Locality via Alternation?

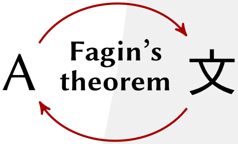
Fabian Reiter

LIGM, Université Gustave Eiffel

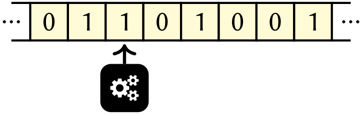
ADGA 2024

Video on YouTube
youtu.be/wCWBSvV7B_I

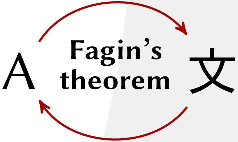
Two characterizations of NP



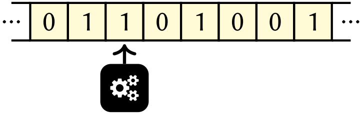
Two characterizations of NP

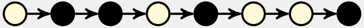


Nondet. polynomial-time
Turing machines



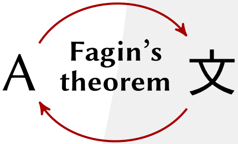
Two characterizations of NP





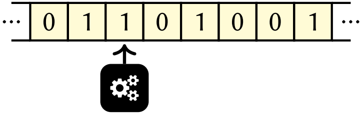
$\exists R\colon \mathit{Bijective}(R) \land \forall x, y\colon R(x, y) \to (\bullet x \leftrightarrow \neg \bullet y)$

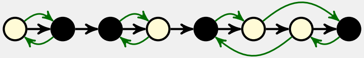
Nondet. polynomial-time Turing machines



Existential fragment of second-order logic

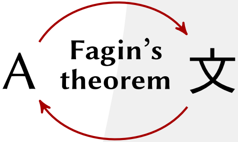
Two characterizations of NP





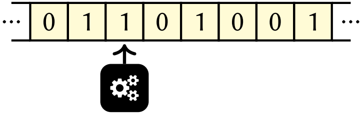
$\exists R\colon \mathit{Bijective}(R) \wedge \forall x, y\colon R(x, y) \rightarrow (\bullet x \leftrightarrow \neg\bullet y)$

Nondet. polynomial-time Turing machines

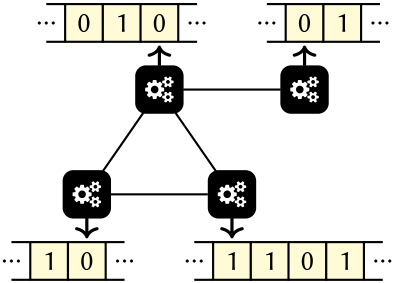


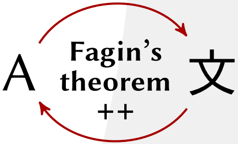
Existential fragment of second-order logic

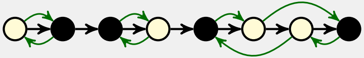
Two characterizations of NP



Nondet. polynomial-time
Turing machines

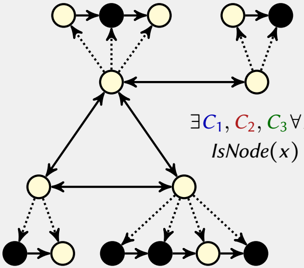






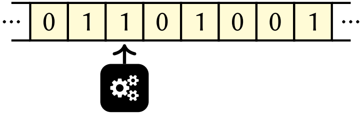
$\exists R\colon \mathit{Bijective}(R) \land \forall x, y\colon R(x, y) \to (\bullet x \leftrightarrow \neg \bullet y)$

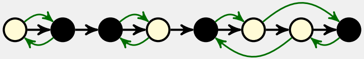
Existential fragment of
second-order logic



$\exists C_1, C_2, C_3 \forall x\colon$
$\mathit{IsNode}(x) \to \mathit{WellColored}(x)$

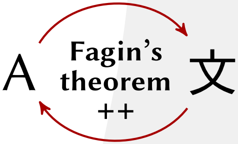
Two characterizations of NP



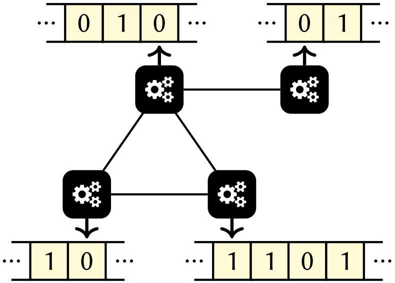


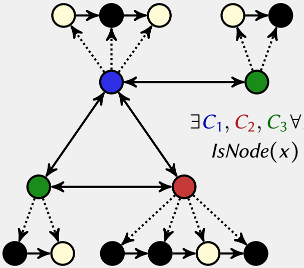
$\exists R\colon \mathit{Bijective}(R) \wedge \forall x, y\colon R(x, y) \rightarrow (\bullet x \leftrightarrow \neg\bullet y)$

Nondet. polynomial-time Turing machines



Existential fragment of second-order logic

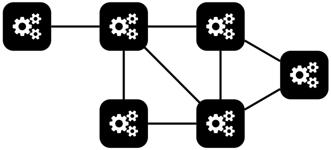




$\exists C_1, C_2, C_3 \forall x\colon$

$\mathit{IsNode}(x) \rightarrow \mathit{WellColored}(x)$

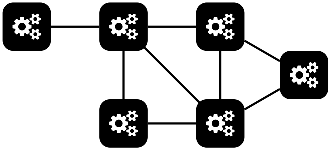
Model of computation



The LOCAL model

- Network of nodes with IDs & labels
- Same algorithm on all nodes
- Synchronous communication rounds

Model of computation



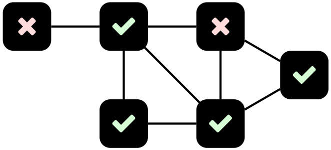
The LOCAL model

- Network of nodes with IDs & labels
- Same algorithm on all nodes
- Synchronous communication rounds

Local distributed decision

- Constant number of rounds
- Graph { accepted unanimously / or rejected by veto

Model of computation



not Eulerian

(some nodes of odd degree)

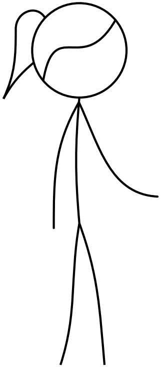
The LOCAL model

- Network of nodes with IDs & labels
- Same algorithm on all nodes
- Synchronous communication rounds

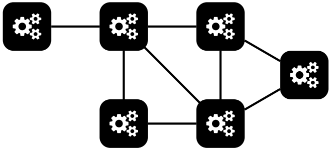
Local distributed decision

- Constant number of rounds
- Graph { accepted unanimously / or rejected by veto }

Model of computation



Eve (∃)



not Eulerian
(some nodes of odd degree)

The LOCAL model

- Network of nodes with IDs & labels
- Same algorithm on all nodes
- Synchronous communication rounds

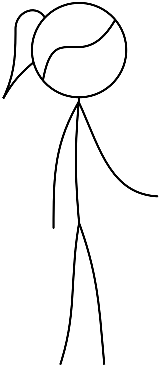
Local distributed decision

- Constant number of rounds
- Graph { accepted unanimously / or rejected by veto }

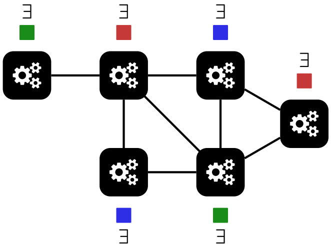
Nondeterministic extension

- Certificates chosen by Eve

Model of computation



Eve (∃)



not Eulerian

(some nodes of odd degree)

3-colorable

(Eve can find a 3-coloring)

The LOCAL model

- Network of nodes with IDs & labels
- Same algorithm on all nodes
- Synchronous communication rounds

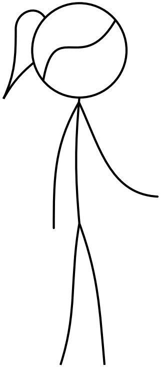
Local distributed decision

- Constant number of rounds
- Graph { accepted unanimously / or rejected by veto }

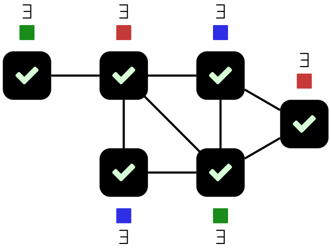
Nondeterministic extension

- Certificates chosen by Eve

Model of computation



Eve (∃)



not Eulerian

(some nodes of odd degree)

3-colorable

(Eve can find a 3-coloring)

The LOCAL model

- Network of nodes with IDs & labels
- Same algorithm on all nodes
- Synchronous communication rounds

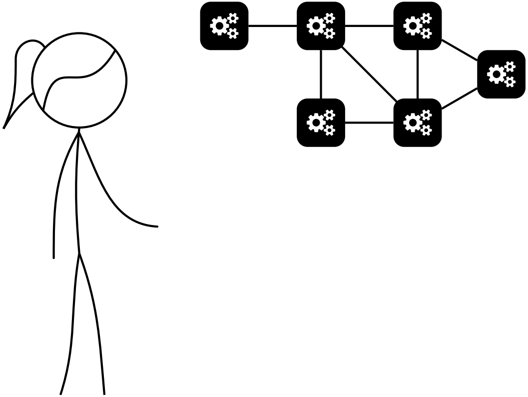
Local distributed decision

- Constant number of rounds
- Graph { accepted unanimously
  or rejected by veto

Nondeterministic extension

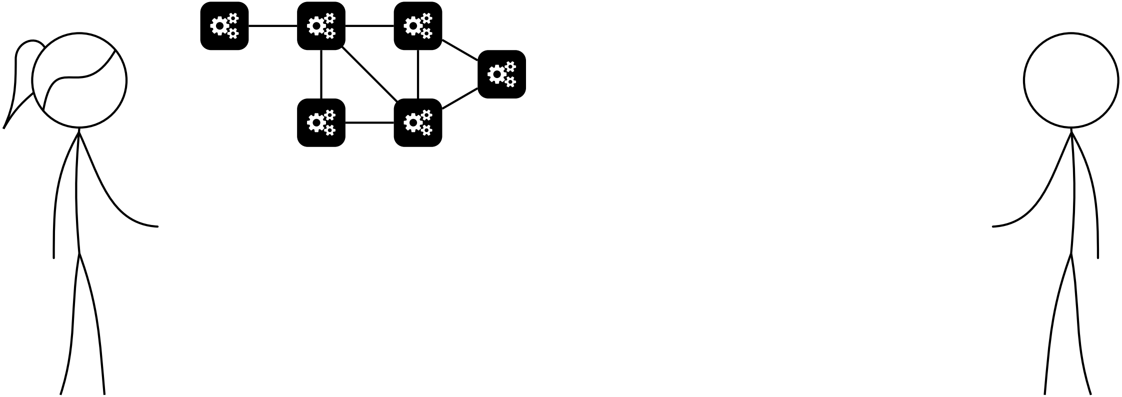
- Certificates chosen by Eve

Alternation



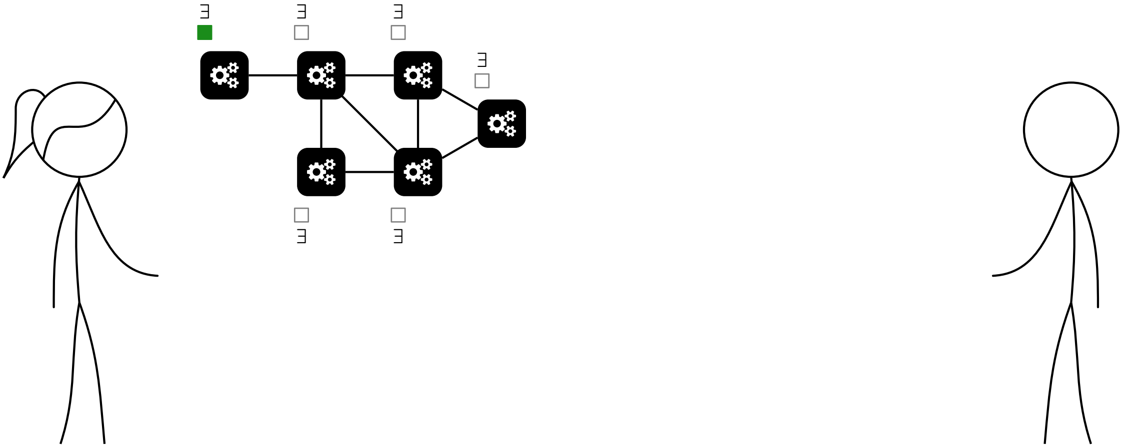
Eve (∃)

Alternation



Eve (∃) Adam (∀)

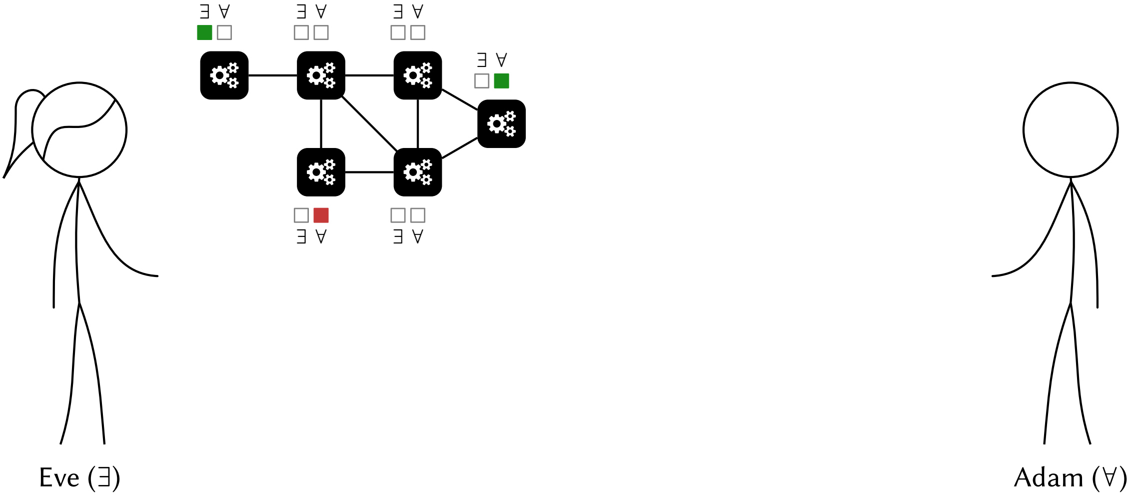
Alternation



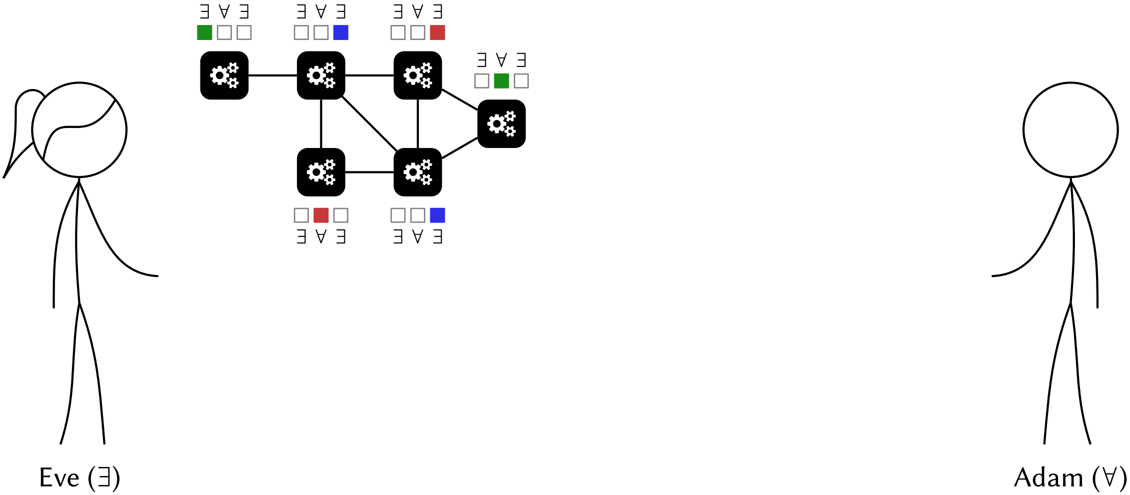
Eve (∃)

Adam (∀)

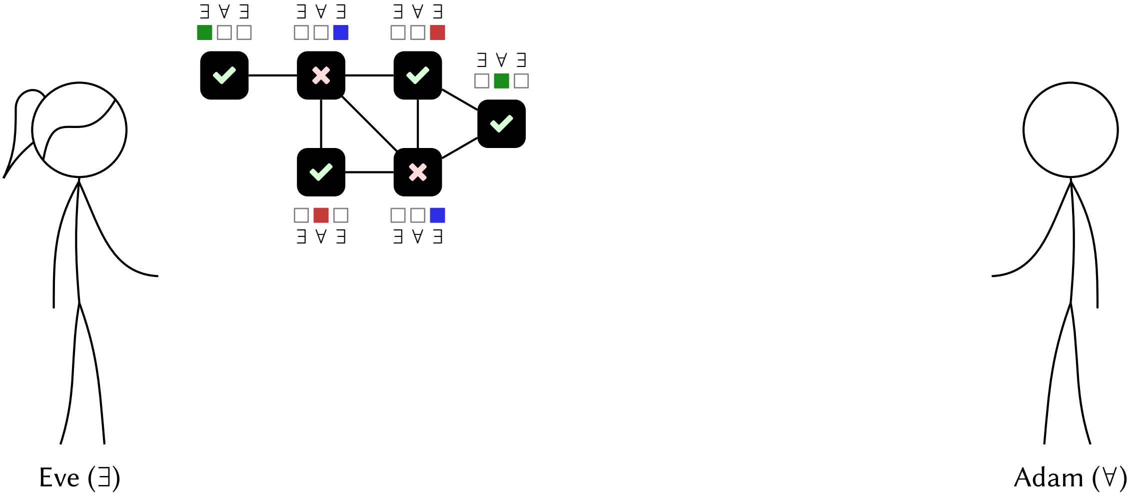
Alternation



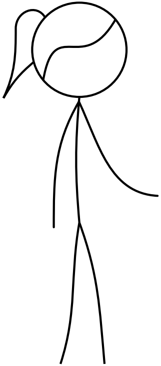
Alternation

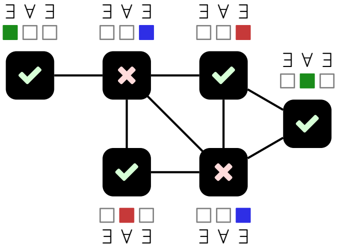


Alternation



Alternation

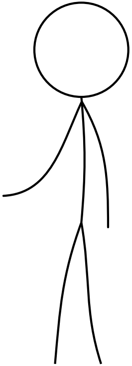




not 3-round 3-colorable

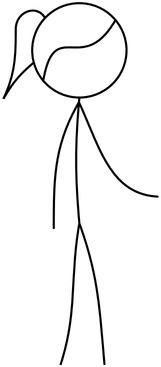
(Adam has a winning strategy)

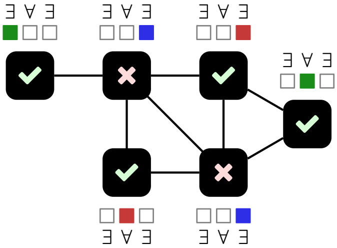
Eve (∃)



Adam (∀)

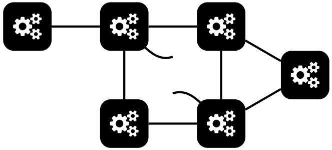
Alternation

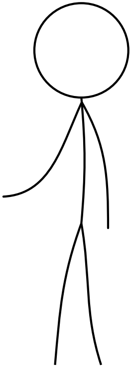




not 3-round 3-colorable

(Adam has a winning strategy)

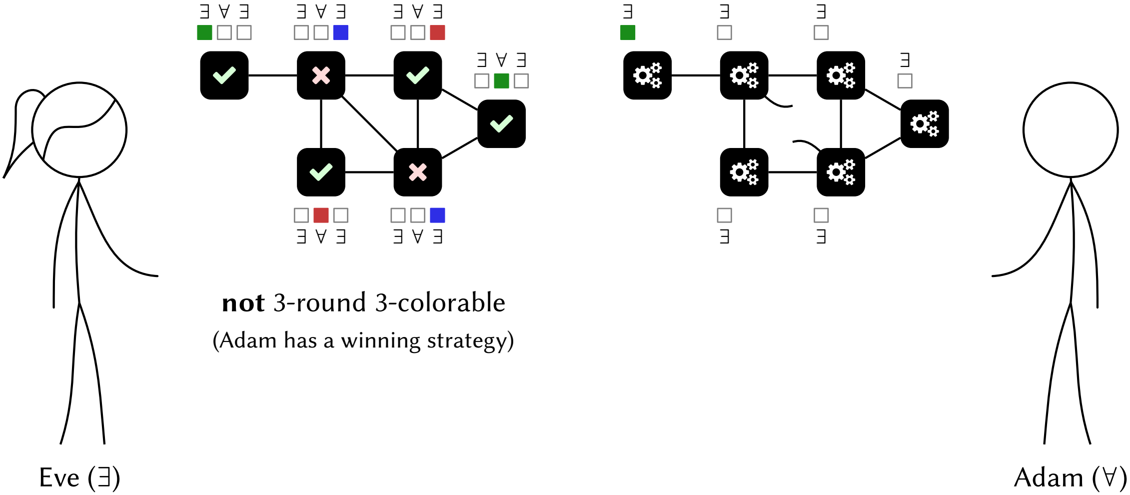




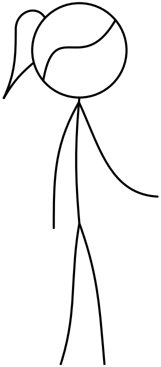
Eve (∃)

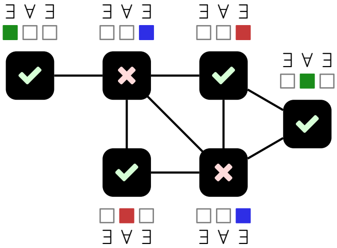
Adam (∀)

Alternation



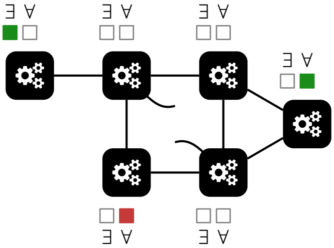
Alternation

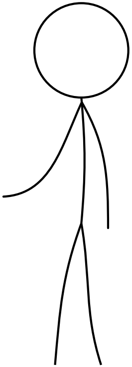




not 3-round 3-colorable

(Adam has a winning strategy)

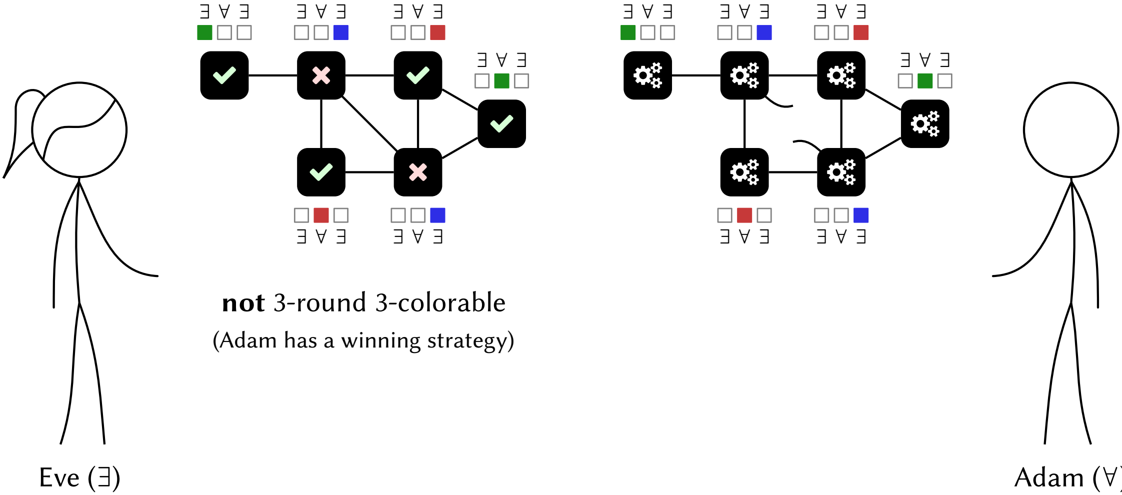




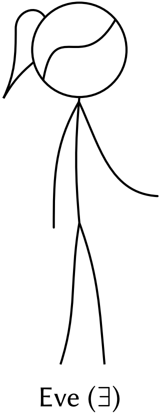
Eve (∃)

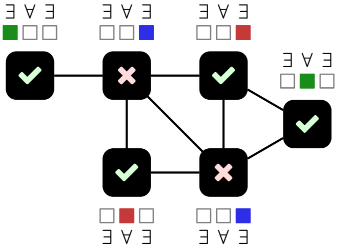
Adam (∀)

Alternation



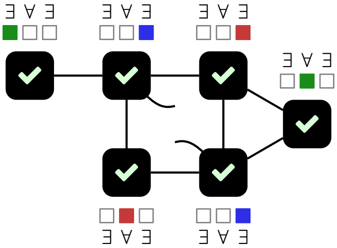
Alternation

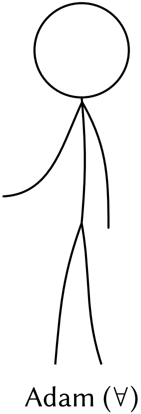




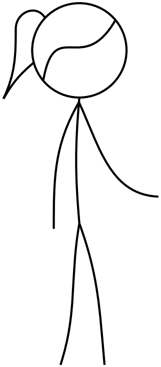
not 3-round 3-colorable

(Adam has a winning strategy)

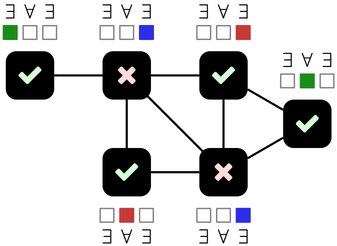




Alternation

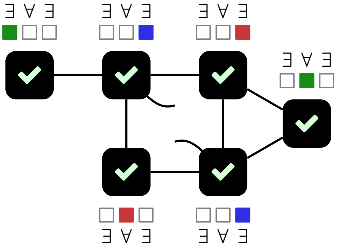


Eve (∃)



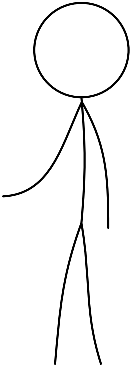
not 3-round 3-colorable

(Adam has a winning strategy)



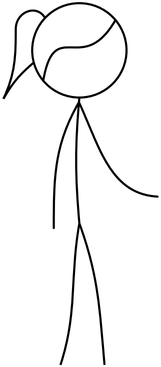
3-round 3-colorable

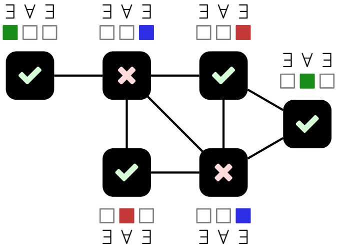
(Eve has a winning strategy)



Adam (∀)

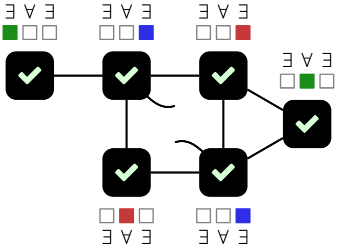
Alternation





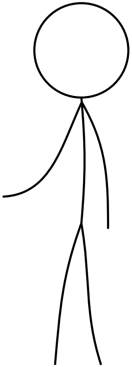
not 3-round 3-colorable

(Adam has a winning strategy)



3-round 3-colorable

(Eve has a winning strategy)



$\boldsymbol{\Sigma}_3$ $\exists\,\forall\,\exists$

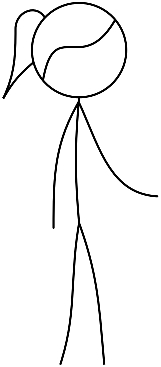
$\boldsymbol{\Sigma}_2$ $\exists\,\forall$

$\boldsymbol{\Sigma}_1$ $\exists$

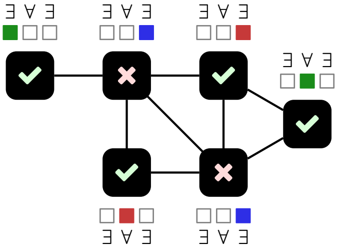
Eve (∃)

Adam (∀)

Alternation

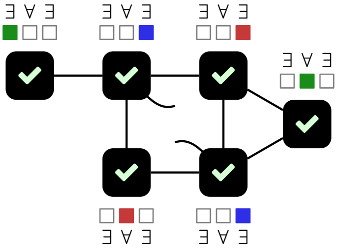


Eve (∃)



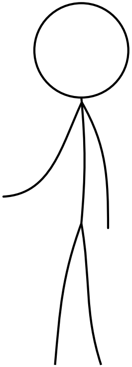
not 3-round 3-colorable

(Adam has a winning strategy)



3-round 3-colorable

(Eve has a winning strategy)



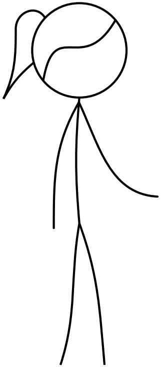
Adam (∀)

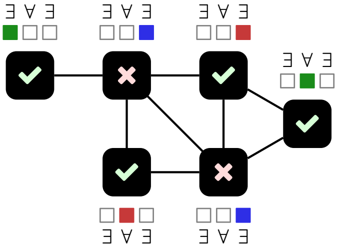
$\longrightarrow$ $\boldsymbol{\Sigma}_3$ ∃ ∀ ∃

$\boldsymbol{\Sigma}_2$ ∃ ∀

$\boldsymbol{\Sigma}_1$ ∃

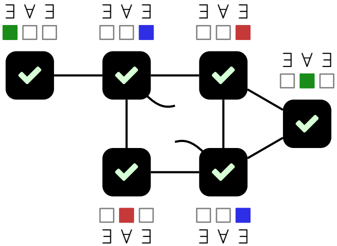
Alternation





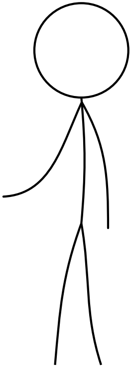
not 3-round 3-colorable

(Adam has a winning strategy)



3-round 3-colorable

(Eve has a winning strategy)



Eve (∃)

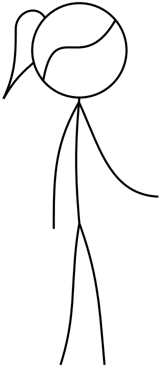
Adam (∀)

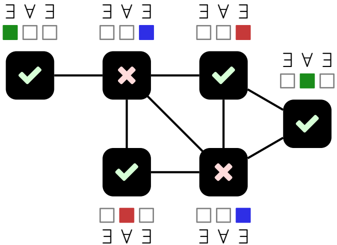
⟶ $\boldsymbol{\Sigma}_3$ ∃ ∀ ∃

$\boldsymbol{\Sigma}_2$ ∃ ∀

⟶ $\boldsymbol{\Sigma}_1$ ∃

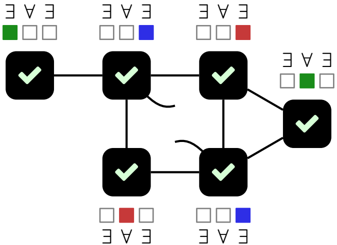
Alternation





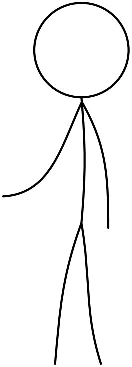
not 3-round 3-colorable

(Adam has a winning strategy)



3-round 3-colorable

(Eve has a winning strategy)

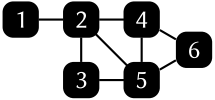


⟶	Σ_3	∃ ∀ ∃	Π_3	∀ ∃ ∀
	Σ_2	∃ ∀	Π_2	∀ ∃
⟶	Σ_1	∃	Π_1	∀

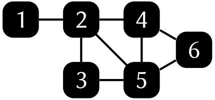
Eve (∃)

Adam (∀)

Related work

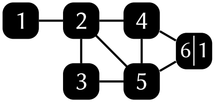
	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work

Related work



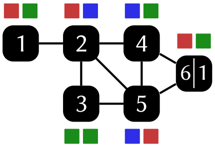
	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	

Related work



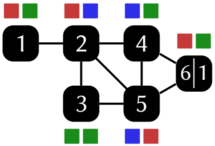
	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	local

Related work



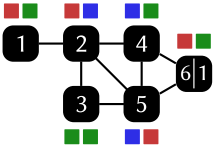
	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	local
IDs in certificates	yes	no	no	

Related work



	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	local
IDs in certificates	yes	no	no	(yes)

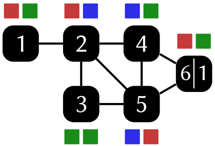
Related work



	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	local
IDs in certificates	yes	no	no	(yes)
Certificate size	$O(\log n)$	unbounded	poly n	

n: number of nodes

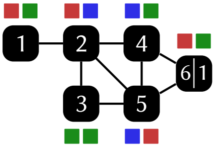
Related work



	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	local
IDs in certificates	yes	no	no	(yes)
Certificate size	$O(\log n)$	unbounded	poly n	poly $\lvert N_r(v) \rvert$

n : number of nodes $\quad\quad$ $\lvert N_r(v) \rvert$: size of node v's r-neighborhood

Related work

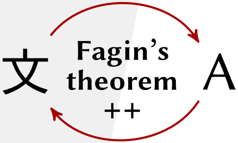


	Feuilloley Fraigniaud Hirvonen (ICALP 2016)	Balliu D'Angelo Fraigniaud Olivetti (STACS 2017)	Aldema Tshuva Oshman (PODC 2022)	This work
ID uniqueness	global	global	global	local
IDs in certificates	yes	no	no	(yes)
Certificate size	$O(\log n)$	unbounded	poly n	poly $\lvert N_r(v)\rvert$
Computation time	unbounded	unbounded	poly n	poly $\lvert N_r(v)\rvert$

n: number of nodes

$\lvert N_r(v)\rvert$: size of node v's r-neighborhood

Using logic and automata theory



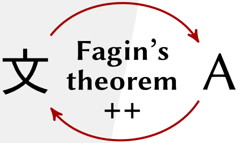
The LOCAL model

- + locally unique IDs
- + local-polynomial bounds

Using logic and automata theory

Monadic second-order logic (MSO)

- *Yields an infinite hierarchy on grids* [1].
- *Satisfies a locality property* [2].



The LOCAL model

+ locally unique IDs
+ local-polynomial bounds

[1] Matz, Schweikardt, Thomas (2002)
[2] Giammarresi, Restivo, Seibert, Thomas (1996)

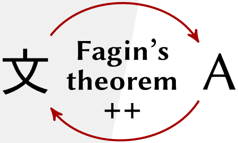
Using logic and automata theory

Monadic second-order logic (MSO)

- *Yields an infinite hierarchy on grids* [1].
- *Satisfies a locality property* [2].

Finite-state automata

- *Satisfy a pumping lemma* [3].
- *Are equivalent to MSO on words* [4].



The LOCAL model

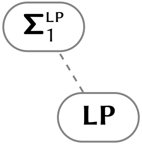
+ locally unique IDs
+ local-polynomial bounds

[1] Matz, Schweikardt, Thomas (2002)
[2] Giammarresi, Restivo, Seibert, Thomas (1996)
[3] Rabin, Scott (1959) & Bar-Hillel, Perles, Shamir (1961)
[4] Büchi (1960) & Elgot (1961) & Trakhtenbrot (1962)

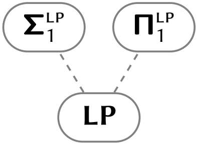
The local-polynomial hierarchy

LP

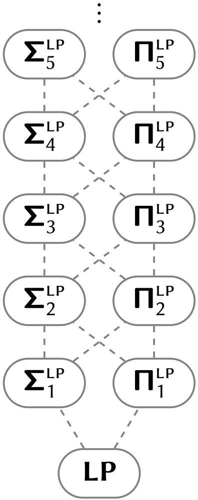
The local-polynomial hierarchy



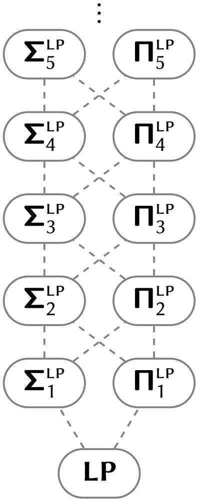
The local-polynomial hierarchy



The local-polynomial hierarchy



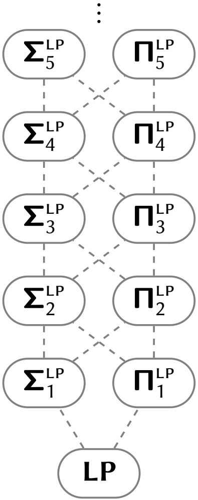
The local-polynomial hierarchy



Connection to classical complexity:

$\Sigma_\ell^{P} = \Sigma_\ell^{LP}|_{NODE}$ $\Pi_\ell^{P} = \Pi_\ell^{LP}|_{NODE}$

The local-polynomial hierarchy



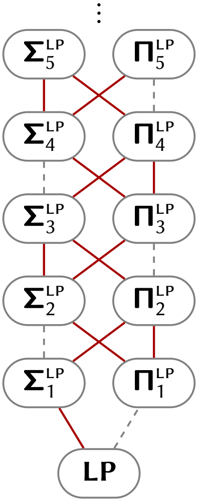
Connection to classical complexity:

$\mathbf{\Sigma}_\ell^{\mathrm{P}} = \mathbf{\Sigma}_\ell^{\mathrm{LP}}\big|_{\mathrm{NODE}}$ $\qquad$ $\mathbf{\Pi}_\ell^{\mathrm{P}} = \mathbf{\Pi}_\ell^{\mathrm{LP}}\big|_{\mathrm{NODE}}$

In particular:

$\mathbf{P} = \mathbf{LP}\big|_{\mathrm{NODE}}$ $\qquad$ $\mathbf{NP} = \mathbf{\Sigma}_1^{\mathrm{LP}}\big|_{\mathrm{NODE}}$

The local-polynomial hierarchy



Connection to classical complexity:

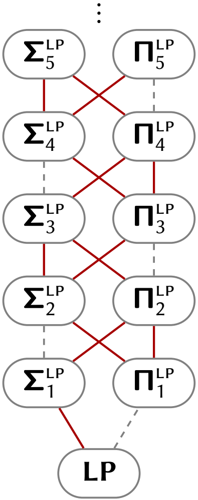
$$\Sigma_\ell^{P} = \Sigma_\ell^{LP}\big|_{\text{NODE}} \qquad \Pi_\ell^{P} = \Pi_\ell^{LP}\big|_{\text{NODE}}$$

In particular:

$$\mathbf{P} = \mathbf{LP}\big|_{\text{NODE}} \qquad \mathbf{NP} = \Sigma_1^{LP}\big|_{\text{NODE}}$$

THEOREM: — Strict inclusions

The local-polynomial hierarchy



Connection to classical complexity:

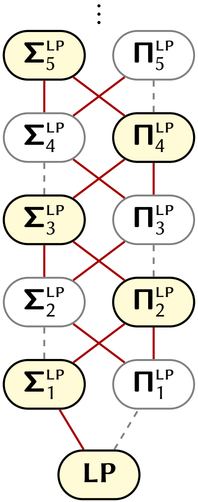
$\boldsymbol{\Sigma}_\ell^{\mathbf{P}} = \boldsymbol{\Sigma}_\ell^{\mathbf{LP}}\big|_{\text{NODE}}$ $\qquad$ $\boldsymbol{\Pi}_\ell^{\mathbf{P}} = \boldsymbol{\Pi}_\ell^{\mathbf{LP}}\big|_{\text{NODE}}$

In particular:

$\mathbf{P} = \mathbf{LP}\big|_{\text{NODE}}$ $\qquad$ $\mathbf{NP} = \boldsymbol{\Sigma}_1^{\mathbf{LP}}\big|_{\text{NODE}}$

THEOREM: — Strict inclusions
--- Equalities iff **P** = **NP**

The local-polynomial hierarchy



Connection to classical complexity:

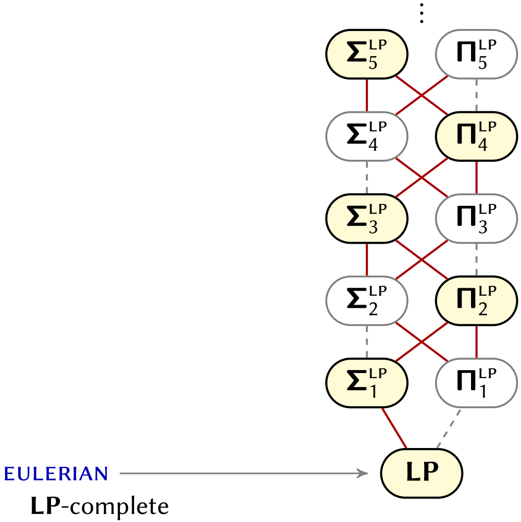
$\Sigma_\ell^{P} = \Sigma_\ell^{LP}|_{\text{NODE}}$ $\qquad$ $\Pi_\ell^{P} = \Pi_\ell^{LP}|_{\text{NODE}}$

In particular:

$\mathbf{P} = \mathbf{LP}|_{\text{NODE}}$ $\qquad$ $\mathbf{NP} = \Sigma_1^{LP}|_{\text{NODE}}$

THEOREM: —— Strict inclusions
- - - Equalities iff $\mathbf{P} = \mathbf{NP}$

The local-polynomial hierarchy



Connection to classical complexity:

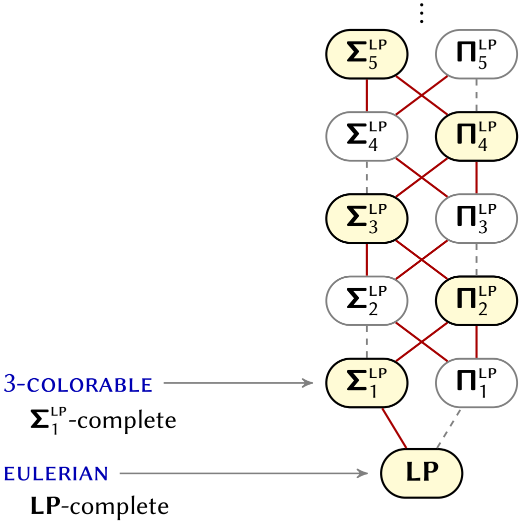
$$\Sigma_\ell^{P} = \Sigma_\ell^{LP}|_{NODE} \qquad \Pi_\ell^{P} = \Pi_\ell^{LP}|_{NODE}$$

In particular:

$$\mathbf{P} = \mathbf{LP}|_{NODE} \qquad \mathbf{NP} = \Sigma_1^{LP}|_{NODE}$$

THEOREM: —— Strict inclusions
- - - Equalities iff **P** = **NP**

The local-polynomial hierarchy



Connection to classical complexity:

$\Sigma_\ell^{P} = \Sigma_\ell^{LP}|_{\text{NODE}}$ $\quad$ $\Pi_\ell^{P} = \Pi_\ell^{LP}|_{\text{NODE}}$

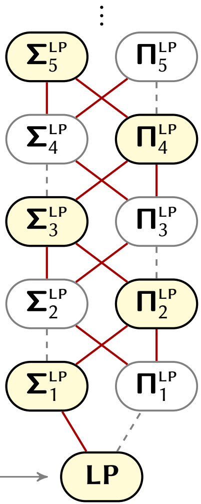
In particular:

$\mathbf{P} = \mathbf{LP}|_{\text{NODE}}$ $\quad$ $\mathbf{NP} = \Sigma_1^{LP}|_{\text{NODE}}$

THEOREM: —— Strict inclusions

- - - Equalities iff **P** = **NP**

The local-polynomial hierarchy



HAMILTONIAN →
LP-hard & **coLP**-hard
In $(\mathbf{\Sigma}_3^{LP} \setminus \mathbf{\Sigma}_1^{LP})$

3-COLORABLE →
$\mathbf{\Sigma}_1^{LP}$-complete

EULERIAN →
LP-complete

Connection to classical complexity:

$$\mathbf{\Sigma}_\ell^{P} = \mathbf{\Sigma}_\ell^{LP}|_{\text{NODE}} \qquad \mathbf{\Pi}_\ell^{P} = \mathbf{\Pi}_\ell^{LP}|_{\text{NODE}}$$

In particular:

$$\mathbf{P} = \mathbf{LP}|_{\text{NODE}} \qquad \mathbf{NP} = \mathbf{\Sigma}_1^{LP}|_{\text{NODE}}$$

THEOREM: — Strict inclusions
--- Equalities iff **P** = **NP**

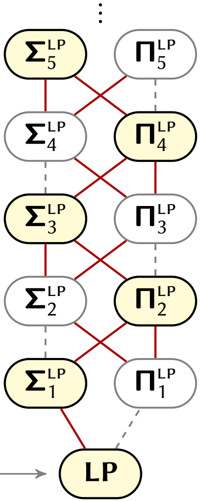
The local-polynomial hierarchy

5-ROUND-3-COLORABLE →
Σ_5^{LP}-complete ?

HAMILTONIAN →
LP-hard & **coLP**-hard
In $(\Sigma_3^{LP} \setminus \Sigma_1^{LP})$

3-COLORABLE →
Σ_1^{LP}-complete

EULERIAN →
LP-complete



Connection to classical complexity:

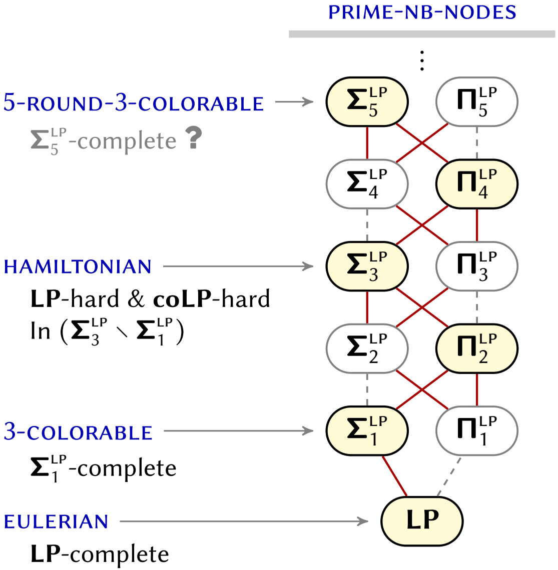
$$\Sigma_\ell^{P} = \Sigma_\ell^{LP}|_{\text{NODE}} \qquad \Pi_\ell^{P} = \Pi_\ell^{LP}|_{\text{NODE}}$$

In particular:

$$\mathbf{P} = \mathbf{LP}|_{\text{NODE}} \qquad \mathbf{NP} = \Sigma_1^{LP}|_{\text{NODE}}$$

THEOREM: —— Strict inclusions
- - - Equalities iff **P** = **NP**

The local-polynomial hierarchy



Connection to classical complexity:

$$\boldsymbol{\Sigma}_\ell^{\mathrm{P}} = \boldsymbol{\Sigma}_\ell^{\mathrm{LP}}|_{\mathrm{NODE}} \qquad \boldsymbol{\Pi}_\ell^{\mathrm{P}} = \boldsymbol{\Pi}_\ell^{\mathrm{LP}}|_{\mathrm{NODE}}$$

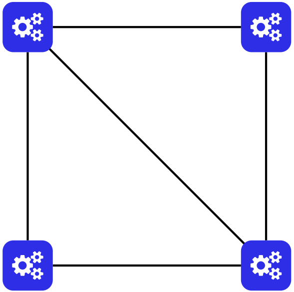
In particular:

$$\mathbf{P} = \mathbf{LP}|_{\mathrm{NODE}} \qquad \mathbf{NP} = \boldsymbol{\Sigma}_1^{\mathrm{LP}}|_{\mathrm{NODE}}$$

THEOREM: —— Strict inclusions
- - - Equalities iff **P** = **NP**

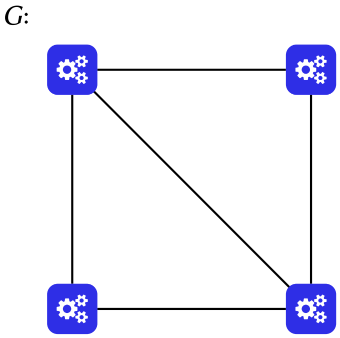
A local-polynomial reduction

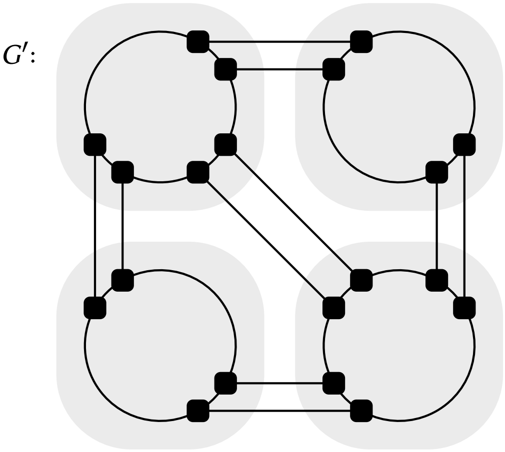
G:



All nodes of G are blue

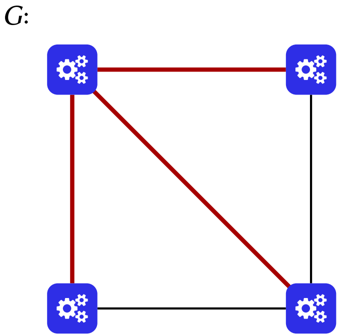
A local-polynomial reduction

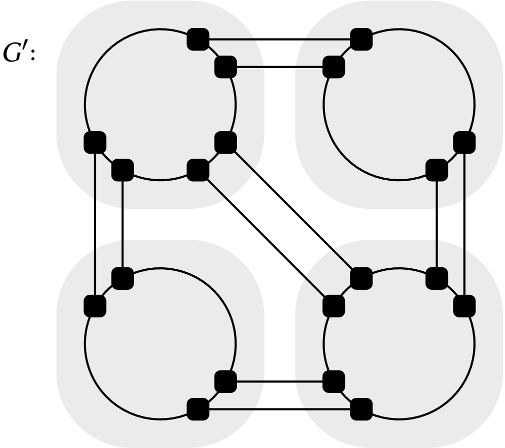




All nodes of G are blue $\iff$ G' contains a Hamiltonian cycle

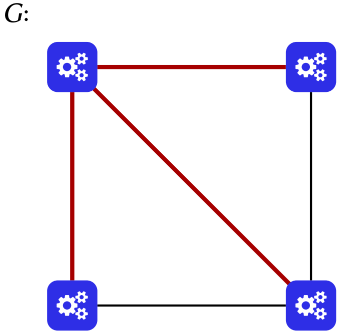
A local-polynomial reduction

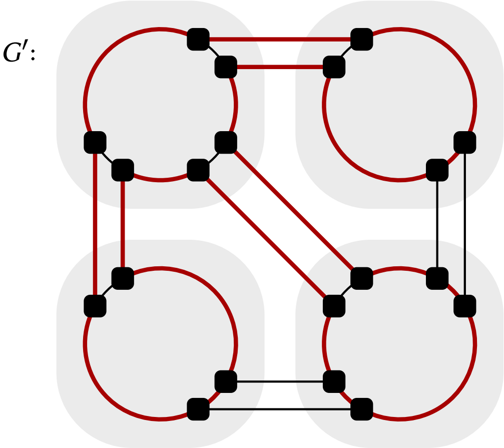




All nodes of G are blue $\iff$ G' contains a Hamiltonian cycle

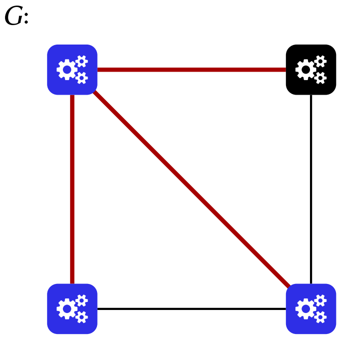
A local-polynomial reduction

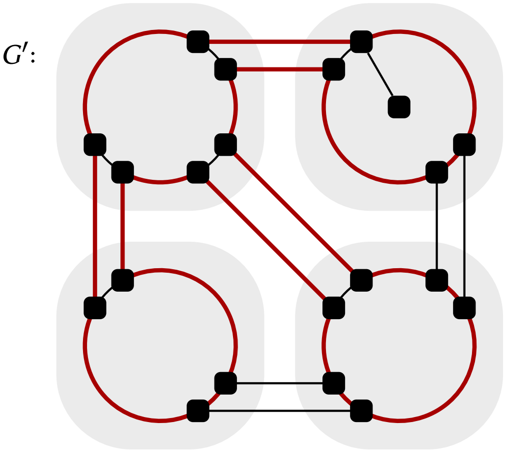




All nodes of G are blue $\iff$ G' contains a Hamiltonian cycle

A local-polynomial reduction





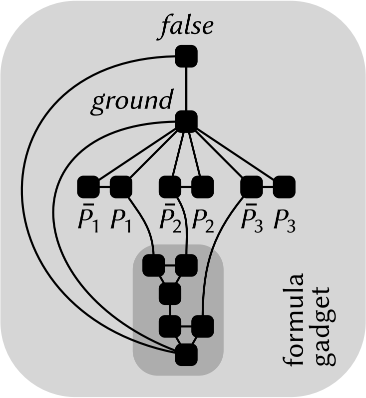
All nodes of G are blue $\iff$ G' contains a Hamiltonian cycle

Extending a classical reduction

G is satisfiable $\iff$ G' is 3-colorable

G: $P_1 \vee \bar{P}_2 \vee \bar{P}_3$

G':

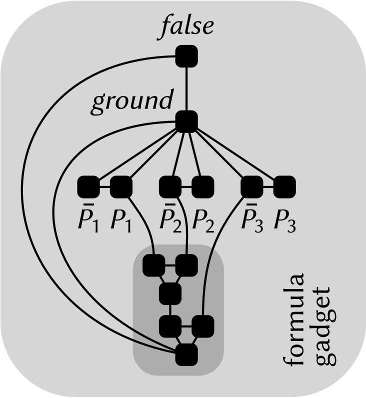


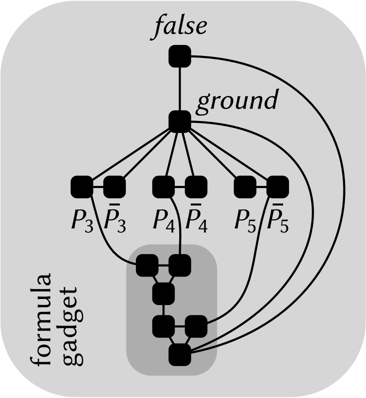
Extending a classical reduction

G is satisfiable $\iff$ G' is 3-colorable

G: $P_1 \vee \bar{P}_2 \vee \bar{P}_3$ $P_3 \vee P_4 \vee \bar{P}_5$

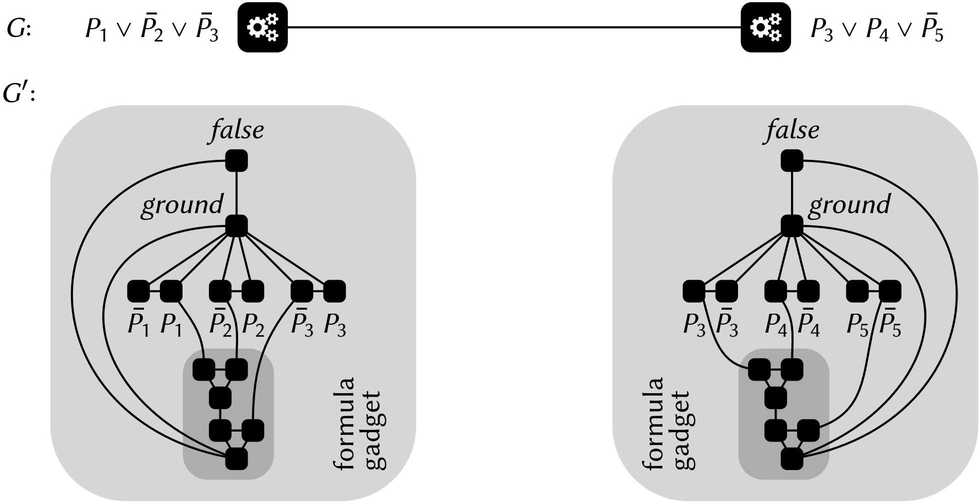
G':





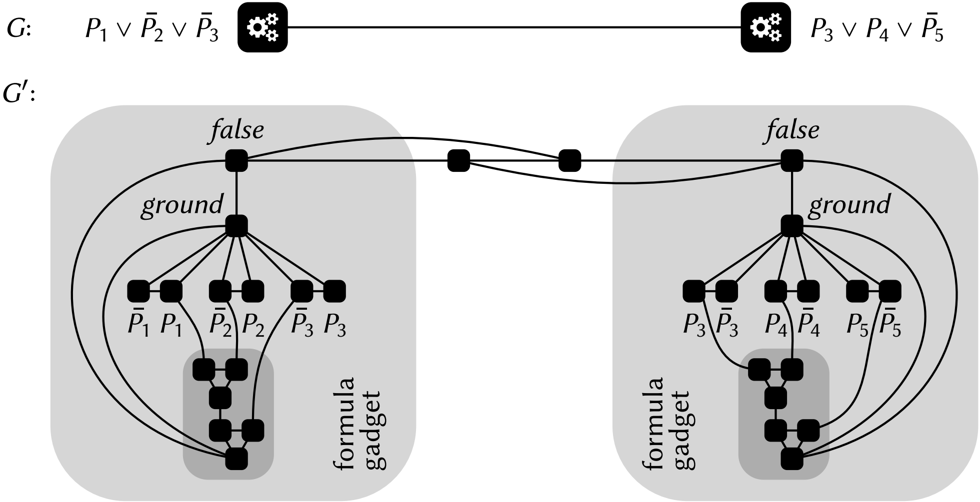
Extending a classical reduction

G is satisfiable $\iff$ G' is 3-colorable



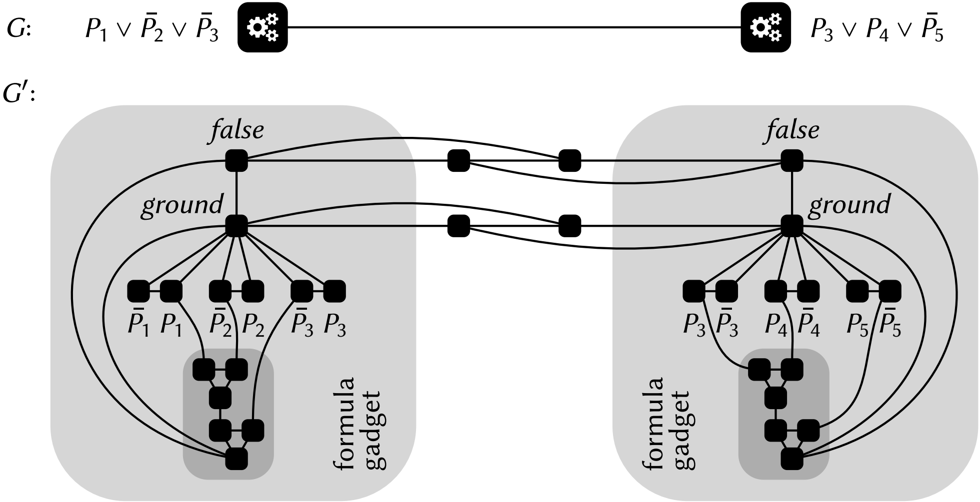
Extending a classical reduction

G is satisfiable $\iff$ G' is 3-colorable



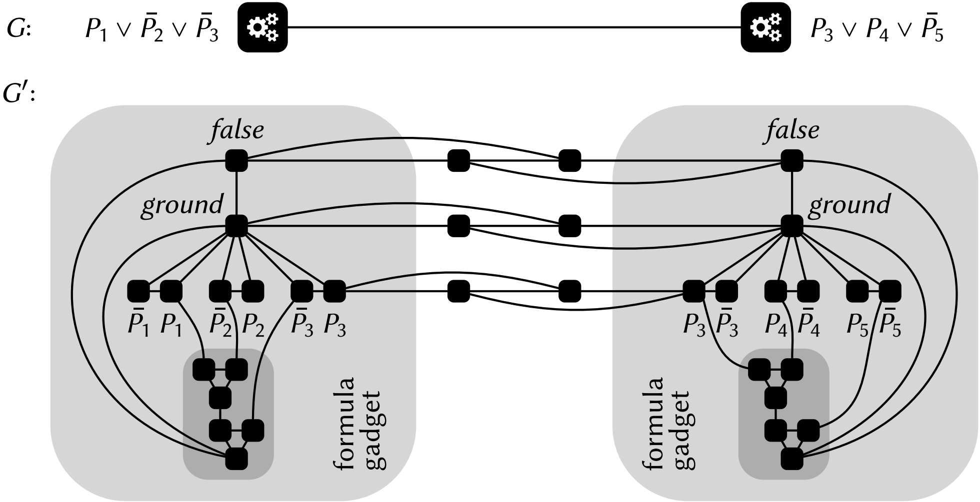
Extending a classical reduction

G is satisfiable $\iff$ G' is 3-colorable



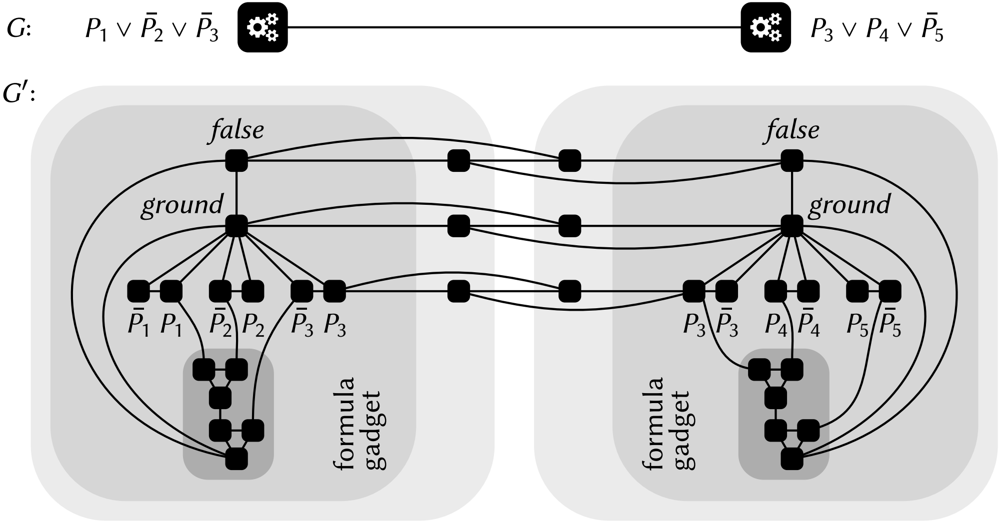
Extending a classical reduction

G is satisfiable $\iff$ G' is 3-colorable



Extending a classical reduction

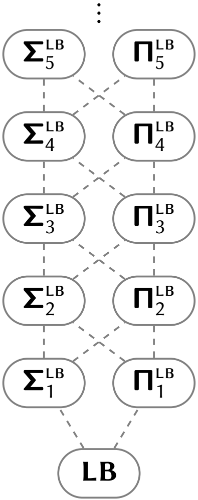
G is satisfiable $\iff$ G' is 3-colorable



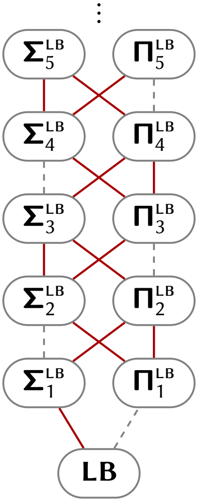
The local-*bounded* hierarchy

LP-hierarchy ⟶ **LB**-hierarchy

polynomial bounds   arbitrary **b**ounds



The local-*bounded* hierarchy

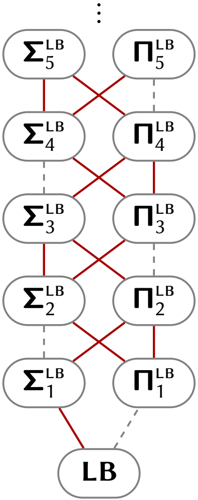


LP-hierarchy ⟶ **LB**-hierarchy

polynomial bounds   arbitrary **b**ounds

THEOREM: — Strict inclusions

The local-*bounded* hierarchy



LP-hierarchy ⟶ **LB**-hierarchy

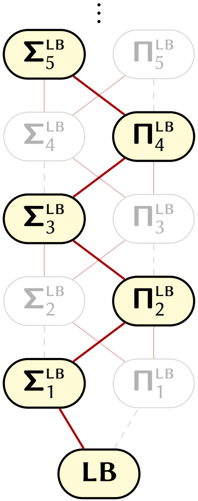
polynomial bounds — arbitrary **b**ounds

THEOREM: —— Strict inclusions
- - - Equalities

The local-*bounded* hierarchy

LP-hierarchy ⟶ **LB**-hierarchy

polynomial bounds — arbitrary **b**ounds



THEOREM: — Strict inclusions
- - - Equalities

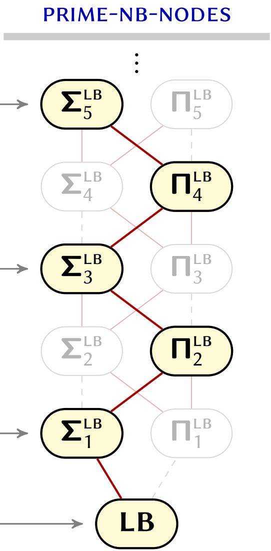
The local-*bounded* hierarchy

5-ROUND-3-COLORABLE →
Σ_5^{LB}-complete ?

HAMILTONIAN →
LB-hard & **coLB**-hard
In $(\Sigma_3^{\mathsf{LB}} \setminus \Sigma_1^{\mathsf{LB}})$

3-COLORABLE →
Σ_1^{LB}-complete

EULERIAN →
LB-complete



LP-hierarchy ⟶ **LB**-hierarchy

polynomial bounds — arbitrary **b**ounds

THEOREM: — Strict inclusions
- - - Equalities

The local-*bounded* hierarchy

LP-hierarchy ⟶ **LB**-hierarchy

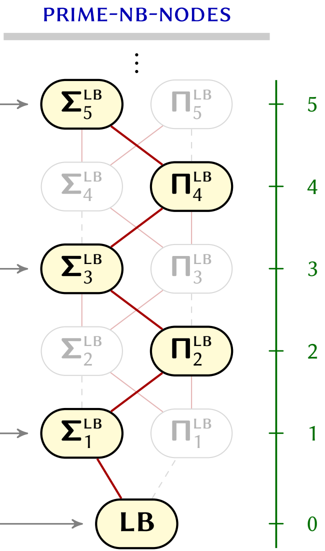
polynomial bounds ⟶ arbitrary **b**ounds

5-ROUND-3-COLORABLE →
Σ_5^{LB}-complete ?

HAMILTONIAN →
LB-hard & **coLB**-hard
In $(\Sigma_3^{LB} \setminus \Sigma_1^{LB})$

3-COLORABLE →
Σ_1^{LB}-complete

EULERIAN →
LB-complete



THEOREM: — Strict inclusions
--- Equalities

A measure of locality?

The local-*bounded* hierarchy

LP-hierarchy ⟶ **LB**-hierarchy

polynomial bounds — arbitrary **b**ounds

5-ROUND-3-COLORABLE

Σ_5^{LB}-complete ?

HAMILTONIAN

LB-hard & **coLB**-hard

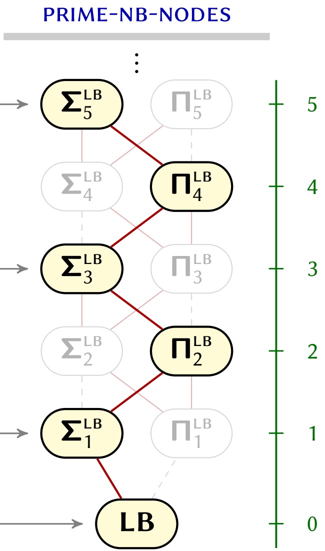
In $(\Sigma_3^{LB} \setminus \Sigma_1^{LB})$

3-COLORABLE

Σ_1^{LB}-complete

EULERIAN

LB-complete



THEOREM: — Strict inclusions - - - Equalities

We lose one thing:
Fagin's theorem

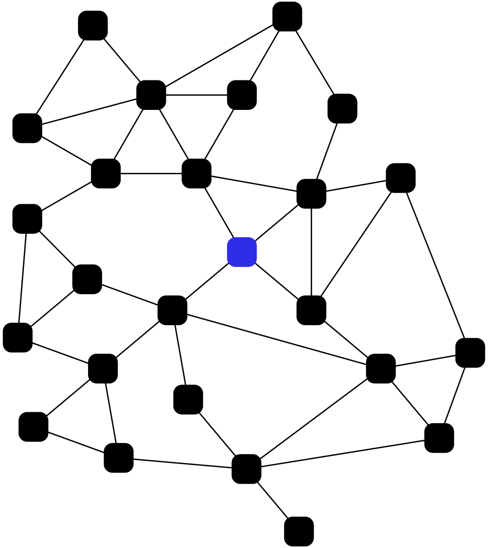
A measure of locality?

Building a theory

Image source: www.rhb.ch

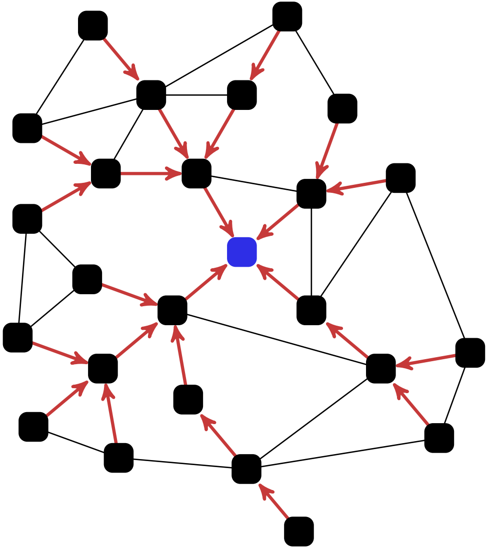
Building a theory

SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** ■:

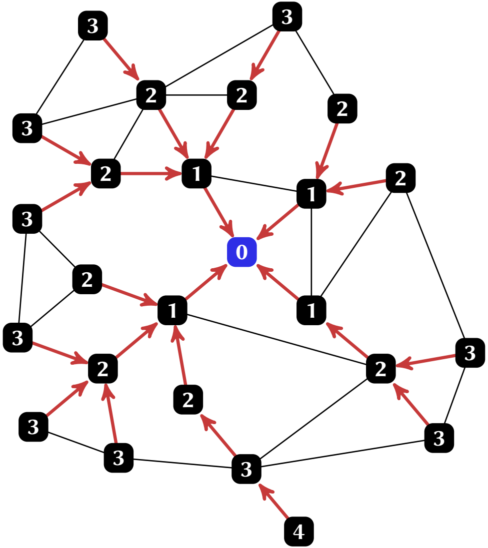
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.

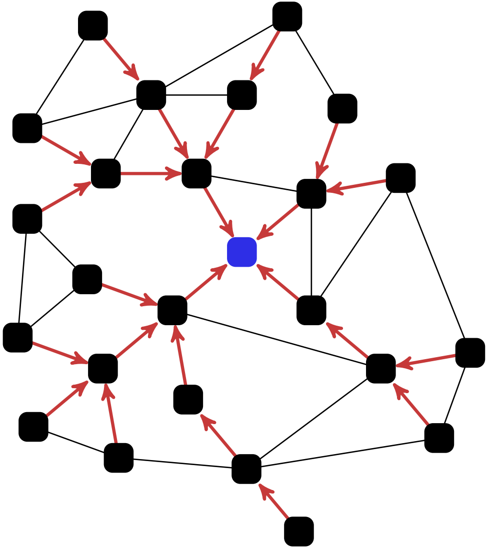
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.

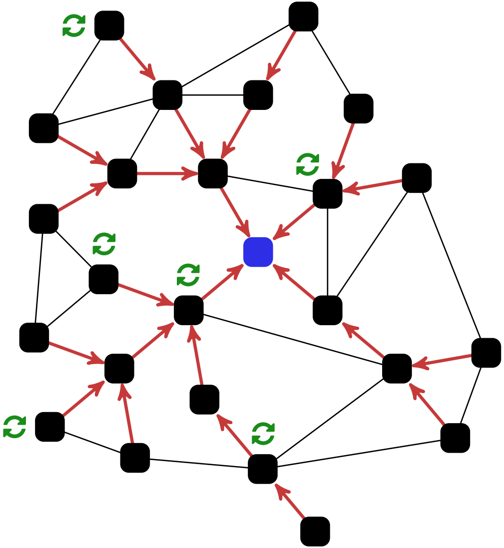
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.

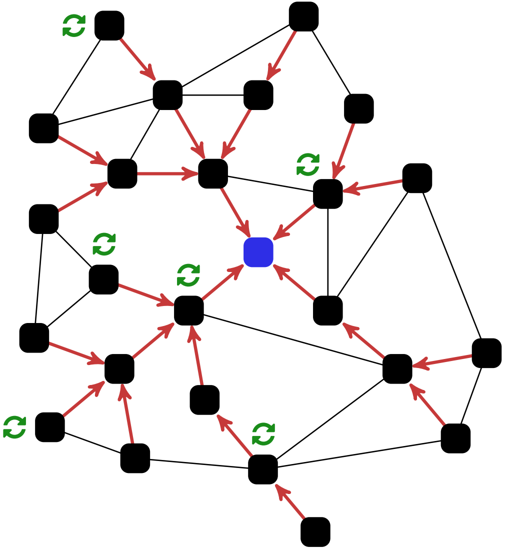
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.

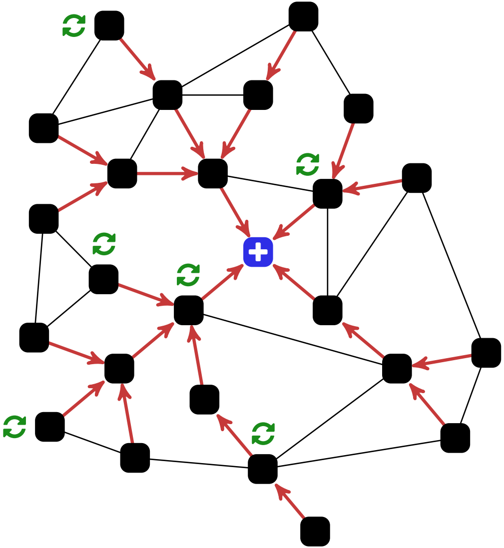
SOME-NODE-BLUE $\in \Sigma_3^{LB}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

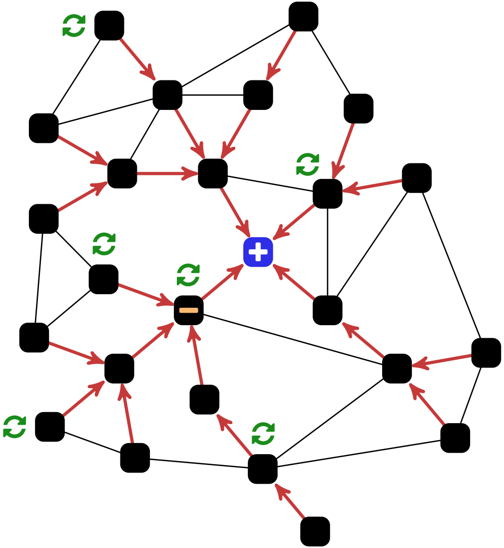
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

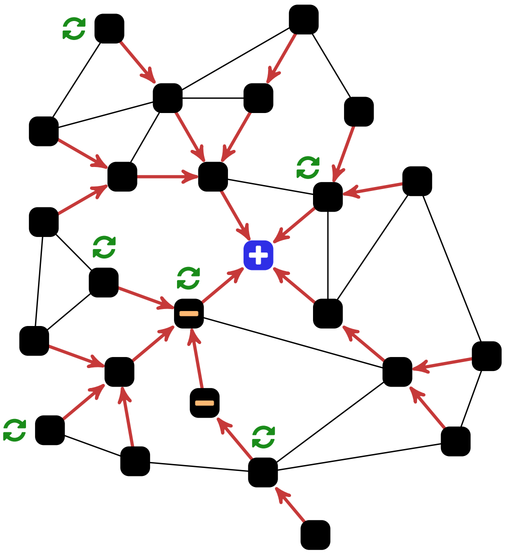
SOME-NODE-BLUE $\in \Sigma_3^{\text{LB}}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ⟳.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

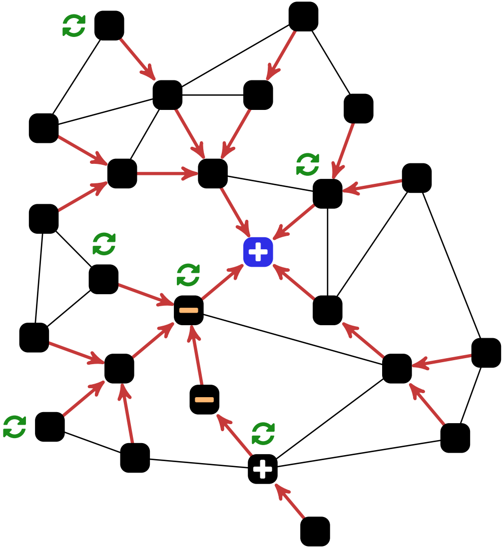
SOME-NODE-BLUE $\in \Sigma_3^{LB}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

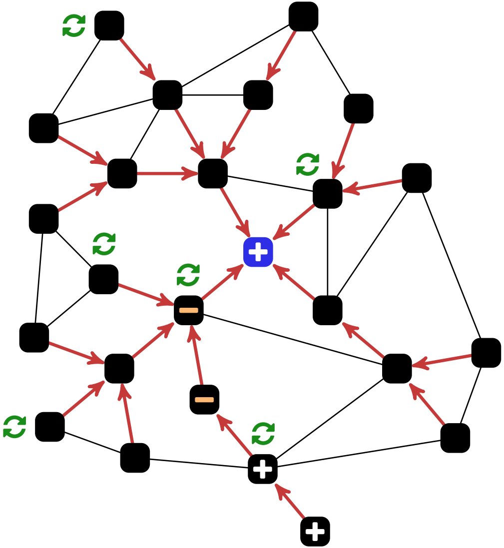
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node**:

1. Eve chooses a **spanning tree** rooted at .
2. Adam chooses a set of **flipping nodes**.
3. Eve **charges nodes** either + or − so that:
   - is charged +.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

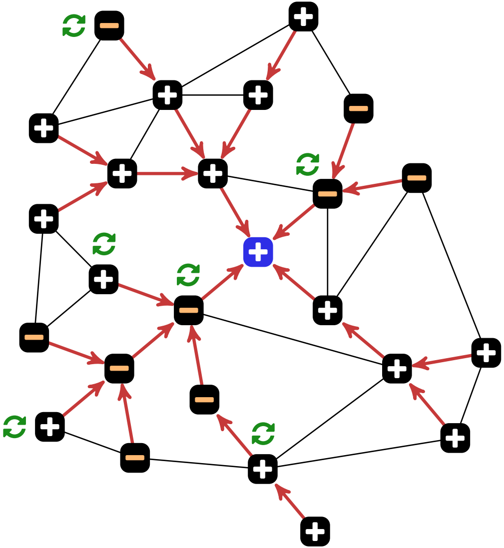
SOME-NODE-BLUE $\in \Sigma_3^{LB}$



To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

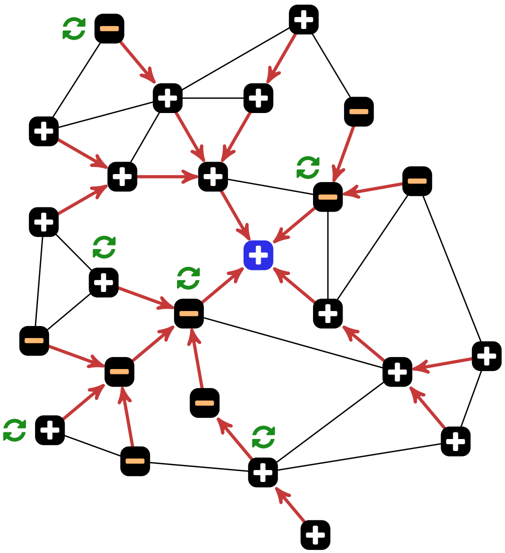
SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$



To prove the existence of a **blue node** :

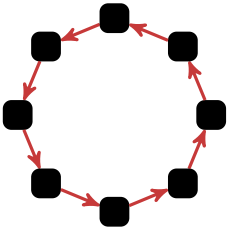
1. Eve chooses a **spanning tree** ↑ rooted at .
2. Adam chooses a set of **flipping nodes** .
3. Eve **charges nodes** either **+** or **−** so that:
   - is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

SOME-NODE-BLUE $\in \Sigma_3^{\mathrm{LB}}$

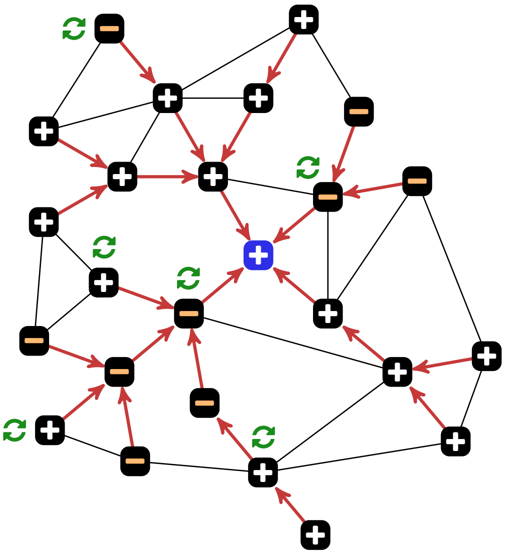


To prove the existence of a **blue node**:

1. Eve chooses a **spanning tree** ↑ rooted at .
2. Adam chooses a set of **flipping nodes**.
3. Eve **charges nodes** either **+** or **−** so that:
   - is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

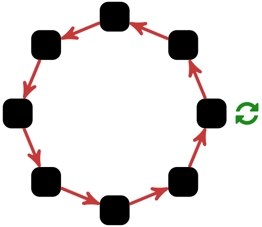


SOME-NODE-BLUE $\in \Sigma_3^{LB}$

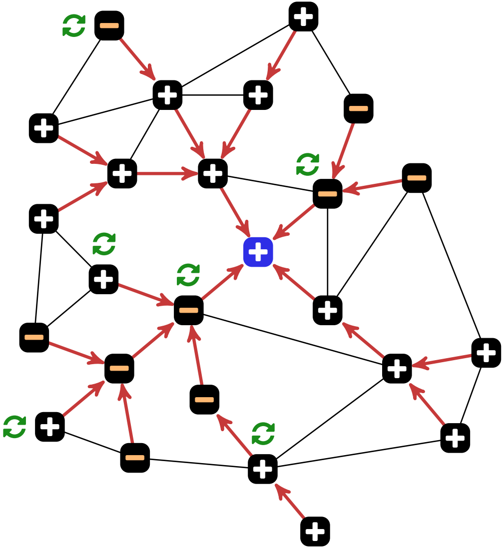


To prove the existence of a **blue node**:

1. Eve chooses a **spanning tree** ↑ rooted at .
2. Adam chooses a set of **flipping nodes**.
3. Eve **charges nodes** either **+** or − so that:
   - is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

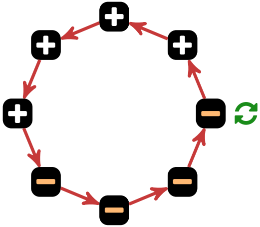


SOME-NODE-BLUE $\in \Sigma_3^{LB}$

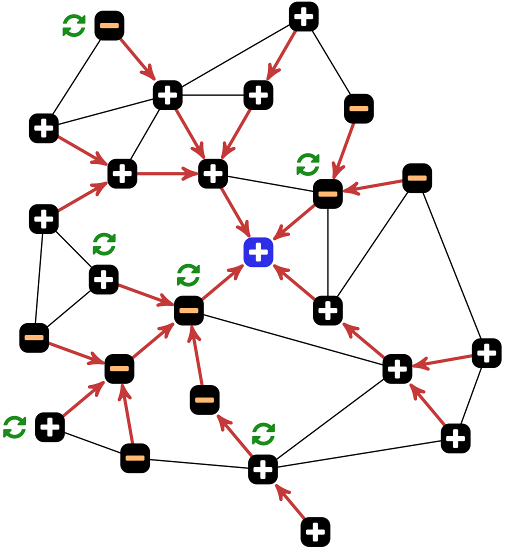


To prove the existence of a **blue node** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

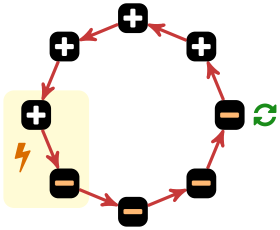


SOME-NODE-BLUE $\in \Sigma_3^{LB}$

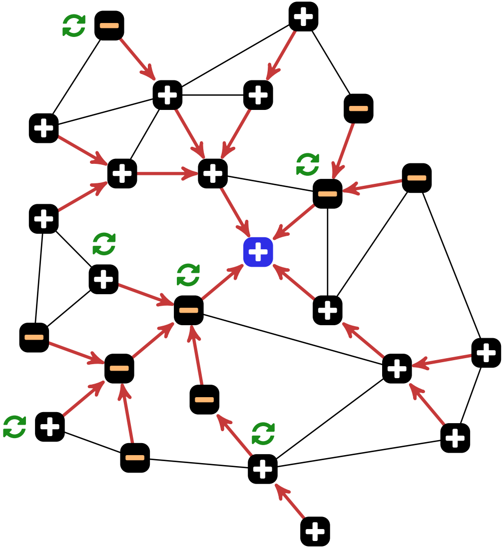


To prove the existence of a **blue node**:

1. Eve chooses a **spanning tree** rooted at .
2. Adam chooses a set of **flipping nodes**.
3. Eve **charges nodes** either + or − so that:
   - is charged +.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.



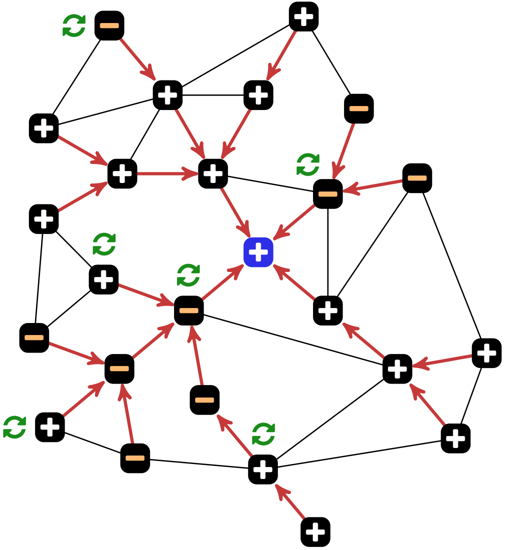
UNIQUE-BLUE-NODE ∈ Σ_3^{LB}



To prove that there is **exactly one** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

UNIQUE-BLUE-NODE $\in \Sigma_3^{\mathrm{LB}}$

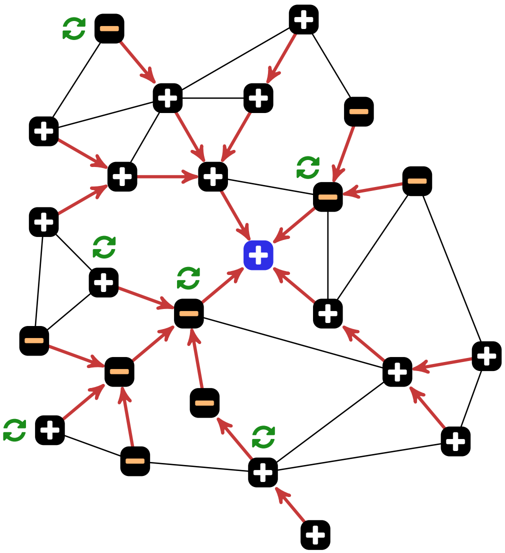


To prove that there is **exactly one** 🟦:

1. Eve chooses a **spanning tree** ↑ rooted at 🟦.
2. Adam chooses a set of **flipping nodes** 🔄.
3. Eve **charges nodes** either **+** or **−** so that:
   - 🟦 is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

Eve tells each node if 🟦 is a flipping node.

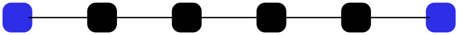
UNIQUE-BLUE-NODE $\in \Sigma_3^{\mathrm{LB}}$



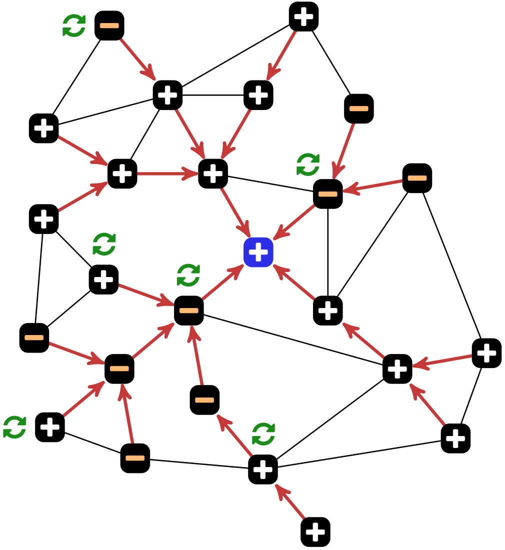
To prove that there is **exactly one** :

1. Eve chooses a **spanning tree** ↑ rooted at .
2. Adam chooses a set of **flipping nodes** .
3. Eve **charges nodes** either **+** or **−** so that:
   - is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

Eve tells each node if is a flipping node.



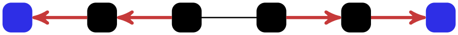
UNIQUE-BLUE-NODE $\in \Sigma_3^{\mathsf{LB}}$



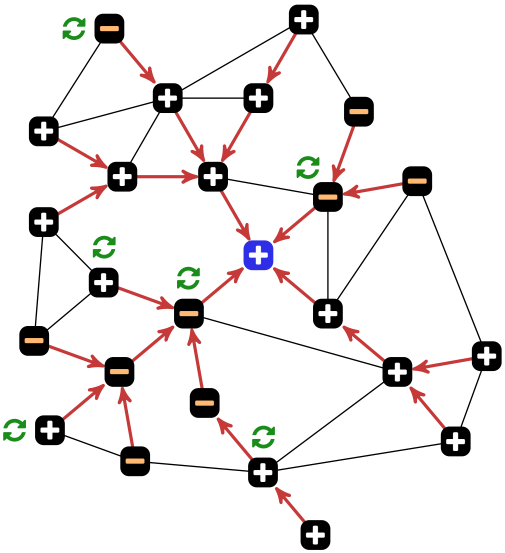
To prove that there is **exactly one** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

Eve tells each node if ■ is a flipping node.



UNIQUE-BLUE-NODE $\in \Sigma_3^{\mathrm{LB}}$

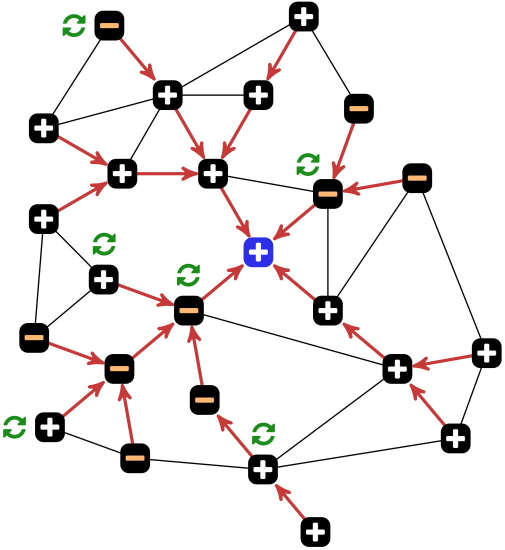


To prove that there is **exactly one** 🟦:

1. Eve chooses a **spanning tree** ↑ rooted at 🟦.
2. Adam chooses a set of **flipping nodes** 🔄.
3. Eve **charges nodes** either **+** or **−** so that:
   - 🟦 is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

Eve tells each node if 🟦 is a flipping node.

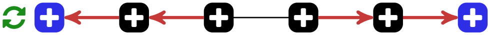
UNIQUE-BLUE-NODE ∈ Σ_3^{LB}



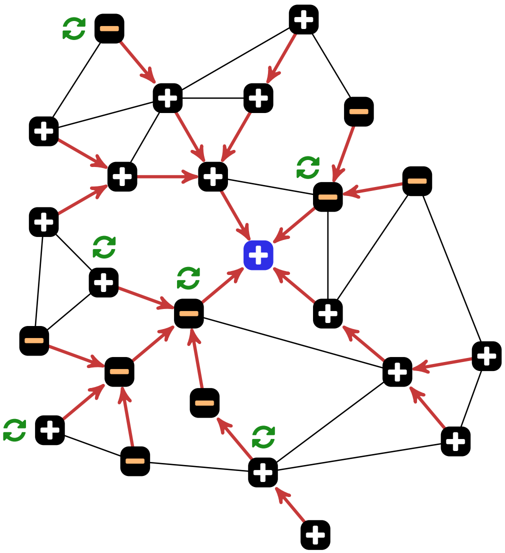
To prove that there is **exactly one** ■:

1. Eve chooses a **spanning tree** ↑ rooted at ■.
2. Adam chooses a set of **flipping nodes** ↻.
3. Eve **charges nodes** either **+** or **−** so that:
   - ■ is charged **+**.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

Eve tells each node if ■ is a flipping node.



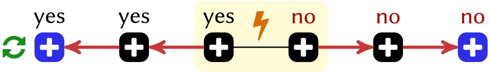
UNIQUE-BLUE-NODE $\in \Sigma_3^{\mathrm{LB}}$



To prove that there is **exactly one** :

1. Eve chooses a **spanning tree** ↑ rooted at .
2. Adam chooses a set of **flipping nodes** 🔄.
3. Eve **charges nodes** either + or − so that:
   - is charged +.
   - Normal nodes inherit their parent's charge.
   - Flipping nodes receive the opposite charge.

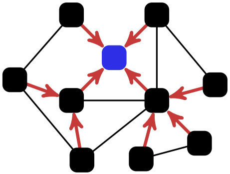
Eve tells each node if is a flipping node.



Properties in Σ_3^{LB}

*With a **spanning tree**, Eve can prove many things:*

UNIQUE-BLUE-NODE

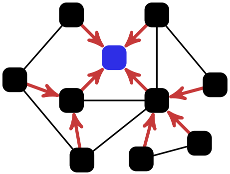


Examples adapted from Göös & Suomela (PODC 2011): *Locally Checkable Proofs*

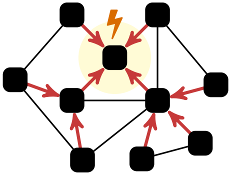
Properties in Σ_3^{LB}

*With a **spanning tree**, Eve can prove many things:*

UNIQUE-BLUE-NODE



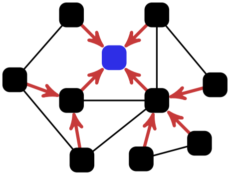
Any property in **coLB**



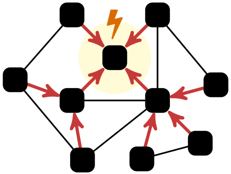
Properties in Σ_3^{LB}

*With a **spanning tree**, Eve can prove many things:*

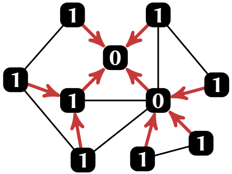
UNIQUE-BLUE-NODE



Any property in **coLB**



EVEN-NB-NODES

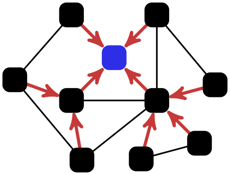


Examples adapted from Göös & Suomela (PODC 2011): *Locally Checkable Proofs*

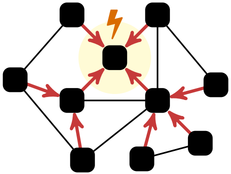
Properties in Σ_3^{LB}

With a ***spanning tree****, Eve can prove many things:*

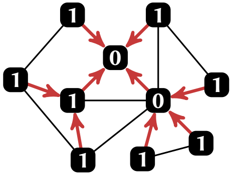
UNIQUE-BLUE-NODE



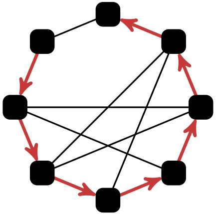
Any property in **coLB**



EVEN-NB-NODES



HAMILTONIAN

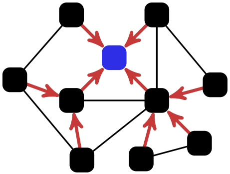


Examples adapted from Göös & Suomela (PODC 2011): *Locally Checkable Proofs*

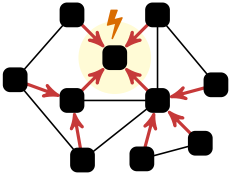
Properties in Σ_3^{LB}

*With a **spanning tree**, Eve can prove many things:*

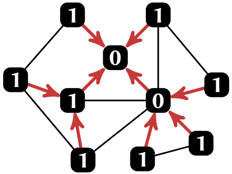
UNIQUE-BLUE-NODE



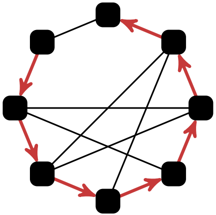
Any property in **coLB**



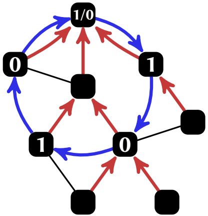
EVEN-NB-NODES



HAMILTONIAN



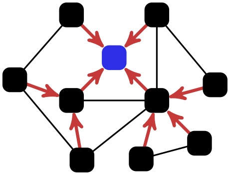
NON-2-COLORABLE



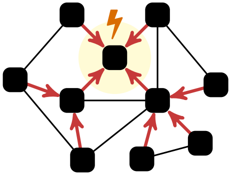
Properties in Σ_3^{LB}

*With a **spanning tree**, Eve can prove many things:*

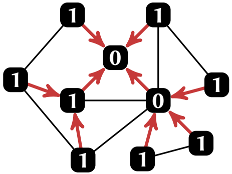
UNIQUE-BLUE-NODE



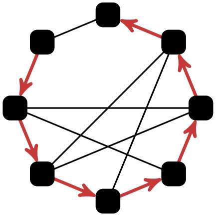
Any property in **coLB**



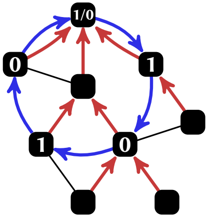
EVEN-NB-NODES



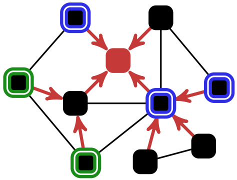
HAMILTONIAN



NON-2-COLORABLE



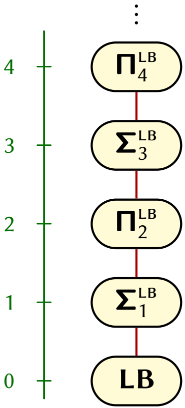
EMSO-definable properties



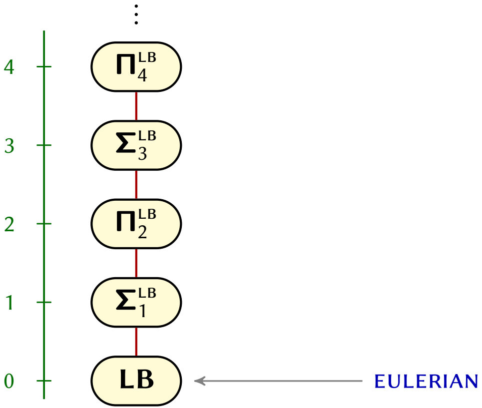
$\exists S_1, \ldots, S_k \, \exists x \, \forall y \, \varphi$

Examples adapted from Göös & Suomela (PODC 2011): *Locally Checkable Proofs*

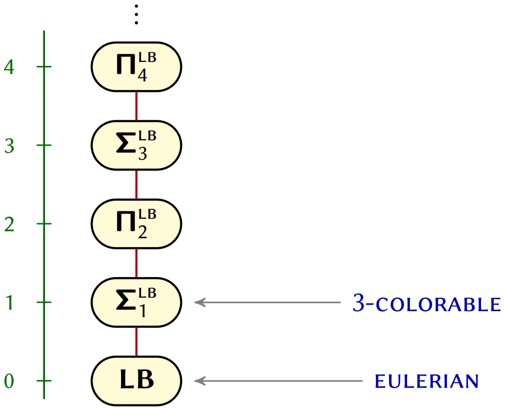
Measuring locality?



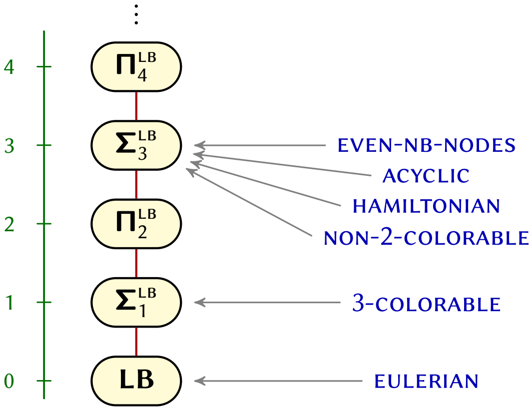
Measuring locality?



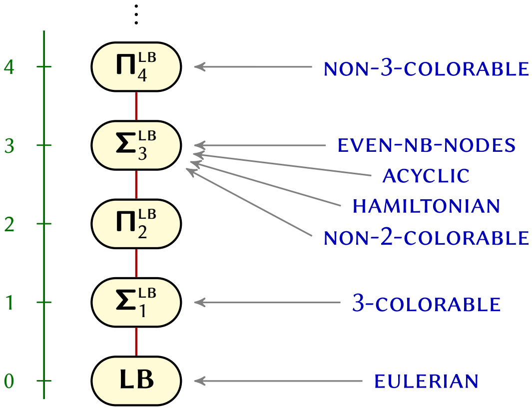
Measuring locality?



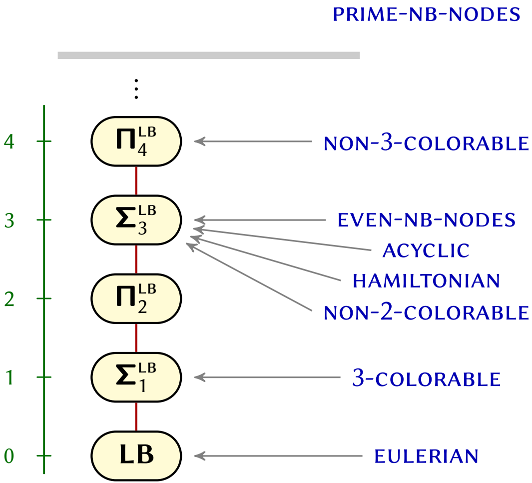
Measuring locality?



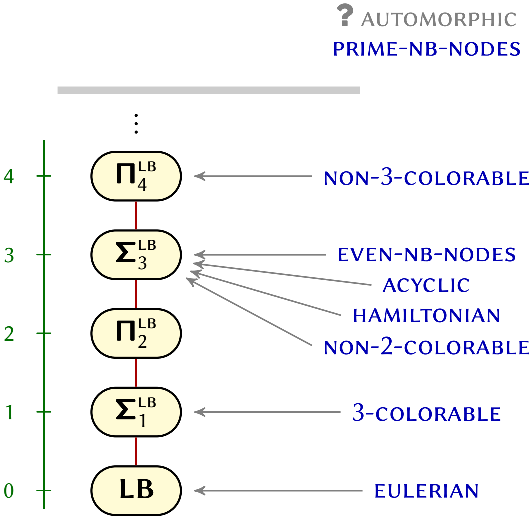
Measuring locality?



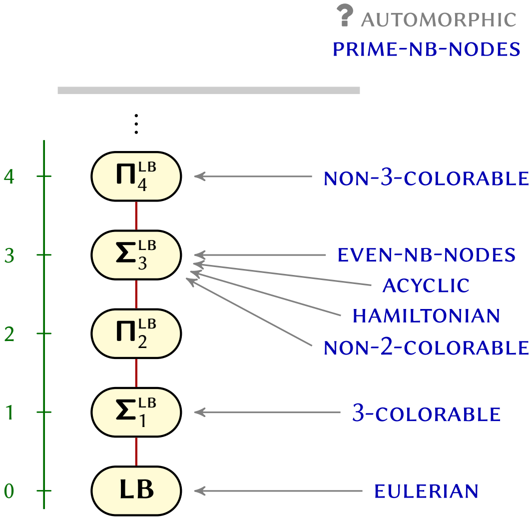
Measuring locality?



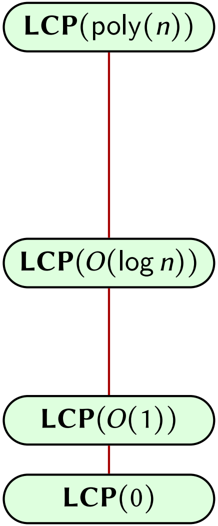
Measuring locality?



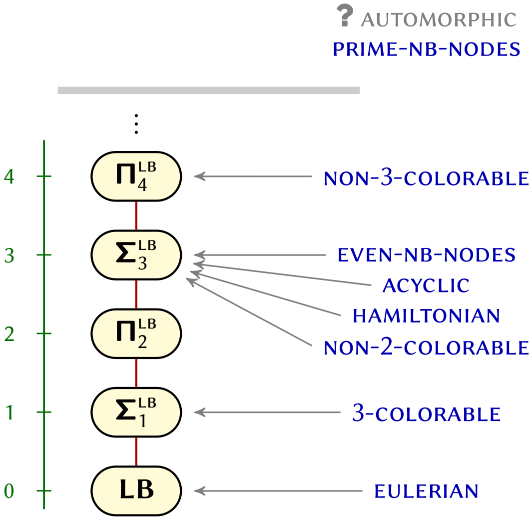
Measuring locality?



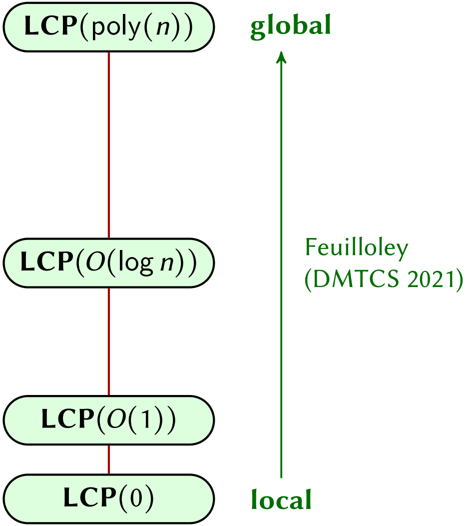
Locally Checkable Proofs
Göös, Suomela (PODC 2011)



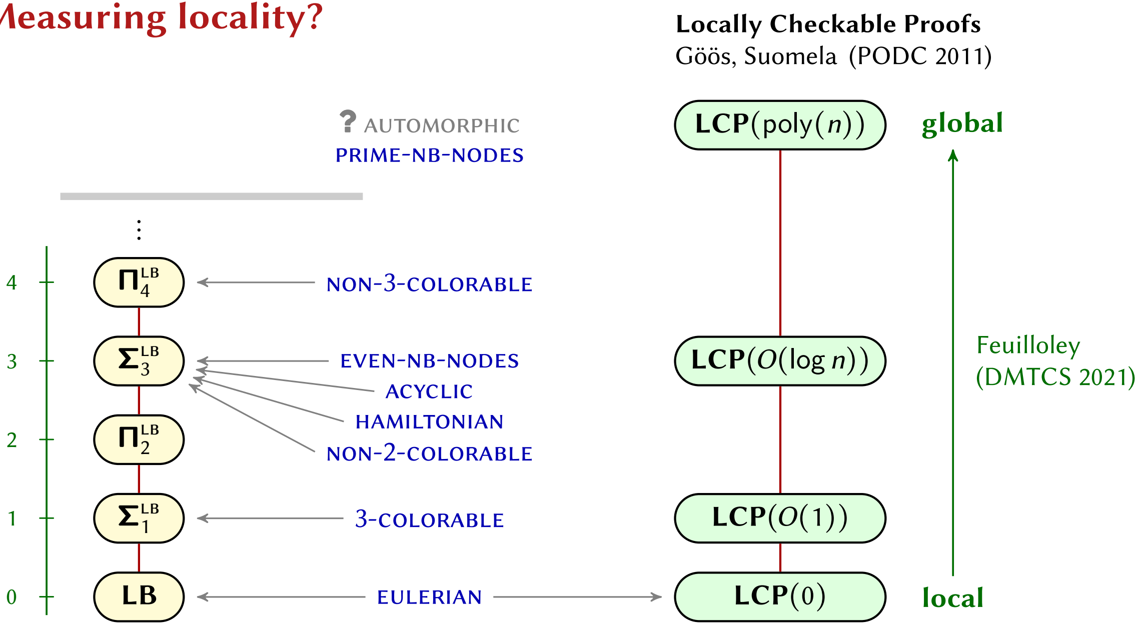
Measuring locality?



Locally Checkable Proofs
Göös, Suomela (PODC 2011)

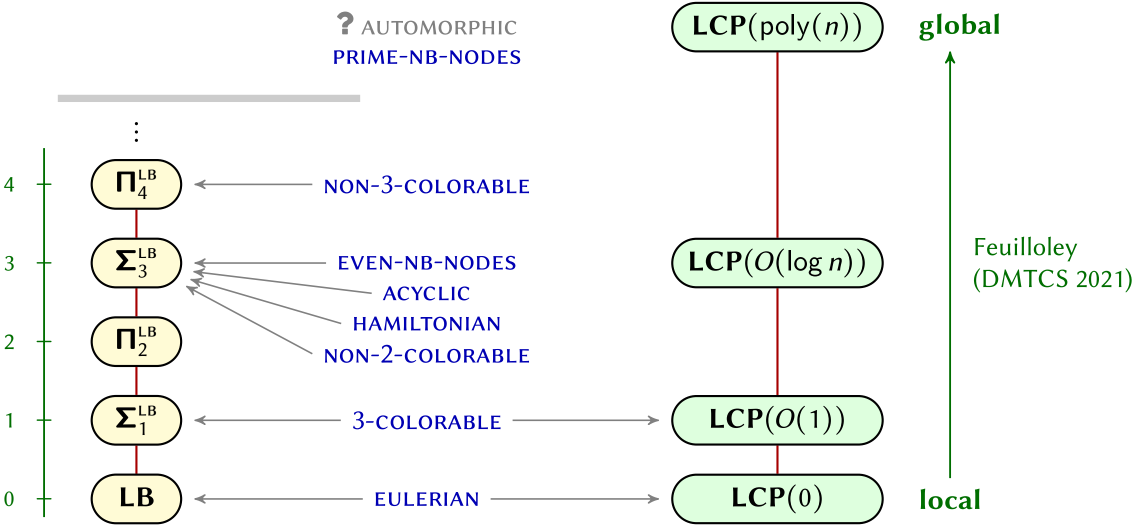


Measuring locality?



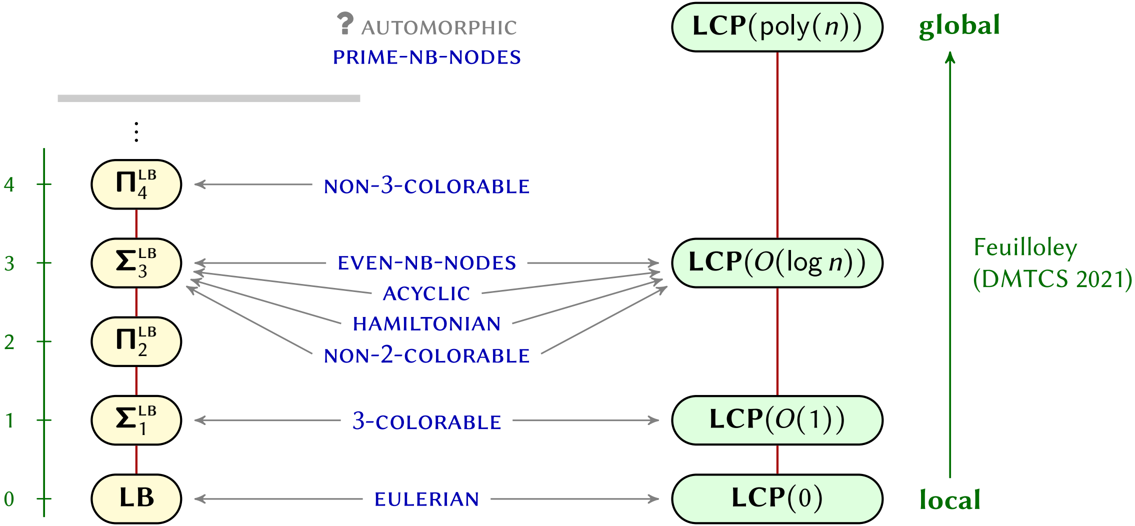
Measuring locality?

Locally Checkable Proofs
Göös, Suomela (PODC 2011)

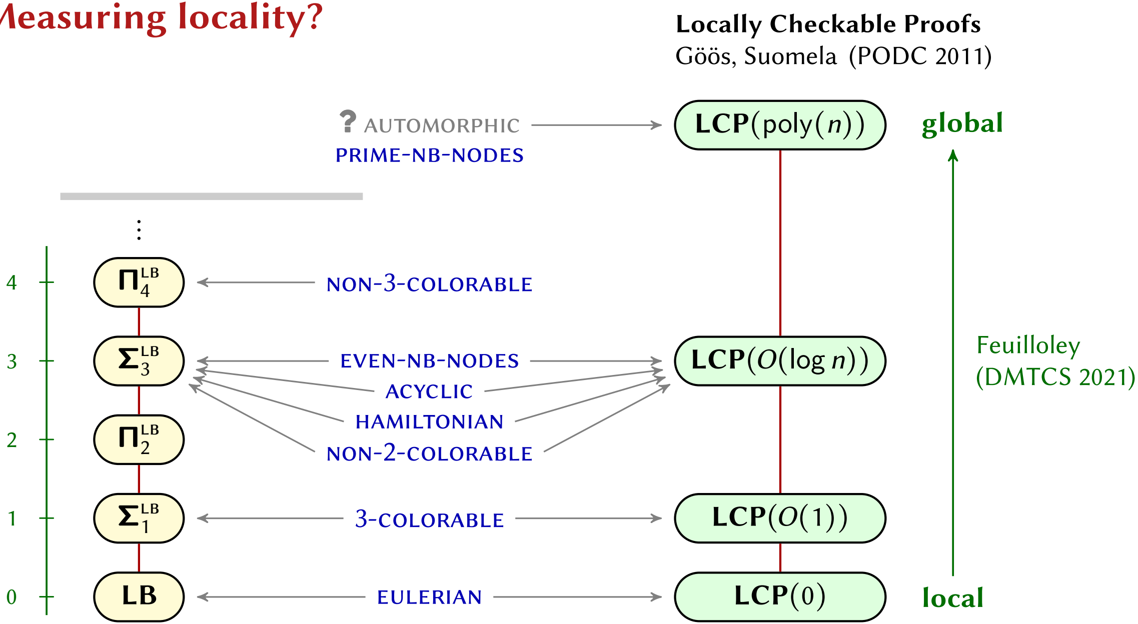


Measuring locality?

Locally Checkable Proofs
Göös, Suomela (PODC 2011)

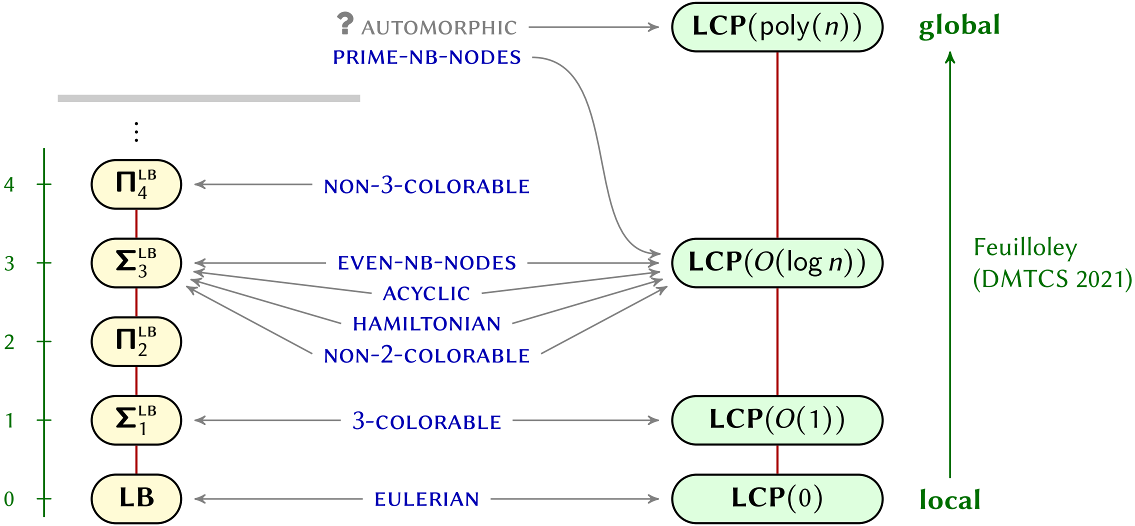


Measuring locality?



Measuring locality?

Locally Checkable Proofs
Göös, Suomela (PODC 2011)



Measuring locality?