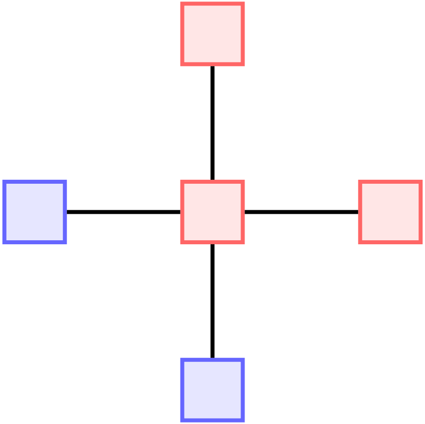
Decision Power of Weak Asynchronous Models of Distributed Computing

Philipp Czerner Roland Guttenberg Javier Esparza Martin Helfrich

Technische Universität München

Common Ground and Differing Aspects

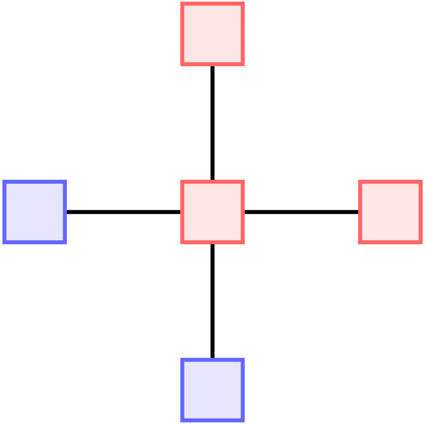
Every node executes identical finite-state machine.



Common Ground and Differing Aspects

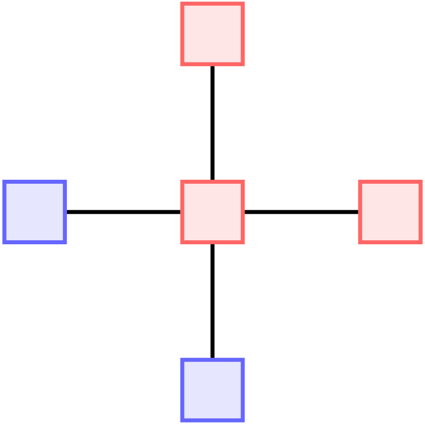
Every node executes identical finite-state machine.

Nodes decide property of graph or initial labeling by consensus.



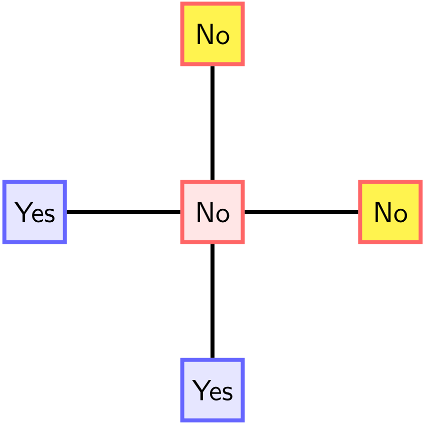
Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



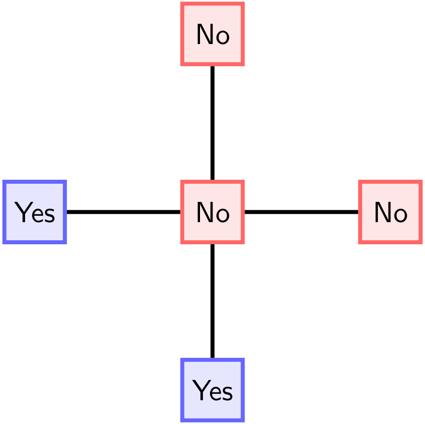
Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



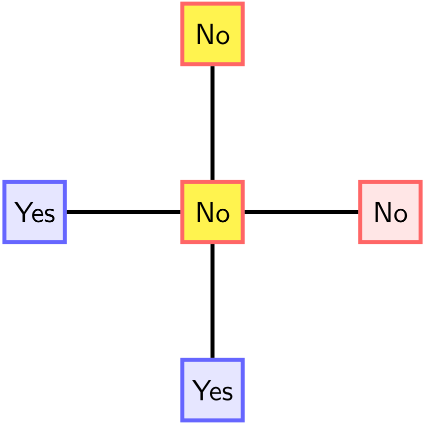
Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



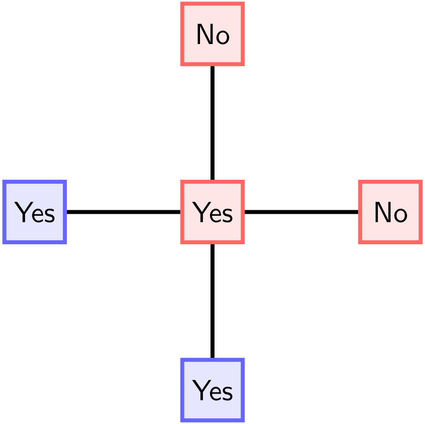
Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



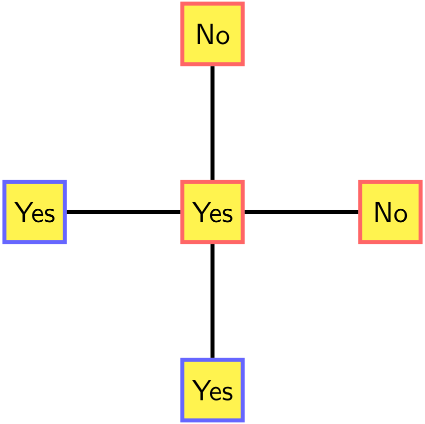
Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



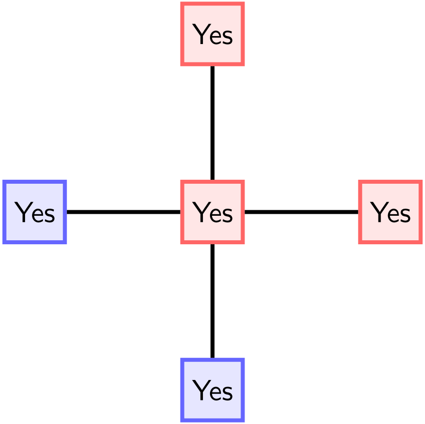
Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



Common Ground and Differing Aspects

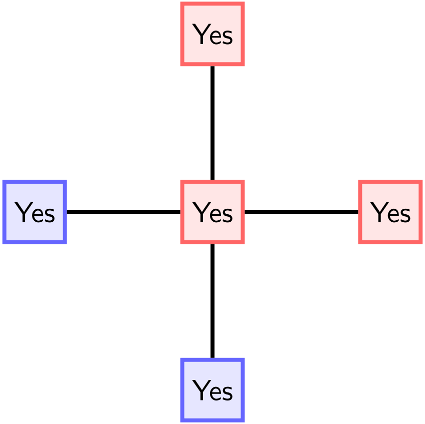
Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

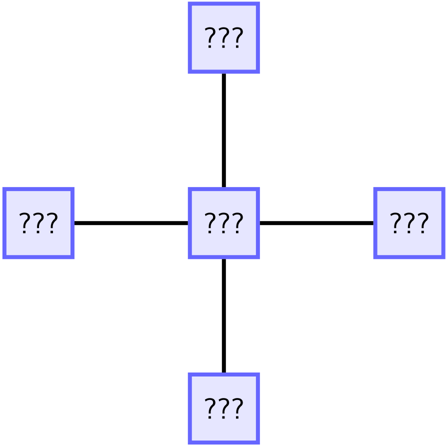


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

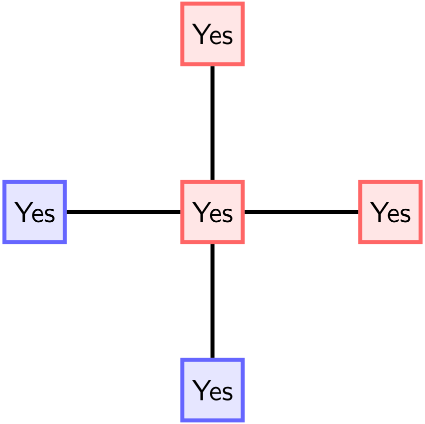


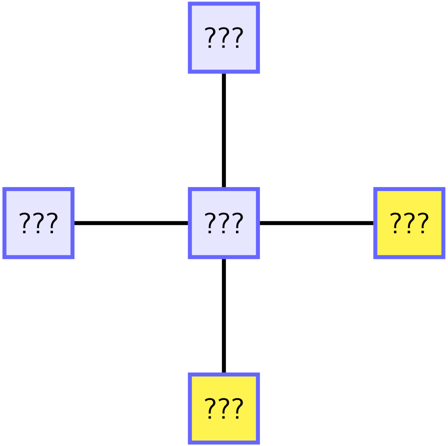


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

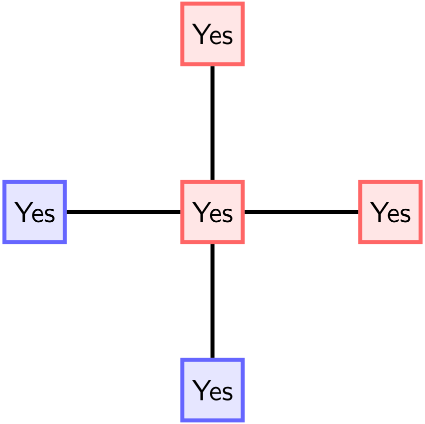


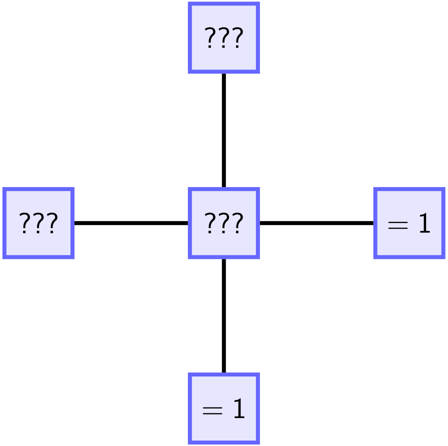


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

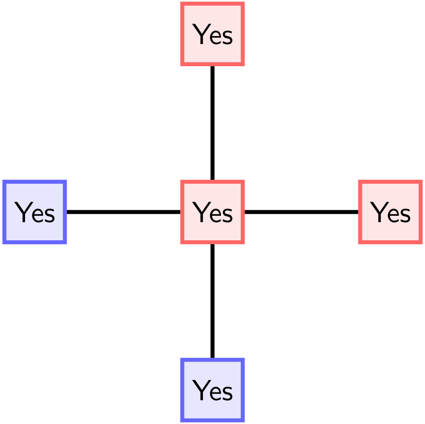


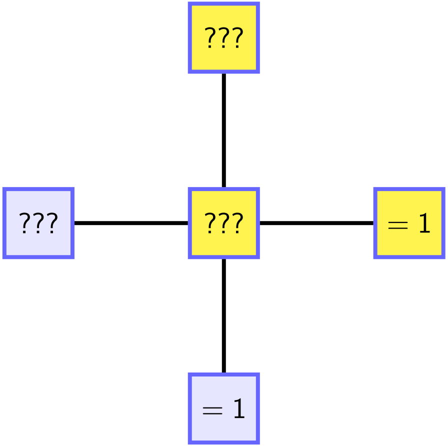


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

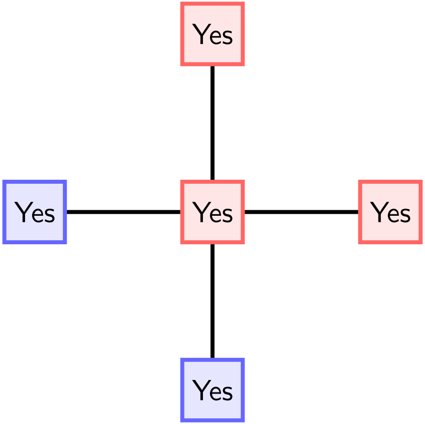




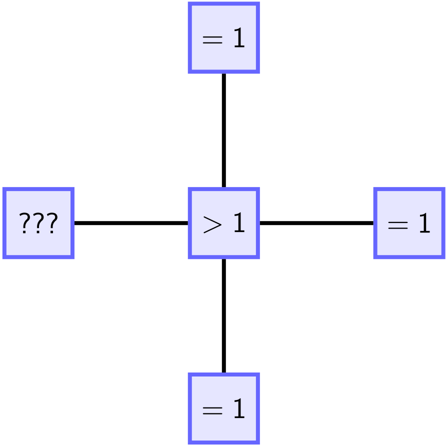
Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

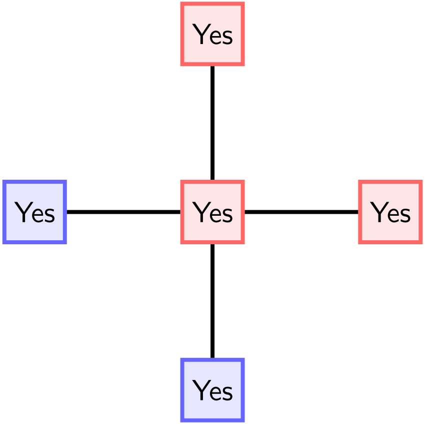


Nodes had to change answer.

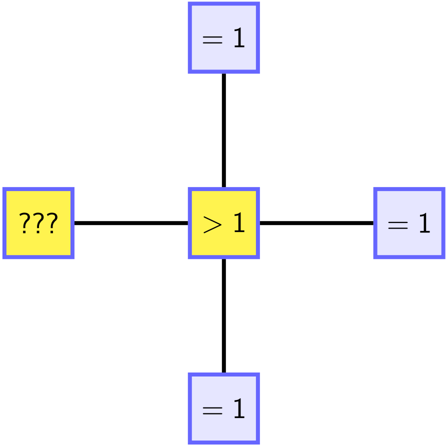


Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

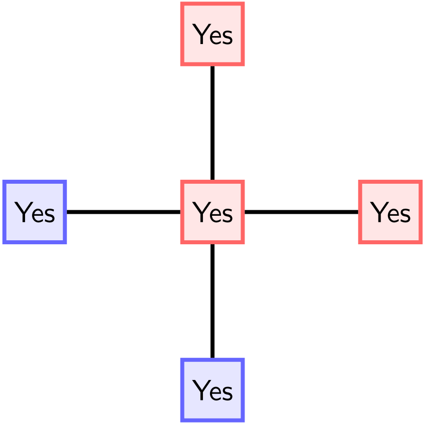


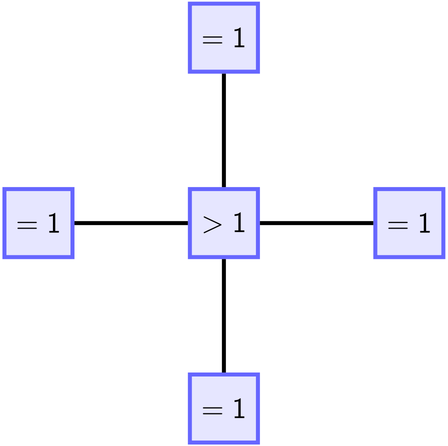
Nodes had to change answer.



Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

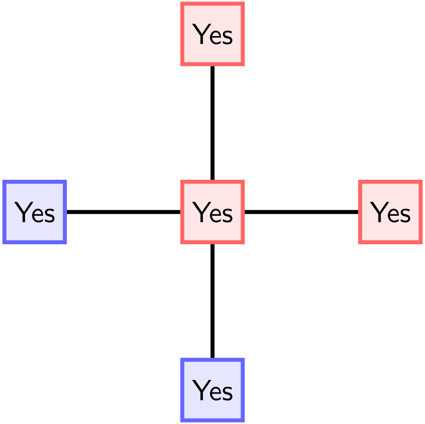


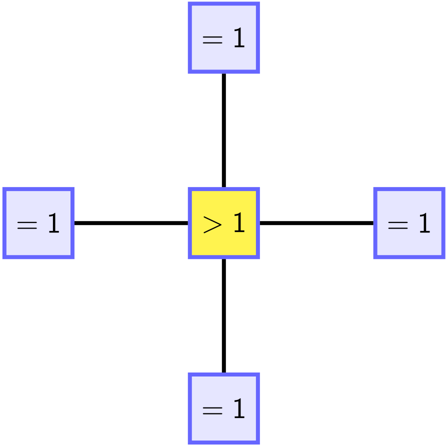


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

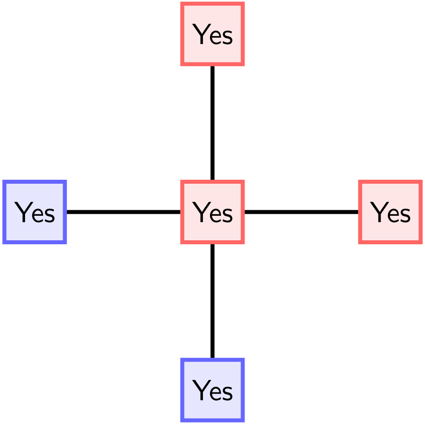


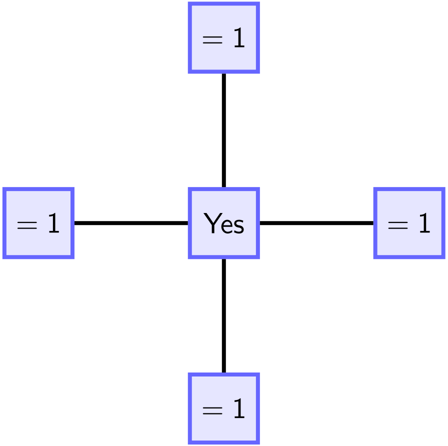


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

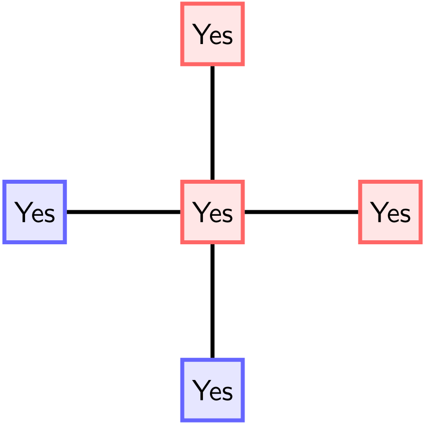


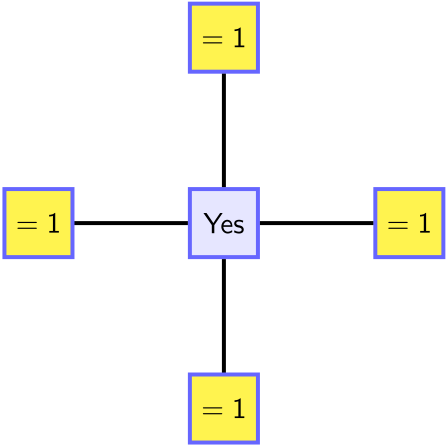


Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).

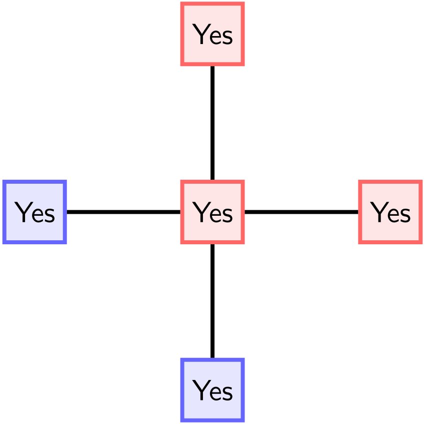




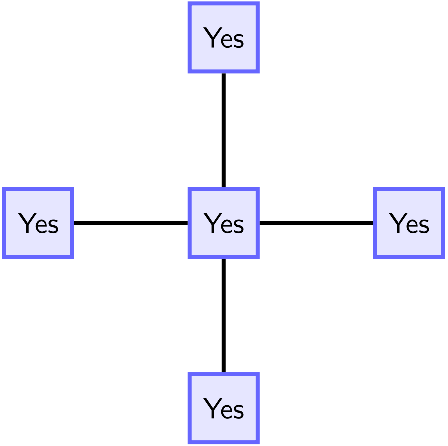
Nodes had to change answer.

Common Ground and Differing Aspects

Every node executes identical finite-state machine.
Nodes decide property of graph or initial labeling by consensus.
For example: (existence of blue node) or (graph is a star).



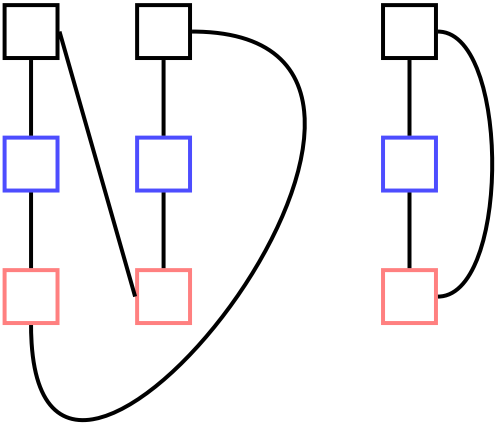
Nodes had to change answer.



Nodes had to count their neighbors.

Distinguish Cycles(?)

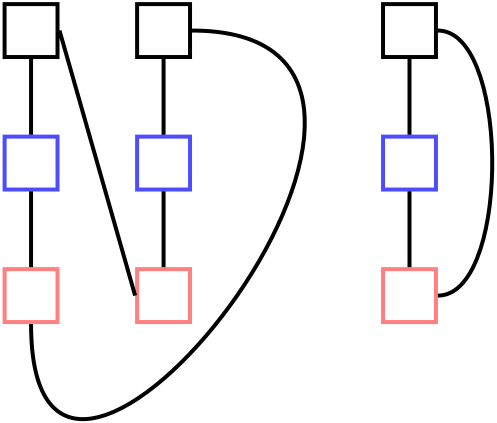
Is it possible to distinguish different length cycles?



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

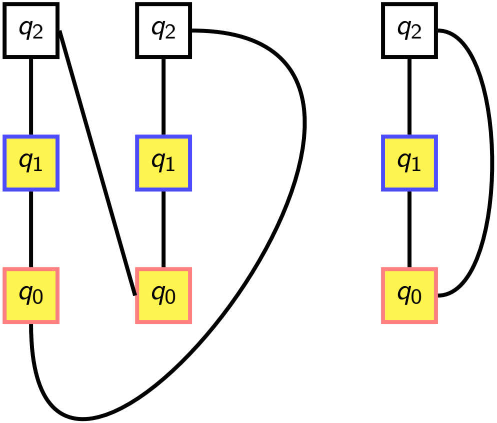
Another assumption comes into play: Nodes are anonymous.



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

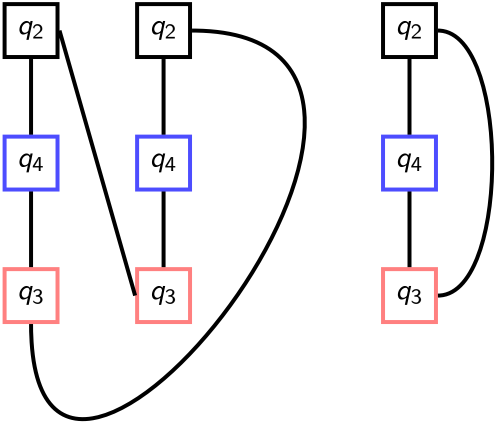
Another assumption comes into play: Nodes are anonymous.



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

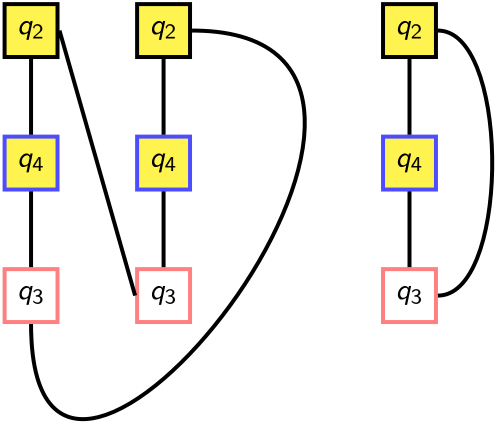
Another assumption comes into play: Nodes are anonymous.



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

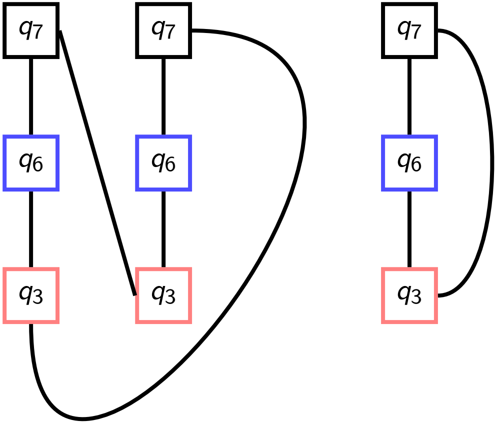
Another assumption comes into play: Nodes are anonymous.



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

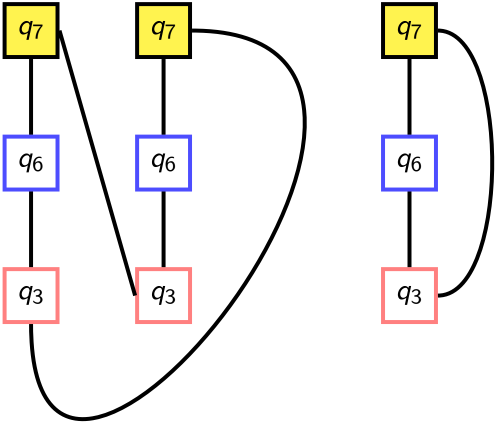
Another assumption comes into play: Nodes are anonymous.



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

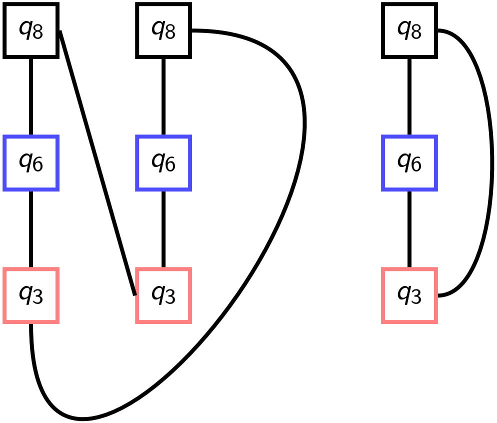
Another assumption comes into play: Nodes are anonymous.



Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

Another assumption comes into play: Nodes are anonymous.

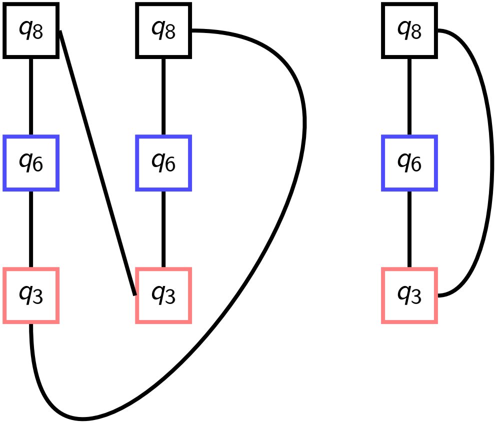


Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

Another assumption comes into play: Nodes are anonymous.

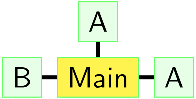
Not if same color nodes are always selected at same time. $\to$ *Fairness*.



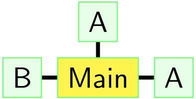
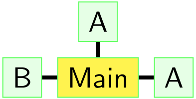
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness

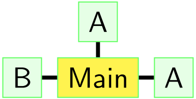
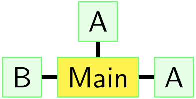
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. 			

The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. 			
Counting: $\{\{A, A, B\}\}$. 			

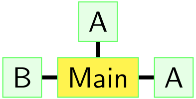
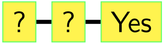
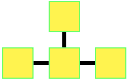
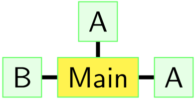
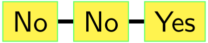
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. 	Halting: Nodes cannot change answer. ? – ? – Yes		
Counting: $\{\{A, A, B\}\}$. 			

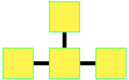
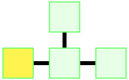
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. A B — Main — A	Halting: Nodes cannot change answer. ? — ? — Yes		
Counting: $\{\{A, A, B\}\}$. A B — Main — A	Stable Consensus: Nodes can change their answer. No — No — Yes		

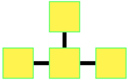
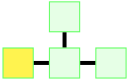
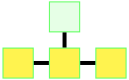
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. 	Halting: Nodes cannot change answer. 	Synchronous: 	
Counting: $\{\{A, A, B\}\}$. 	Stable Consensus: Nodes can change their answer. 		

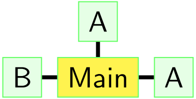
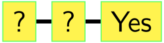
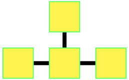
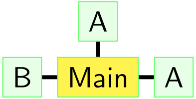
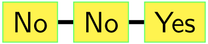
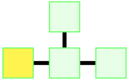
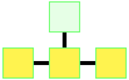
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. A B — Main — A	Halting: Nodes cannot change answer. ? — ? — Yes	Synchronous: 	
Counting: $\{\{A, A, B\}\}$. A B — Main — A	Stable Consensus: Nodes can change their answer. No — No — Yes	Exclusive: 	

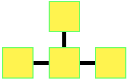
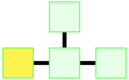
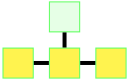
The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. A; B – Main – A	Halting: Nodes cannot change answer. ? – ? – Yes	Synchronous: 	
Counting: $\{\{A, A, B\}\}$. A; B – Main – A	Stable Consensus: Nodes can change their answer. No – No – Yes	Exclusive: 	
		Liberal: 	

The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. 	Halting: Nodes cannot change answer. 	Synchronous: 	Adversarial Scheduling: Every node v is selected infinitely often.
Counting: $\{\{A, A, B\}\}$. 	Stable Consensus: Nodes can change their answer. 	Exclusive: 	
		Liberal: 	

The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
Non-Counting: $\{A, B\}$. A B – Main – A	Halting: Nodes cannot change answer. ? – ? – Yes	Synchronous: 	Adversarial Scheduling: Every node v is selected infinitely often.
Counting: $\{\{A, A, B\}\}$. A B – Main – A	Stable Consensus: Nodes can change their answer. No – No – Yes	Exclusive: 	Pseudo-Stochastic: Every finite sequence of selections occurs infinitely often.
		Liberal: 	

Classification

Prior Research: Choice in Selection Aspect does not influence decision power.

Classification

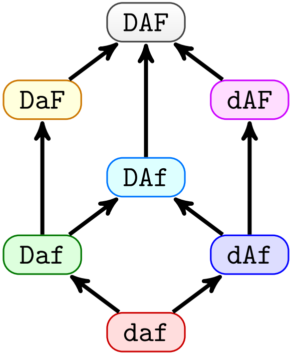
Prior Research: Choice in Selection Aspect does not influence decision power.

Detection	*Acceptance*	*Fairness*
`d`: non-counting	`a`: halting	`f`: adversarial scheduling
`D`: counting	`A`: stable consensus	`F`: pseudo-stochastic scheduling

Classification

Prior Research: Choice in Selection Aspect does not influence decision power.

Detection	*Acceptance*	*Fairness*
`d`: non-counting	`a`: halting	`f`: adversarial scheduling
`D`: counting	`A`: stable consensus	`F`: pseudo-stochastic scheduling



Labeling Properties

We determine the decision power for labeling properties.

Labeling Properties

We determine the decision power for labeling properties.

For every colored graph: color count $=$ function assigning to a color the number of nodes.

Labeling Properties

We determine the decision power for labeling properties.

For every colored graph: color count $=$ function assigning to a color the number of nodes.

A property φ is called a labeling property if it only depends on the color counts.

Labeling Properties

We determine the decision power for labeling properties.
For every colored graph: color count = function assigning to a color the number of nodes.
A property φ is called a labeling property if it only depends on the color counts.

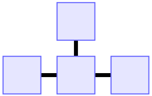
Non-Example	The graph is a star.

Labeling Properties

We determine the decision power for labeling properties.

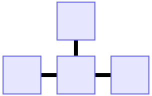
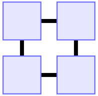
For every colored graph: color count $=$ function assigning to a color the number of nodes.

A property φ is called a labeling property if it only depends on the color counts.

Non-Example	The graph is a star.
Accepted	

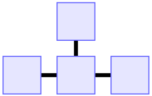
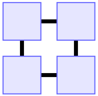
Labeling Properties

We determine the decision power for labeling properties.
For every colored graph: color count $=$ function assigning to a color the number of nodes.
A property φ is called a labeling property if it only depends on the color counts.

Non-Example	The graph is a star.
Accepted	
Rejected	

Labeling Properties

We determine the decision power for labeling properties.
For every colored graph: color count $=$ function assigning to a color the number of nodes.
A property φ is called a labeling property if it only depends on the color counts.

Non-Example	The graph is a star.
Accepted	
Rejected	

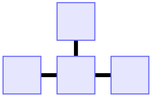
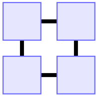
Example	There exists a blue node.

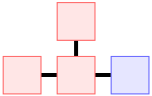
Labeling Properties

We determine the decision power for labeling properties.

For every colored graph: color count = function assigning to a color the number of nodes.

A property φ is called a labeling property if it only depends on the color counts.

Non-Example	The graph is a star.
Accepted	
Rejected	

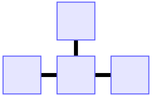
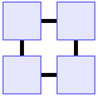
Example	There exists a blue node.
Accepted	

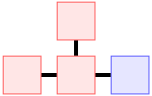
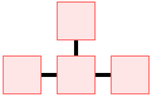
Labeling Properties

We determine the decision power for labeling properties.

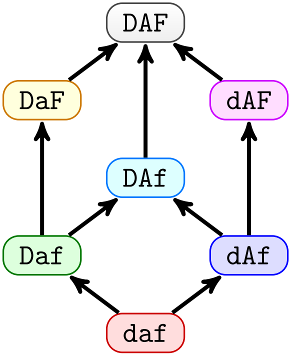
For every colored graph: color count = function assigning to a color the number of nodes.

A property φ is called a labeling property if it only depends on the color counts.

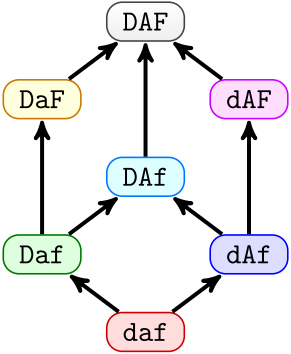
Non-Example	The graph is a star.
Accepted	
Rejected	

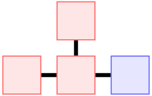
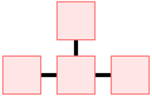
Example	There exists a blue node.
Accepted	
Rejected	

Our Results/Unrestricted Set of Graphs

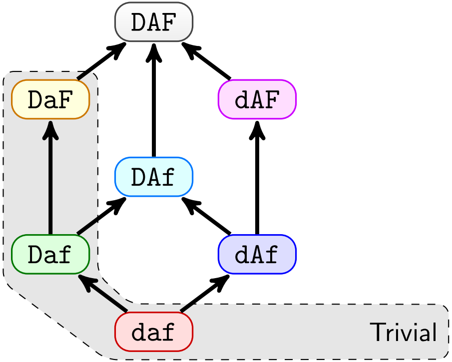


Our Results/Unrestricted Set of Graphs



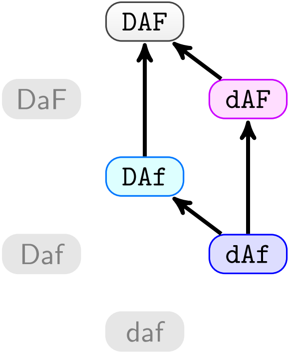
Property	There exists one blue node.
Accepted	
Rejected	

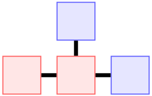
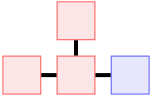
Our Results/Unrestricted Set of Graphs



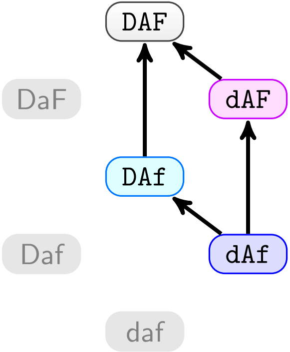
Class	Trivial
includes	True, False

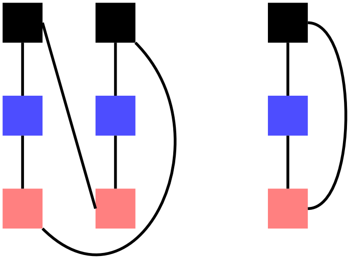
Our Results/Unrestricted Set of Graphs



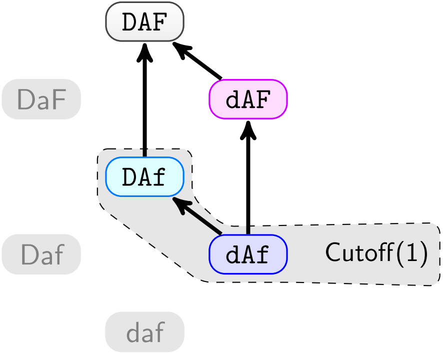
Property	There exist two blue nodes.
Accepted	
Rejected	

Our Results/Unrestricted Set of Graphs



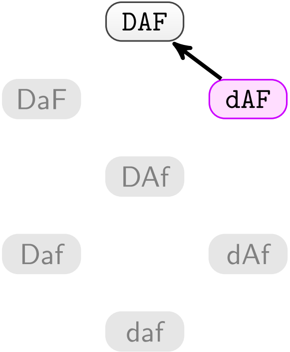


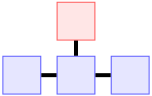
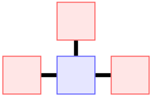
Our Results/Unrestricted Set of Graphs



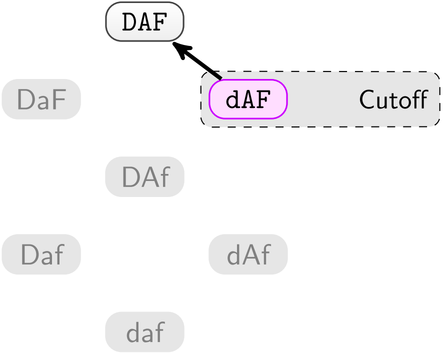
Class	Cutoff(1)
included	There exists both one red and one blue node.
not included	There exist two blue nodes.

Our Results/Unrestricted Set of Graphs



Property	There exist more blue nodes than red nodes.
Accepted	
Rejected	

Our Results/Unrestricted Set of Graphs



Class	Cutoff
included	There exist three blue nodes and there exist two red nodes.
not included	There exist more blue nodes than red nodes.

Our Results/Unrestricted Set of Graphs

DAF

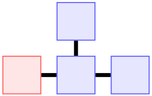
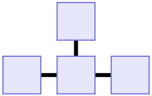
DaF dAF

DAf

Daf dAf

daf

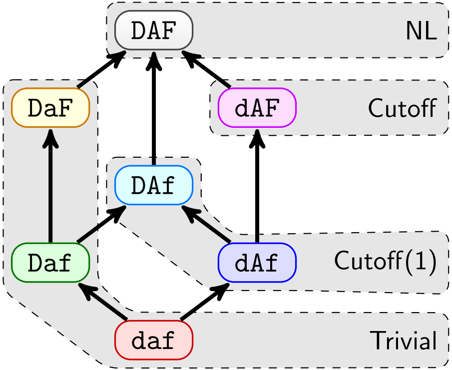
Where is the Limit? What about PRIMES?

Property	The number of blue nodes is a prime number.
Accepted	
Rejected	

Our Results/Unrestricted Set of Graphs

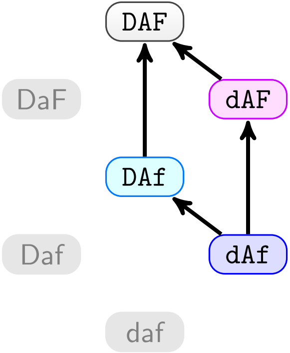
Class	NL = NSPACE($\log n$)
inputs	n blue nodes means input size $\Theta(n)$, i.e. input in unary!
includes	There exist more blue nodes than red nodes. The number of blue nodes is a prime number.

Our Results/Unrestricted Set of Graphs

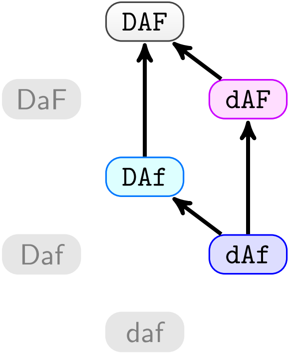


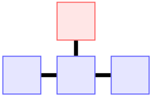
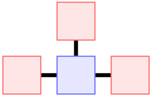
Class	Distinguishing Property
Cutoff(1)	There exists one blue node.
Cutoff	There exist two blue nodes.
NL	There exist more blue nodes than red nodes.

Our Results/k-Degree-Bounded Graphs

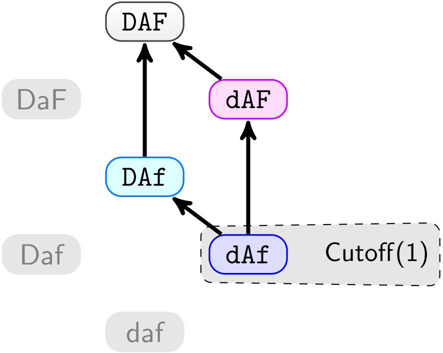


Our Results/k-Degree-Bounded Graphs



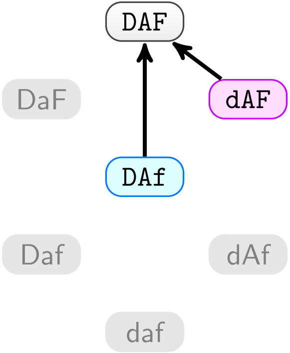
Property	There exist more blue nodes than red nodes.
Accepted	
Rejected	

Our Results/k-Degree-Bounded Graphs



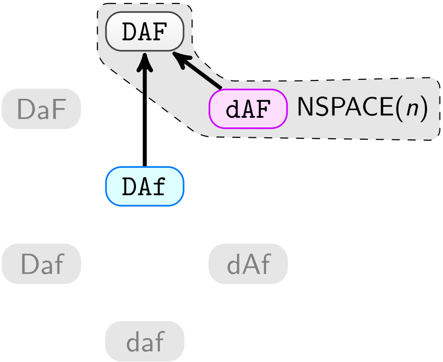
Class	Cutoff(1)
included	There exists both one red and one blue node.
not included	There are two blue nodes. There are more blue nodes than red nodes.

Our Results/k-Degree-Bounded Graphs



What is the new limit for the strongest model?
Has to be at least NL.

Our Results/k-Degree-Bounded Graphs



Class	NSPACE(n)
How huge?	gigantic. theoretical maximum.

Our Results/k-Degree-Bounded Graphs

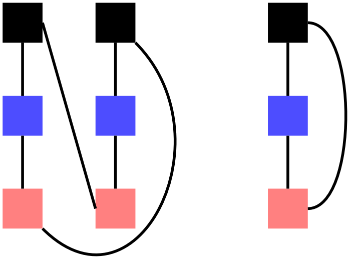
DAF

DaF dAF

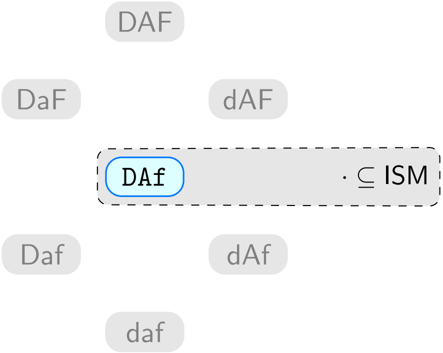
DAf

Daf dAf

daf



Our Results/k-Degree-Bounded Graphs



Class	Invariant under Scalar Multiplication
Definition	$\varphi \in \mathsf{ISM} \Leftrightarrow \forall \lambda \in \mathbb{N}_{>0} : \varphi(L) = \varphi(\lambda \cdot L)$.
included	There exist more blue nodes than red nodes.
not included	There exist two blue nodes.

Our Results/k-Degree-Bounded Graphs

DAF

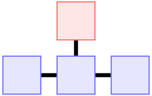
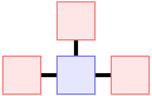
DaF dAF

DAf

Daf dAf

daf

Priority: Can we decide Majority with `DAf`?

Property	There exist more blue nodes than red nodes.
Accepted	
Rejected	

Our Results/k-Degree-Bounded Graphs

DAF

DaF dAF

`DAf` Maj $\subseteq \cdot$

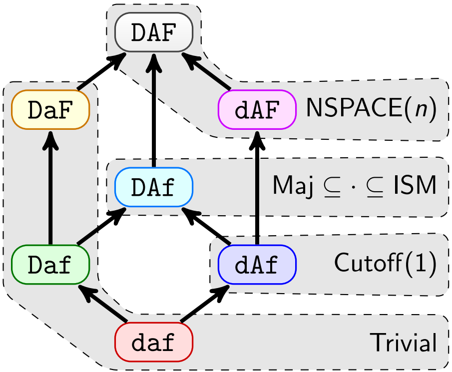
Daf dAf

daf

Yes!

Can also decide $\sum_i a_i x_i \geq 0$ with $a_i \in \mathbb{Z}$.

Our Results/k-Degree-Bounded Graphs



Class	Distinguishing Property
Cutoff(1)	There exists one blue node.
ISM	There exist more blue nodes than red nodes.
NSPACE(n)	The number of blue nodes is a prime number.

Thank you for your Attention!