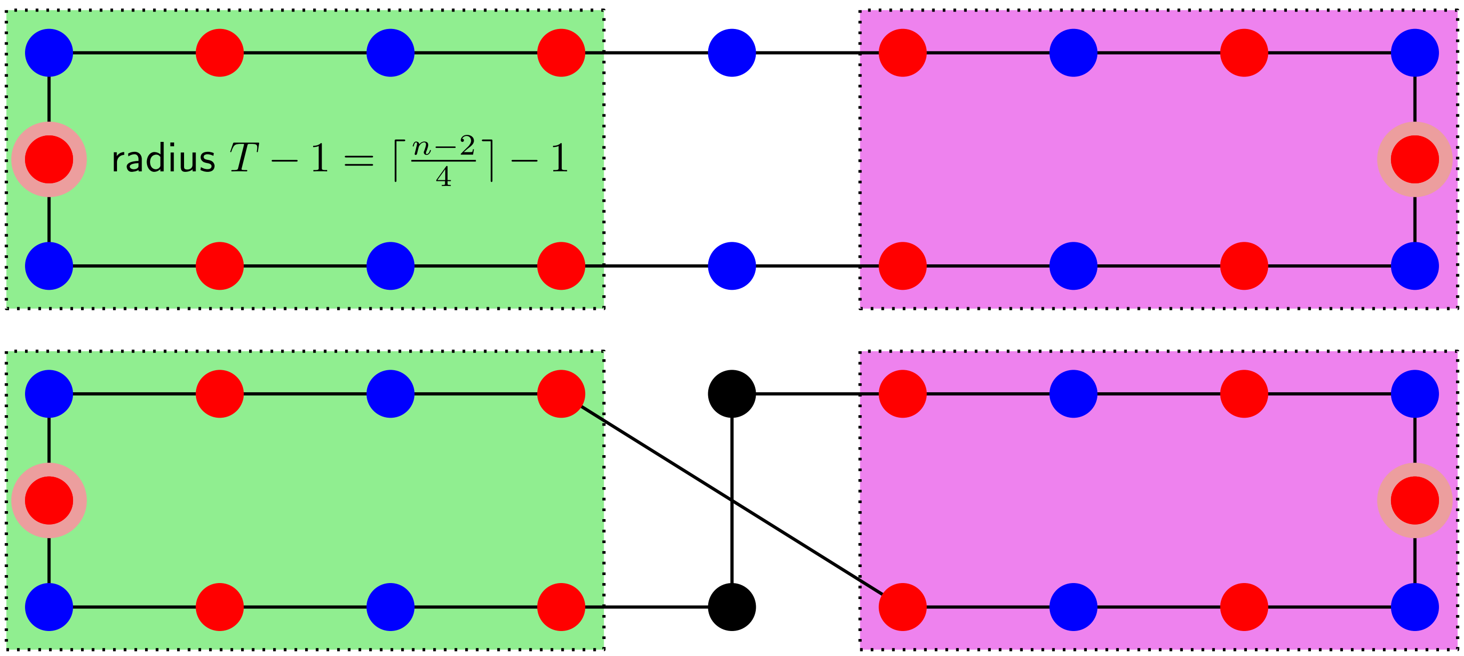
- non-signaling beyond distance $T = \lceil \frac{n-2}{4} \rceil$

Non-signaling outcome

- An outcome is **non-signaling beyond distance** T if for all $S \subseteq V$, for all inputs $(G, x), (H, y)$ on V with $G[\mathcal{N}_T(S)] \sim H[\mathcal{N}_T(S)]$, the distributions restricted to $G[S]$ and $H[S]$ are the same

Example: 2-coloring cycles

- Outcome: sample uniformly at random a consistent 2-coloring



- non-signaling beyond distance $T = \lceil \frac{n-2}{4} \rceil$

The φ-LOCAL and non-signaling LOCAL models

[Gavoille et al., DISC '09]

- The φ-**LOCAL** model is a computational model that produces non-signaling outcomes
- class $\Phi[T]$ consists of problems solvable by outcomes that are non-signaling beyond distance T

The φ-LOCAL and non-signaling LOCAL models

[Gavoille et al., DISC '09]

- The φ-**LOCAL** model is a computational model that produces non-signaling outcomes

\- class $\Phi[T]$ consists of problems solvable by outcomes that are non-signaling beyond distance T

[Arfaoui and Fraigniaud, PODC '12]

- The **non-signaling** (NS) **LOCAL** model is a computational model that produces non-signaling outcomes defined for every possible input graph

\- class $\mathcal{NS}[T]$ defined analogously

The φ-LOCAL and non-signaling LOCAL models

[Gavoille et al., DISC '09]

- The φ-**LOCAL** model is a computational model that produces non-signaling outcomes

- class $\Phi[T]$ consists of problems solvable by outcomes that are non-signaling beyond distance T

[Arfaoui and Fraigniaud, PODC '12]

- The **non-signaling** (NS) **LOCAL** model is a computational model that produces non-signaling outcomes defined for every possible input graph

- class $\mathcal{NS}[T]$ defined analogously

- $\mathcal{P} \in \mathcal{NS}[T] \implies \mathcal{P} \in \Phi[T]$

Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

- Ramanujan graphs of n nodes with the following properties
  - $2c^2$-regular
  - girth $\Omega(\log_c n)$
  - chromatic number $> c$

Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

- Ramanujan graphs of n nodes with the following properties
  - $2c^2$-regular
  - girth $\Omega(\log_c n)$
  - chromatic number $> c$
- **By contradiction**: det-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$

Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

- Ramanujan graphs of n nodes with the following properties
  - $2c^2$-regular
  - girth $\Omega(\log_c n)$
  - chromatic number $> c$

- **By contradiction**: det-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
  - run $\mathcal{A}$ in the Ramanujan graph R

R

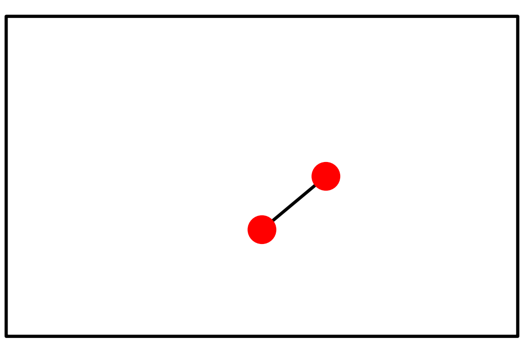
Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

- Ramanujan graphs of n nodes with the following properties
  - $2c^2$-regular
  - girth $\Omega(\log_c n)$
  - chromatic number $> c$

- **By contradiction**: det-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere



R

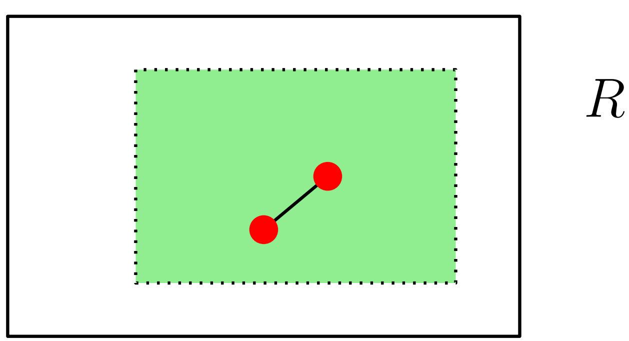
Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

- Ramanujan graphs of n nodes with the following properties
  - $2c^2$-regular
  - girth $\Omega(\log_c n)$
  - chromatic number $> c$

- **By contradiction**: det-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere
- T-radius neighborhood around the failing region



Indistinguishability & graph-theory: det-LOCAL

- [Linial, FOCS '87] combines the indistinguishability argument with some graph-theoretic results

Problem: c-coloring trees: $T = \Omega(\log_c n)$

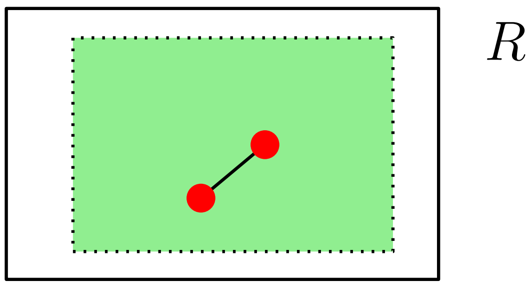
- Ramanujan graphs of n nodes with the following properties
  - $2c^2$-regular
  - girth $\Omega(\log_c n)$
  - chromatic number $> c$

- **By contradiction**: det-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere
- T-radius neighborhood around the failing region
- it's a c-regular tree (completable to n nodes)

Indistinguishability & graph-theory: rand-LOCAL

- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$

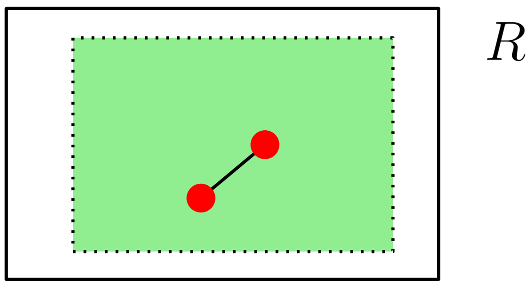
- run $\mathcal{A}$ in the Ramanujan graph R



Indistinguishability & graph-theory: rand-LOCAL

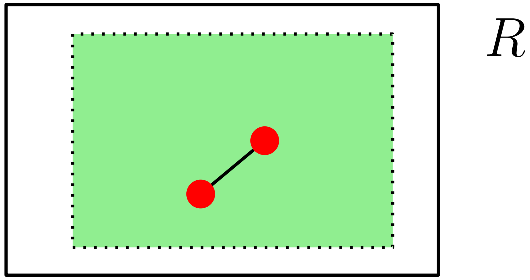
- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$

- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere, but then?



Indistinguishability & graph-theory: rand-LOCAL

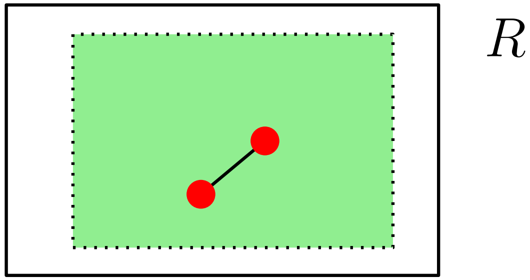
- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere, but then?



- Suppose R can admits a family of subgraphs $\{R_i\}_{i=1}^k$ such that
- $\cup_{i=1}^k R_i = R$

Indistinguishability & graph-theory: rand-LOCAL

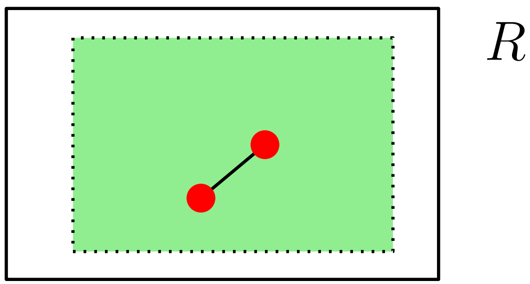
- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere, but then?



- Suppose R can admits a family of subgraphs $\{R_i\}_{i=1}^k$ such that
- $\cup_{i=1}^k R_i = R$
- $\implies \exists i^\star \in [k]$ such that $\mathcal{A}$ fails in R_i with probability at least $1/k$

Indistinguishability & graph-theory: rand-LOCAL

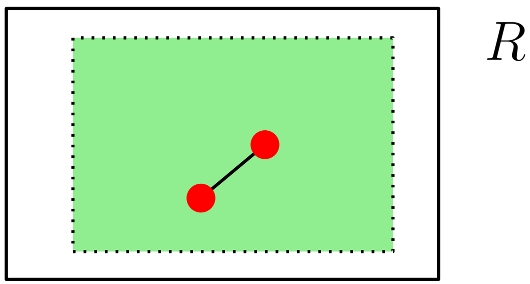
- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere, but then?



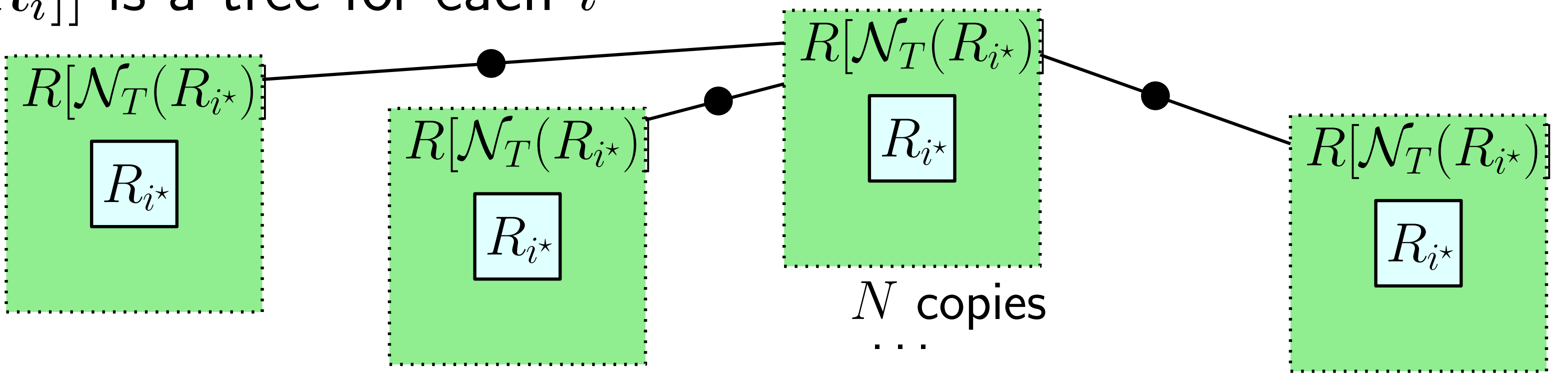
- Suppose R can admits a family of subgraphs $\{R_i\}_{i=1}^k$ such that
- $\cup_{i=1}^k R_i = R$
- $\implies \exists i^\star \in [k]$ such that $\mathcal{A}$ fails in R_i with probability at least $1/k$
- $R[\mathcal{N}_T[R_i]]$ is a tree for each i

Indistinguishability & graph-theory: rand-LOCAL

- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere, but then?

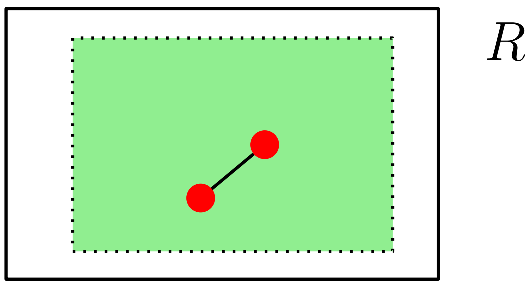


- Suppose R can admits a family of subgraphs $\{R_i\}_{i=1}^k$ such that
- $\cup_{i=1}^k R_i = R$
- $\implies \exists i^\star \in [k]$ such that $\mathcal{A}$ fails in R_i with probability at least $1/k$
- $R[\mathcal{N}_T[R_i]]$ is a tree for each i

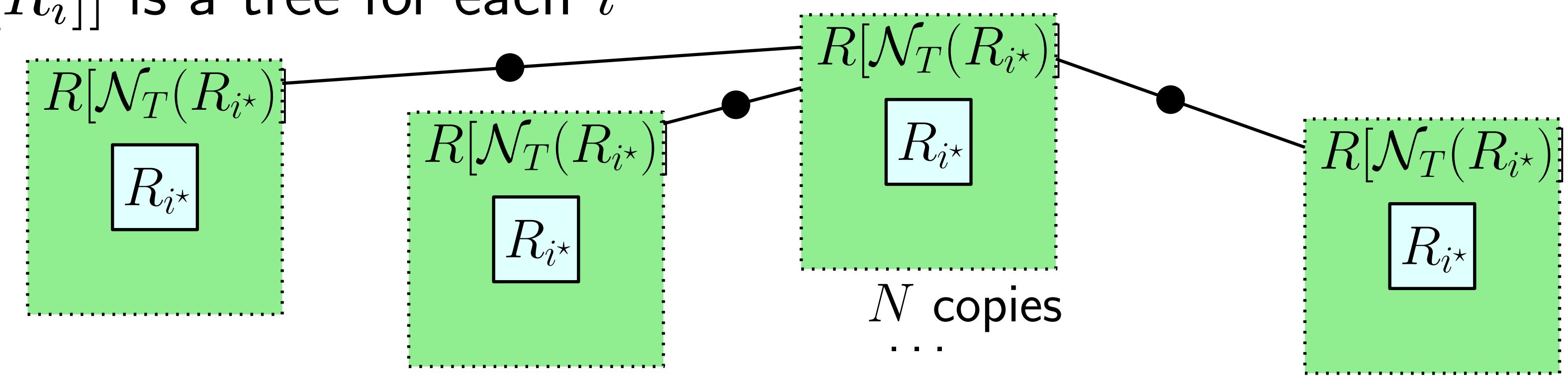


Indistinguishability & graph-theory: rand-LOCAL

- **By contradiction**: rand-LOCAL algorithm $\mathcal{A}$ with $T = \mathcal{O}(\log_c n)$
- run $\mathcal{A}$ in the Ramanujan graph R
- $\mathcal{A}$ fails somewhere, but then?



- Suppose R can admits a family of subgraphs $\{R_i\}_{i=1}^k$ such that
- $\cup_{i=1}^k R_i = R$
- $\implies \exists i^\star \in [k]$ such that $\mathcal{A}$ fails in R_i with probability at least $1/k$
- $R[\mathcal{N}_T[R_i]]$ is a tree for each i



- Failing prob.: $1 - (1 - \frac{1}{k})^N$
- lower bound $T_{\text{new}}(n) = T(n/N)$

Indistinguishability & graph-theory: NS-LOCAL

Issues

- Possible dependencies: output distributions for two subsets of nodes X and Y, then even if X and Y are far from each other, the outputs of these subsets might be dependent

Indistinguishability & graph-theory: NS-LOCAL

Issues

- Possible dependencies: output distributions for two subsets of nodes X and Y, then even if X and Y are far from each other, the outputs of these subsets might be dependent
- No-cloning principle: if two graphs G and H have different sizes, then even if they share two identical subgraphs G' and H' with isomorphic radius-T neighborhoods, the output distribution is not guaranteed to be identical over G' and H'

Indistinguishability & graph-theory: NS-LOCAL

Issues

- Possible dependencies: output distributions for two subsets of nodes X and Y, then even if X and Y are far from each other, the outputs of these subsets might be dependent
- No-cloning principle: if two graphs G and H have different sizes, then even if they share two identical subgraphs G' and H' with isomorphic radius-T neighborhoods, the output distribution is not guaranteed to be identical over G' and H'

- Can we do something similar?

Indistinguishability & graph-theory: NS-LOCAL

Issues

- Possible dependencies: output distributions for two subsets of nodes X and Y, then even if X and Y are far from each other, the outputs of these subsets might be dependent
- No-cloning principle: if two graphs G and H have different sizes, then even if they share two identical subgraphs G' and H' with isomorphic radius-T neighborhoods, the output distribution is not guaranteed to be identical over G' and H'

- Can we do something similar?
- We need to refine the definition of **cheating graph**

Cheating graph

Fix an LCL problem $\mathcal{P}$ over some graph family $\mathcal{F}$

Cheating graph:

Cheating graph

Fix an LCL problem $\mathcal{P}$ over some graph family $\mathcal{F}$

Cheating graph:

- For some n, there exist a value $T = T(n)$ and a graph G_n on n nodes with a family of subgraphs $\{G_n^{(i)}\}_{i=1}^k$ such that

Cheating graph

Fix an LCL problem $\mathcal{P}$ over some graph family $\mathcal{F}$

Cheating graph:

- For some n, there exist a value $T = T(n)$ and a graph G_n on n nodes with a family of subgraphs $\{G_n^{(i)}\}_{i=1}^{k}$ such that

- $\cup_{i=1}^{k} G_n^{(i)} = G_n$

Cheating graph

Fix an LCL problem $\mathcal{P}$ over some graph family $\mathcal{F}$

Cheating graph:

- For some n, there exist a value $T = T(n)$ and a graph G_n on n nodes with a family of subgraphs $\{G_n^{(i)}\}_{i=1}^{k}$ such that

- $\cup_{i=1}^{k} G_n^{(i)} = G_n$

- $\mathcal{P}$ is solvable on each $G_n[\mathcal{N}_T(G_n^{(i)})]$

Cheating graph

Fix an LCL problem $\mathcal{P}$ over some graph family $\mathcal{F}$

Cheating graph:

- For some n, there exist a value $T = T(n)$ and a graph G_n on n nodes with a family of subgraphs $\{G_n^{(i)}\}_{i=1}^k$ such that

- $\cup_{i=1}^k G_n^{(i)} = G_n$
- $\mathcal{P}$ is solvable on each $G_n[\mathcal{N}_T(G_n^{(i)})]$
- for some $N > 0$, and all $\mathbf{x} \in \{1, \ldots, k\}^N$, $\mathcal{F}$ contains a graph $H_{\mathbf{x}}$ on $n \cdot N$ nodes which admits $\sqcup_{j=1}^N G_n^{(\mathbf{x}_j)}$ as induced subgraph $H'_{\mathbf{x}}$ so that $H_{\mathbf{x}}[\mathcal{N}_T(H'_{\mathbf{x}})]$ and $\sqcup_{j=1}^N G_n[\mathcal{N}_T(G_n^{(\mathbf{x}_j)})]$ are isomorphic

Cheating graph

Fix an LCL problem $\mathcal{P}$ over some graph family $\mathcal{F}$

Cheating graph:

- For some n, there exist a value $T = T(n)$ and a graph G_n on n nodes with a family of subgraphs $\{G_n^{(i)}\}_{i=1}^k$ such that

- $\cup_{i=1}^k G_n^{(i)} = G_n$

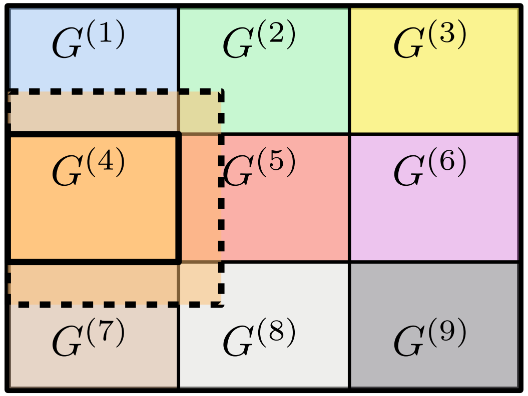
- $\mathcal{P}$ is solvable on each $G_n[\mathcal{N}_T(G_n^{(i)})]$

- for some $N > 0$, and all $\mathbf{x} \in \{1, \ldots, k\}^N$, $\mathcal{F}$ contains a graph $H_{\mathbf{x}}$ on $n \cdot N$ nodes which admits $\sqcup_{j=1}^N G_n^{(\mathbf{x}_j)}$ as induced subgraph $H'_{\mathbf{x}}$ so that $H_{\mathbf{x}}[\mathcal{N}_T(H'_{\mathbf{x}})]$ and $\sqcup_{j=1}^N G_n[\mathcal{N}_T(G_n^{(\mathbf{x}_j)})]$ are isomorphic

Theorem: failing probability $\geq 1 - (1 - \frac{1}{k})^N$ and $T_{\text{new}}(n) = T(\frac{n}{N})$

Cheating graph

Cheating graph G

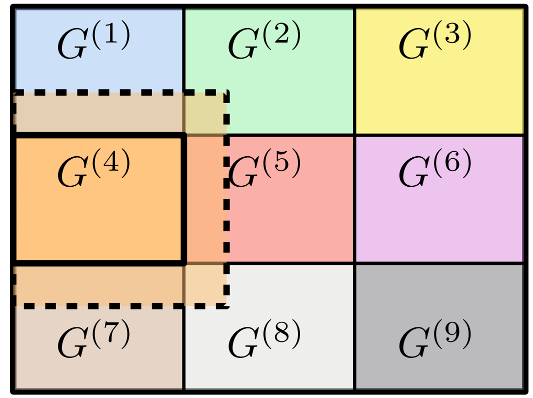


Locally, $\mathcal{P}$ solvable

Globally, $\mathcal{P}$ not solvable

Cheating graph

Cheating graph G

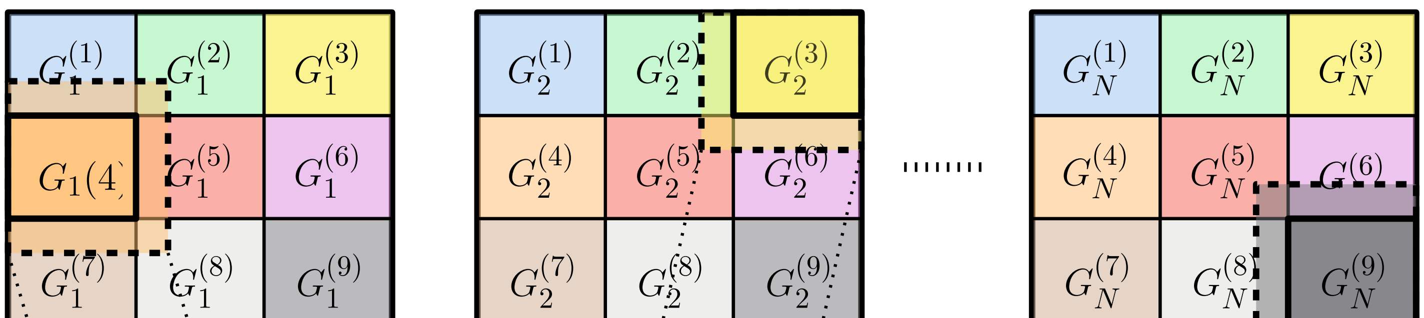


Locally, $\mathcal{P}$ solvable

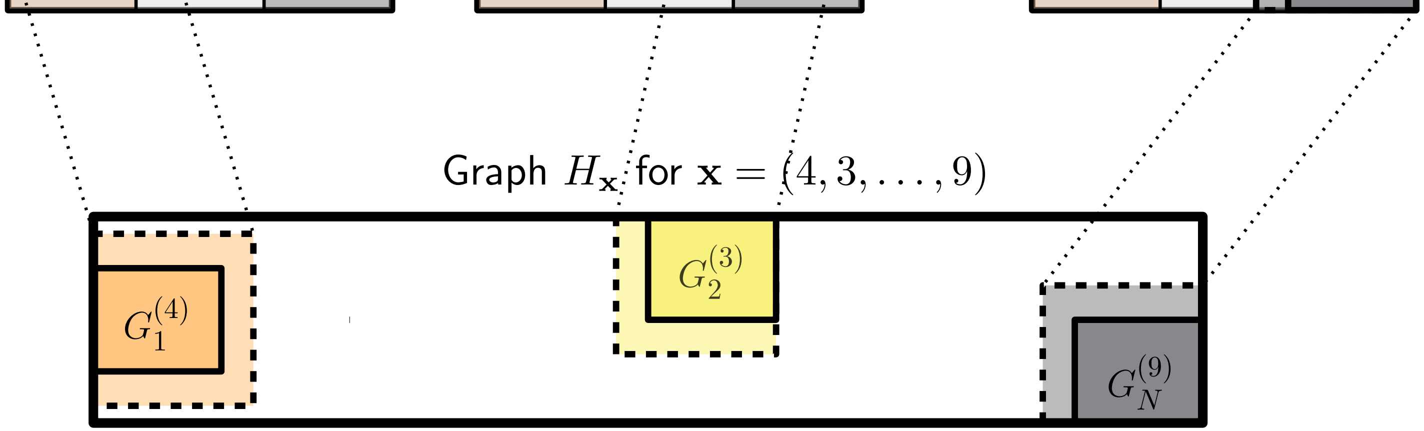
Globally, $\mathcal{P}$ not solvable

N copies of the cheating graph G

Cheating graph G_1 | Cheating graph G_2 | Cheating graph G_N



Graph $H_{\mathbf{x}}$ for $\mathbf{x} = (4, 3, \ldots, 9)$



Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

- Ramanujan graphs

Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

- Ramanujan graphs

- **Result 2**: 3-coloring $n \times n$ grids with success probability $\geq \varepsilon$ requires locality $T = \Omega(\frac{n}{\log \frac{1}{\varepsilon}})$

Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

\- Ramanujan graphs

- **Result 2**: 3-coloring $n \times n$ grids with success probability $\geq \varepsilon$ requires locality $T = \Omega(\frac{n}{\log \frac{1}{\varepsilon}})$

\- quadrangulations of the Klein bottle

Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

- Ramanujan graphs

- **Result 2**: 3-coloring $n \times n$ grids with success probability $\geq \varepsilon$ requires locality $T = \Omega(\frac{n}{\log \frac{1}{\varepsilon}})$

- quadrangulations of the Klein bottle

- **Result 3**: c-coloring χ-chromatic graphs with success probability $\geq \varepsilon$ requires locality $T = \Omega((\frac{n}{\log \frac{1}{\varepsilon}})^{1/\alpha})$ where $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$

Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

- Ramanujan graphs

- **Result 2**: 3-coloring $n \times n$ grids with success probability $\geq \varepsilon$ requires locality $T = \Omega(\frac{n}{\log \frac{1}{\varepsilon}})$

- quadrangulations of the Klein bottle

- **Result 3**: c-coloring χ-chromatic graphs with success probability $\geq \varepsilon$ requires locality $T = \Omega((\frac{n}{\log \frac{1}{\varepsilon}})^{1/\alpha})$ where $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$

- construction in [Bogdanov '13]

Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

- Ramanujan graphs

- **Result 2**: 3-coloring $n \times n$ grids with success probability $\geq \varepsilon$ requires locality $T = \Omega(\frac{n}{\log \frac{1}{\varepsilon}})$

- quadrangulations of the Klein bottle

- **Result 3**: c-coloring χ-chromatic graphs with success probability $\geq \varepsilon$ requires locality $T = \Omega((\frac{n}{\log \frac{1}{\varepsilon}})^{1/\alpha})$ where $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$

- construction in [Bogdanov '13]
- deterministic upper bound $T = \mathcal{O}(n^{\frac{1}{\alpha}} \text{polylog}(n))$

Some applications

The following holds in NS-LOCAL

- **Result 1**: c-coloring trees with success probability ε requires locality $T = \Omega(\log_c n - \log_c \log \frac{1}{\varepsilon})$

- Ramanujan graphs

- **Result 2**: 3-coloring $n \times n$ grids with success probability $\geq \varepsilon$ requires locality $T = \Omega(\frac{n}{\log \frac{1}{\varepsilon}})$

- quadrangulations of the Klein bottle

- **Result 3**: c-coloring χ-chromatic graphs with success probability $\geq \varepsilon$ requires locality $T = \Omega((\frac{n}{\log \frac{1}{\varepsilon}})^{1/\alpha})$ where $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$

- construction in [Bogdanov '13]
- deterministic upper bound $T = \mathcal{O}(n^{\frac{1}{\alpha}} \text{polylog}(n))$

Approximate graph coloring

Problem: Let $2 \leq \chi \leq c$. The approximate graph coloring problem asks to c-color χ-chromatic graphs.

- **Previous results** (classical LOCAL model)
  - $c = 2, \chi = 2 \implies T = \Theta(n)$
  - $c = 3, \chi = 2 \implies T = \Omega(\sqrt{n})$ [Brandt et al., PODC '17]
  - $c \geq 4, \chi = 2 \implies T = \Omega(\log(n))$ [Linial, FOCS '87]

- **Our results** (All LOCAL models – from det. to non-signaling)
  - $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor \implies T = \Omega(n^{\frac{1}{\alpha}}), T = \mathcal{O}(n^{\frac{1}{\alpha}} \text{polylog}(n))$
  - $\implies$ no quantum advantage

Bogdanov's construction

- **Theorem**: Let $\chi \geq 2$, $r \geq 1$, $\alpha \geq 0$ be integers. There exists a graph $G_{\alpha,\chi}$ such that

- $\mathcal{LX}_r(G_{\alpha,\chi}) = \chi$

Bogdanov's construction

- **Theorem**: Let $\chi \geq 2$, $r \geq 1$, $\alpha \geq 0$ be integers. There exists a graph $G_{\alpha,\chi}$ such that

- $\mathcal{L}\mathcal{X}_r(G_{\alpha,\chi}) = \chi$
- $\mathcal{X}(G_{\alpha,\chi}) = (\alpha+1)(\chi-1)+1$

Bogdanov's construction

- **Theorem**: Let $\chi \geq 2$, $r \geq 1$, $\alpha \geq 0$ be integers. There exists a graph $G_{\alpha,\chi}$ such that

- $\mathcal{LX}_r(G_{\alpha,\chi}) = \chi$
- $\mathcal{X}(G_{\alpha,\chi}) = (\alpha + 1)(\chi - 1) + 1$
- $|V(G_{\alpha,\chi})| = n = \frac{(2r\chi+1)^{\alpha+1}-1}{2r}$

Bogdanov's construction

- **Theorem**: Let $\chi \geq 2$, $r \geq 1$, $\alpha \geq 0$ be integers. There exists a graph $G_{\alpha,\chi}$ such that

- $\mathcal{LX}_r(G_{\alpha,\chi}) = \chi$
- $\mathcal{X}(G_{\alpha,\chi}) = (\alpha + 1)(\chi - 1) + 1$
- $|V(G_{\alpha,\chi})| = n = \frac{(2r\chi+1)^{\alpha+1}-1}{2r}$

- **Remark 1**: Let $c \geq \chi \geq 2$. If $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$, then $\alpha(\chi - 1) + 1 \leq c < (\alpha + 1)(\chi - 1) + 1$

Bogdanov's construction

- **Theorem**: Let $\chi \geq 2$, $r \geq 1$, $\alpha \geq 0$ be integers. There exists a graph $G_{\alpha,\chi}$ such that

- $\mathcal{LX}_r(G_{\alpha,\chi}) = \chi$
- $\mathcal{X}(G_{\alpha,\chi}) = (\alpha+1)(\chi-1)+1$
- $|V(G_{\alpha,\chi})| = n = \frac{(2r\chi+1)^{\alpha+1}-1}{2r}$

- **Remark 1**: Let $c \geq \chi \geq 2$. If $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$, then $\alpha(\chi-1)+1 \leq c < (\alpha+1)(\chi-1)+1$

- **Remark 2**: $r = \Theta(n^{\frac{1}{\alpha}}/\chi^{1+\frac{1}{\alpha}})$

Bogdanov's construction

- **Theorem**: Let $\chi \geq 2$, $r \geq 1$, $\alpha \geq 0$ be integers. There exists a graph $G_{\alpha,\chi}$ such that

- $\mathcal{LX}_r(G_{\alpha,\chi}) = \chi$
- $\mathcal{X}(G_{\alpha,\chi}) = (\alpha+1)(\chi-1)+1$
- $|V(G_{\alpha,\chi})| = n = \frac{(2r\chi+1)^{\alpha+1}-1}{2r}$

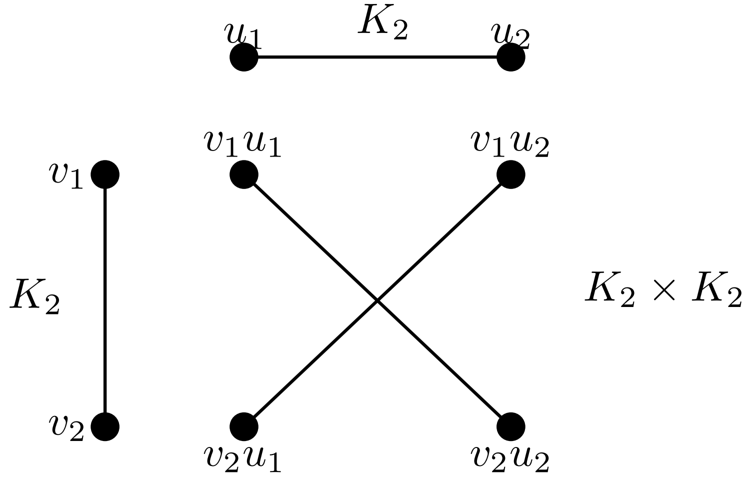
- **Remark 1**: Let $c \geq \chi \geq 2$. If $\alpha = \lfloor \frac{c-1}{\chi-1} \rfloor$, then $\alpha(\chi-1)+1 \leq c < (\alpha+1)(\chi-1)+1$

- **Remark 2**: $r = \Theta(n^{\frac{1}{\alpha}}/\chi^{1+\frac{1}{\alpha}})$

$G_{\alpha,\chi}$ is constructed recursively in α

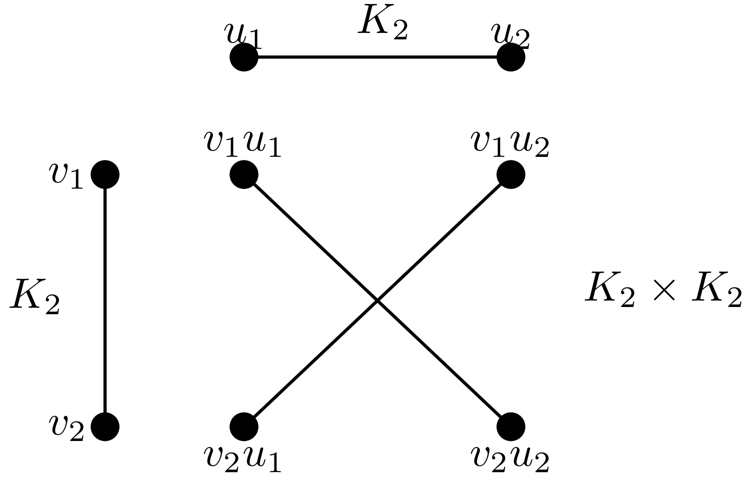
Bogdanov's construction

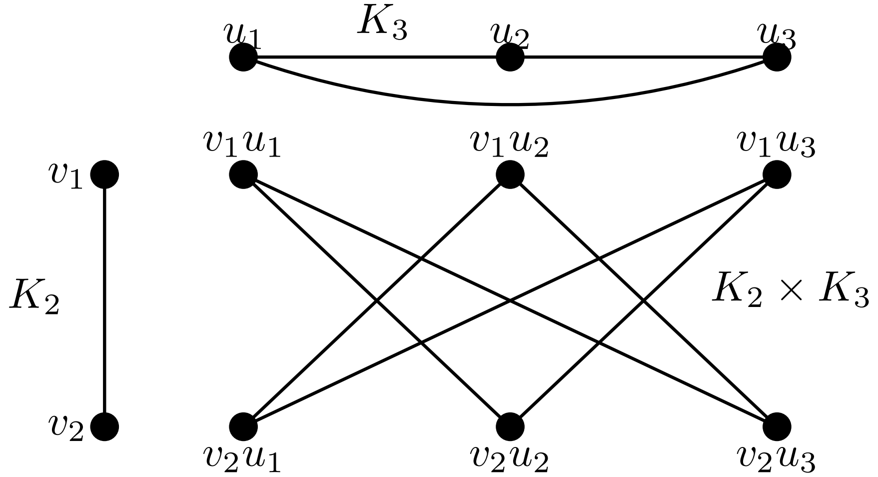
- **Tensor product** of graphs



Bogdanov's construction

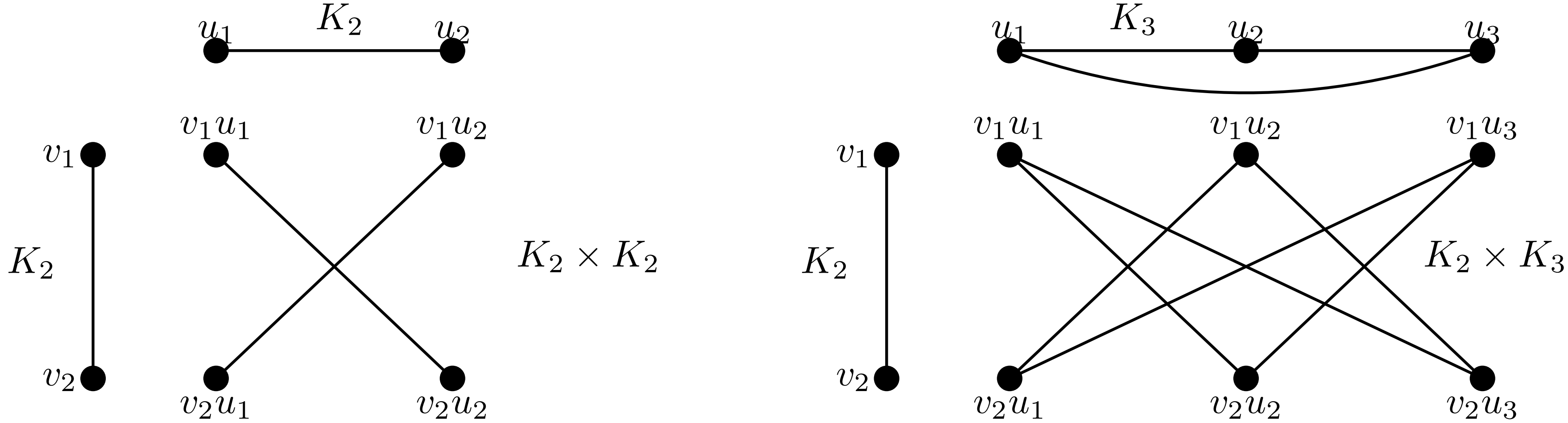
- **Tensor product** of graphs





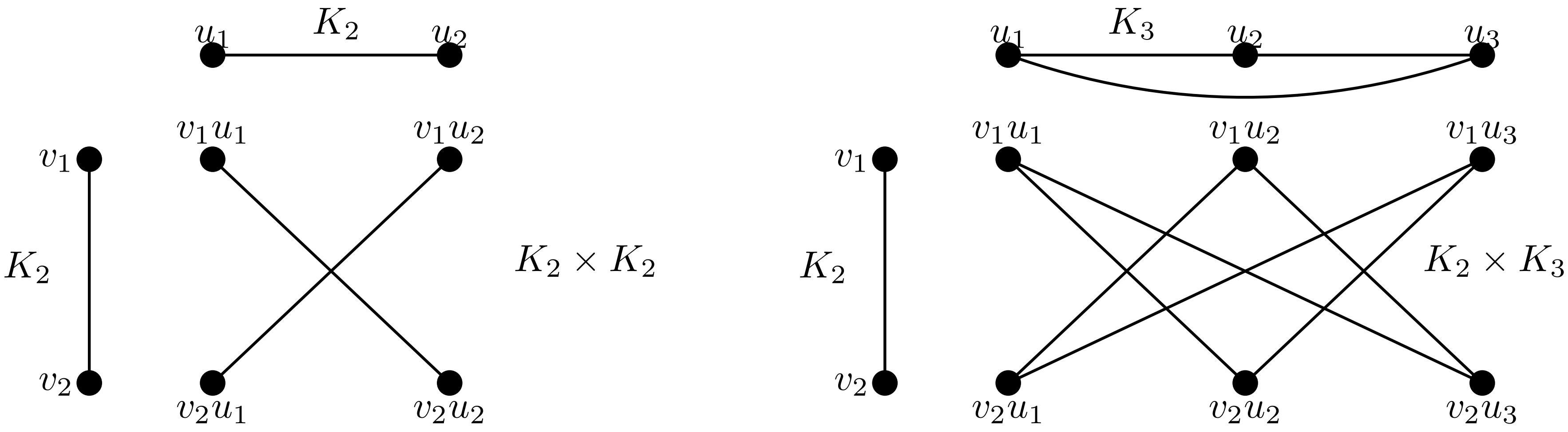
Bogdanov's construction

- **Tensor product** of graphs



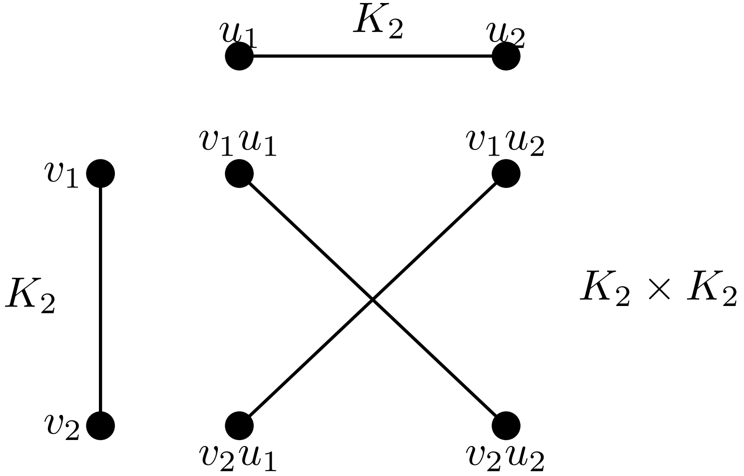
Bogdanov's construction

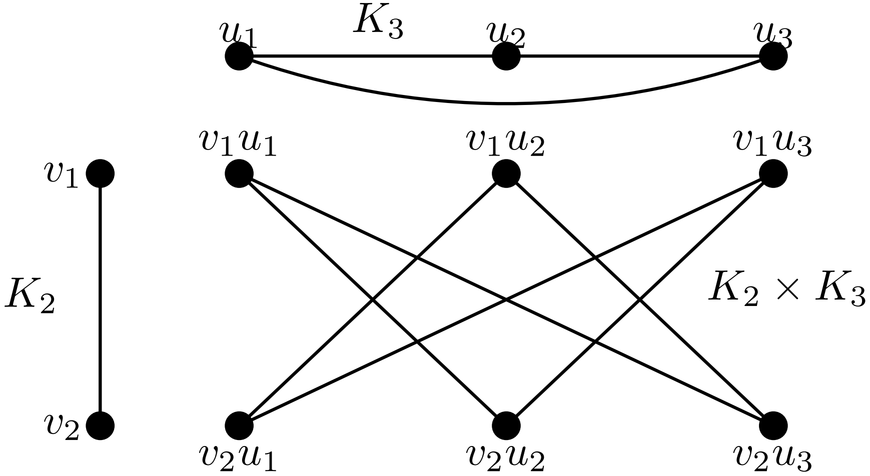
- **Tensor product** of graphs



Bogdanov's construction

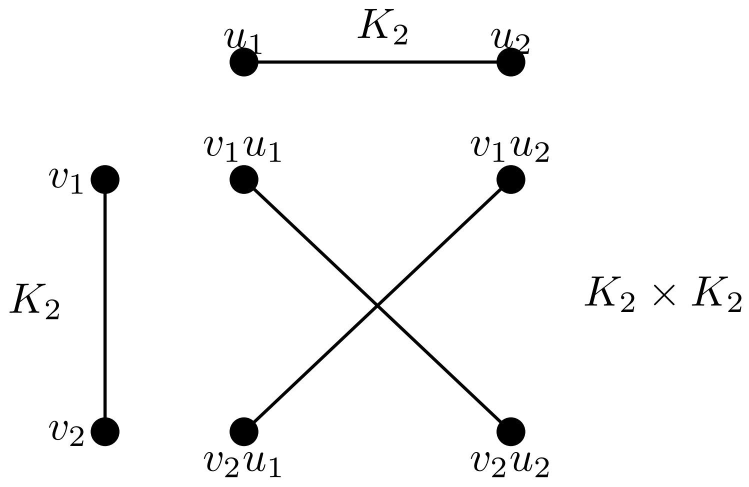
- **Tensor product** of graphs

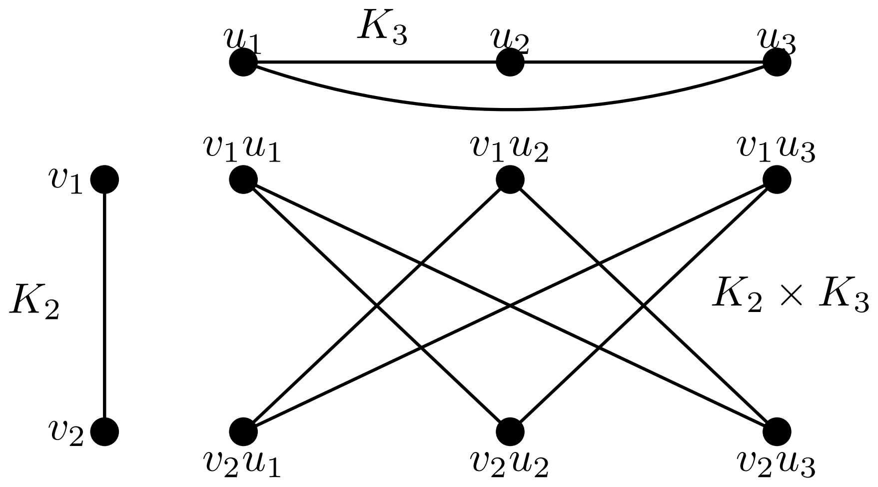




Bogdanov's construction

- **Tensor product** of graphs





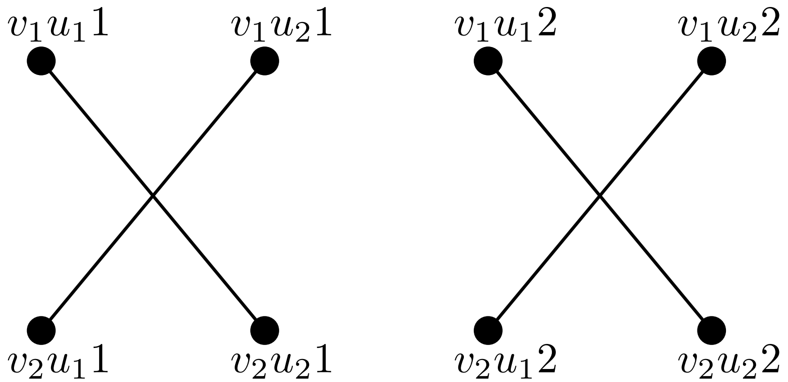
r-join of graphs

- **r-join** of graphs $(r = 2)$ $\quad K_2^{(1)} \star_2 K_2^{(2)}$

r-join of graphs

- **r-join** of graphs $(r = 2)$

$K_2^{(1)} \star_2 K_2^{(2)}$

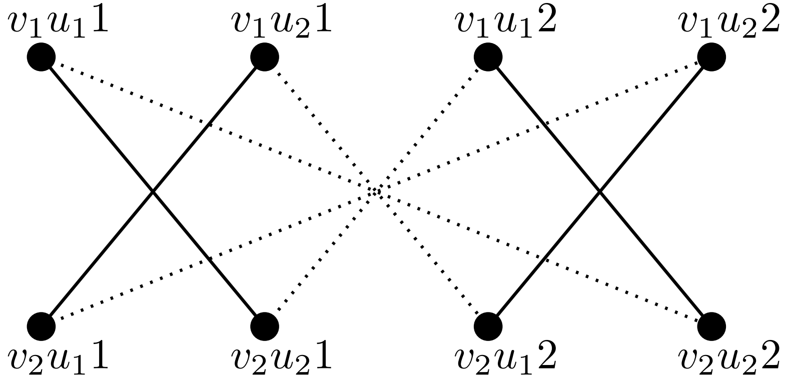


$K_2^{(1)} \times K_2^{(2)} \times \{1\}$ $\quad$ $K_2^{(1)} \times K_3^{(2)} \times \{2\}$

r-join of graphs

- **r-join** of graphs $(r = 2)$

$K_2^{(1)} \star_2 K_2^{(2)}$

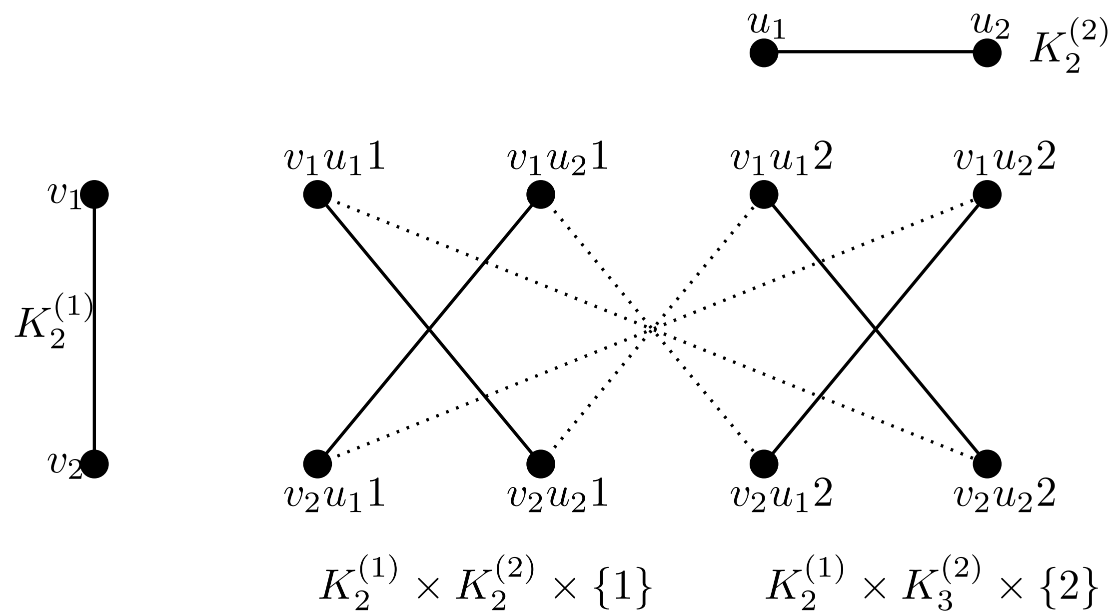


$K_2^{(1)} \times K_2^{(2)} \times \{1\}$ $\quad$ $K_2^{(1)} \times K_3^{(2)} \times \{2\}$

r-join of graphs

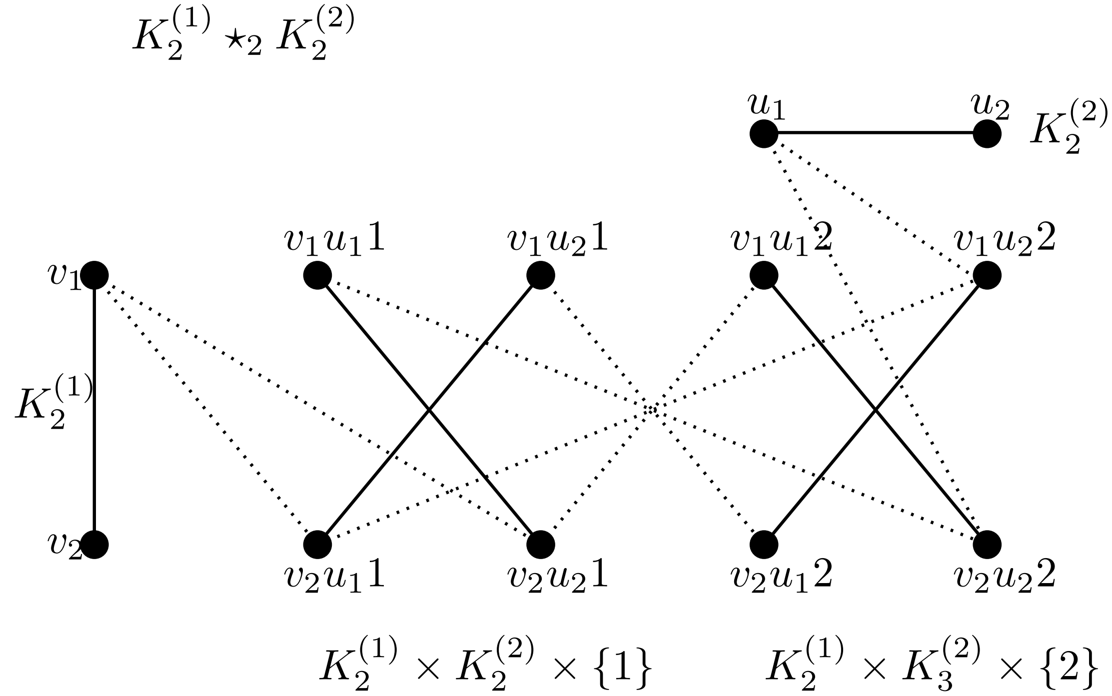
- $\mathbf{r}$-**join** of graphs $(r = 2)$

$K_2^{(1)} \star_2 K_2^{(2)}$



r-join of graphs

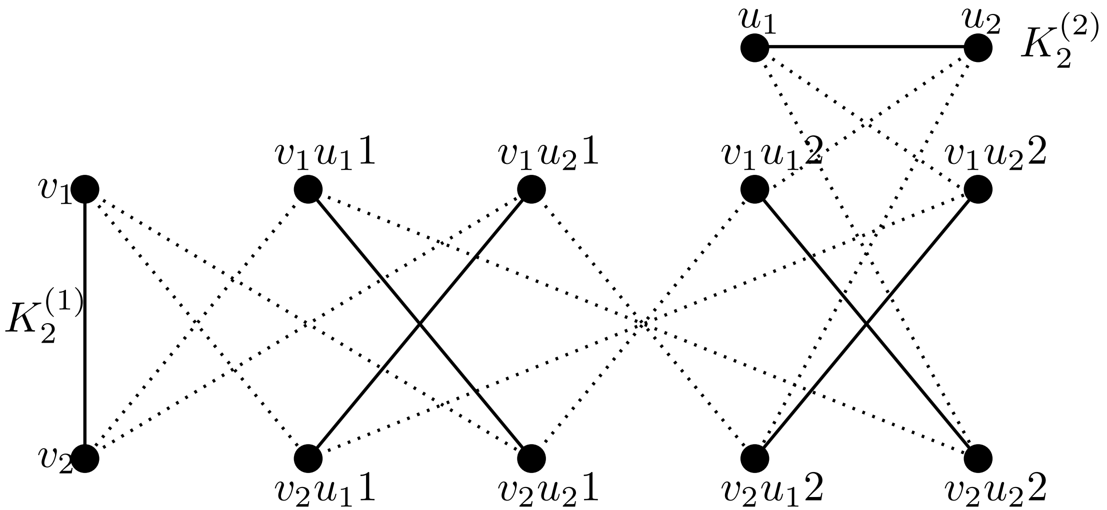
- **r-join** of graphs $(r = 2)$



r-join of graphs

- $\mathbf{r}$-**join** of graphs $(r = 2)$

$K_2^{(1)} \star_2 K_2^{(2)}$

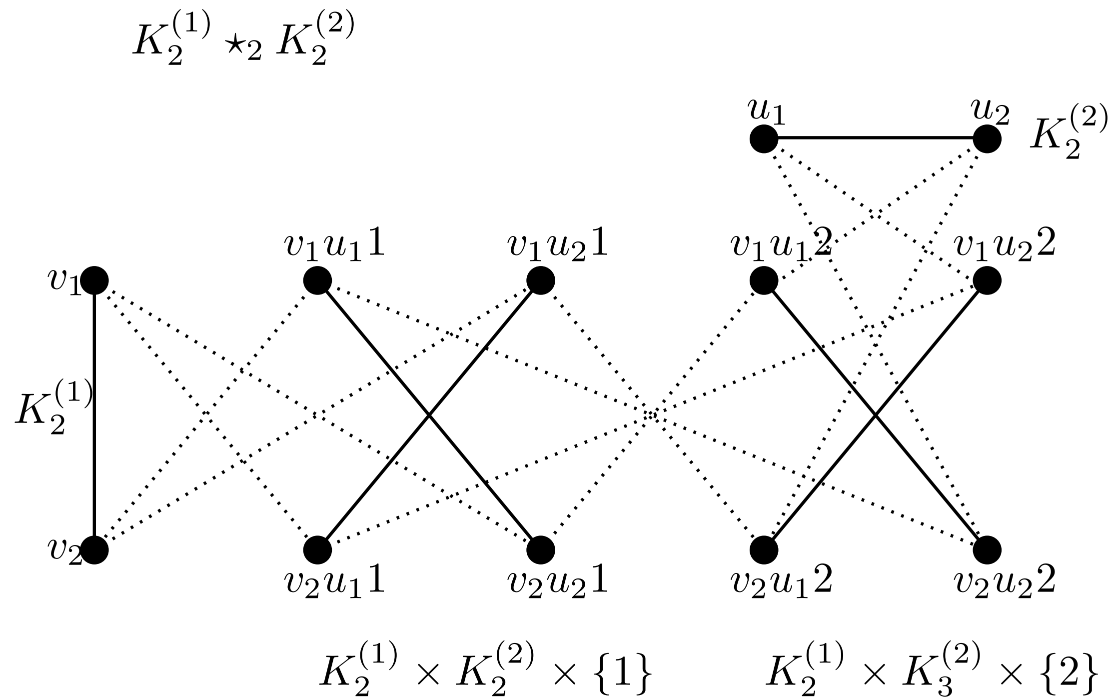


$K_2^{(1)} \times K_2^{(2)} \times \{1\}$ $K_2^{(1)} \times K_3^{(2)} \times \{2\}$

r-join of graphs

- **r-join** of graphs $(r = 2)$

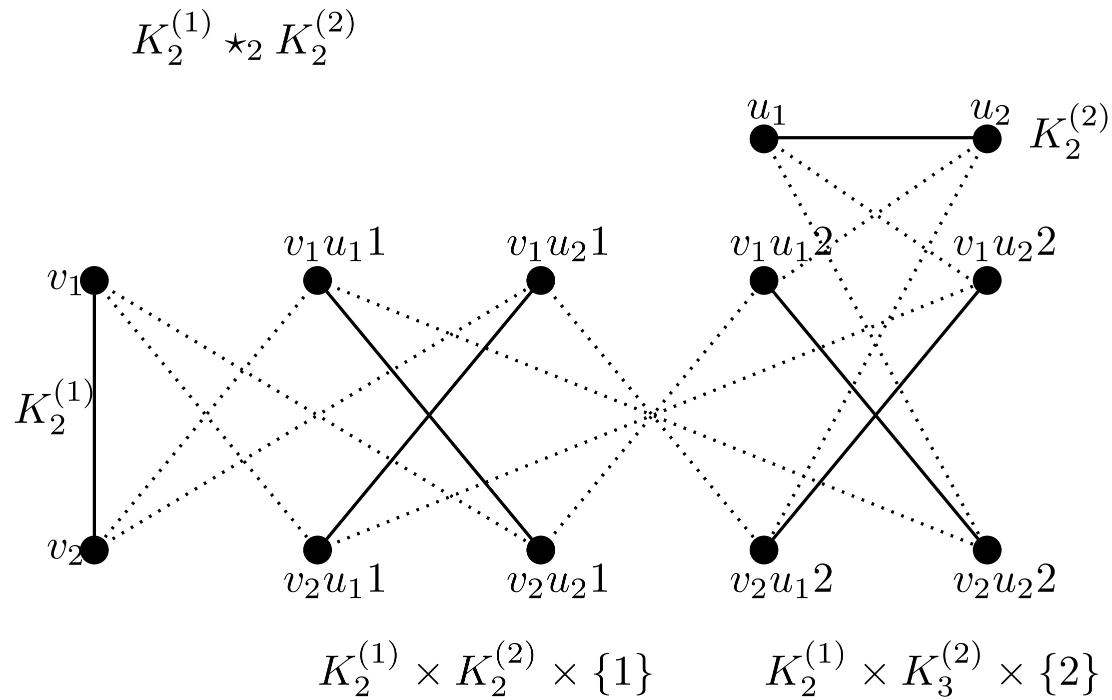
  - $\mathcal{X}(K_2^{(1)} \star_2 K_2^{(2)}) = 3$



r-join of graphs

- **r-join** of graphs $(r = 2)$

  - $\mathcal{X}(K_2^{(1)} \star_2 K_2^{(2)}) = 3$
  - $G_{\alpha,\chi} = K_2^{(1)} \star_2 K_2^{(2)}$ for $(\alpha, \chi) = (1, 2)$

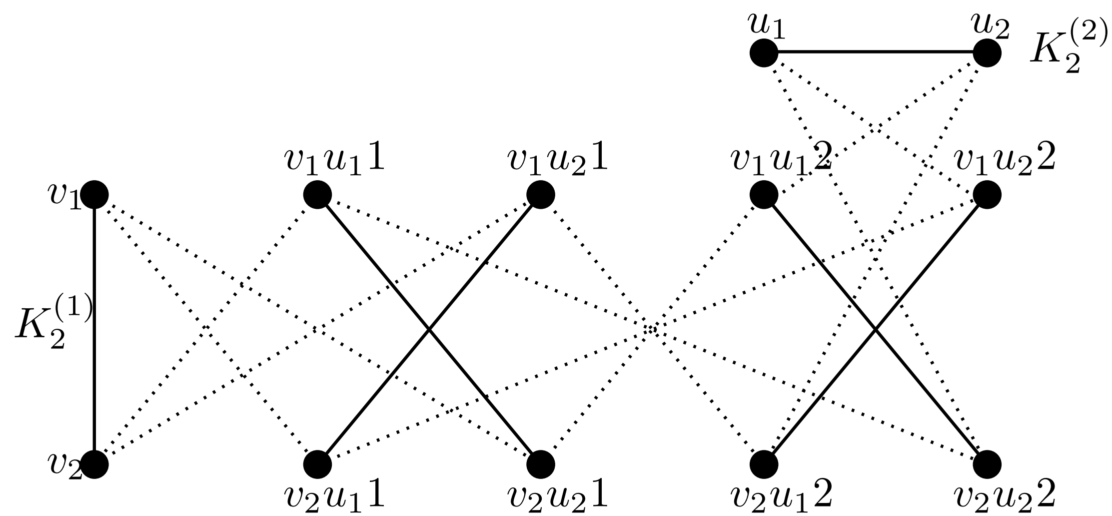


r-join of graphs

- **r-join** of graphs $(r = 2)$

  - $\mathcal{X}(K_2^{(1)} \star_2 K_2^{(2)}) = 3$
  - $G_{\alpha,\chi} = K_2^{(1)} \star_2 K_2^{(2)}$ for $(\alpha, \chi) = (1, 2)$
  - $G_{\alpha,\chi} = G_{\alpha-1,\chi} \star_r K_\chi^{(\alpha)}$

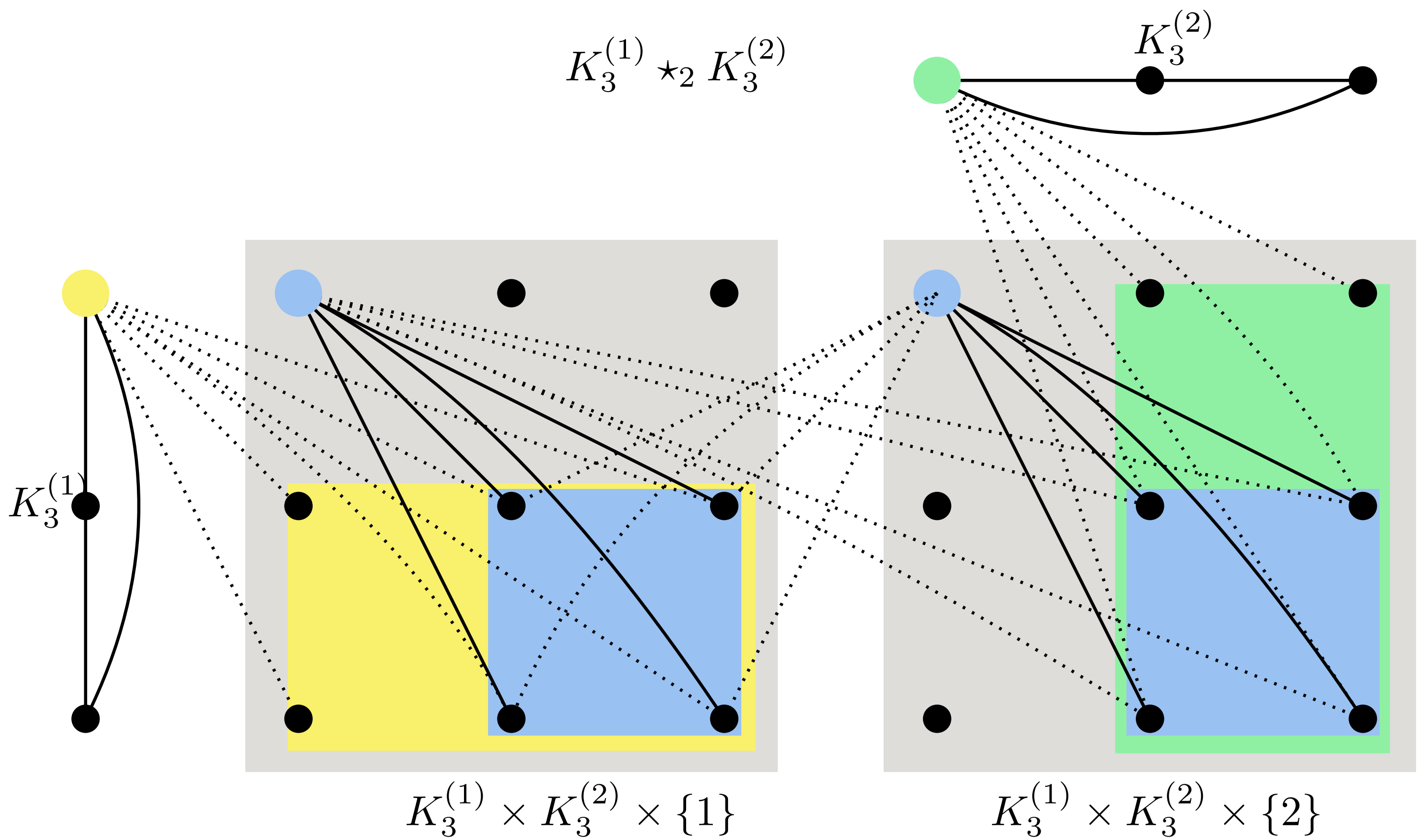
$K_2^{(1)} \star_2 K_2^{(2)}$



$K_2^{(1)} \times K_2^{(2)} \times \{1\}$ $K_2^{(1)} \times K_3^{(2)} \times \{2\}$

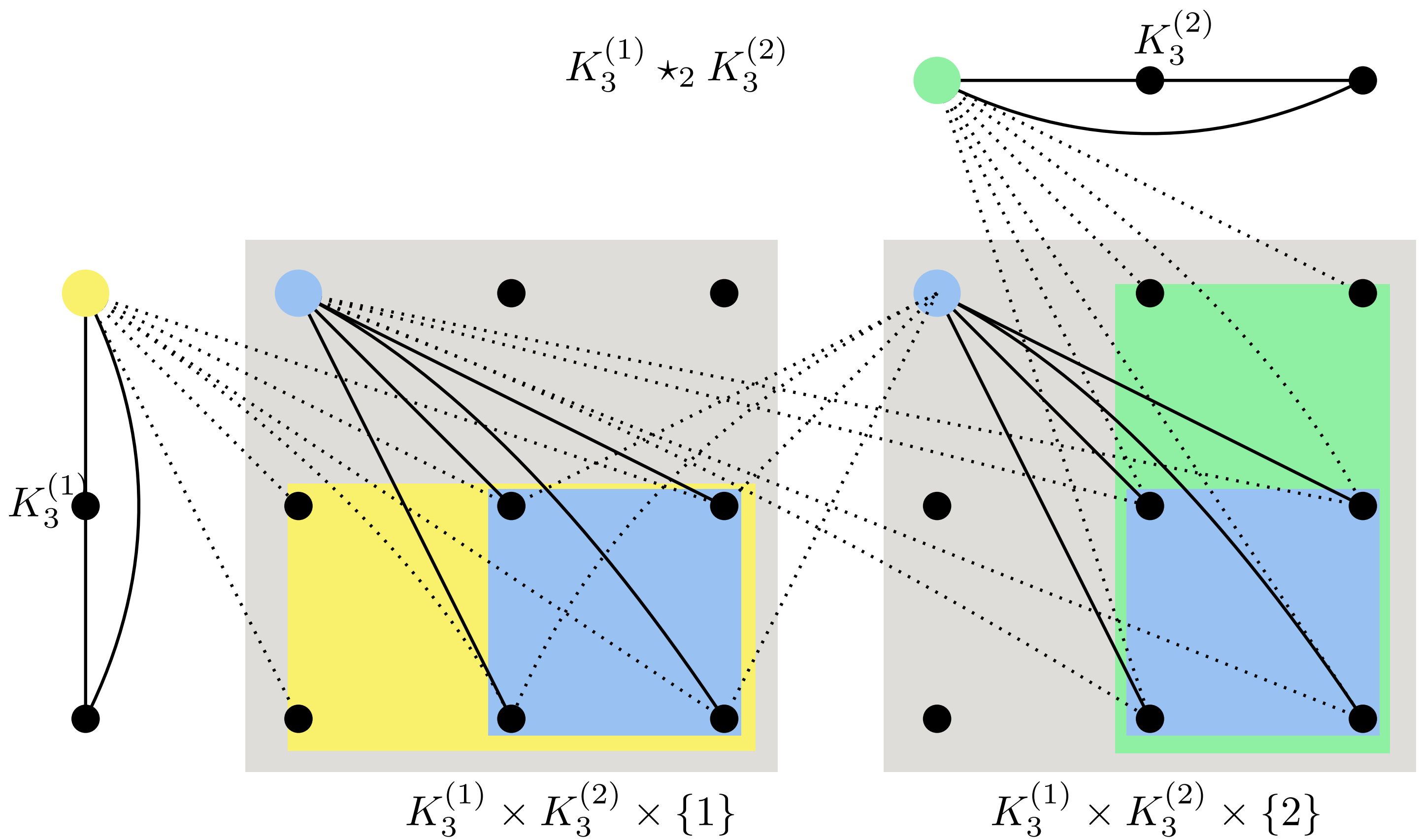
Bogdanov's construction

- **r-join** of graphs ($r = 2$), example for $\chi = 3$



Bogdanov's construction

- **r-join** of graphs $(r = 2)$, example for $\chi = 3$



- $\mathcal{X}(K_3^{(1)} \star_2 K_3^{(2)}) = 5$

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs
- We prove that the following LCLs cannot admit quantum advantage

- c-coloring trees
- 3-coloring grids
- c-coloring χ chromatic graphs
- how? Extending indistinguishability and graph existential arguments all the way up to NS-LOCAL

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs
- We prove that the following LCLs cannot admit quantum advantage

- c-coloring trees
- 3-coloring grids
- c-coloring χ chromatic graphs
- how? Extending indistinguishability and graph existential arguments all the way up to NS-LOCAL

- Under development (spoiler alert)

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs
- We prove that the following LCLs cannot admit quantum advantage

- c-coloring trees
- 3-coloring grids
- c-coloring χ chromatic graphs
- how? Extending indistinguishability and graph existential arguments all the way up to NS-LOCAL

- Under development (spoiler alert)

- all problems' complexities $\mathcal{O}(\log^\star n)$ in classical LOCAL become $\mathcal{O}(1)$ in NS-LOCAL (plus independence!)

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs
- We prove that the following LCLs cannot admit quantum advantage
- c-coloring trees
- 3-coloring grids
- c-coloring χ chromatic graphs
- how? Extending indistinguishability and graph existential arguments all the way up to NS-LOCAL
- Under development (spoiler alert)
- all problems' complexities $\mathcal{O}(\log^\star n)$ in classical LOCAL become $\mathcal{O}(1)$ in NS-LOCAL (plus independence!)
- (paths, cycles, trees): all problems' complexities $\mathcal{O}(1)$ in φ-LOCAL become $\mathcal{O}(\log^\star n)$ in classical LOCAL

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs
- We prove that the following LCLs cannot admit quantum advantage
- c-coloring trees
- 3-coloring grids
- c-coloring χ chromatic graphs
- how? Extending indistinguishability and graph existential arguments all the way up to NS-LOCAL
- Under development (spoiler alert)
- all problems' complexities $\mathcal{O}(\log^\star n)$ in classical LOCAL become $\mathcal{O}(1)$ in NS-LOCAL (plus independence!)
- (paths, cycles, trees): all problems' complexities $\mathcal{O}(1)$ in φ-LOCAL become $\mathcal{O}(\log^\star n)$ in classical LOCAL
- (paths, cycles, rooted trees): there cannot be quantum advantage beyond complexity $\mathcal{O}(\log^\star n)$

Summary

- To date, no quantum advantage has been shown in the LOCAL model for LCLs
- We prove that the following LCLs cannot admit quantum advantage
- c-coloring trees
- 3-coloring grids
- c-coloring χ chromatic graphs
- how? Extending indistinguishability and graph existential arguments all the way up to NS-LOCAL
- Under development (spoiler alert)
- all problems' complexities $\mathcal{O}(\log^\star n)$ in classical LOCAL become $\mathcal{O}(1)$ in NS-LOCAL (plus independence!)
- (paths, cycles, trees): all problems' complexities $\mathcal{O}(1)$ in φ-LOCAL become $\mathcal{O}(\log^\star n)$ in classical LOCAL
- (paths, cycles, rooted trees): there cannot be quantum advantage beyond complexity $\mathcal{O}(\log^\star n)$

THANKS!