Ability to Count Messages Is Worth $\Theta(\Delta)$ Rounds in Distributed Computing

Tuomo Lempiäinen

Aalto University, Finland

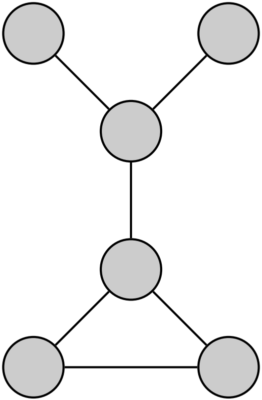
LICS 2016

July 7, 2016 @ New York

Outline

1. Introduction to distributed computing
2. Different models of computation
3. New result: a tight lower bound for simulating one model in another (by using bisimulation)

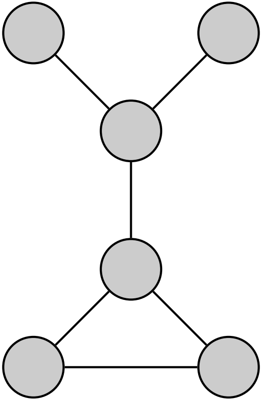
Distributed system



A simple finite undirected graph, whose each node is a deterministic state machine that

- runs the same algorithm,
- can communicate with its neighbours,
- produces a local output.

Distributed system

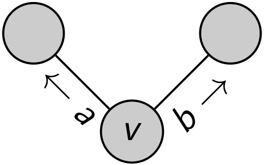


A simple finite undirected graph, whose each node is a deterministic state machine that

- runs the same algorithm,
- can communicate with its neighbours,
- produces a local output.

Anonymous nodes $\Rightarrow$ a weak model of computation.

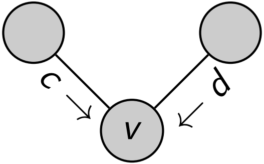
Communication in synchronous rounds



In every round, each node v

1. sends messages to its neighbours,
2. receives messages from its neighbours,
3. updates its state.

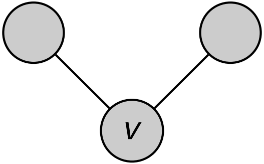
Communication in synchronous rounds



In every round, each node v

1. sends messages to its neighbours,
2. receives messages from its neighbours,
3. updates its state.

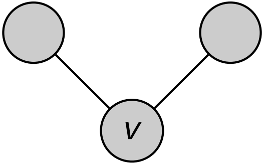
Communication in synchronous rounds



In every round, each node v

1. sends messages to its neighbours,
2. receives messages from its neighbours,
3. updates its state.

Communication in synchronous rounds

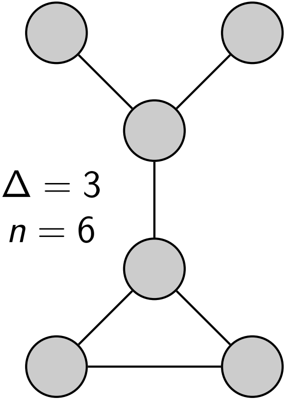


In every round, each node v

1. sends messages to its neighbours,
2. receives messages from its neighbours,
3. updates its state.

Eventually, each node halts and announces its own local output.

Focus on communication, not computation



The running time of an algorithm is the *number of communications rounds*.

The running time may depend on two parameters:

- the maximum degree of the graph, Δ,
- the number of nodes, n.

Variants of the model of computation

Options for sending messages:

- a port number for each neighbour,
- broadcast the same message to all neighbours.

Variants of the model of computation

Options for sending messages:

- a port number for each neighbour,
- broadcast the same message to all neighbours.

Options for receiving messages:

- a port number for each neighbour,
- receive a multiset of messages,
- receive a set of messages.

Variants of the model of computation

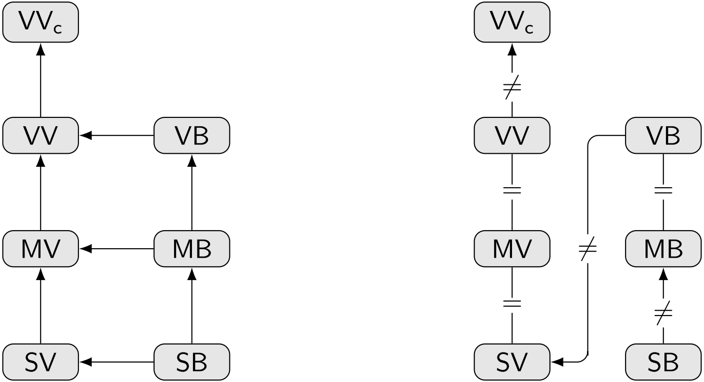
Options for sending messages:

- a port number for each neighbour,
- broadcast the same message to all neighbours.

Options for receiving messages:

- a port number for each neighbour,
- receive a multiset of messages,
- receive a set of messages.

A hierarchy of models



Hella et al. (PODC 2012):

$$\mathsf{SB} \subsetneq \mathsf{MB} = \mathsf{VB} \subsetneq \mathsf{SV} = \mathsf{MV} = \mathsf{VV} \subsetneq \mathsf{VV_c}.$$

Graph problems

We study *graph problems* where

- problem instance is the communication graph $G = (V, E)$,
- the local outputs together define a solution $S\colon V \to Y$, where Y is a finite set of local outputs.

Graph problems

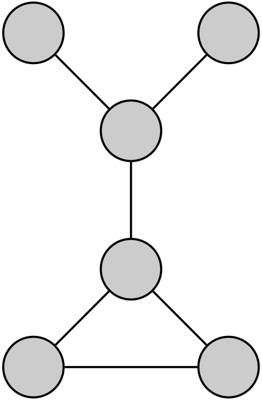
We study *graph problems* where

- problem instance is the communication graph $G = (V, E)$,
- the local outputs together define a solution $S \colon V \to Y$, where Y is a finite set of local outputs.

Algorithm $\mathcal{A}$ solves problem Π *in time T* if for all input graphs G of maximum degree at most Δ:

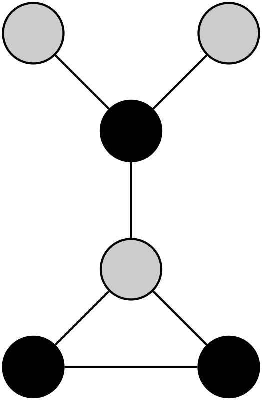
1. $\mathcal{A}$ stops after at most $T(\Delta, n)$ rounds in each node of G.
2. The output S of $\mathcal{A}$ in G is a valid solution for Π.

Example: graph problems



Often the solution $S\colon V \to Y$ is an encoding of a subset of vertices or edges of the graph.

Example: graph problems



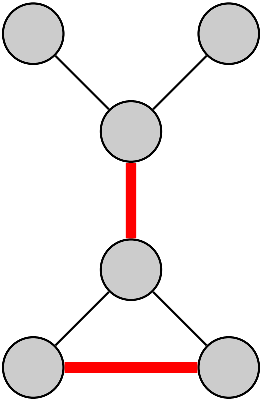
$Y = \{0, 1\}$

Often the solution $S\colon V \to Y$ is an encoding of a subset of vertices or edges of the graph.

Example problems:

- minimum vertex cover,

Example: graph problems



$Y = \{1, 2, 3\}$

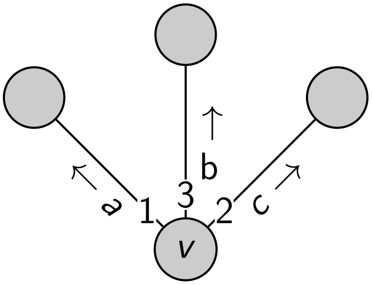
Often the solution $S\colon V \to Y$ is an encoding of a subset of vertices or edges of the graph.

Example problems:

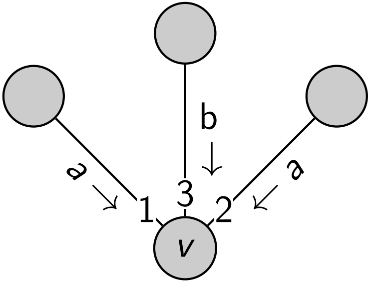
- minimum vertex cover,
- maximal matching.

The models MV and SV

Node v sends a vector (a, c, b).



The models MV and SV

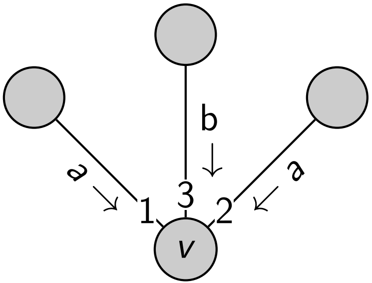


Node v sends a vector (a, c, b).

Node v receives

- a multiset $\{a, a, b\}$ in model MV,
- a set $\{a, b\}$ in model SV.

The models MV and SV



Node v sends a vector (a, c, b).

Node v receives

- a multiset $\{a, a, b\}$ in model MV,
- a set $\{a, b\}$ in model SV.

Formally, MV and SV denote the classes of graph problems solvable in the corresponding models.

The relationship of MV and SV

Trivially SV $\subseteq$ MV.

Hella, Järvisalo, Kuusisto, Laurinharju, L., Luosto, Suomela, Virtema (PODC 2012):

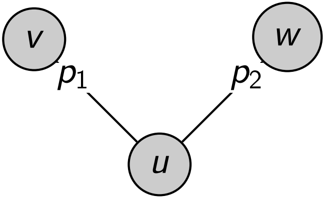
Theorem

Assume that there is an MV*-algorithm* $\mathcal{A}$ *that solves a problem* Π *in time* T*. Then there is an* SV*-algorithm* $\mathcal{B}$ *that solves* Π *in time* $T + 2\Delta - 2$*.*

It follows that SV $=$ MV.

Idea behind the previous theorem

First, solve the following *simulation problem* by an SV-algorithm:



If $p_1 = p_2$, then label(v) $\neq$ label(w).

Now the pair

(label, port number)

is distinct for each neighbour.

Then, simulate the MV-algorithm by attaching the above pair to each message.

Overhead required to simulate MV in SV

PODC 2012: The previous problem can be solved in $2\Delta - 2$ communication rounds.

Is this result tight?

Overhead required to simulate MV in SV

PODC 2012: The previous problem can be solved in $2\Delta - 2$ communication rounds.

Is this result tight?

This work: YES

Theorem

For each $\Delta \geq 2$ there is a port-numbered graph G_Δ with nodes u, v, w such that when executing any SV-algorithm $\mathcal{A}$ in G_Δ, u receives identical messages from its neighbours v and w in rounds $1, 2, \ldots, 2\Delta - 2$.

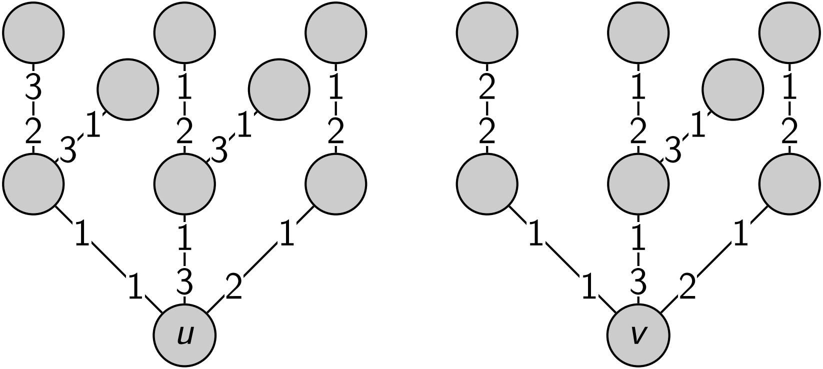
Overhead required to simulate MV in SV

We can also separate the models by a graph problem:

Theorem

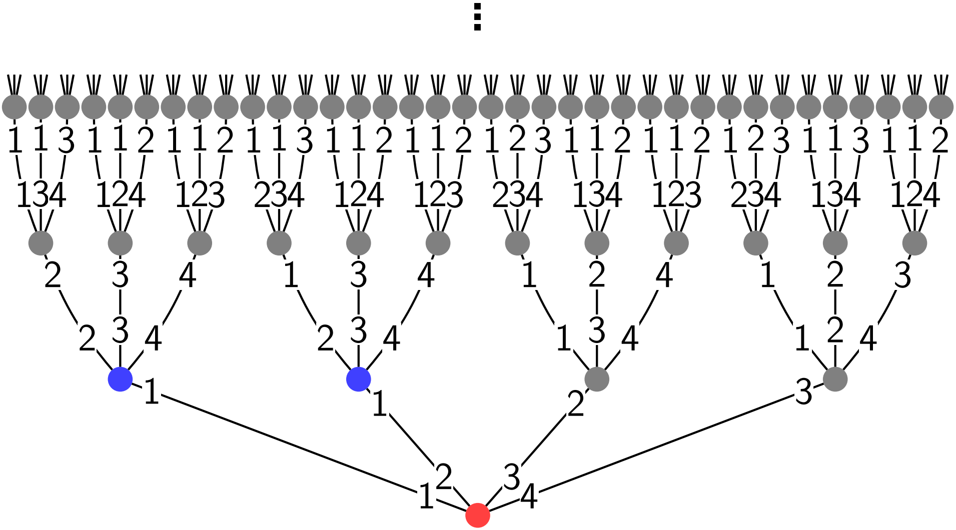
There is a graph problem Π that can be solved in one round by an MV*-algorithm but that requires at least $\Delta - 1$ rounds for all $\Delta \geq 2$, when solved by an* SV*-algorithm.*

Example: separating SV and MV



Output 1 if there is an even number of neighbours of even degree, 0 otherwise.

Generalisation: graph G_Δ (here $\Delta = 4$)



Proof idea

1. Investigate walks that start from the blue nodes and follow an identical sequence of port numbers.
   1. In which cases we cannot extend the walks in a consistent manner?
   2. What is the length of such maximal walks?

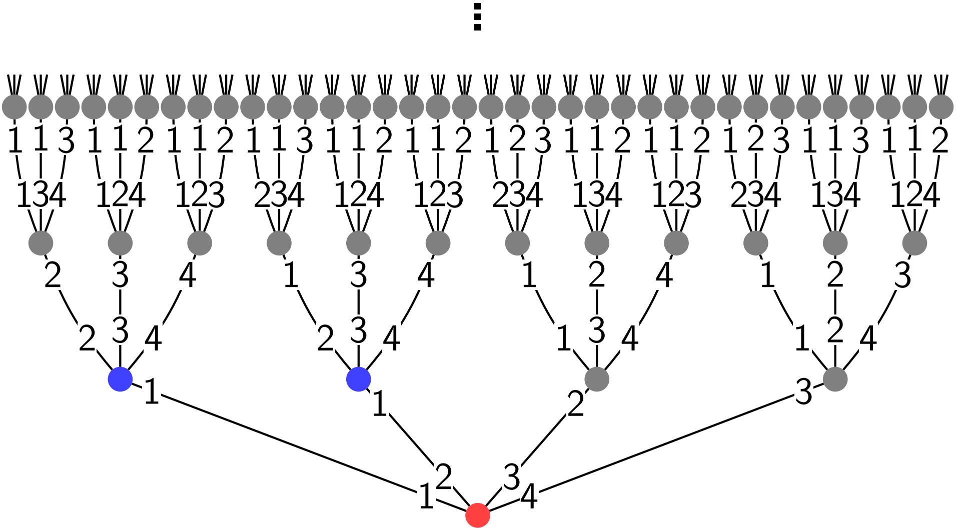
Proof idea

1. Investigate walks that start from the blue nodes and follow an identical sequence of port numbers.
    1. In which cases we cannot extend the walks in a consistent manner?
    2. What is the length of such maximal walks?
2. Prove a lower bound for the length of the walks.

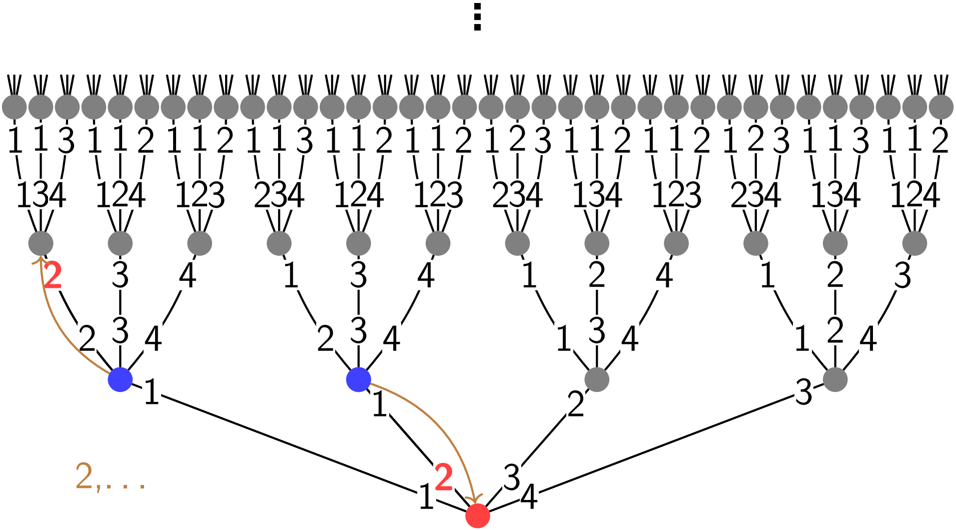
Proof idea

1. Investigate walks that start from the blue nodes and follow an identical sequence of port numbers.
    1. In which cases we cannot extend the walks in a consistent manner?
    2. What is the length of such maximal walks?
2. Prove a lower bound for the length of the walks.
3. Show that the lower bound on walks implies *bisimilarity* of the blue nodes up to a certain distance.
4. Bisimilarity entails a lower bound for the running time of any distributed algorithm that is able to distinguish the nodes.

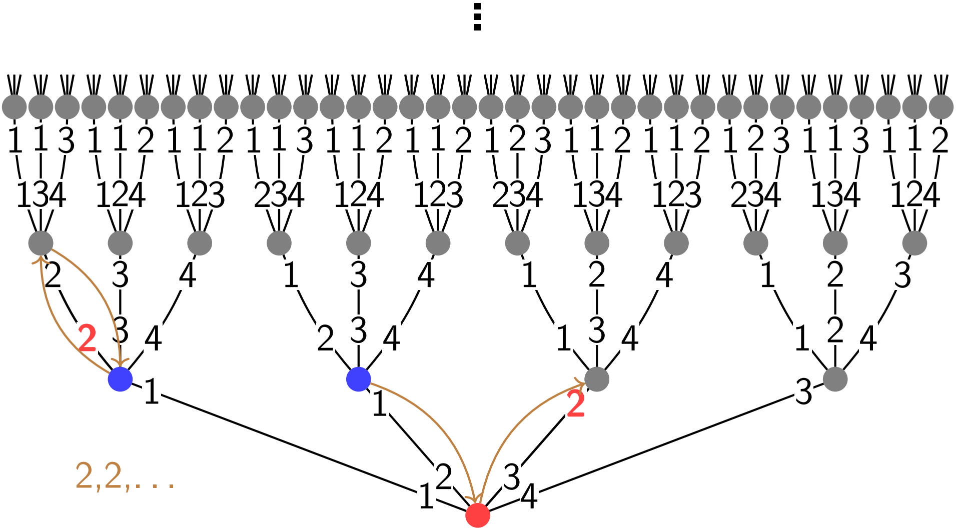
A pair of separating walks in G_4



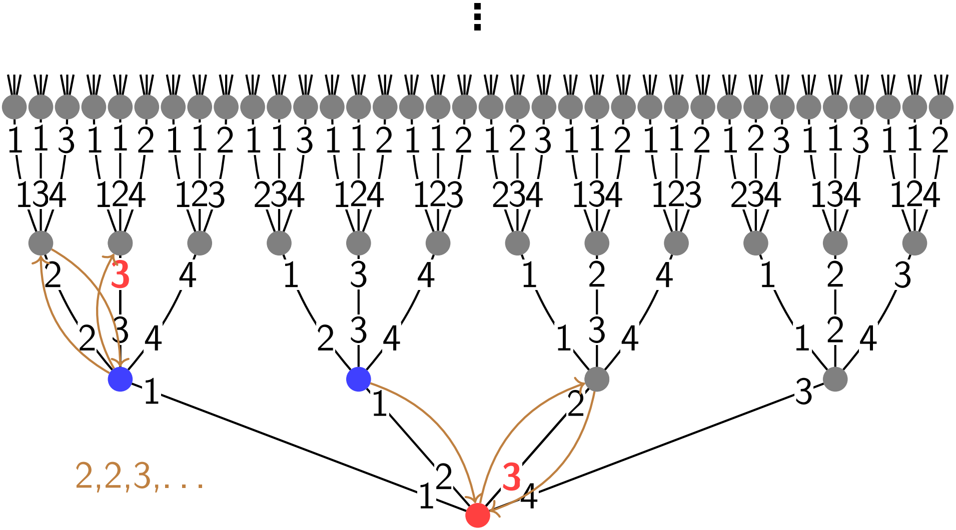
A pair of separating walks in G_4



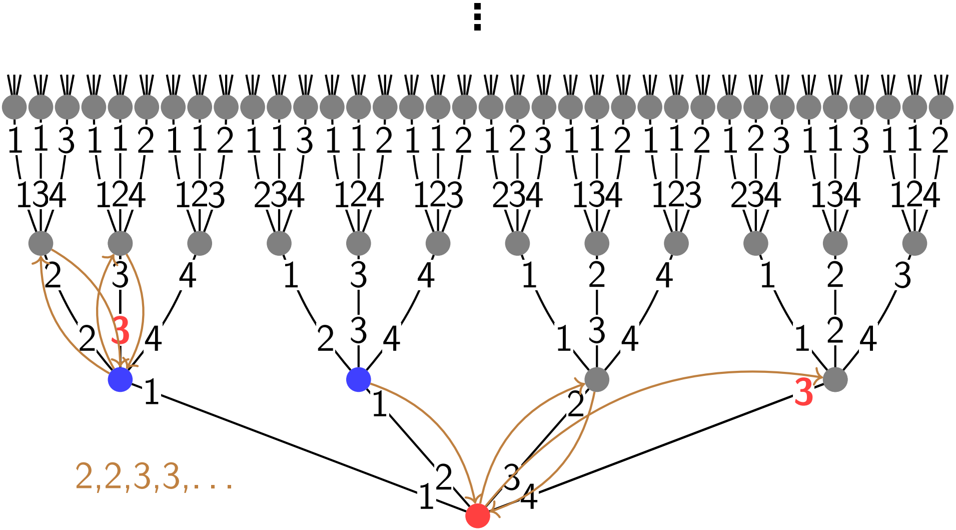
A pair of separating walks in G_4



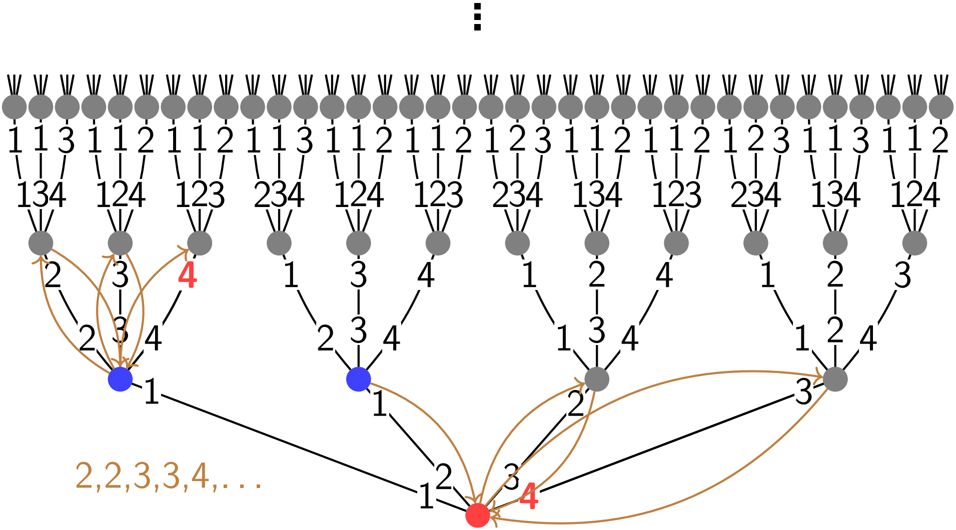
A pair of separating walks in G_4



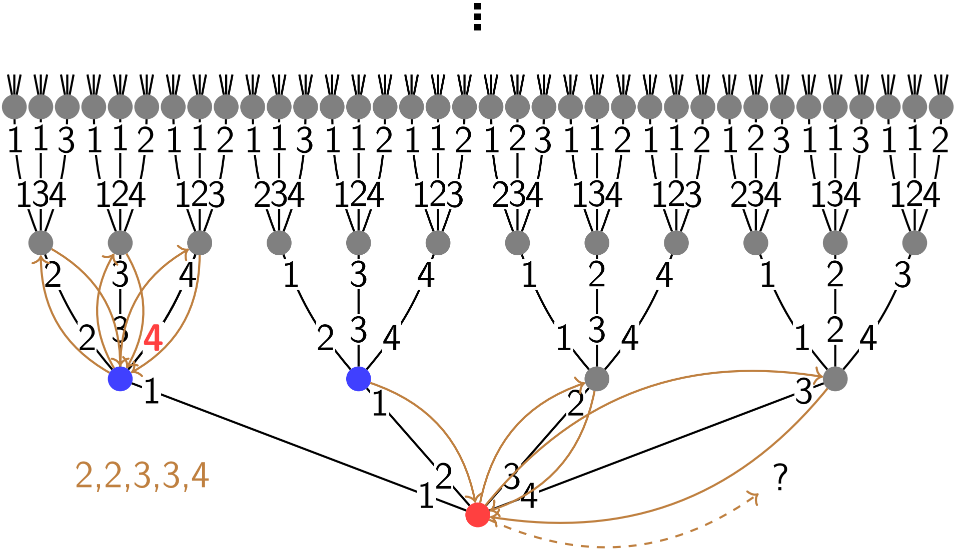
A pair of separating walks in G_4



A pair of separating walks in G_4



A pair of separating walks in G_4



Connections to modal logic

Hella et al. (PODC 2012):

- Logical characterisations for constant-time variants of the problem classes.
- In a certain class of structures, SV corresponds to *multimodal logic*
- . . . and MV corresponds to *graded multimodal logic*.

Connections to modal logic

Hella et al. (PODC 2012):

- Logical characterisations for constant-time variants of the problem classes.
- In a certain class of structures, SV corresponds to *multimodal logic*
- ... and MV corresponds to *graded multimodal logic*.
- Our result: When given a formula ϕ of graded multimodal logic, we can find an equivalent formula ψ of multimodal logic, but in general, the modal depth $\mathrm{md}(\psi)$ of ψ has to be at least $\mathrm{md}(\phi) + \Delta - 1$.

Conclusion

MV: Send a vector, receive a multiset.
SV: Send a vector, receive a set.

Previously:

- It is possible to simulate MV in SV by using $2\Delta - 2$ extra rounds.

This work:

- $2\Delta - 2$ rounds are *necessary*.
- Linear-in-Δ separation by a graph problem.

Conclusion

MV: Send a vector, receive a multiset.
SV: Send a vector, receive a set.

Previously:

- It is possible to simulate MV in SV by using $2\Delta - 2$ extra rounds.

This work:

- $2\Delta - 2$ rounds are *necessary*.
- Linear-in-Δ separation by a graph problem.

Thanks!