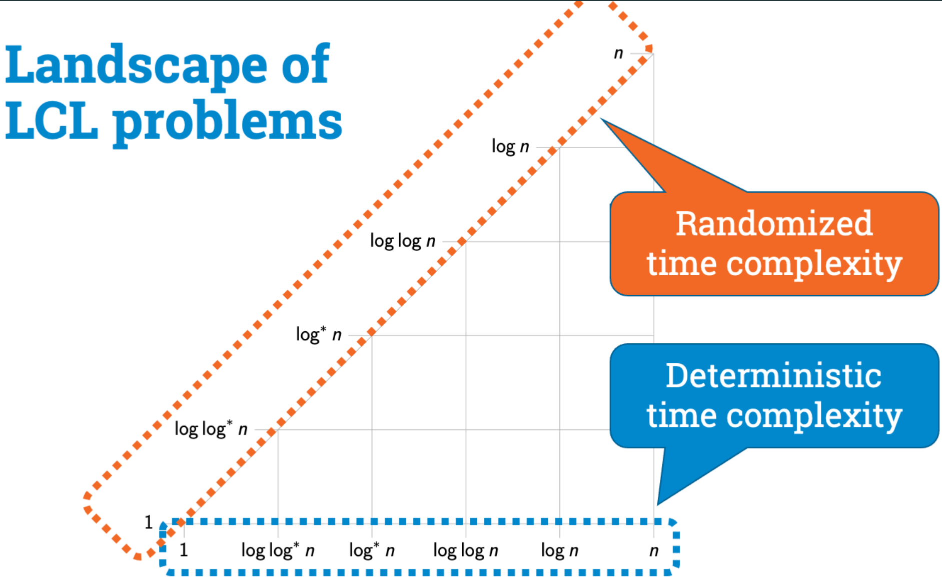
Distributed Computing

10 - Complexity Classes

Mikaël Rabie

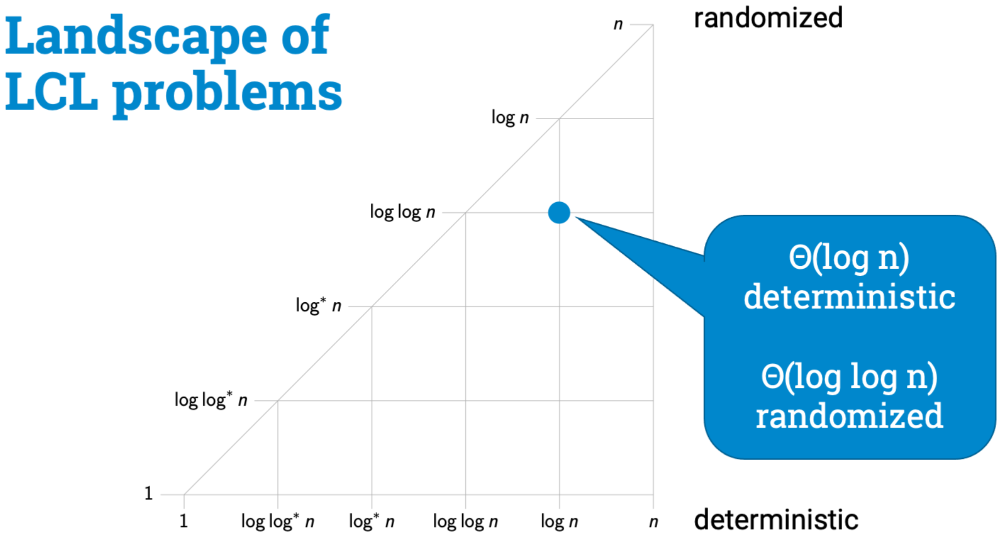
Université Paris-Cité, IRIF

Complexity Landscapes



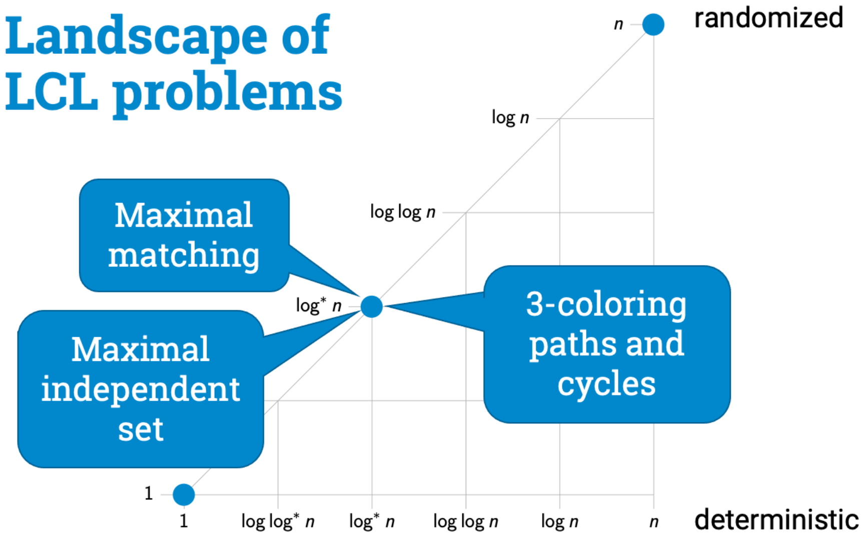
Figures from https ://jukkasuomela.fi/landscape-of-locality/

Complexities on Graphs with the LOCAL Model



Figures from https ://jukkasuomela.fi/landscape-of-locality/

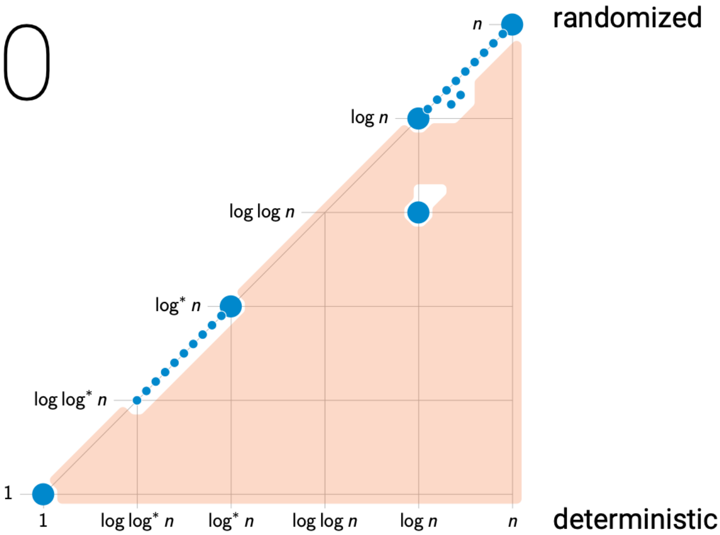
Complexities on Graphs with the LOCAL Model



Figures from https ://jukkasuomela.fi/landscape-of-locality/

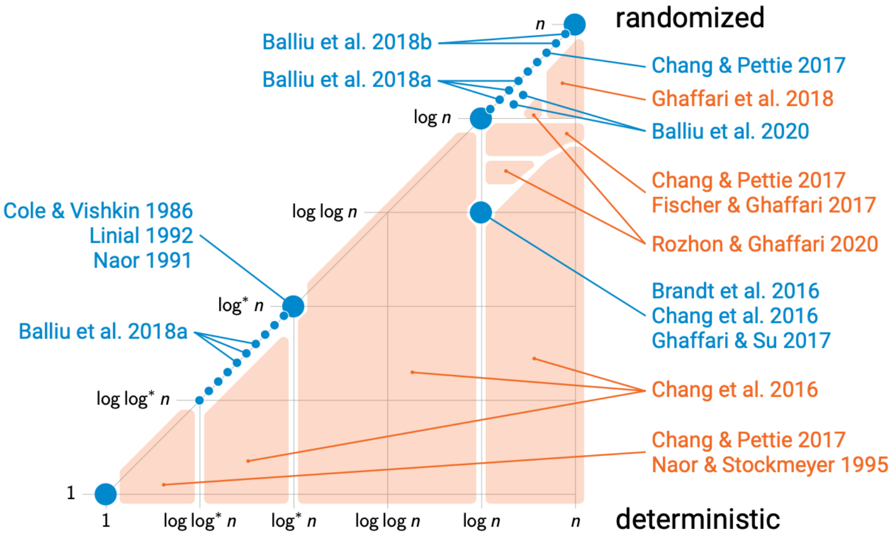
Complexities on Graphs with the LOCAL Model

2020



Figures from https ://jukkasuomela.fi/landscape-of-locality/

Complexities on Graphs with the LOCAL Model



Figures from https ://jukkasuomela.fi/landscape-of-locality/

Path and Cycle Complexities

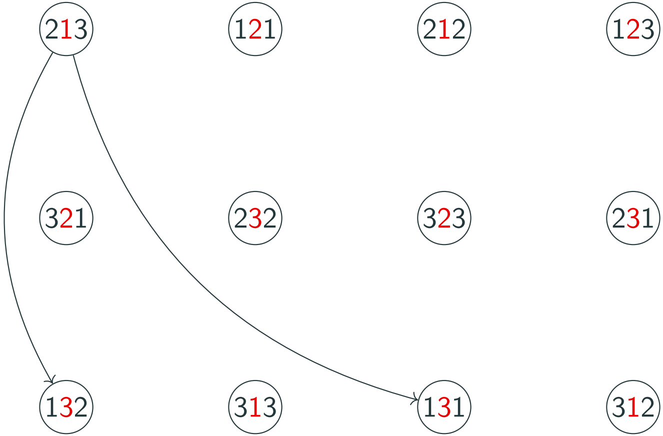
3-coloring neighborhoods

213	121	212	123
321	232	323	231
132	313	131	312

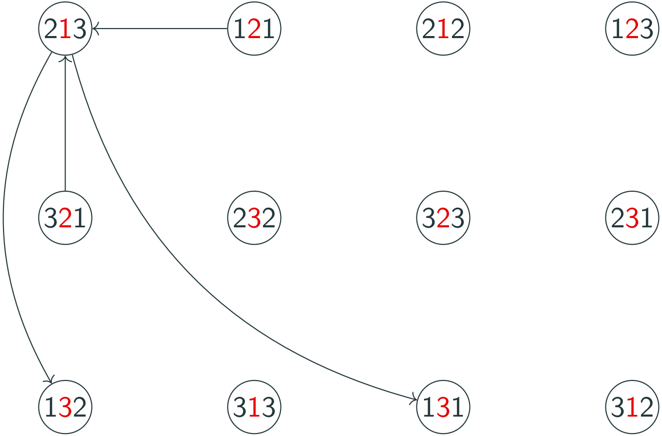
3-coloring neighborhoods

213	121	212	123
321	232	323	231
132	313	131	312

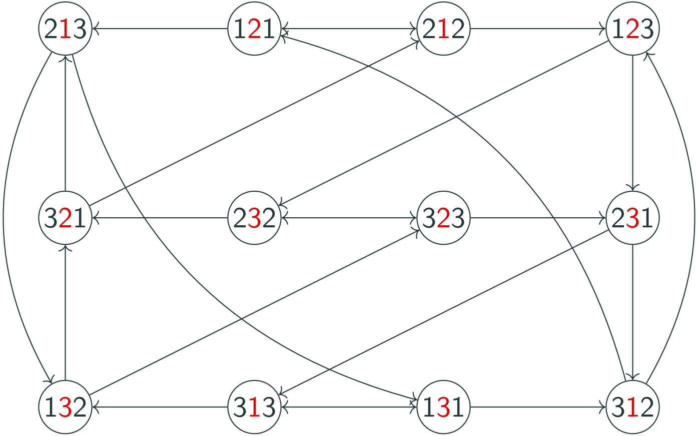
3-coloring neighborhoods



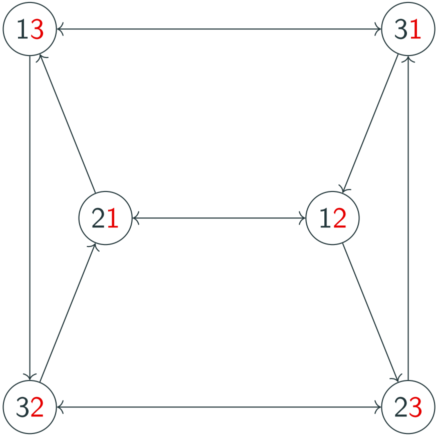
3-coloring neighborhoods



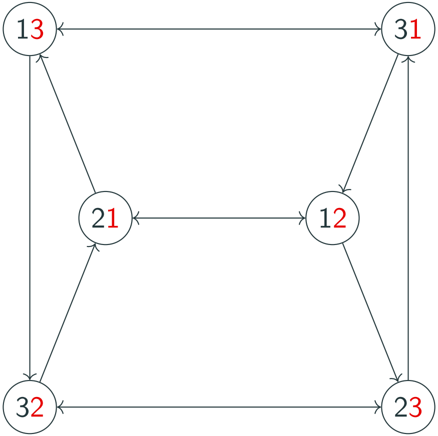
3-coloring neighborhoods

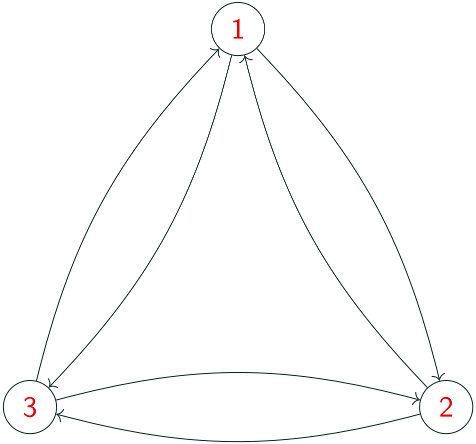


3-coloring neighborhoods

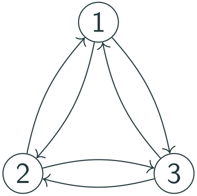


3-coloring neighborhoods



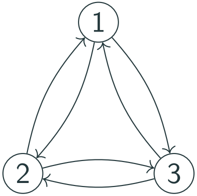


3-coloring



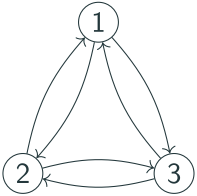
Transition Automata

3-coloring Independent Set



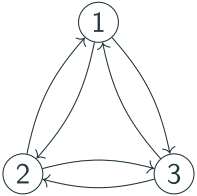
Transition Automata

3-coloring Independent Set



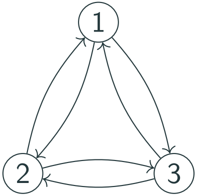
Transition Automata

3-coloring　　Independent Set　　MIS



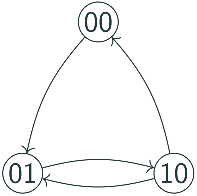
Transition Automata

3-coloring



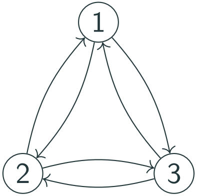
Independent Set

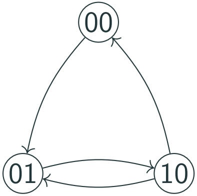
MIS



Transition Automata

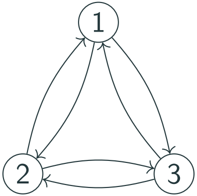
3-coloring | Independent Set | MIS | 2-coloring





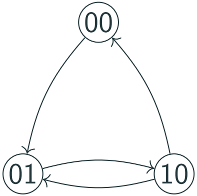
Transition Automata

3-coloring



Independent Set

MIS

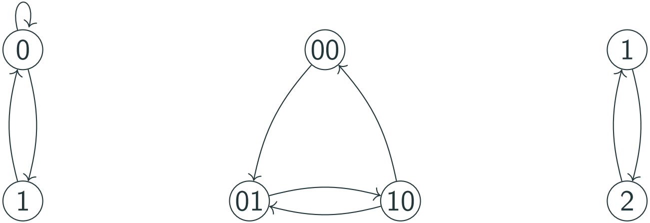


2-coloring

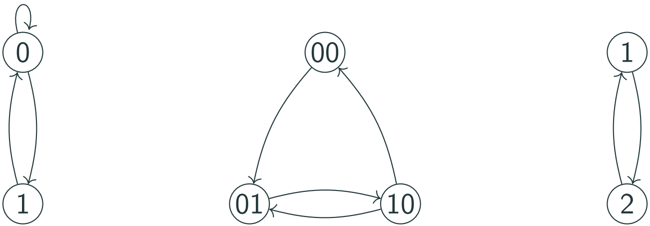
Complexity Separation on Paths

Naor, Stockmeyer (1995)
If the input graph is an unlabeled path or cycle, the time complexity is decidable.

The different time complexities are $O(1)$, $\Theta(\log^* n)$ and $\Omega(n)$.

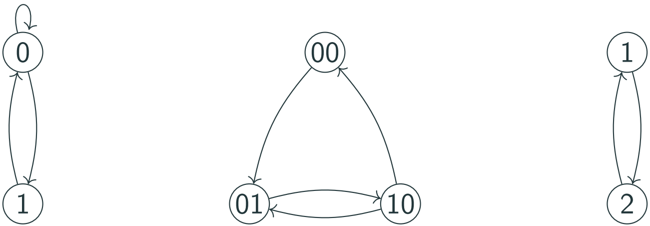


How to Decide the Complexity



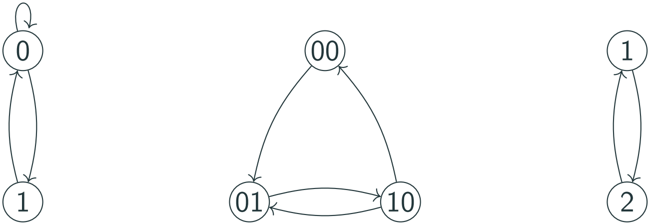
- $O(1)$

How to Decide the Complexity



- $O(1)$: There exists a self-loop.

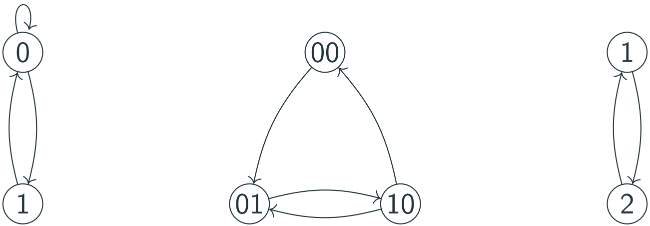
How to Decide the Complexity



- $O(1)$: There exists a self-loop.

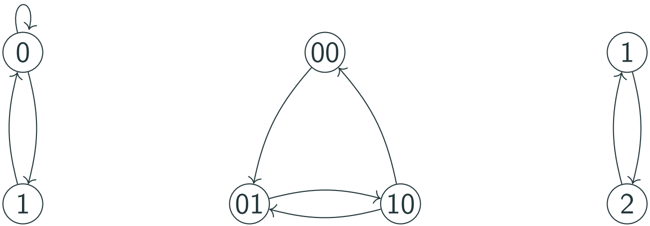
Naor, Stockmeyer (1995)
If a problem on paths can be computed in $o(\log^* n)$ rounds, there exists an algorithm with the same complexity stable under relative order.

How to Decide the Complexity



- $O(1)$: There exists a self-loop.
- $\Theta(\log^* n)$

How to Decide the Complexity

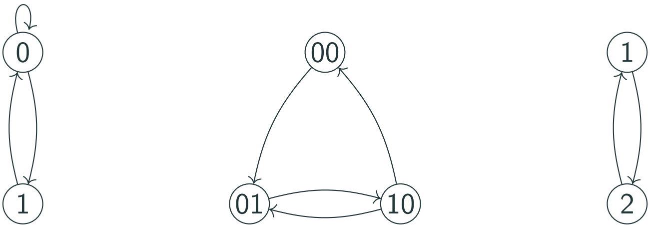


- $O(1)$: There exists a self-loop.

- $\Theta(\log^* n)$: There exists a node v and $k > 0$ such as $\forall i \geq k$, there exists a path of length i from v to v.

- $\Omega(n)$

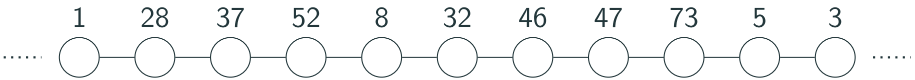
How to Decide the Complexity



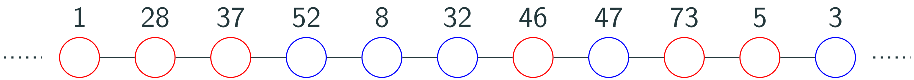
- $O(1)$: There exists a self-loop.
- $\Theta(\log^* n)$: There exists a node v and $k > 0$ such as $\forall i \geq k$, there exists a path of length i from v to v.
- $\Omega(n)$: For all nodes v, either v cannot be reached from v, or there exists $k > 1$ such as any path of length $i > 0$ from v to v is such as k divides i.

Decidability on Paths with Inputs

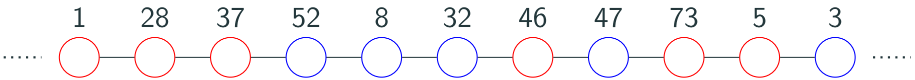
Problem on Paths with Inputs



Problem on Paths with Inputs

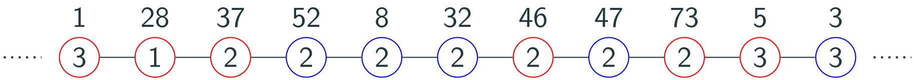


Problem on Paths with Inputs



- 3-color the red nodes.
- Carry the color through the blue nodes.

Problem on Paths with Inputs



- 3-color the red nodes.
- Carry the color through the blue nodes.

Complexity Separation on Paths with Inputs

Balliu *et. al* **(2019)**
For any LCL problem on cycle graphs, its complexity is either $\Omega(n)$ or $O(\log^* n)$. Moreover, there is an algorithm that decides whether the problem has complexity $\Omega(n)$ or $O(\log^* n)$ on cycle graphs ; for the case the complexity is $O(\log^* n)$, the algorithm outputs a description of an $O(\log^* n)$-round deterministic LOCAL algorithm that solves it.

For any LCL problem on cycle graphs, its complexity is either $\Omega(\log^* n)$ or $O(1)$. Moreover, there is an algorithm that decides whether the problem has complexity $\Omega(\log^* n)$ or $O(1)$ on cycle graphs ; for the case the complexity is $O(1)$, the algorithm outputs a description of an $O(1)$-round deterministic LOCAL algorithm that solves it.

Complexity Separation on Paths with Inputs

Balliu *et. al* **(2019)**
It is PSPACE-hard to distinguish whether a given LCL problem with input labels can be solved in $O(1)$ time or needs $\Omega(n)$ time on globally oriented path graphs.

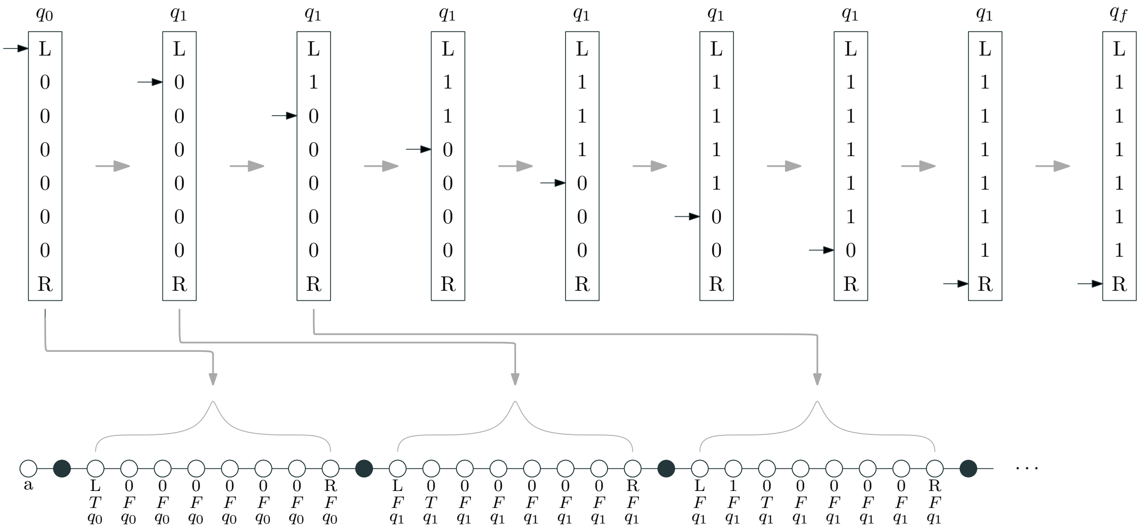
Complexity Separation on Paths with Inputs

Balliu *et. al* (2019)

It is PSPACE-hard to distinguish whether a given LCL problem with input labels can be solved in $O(1)$ time or needs $\Omega(n)$ time on globally oriented path graphs.

- Input : nodes in states a, b, or representing a Turing Machine tape on space k
- Output :
  - Tape nodes carry the a or b closest from the left
  - Tape nodes prove that the tape encodes a bad TM simulation and carry state E after the mistake

Turing Machine Encoding



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
			q_0	q_0	q_0	q_0	q_0	q_0	q_0	q_0		q_1	q_1	q_1	q_1	q_1	q_1	q_1	q_1		q_1	q_1	q_1	q_1	q_1	q_1	q_1	q_1		
Input			T	F	F	F	F	F	F	F		F	T	F	F	F	F	F	F		F	F	T	F	F	F	F	F		
	a		L	0	0	0	0	0	0	R		L	0	0	0	0	1	0	R		L	1	0	0	0	0	0	R		
	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	…
Output	a	a	a	a	a	a	a	E^2	E^2	E^2	E^2	E^2	E^2	E^2	E^2	E^2	E^2	E	E	E	E	E	E	E	E	E	E	E	E	
								0	0	0	0	0	0	0	0	0	0													
								0	1	2	3	4	5	6	7	8	9													

Complexity Separation on Trees

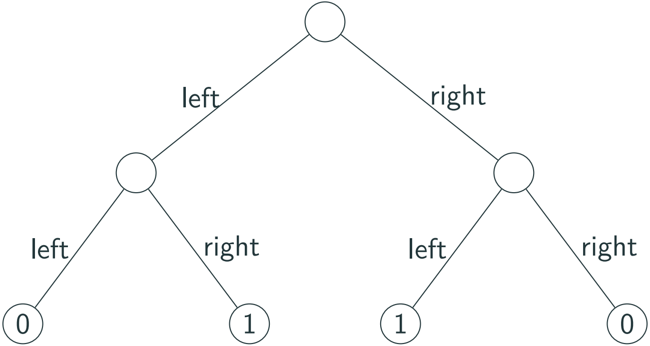
Balliu *et. al* (2019)
It is PSPACE-hard to distinguish whether a given LCL problem without input labels can be solved in $O(1)$ time or needs $\Omega(n)$ time on trees with degree $\Delta = 3$.

Encoding a Number in a Tree

Encoding $6 = 0110_2$

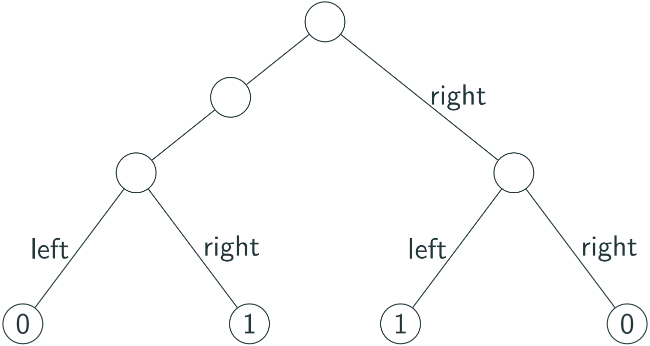
Encoding a Number in a Tree

Encoding $6 = 0110_2$



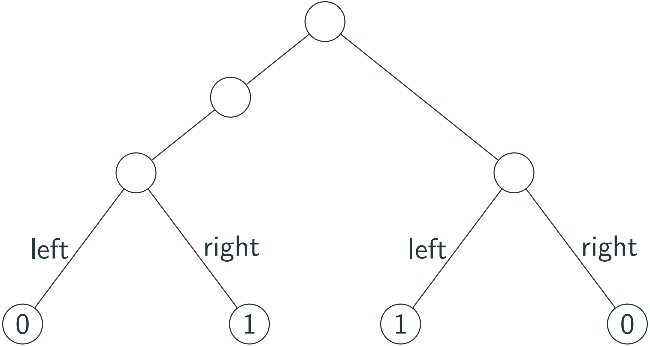
Encoding a Number in a Tree

Encoding $6 = 0110_2$



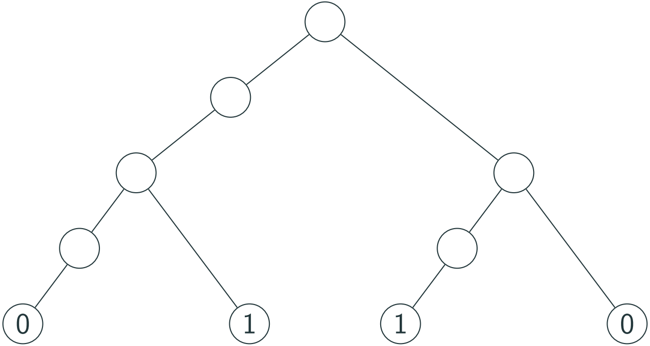
Encoding a Number in a Tree

Encoding $6 = 0110_2$



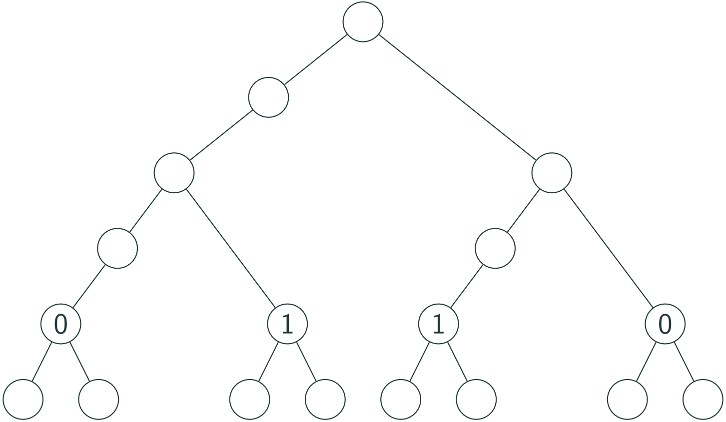
Encoding a Number in a Tree

Encoding $6 = 0110_2$



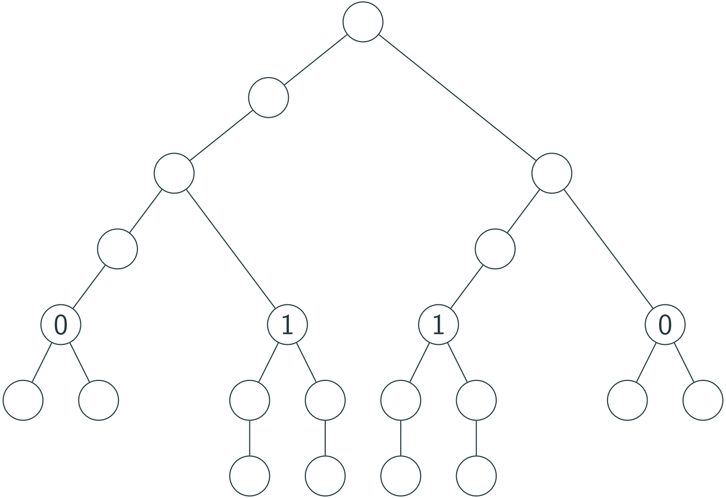
Encoding a Number in a Tree

Encoding $6 = 0110_2$



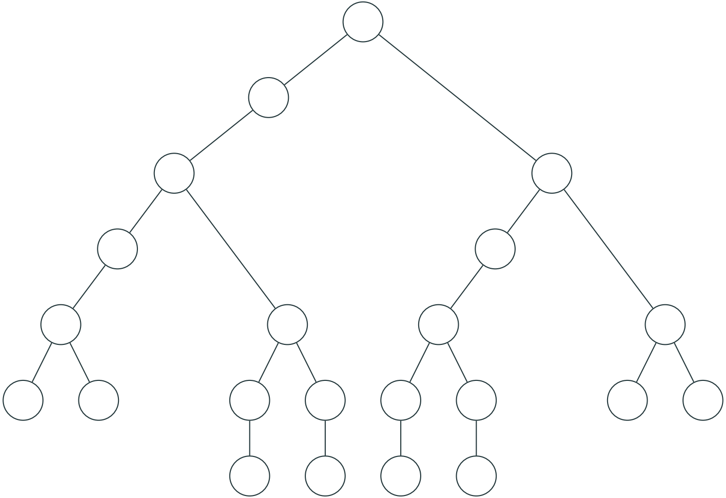
Encoding a Number in a Tree

Encoding $6 = 0110_2$



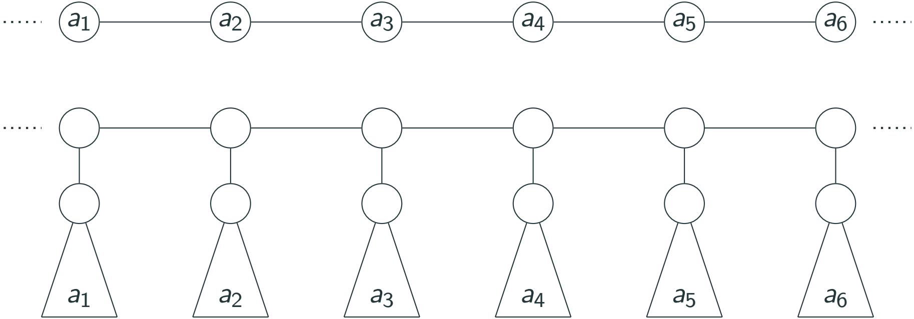
Encoding a Number in a Tree

Encoding $6{=}0110_2$



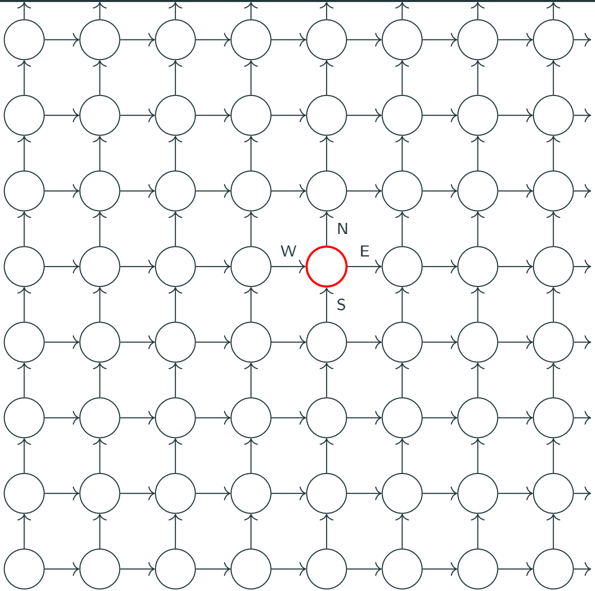
Encoding the Input of the Path

Encoding the Input of the Path



Toroidal Grids

Toroidal Grid Graphs

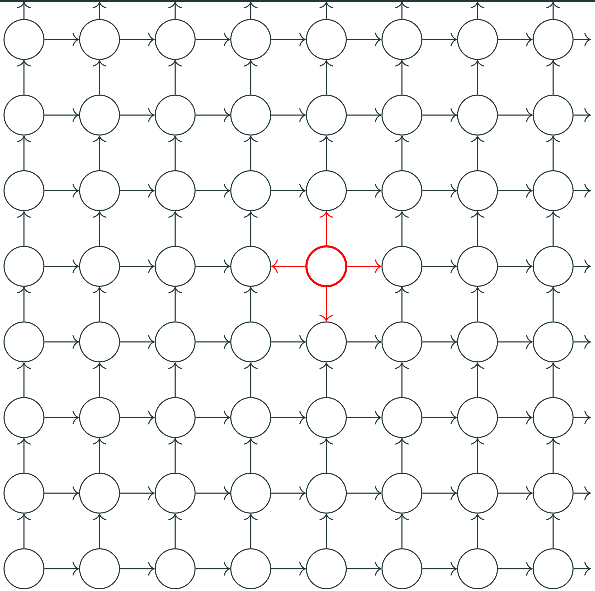


Grid Graphs :

- Toroidal $n \times n$ grid
- Consistent orientation N/S/E/W

Brandt *et. al* (2017)
Given any LCL problem P with an algorithm A that solves P in time $T = o(n)$, there exists an algorithm B that solves P and has running time $O(\log^* n)$.

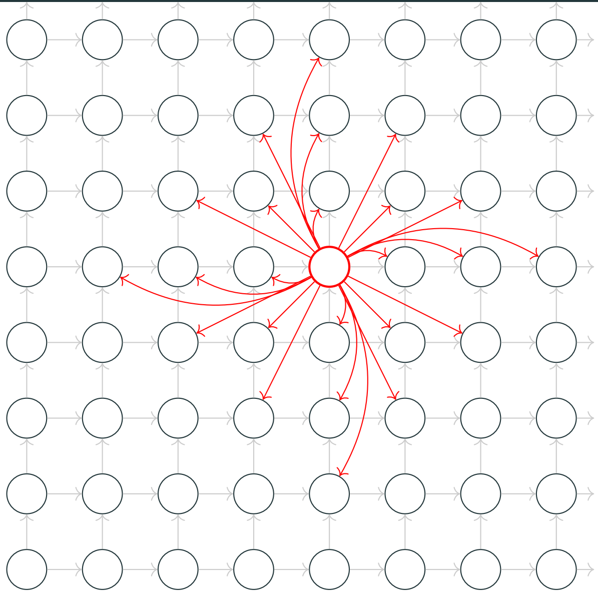
Speed Up Algorithm



Algorithm A in time $T(n) = o(n)$:

- Take k such as $T(k) < k/4 - 4$
- Compute Independent set I of $G^{k/2}$ in time $O(\log^* n)$
- Compute new identifiers in $[k^2]$
  - Each node u finds its closest anchor $i_u \in I$
  - $id'_u = (x_{i_u} - x_u, y_{i_u} - y_u)$
  - New identifiers form a distance-$k/2$ coloring
- Simulate A with the new identifiers in time $T(k)$

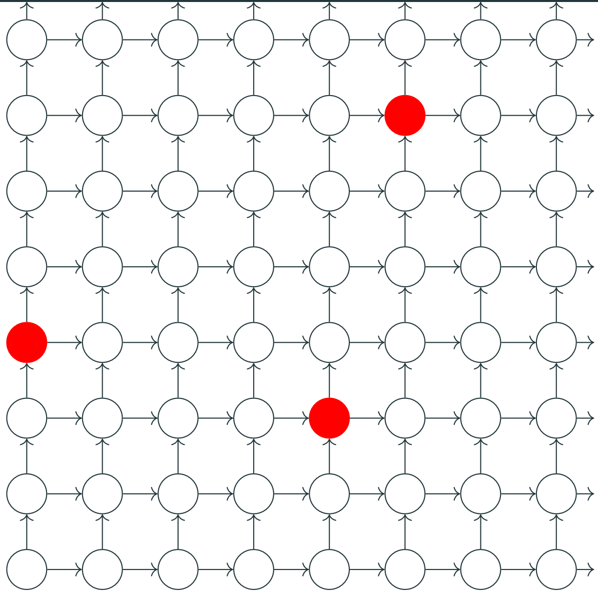
Speed Up Algorithm



Algorithm A in time $T(n) = o(n)$:

- Take k such as $T(k) < k/4 - 4$
- Compute Independent set I of $G^{k/2}$ in time $O(\log^* n)$
- Compute new identifiers in $[k^2]$
  - Each node u finds its closest anchor $i_u \in I$
  - $id'_u = (x_{i_u} - x_u, y_{i_u} - y_u)$
  - New identifiers form a distance-$k/2$ coloring
- Simulate A with the new identifiers in time $T(k)$

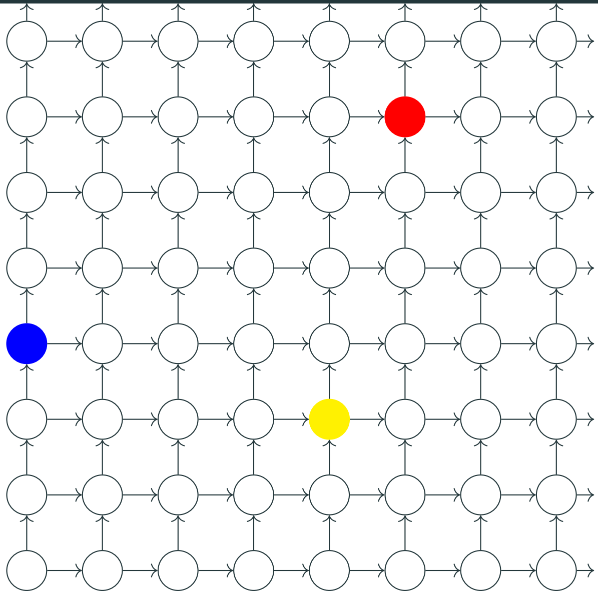
Speed Up Algorithm



Algorithm A in time $T(n) = o(n)$:

- Take k such as $T(k) < k/4 - 4$
- Compute Independent set I of $G^{k/2}$ in time $O(\log^* n)$
- Compute new identifiers in $[k^2]$
  - Each node u finds its closest anchor $i_u \in I$
  - $id'_u = (x_{i_u} - x_u, y_{i_u} - y_u)$
  - New identifiers form a distance-$k/2$ coloring
- Simulate A with the new identifiers in time $T(k)$

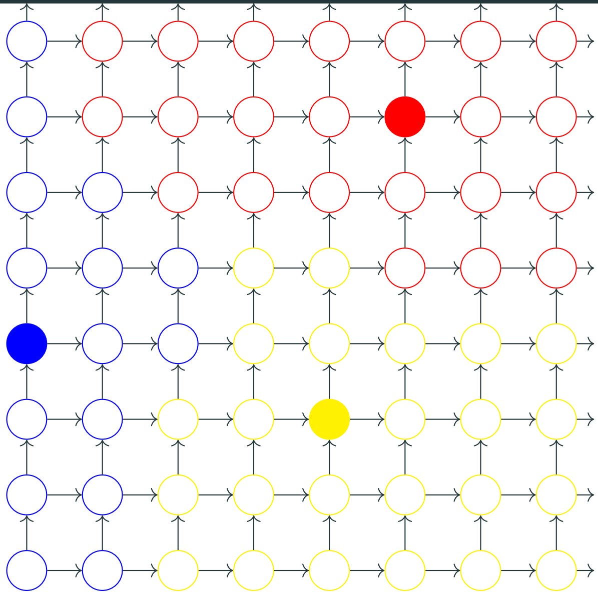
Speed Up Algorithm



Algorithm A in time $T(n) = o(n)$:

- Take k such as $T(k) < k/4 - 4$
- Compute Independent set I of $G^{k/2}$ in time $O(\log^* n)$
- Compute new identifiers in $[k^2]$
  - Each node u finds its closest anchor $i_u \in I$
  - $id'_u = (x_{i_u} - x_u, y_{i_u} - y_u)$
  - New identifiers form a distance-$k/2$ coloring
- Simulate A with the new identifiers in time $T(k)$

Speed Up Algorithm



Algorithm A in time $T(n) = o(n)$:

- Take k such as $T(k) < k/4 - 4$
- Compute Independent set I of $G^{k/2}$ in time $O(\log^* n)$
- Compute new identifiers in $[k^2]$
  - Each node u finds its closest anchor $i_u \in I$
  - $id'_u = (x_{i_u} - x_u, y_{i_u} - y_u)$
  - New identifiers form a distance-$k/2$ coloring
- Simulate A with the new identifiers in time $T(k)$

Complexity Separation on Grids

Brandt *et. al* (2017)

The problem of deciding whether a given LCL can be solved in time $\Theta(\log^* n)$ or $\Theta(n)$ on grids is

Complexity Separation on Grids

Brandt *et. al* (2017)
The problem of deciding whether a given LCL can be solved in time $\Theta(\log^* n)$ or $\Theta(n)$ on grids is undecidable.

Complexity Separation on Grids

Brandt *et. al* (2017)
The problem of deciding whether a given LCL can be solved in time $\Theta(\log^* n)$ or $\Theta(n)$ on grids is undecidable.

From a given Turing Machine :

- Find an area to simulate the execution :
  - Put an anchor that starts the computation (line 0)
  - Line i corresponds to execution step i
  - Possibility to check locally that the execution is right
- Solve a $\Theta(n)$ problem (3-coloring) otherwise

Complexity Separation on Grids

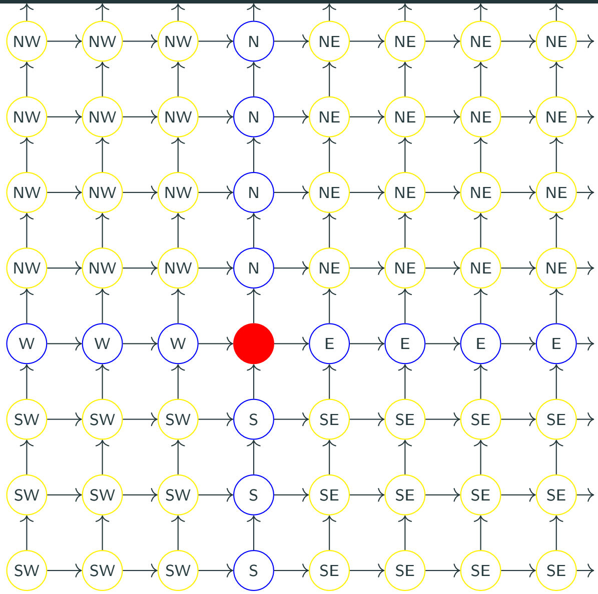
Brandt *et. al* **(2017)**
The problem of deciding whether a given LCL can be solved in time $\Theta(\log^* n)$ or $\Theta(n)$ on grids is undecidable.

From a given Turing Machine :

- Find an area to simulate the execution :
  - Put an anchor that starts the computation (line 0)
  - Line i corresponds to execution step i
  - Possibility to check locally that the execution is right
- Solve a $\Theta(n)$ problem (3-coloring) otherwise

If the Turing Machine halts, the first solution takes a constant space, and can be repeated.
If the Turing Machine does not halt, the global problem must be solved.

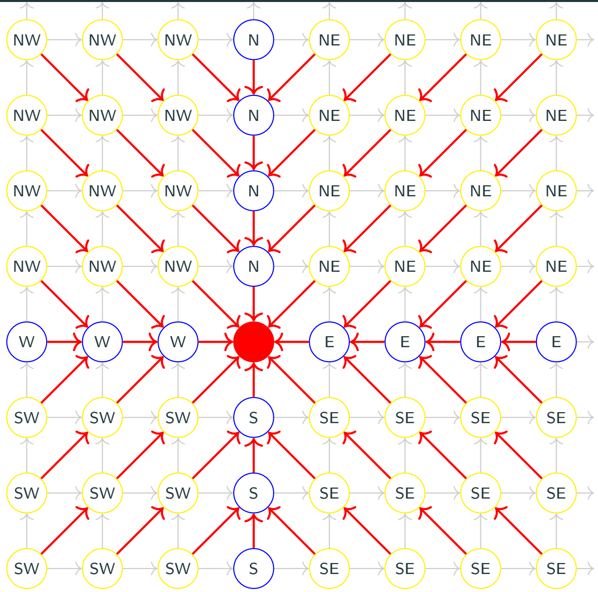
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation

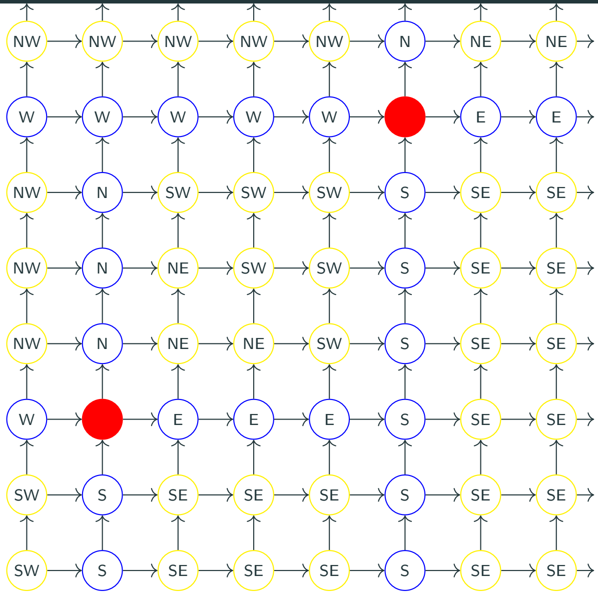
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation

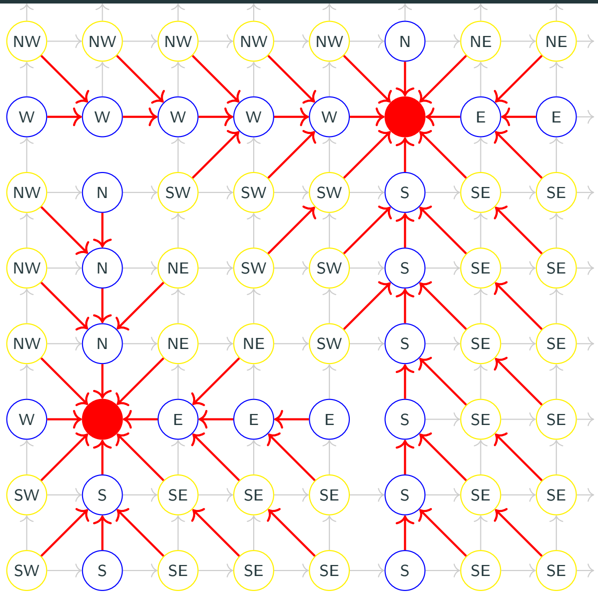
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation

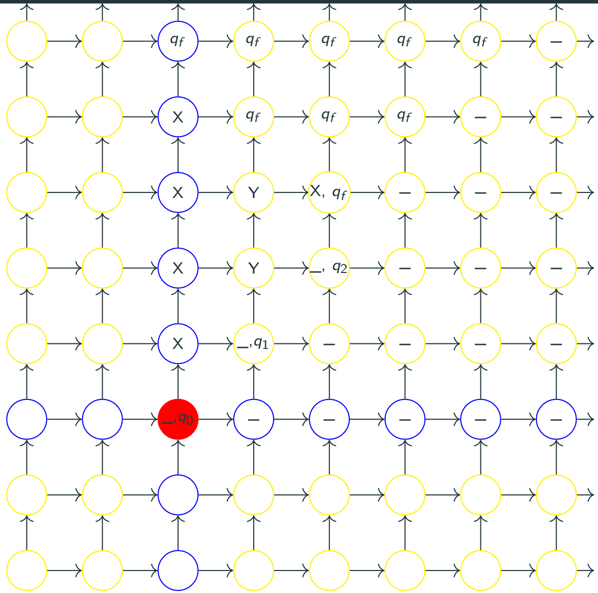
Turing Machine Simulation



Turing Machine Simulation :

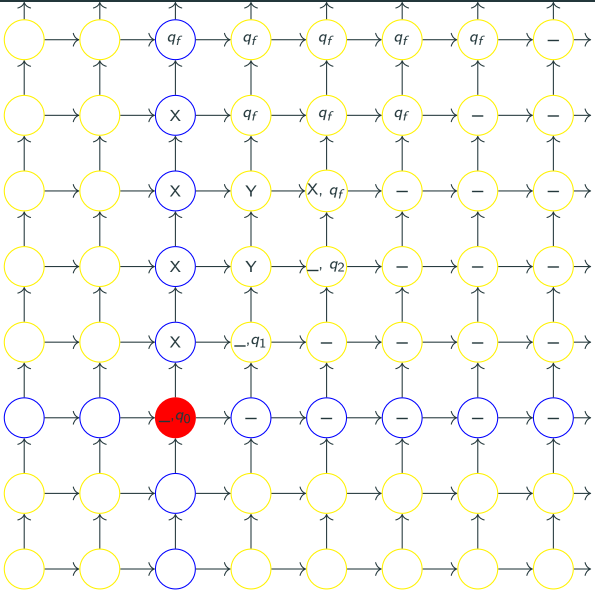
- Consistent labelling around anchors
- Tape simulation

Turing Machine Simulation



Turing Machine Simulation :

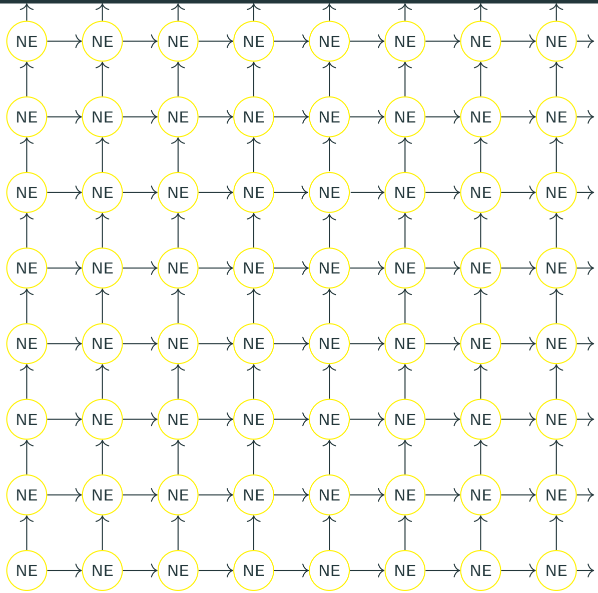
- Consistent labelling around anchors
- Tape simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation
- Problem ?

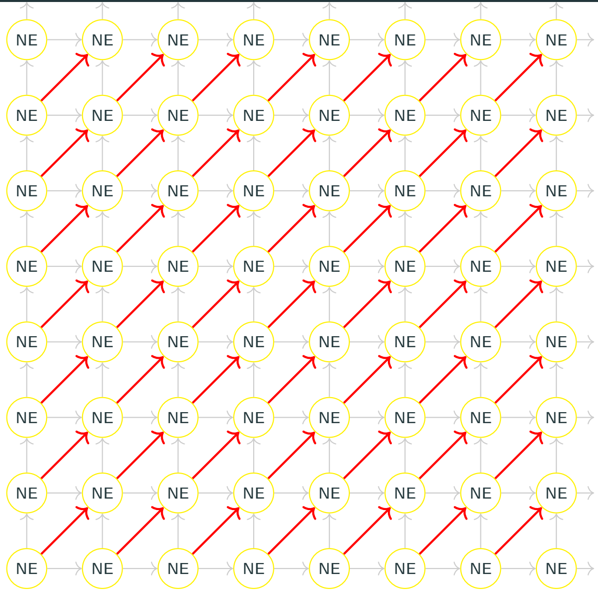
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation
- Problem ?

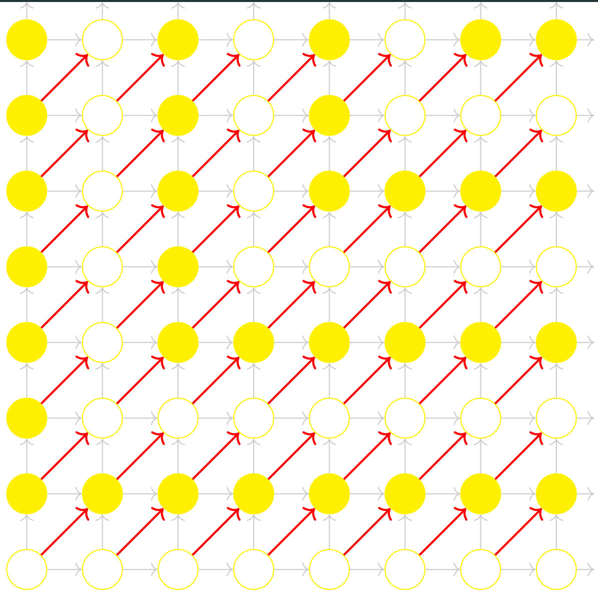
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation
- 2-coloring along diagonals

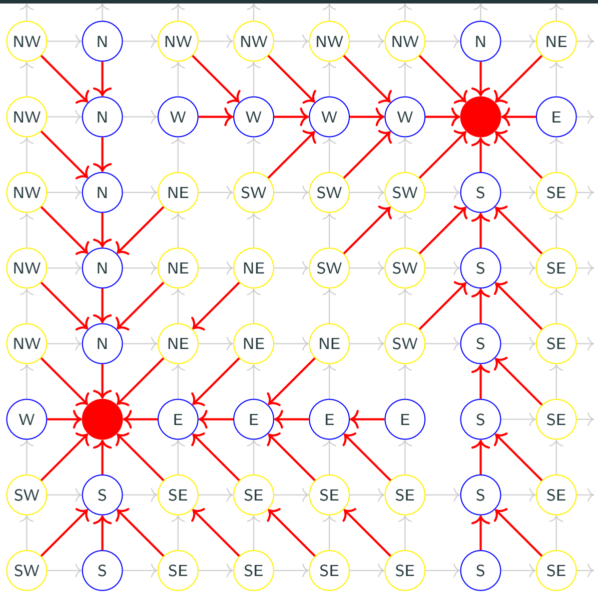
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation
- 2-coloring along diagonals

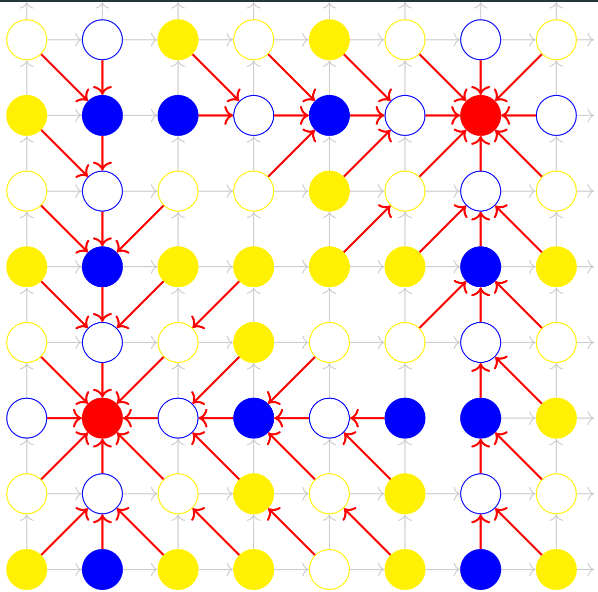
Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation
- 2-coloring along diagonals

Turing Machine Simulation



Turing Machine Simulation :

- Consistent labelling around anchors
- Tape simulation
- 2-coloring along diagonals

Bibliography

- https ://jukkasuomela.fi/landscape-of-locality/
- Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Mikaël Rabie, Jukka Suomela. **The Distributed Complexity of Locally Checkable Problems on Paths is Decidable**. In Proc. PODC 2019.
- Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, Jukka Suomela. **Lower Bounds for Maximal Matchings and Maximal Independent Sets**. In Proc. FOCS 2019.
- Moni Naor, Larry J. Stockmeyer. **What Can be Computed Locally ?** In SIAM J. Comput. 24,1995.
- Sebastian Brandt, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Patric R. J. Östergård, Christopher Purcell, Joel Rybicki, Jukka Suomela, Przemyslaw Uznanski. **LCL Problems on Grids**. In PODC 2017.