

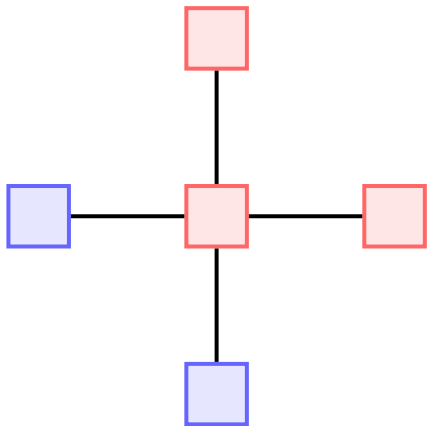
# Decision Power of Weak Asynchronous Models of Distributed Computing

Philipp Czerner   Roland Guttenberg   Javier Esparza   Martin Helfrich

Technische Universität München

# Common Ground and Differing Aspects

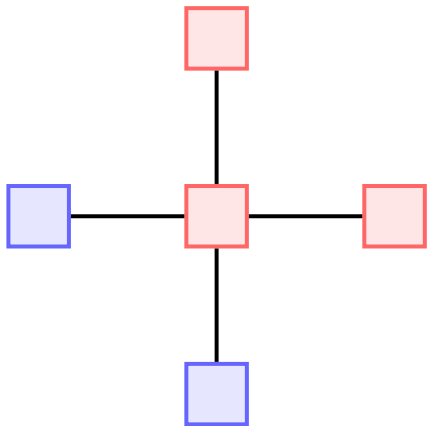
Every node executes **identical** **finite-state machine**.



# Common Ground and Differing Aspects

Every node executes **identical finite-state machine**.

Nodes **decide property** of graph or initial labeling **by consensus**.

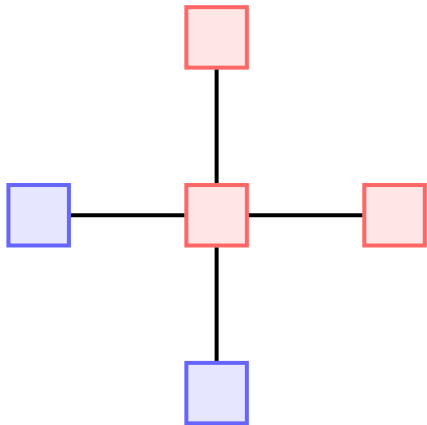


# Common Ground and Differing Aspects

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**).

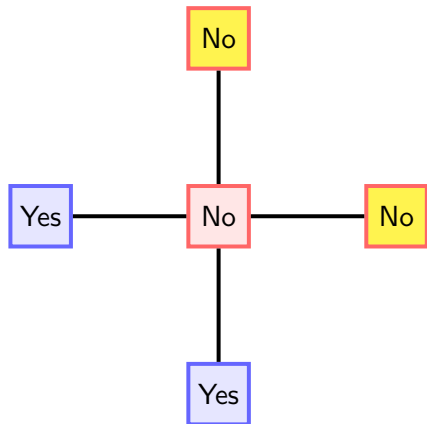


# Common Ground and Differing Aspects

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**).

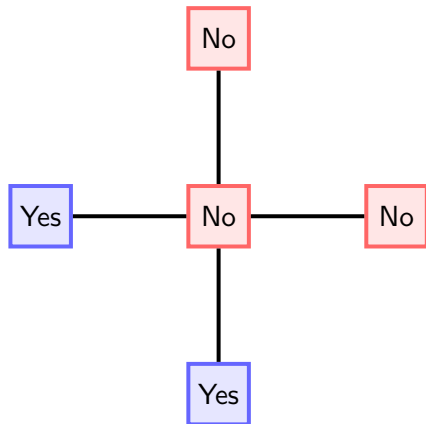


# Common Ground and Differing Aspects

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**).

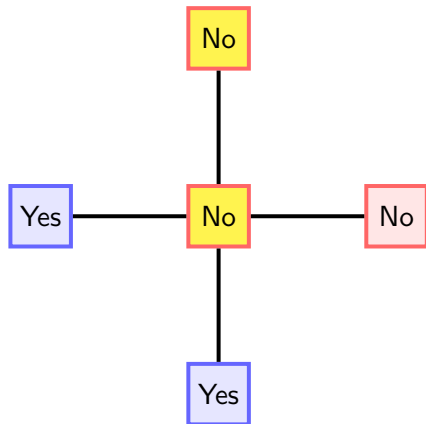


# Common Ground and Differing Aspects

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**).

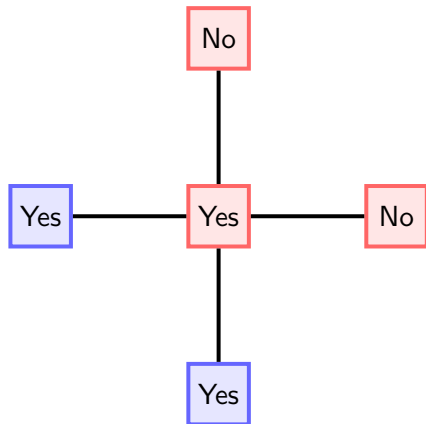


# Common Ground and Differing Aspects

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**).



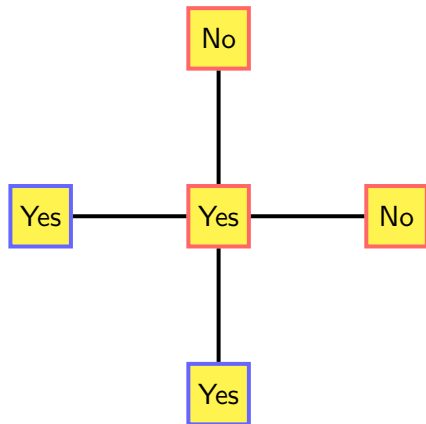


# Common Ground and Differing Aspects

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**).

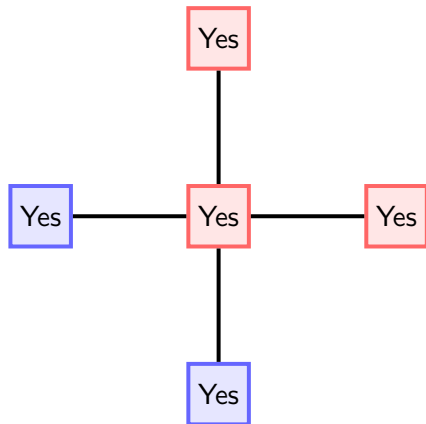


# Common Ground and Differing Aspects

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**).



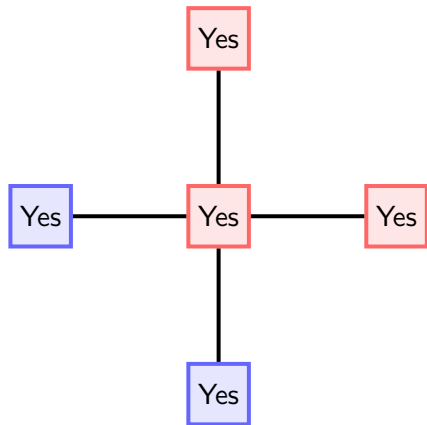
Nodes had to **change answer**.

# Common Ground and Differing Aspects

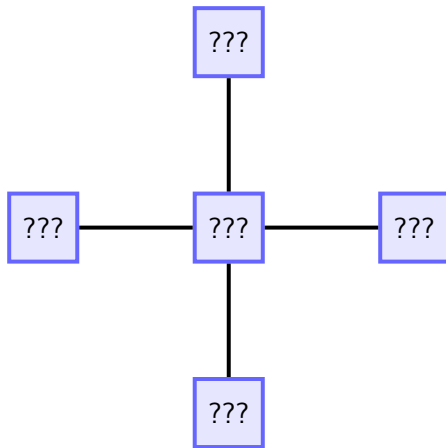
Every node executes **identical finite-state machine**.

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**.

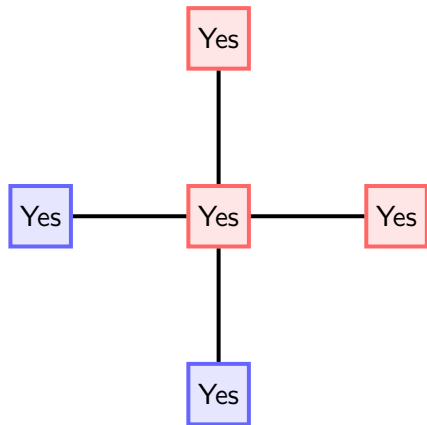


# Common Ground and Differing Aspects

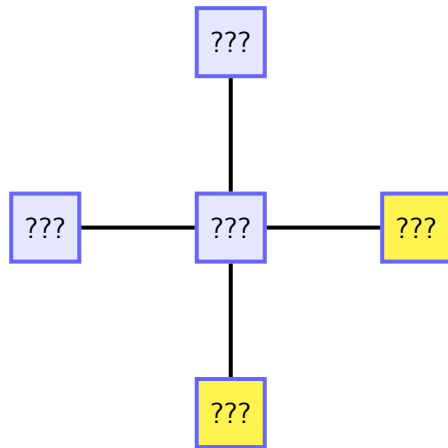
Every node executes **identical finite-state machine**.

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**.

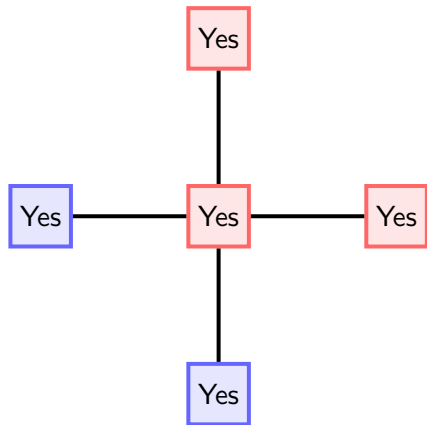


# Common Ground and Differing Aspects

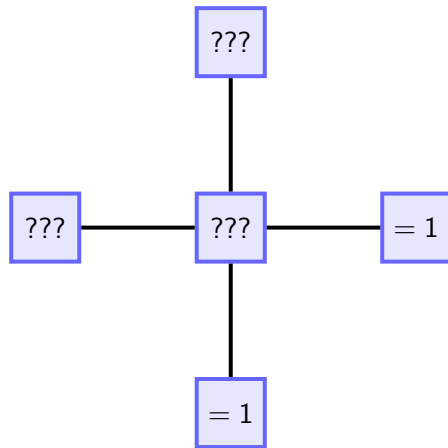
Every node executes **identical finite-state machine**.

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**.

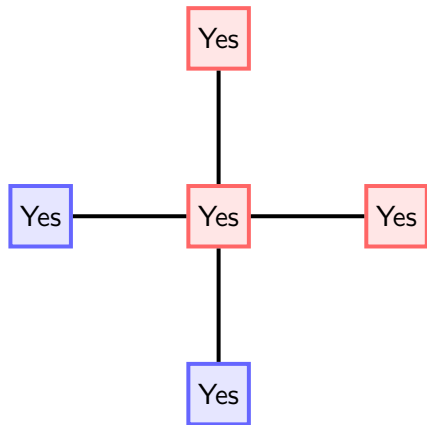


# Common Ground and Differing Aspects

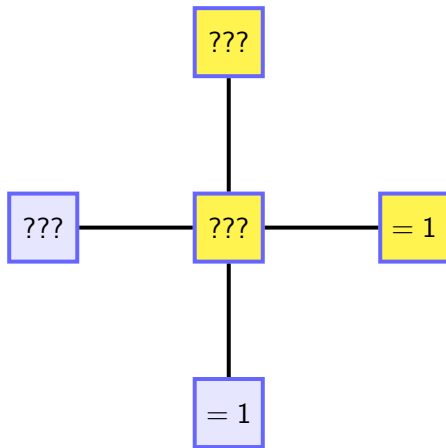
Every node executes **identical finite-state machine**.

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**.

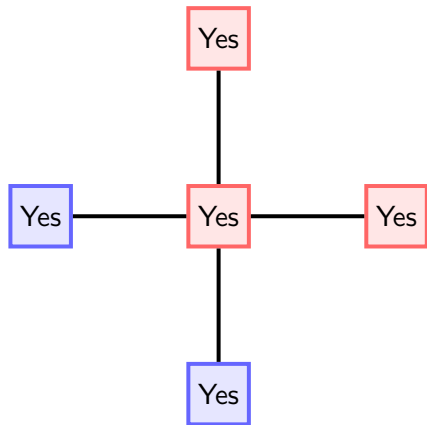


# Common Ground and Differing Aspects

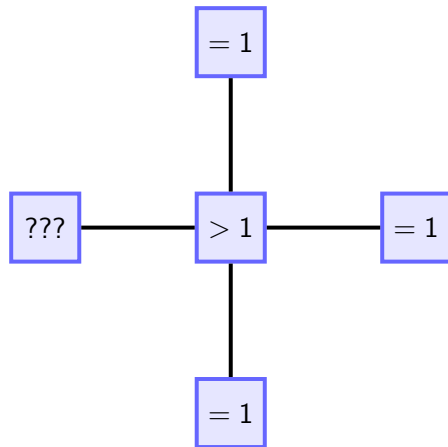
Every node executes **identical finite-state machine**.

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**.

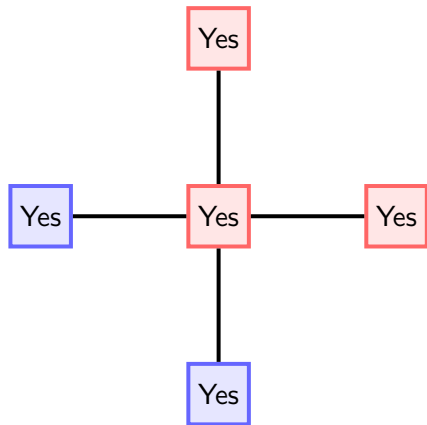


# Common Ground and Differing Aspects

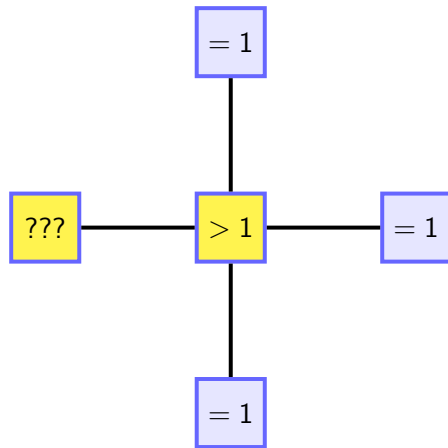
Every node executes **identical finite-state machine**.

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**.



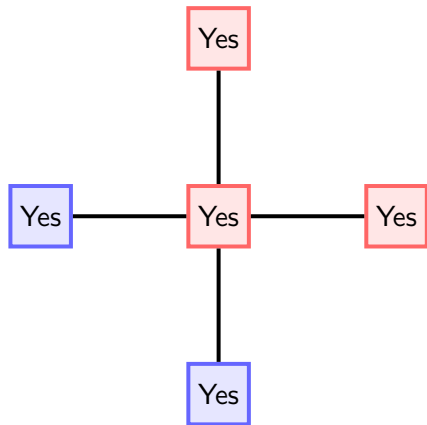


# Common Ground and Differing Aspects

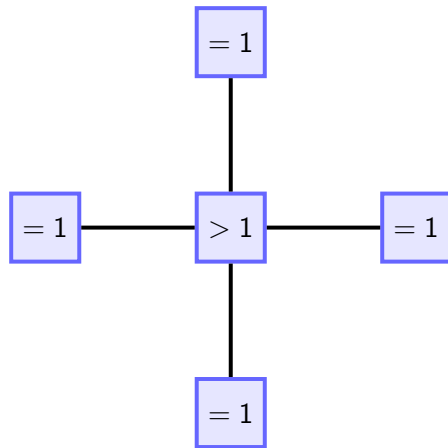
Every node executes **identical finite-state machine**.

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**.

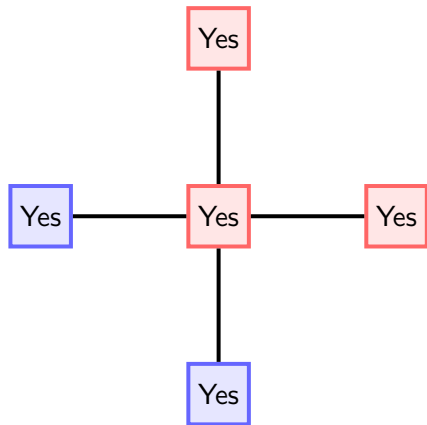


# Common Ground and Differing Aspects

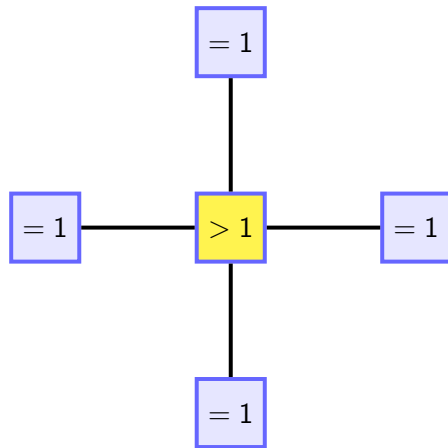
Every node executes **identical finite-state machine**.

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**.

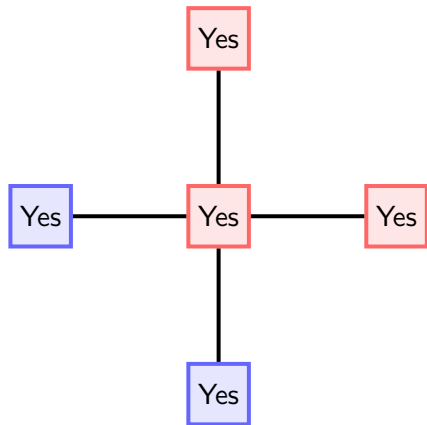


# Common Ground and Differing Aspects

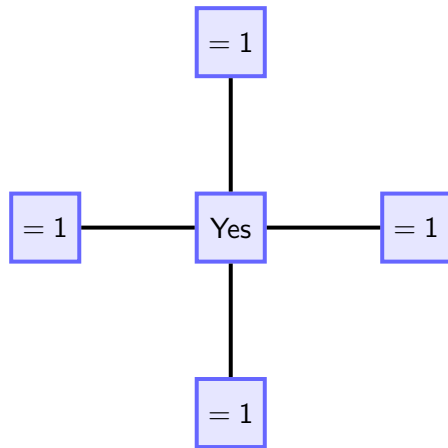
Every node executes **identical finite-state machine**.

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**.

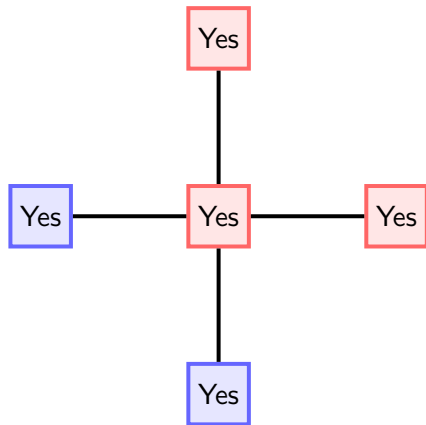


# Common Ground and Differing Aspects

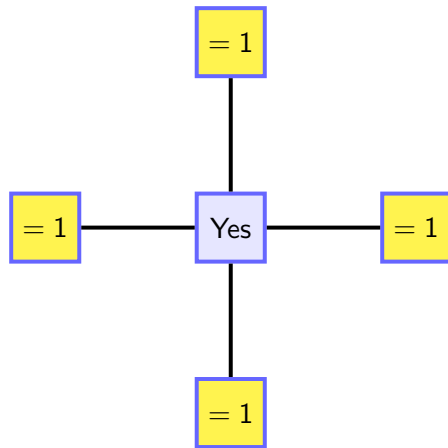
Every node executes **identical finite-state machine**.

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**.

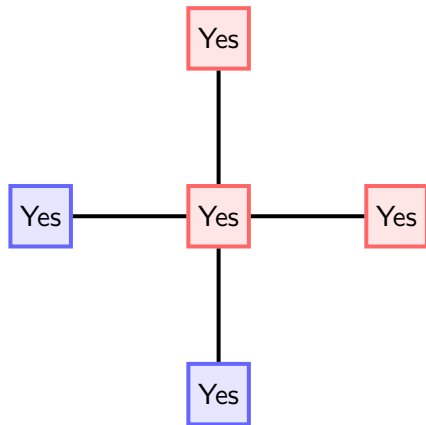


# Common Ground and Differing Aspects

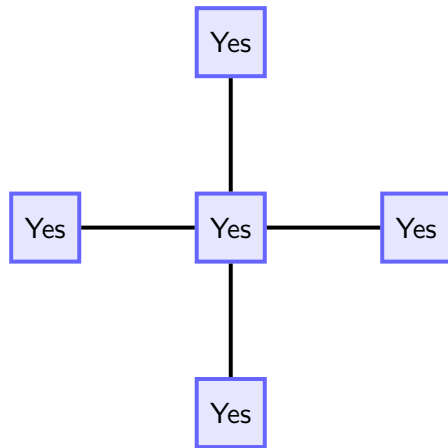
Every node executes **identical finite-state machine**.

Nodes **decide property** of graph or initial labeling **by consensus**.

For example: (existence of **blue node**) or (graph is a **star**).



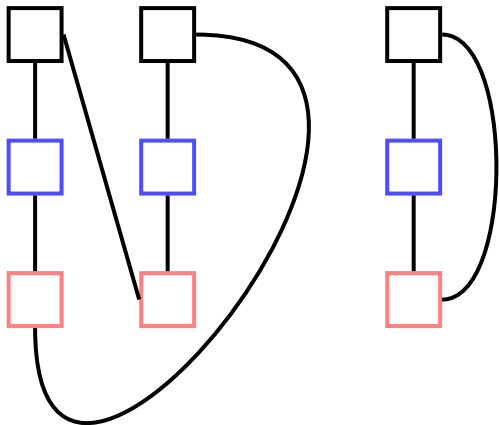
Nodes had to **change answer**.



Nodes had to **count** their neighbors.

# Distinguish Cycles(?)

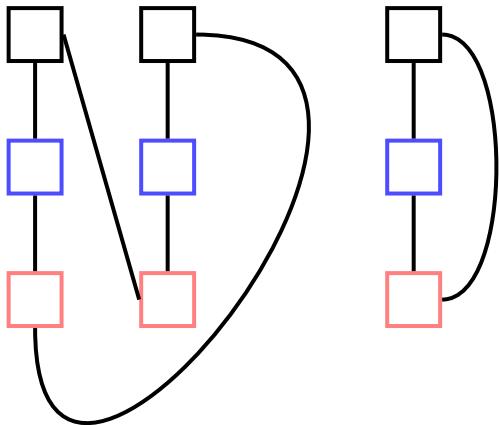
Is it possible to distinguish different length cycles?



# Distinguish Cycles(?)

Is it possible to distinguish **different length cycles**?

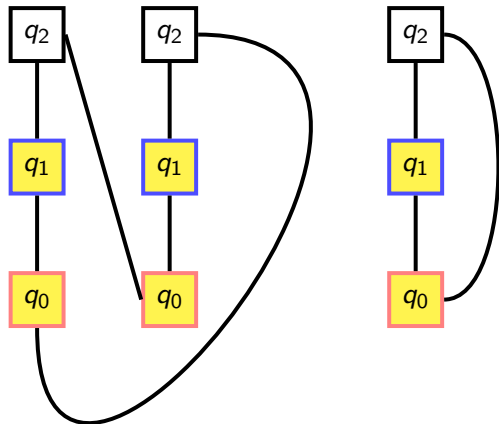
Another **assumption** comes into play: Nodes are **anonymous**.



# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

Another assumption comes into play: Nodes are anonymous.

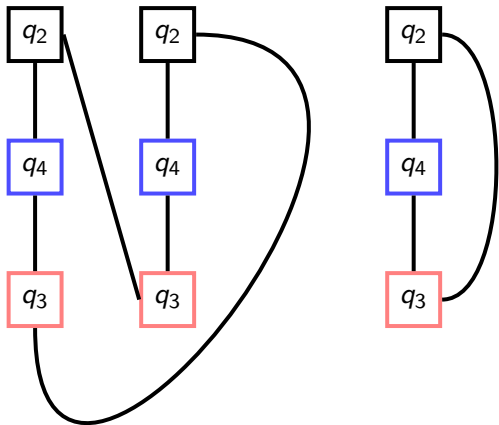




# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

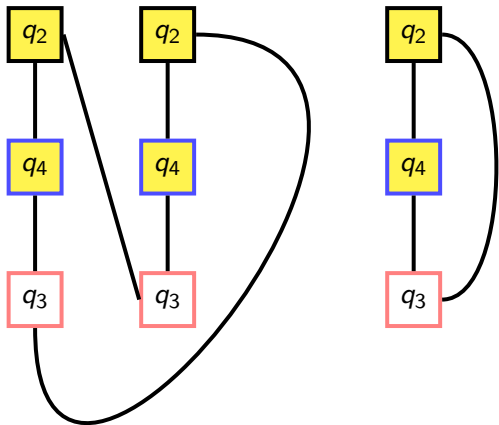
Another assumption comes into play: Nodes are anonymous.



# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

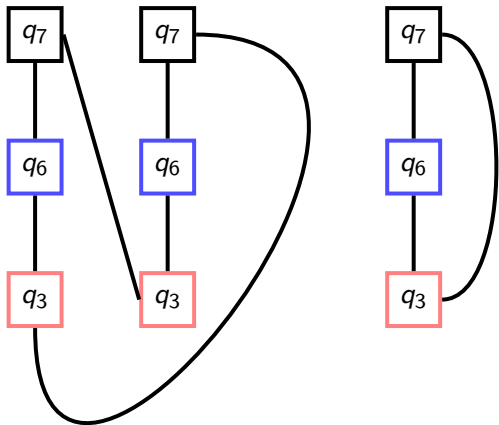
Another assumption comes into play: Nodes are anonymous.



# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

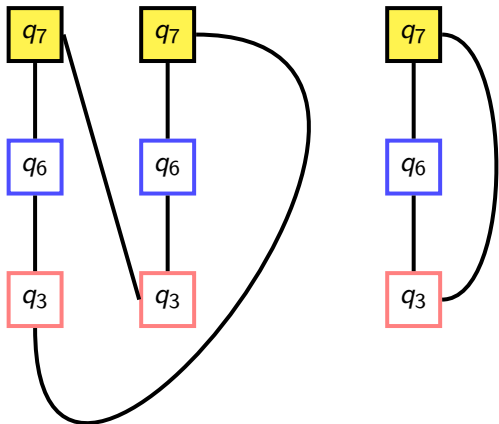
Another assumption comes into play: Nodes are anonymous.



# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

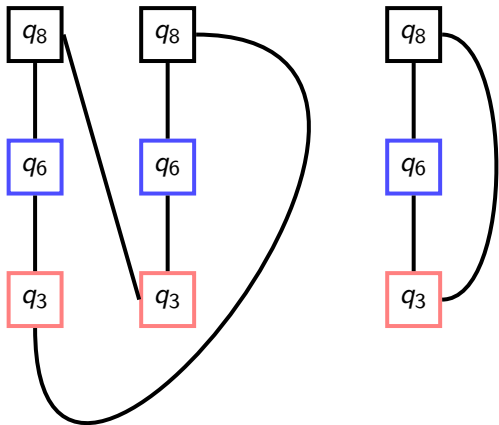
Another assumption comes into play: Nodes are anonymous.



# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

Another assumption comes into play: Nodes are anonymous.

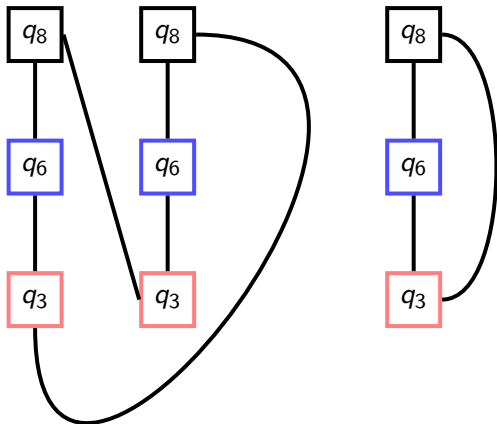


# Distinguish Cycles(?)

Is it possible to distinguish different length cycles?

Another assumption comes into play: Nodes are anonymous.

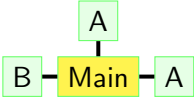
Not if same color nodes are always selected at same time. → *Fairness*.



# The Four Distinguishing Aspects of the Models

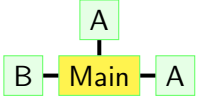
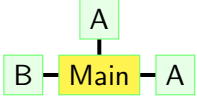
Detection	Acceptance	Selection	Fairness

# The Four Distinguishing Aspects of the Models

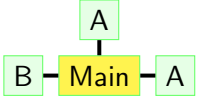
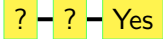
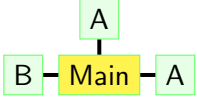
Detection	Acceptance	Selection	Fairness
<p data-bbox="132 308 538 350">Non-Counting: <math>\{A, B\}</math>.</p>  <pre data-bbox="132 355 393 487">graph TD; Main[Main] --- A1[A]; Main --- B[B]; Main --- A2[A];</pre>			



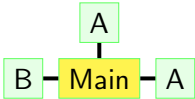
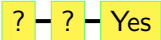
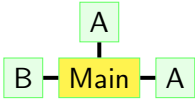
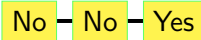
# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p>Non-Counting: <math>\{A, B\}</math>.</p> 			
<p>Counting: <math>\{\{A, A, B\}\}</math>.</p> 			

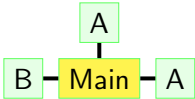
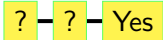

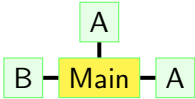
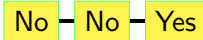
# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p><b>Non-Counting:</b> <math>\{A, B\}</math>.</p>  <pre>graph TD; A[A] --- Main[Main]; Main --- B[B]; Main --- A2[A];</pre>	<p><b>Halting:</b> Nodes cannot change answer.</p>  <pre>graph LR; Q1[?] --- Q2[?] --- Yes[Yes];</pre>		
<p><b>Counting:</b> <math>\{\{A, A, B\}\}</math>.</p>  <pre>graph TD; A[A] --- Main[Main]; Main --- B[B]; Main --- A2[A];</pre>			

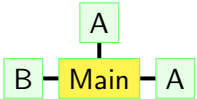
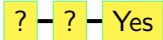

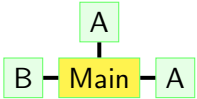
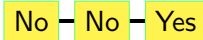
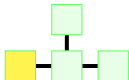
# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p><b>Non-Counting:</b> <math>\{A, B\}</math>.</p>  <p>A network diagram with a central node labeled 'Main' in a yellow box. It is connected to three other nodes: 'A' (green box) above, 'B' (green box) to the left, and 'A' (green box) to the right.</p>	<p><b>Halting:</b> Nodes cannot change answer.</p>  <p>A sequence of three yellow boxes connected by horizontal lines, containing the characters '?', '?', and 'Yes' from left to right.</p>		
<p><b>Counting:</b> <math>\{\{A, A, B\}\}</math>.</p>  <p>A network diagram with a central node labeled 'Main' in a yellow box. It is connected to three other nodes: 'A' (green box) above, 'B' (green box) to the left, and 'A' (green box) to the right.</p>	<p><b>Stable Consensus:</b> Nodes can change their answer.</p>  <p>A sequence of three yellow boxes connected by horizontal lines, containing the characters 'No', 'No', and 'Yes' from left to right.</p>		

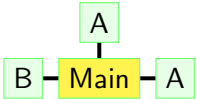
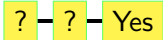
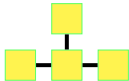
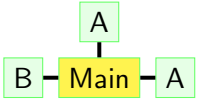
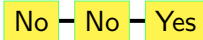
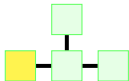
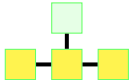
# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p><b>Non-Counting:</b> <math>\{A, B\}</math>.</p> 	<p><b>Halting:</b> Nodes cannot change answer.</p> 	<p><b>Synchronous:</b></p> 	
<p><b>Counting:</b> <math>\{\{A, A, B\}\}</math>.</p> 	<p><b>Stable Consensus:</b> Nodes can change their answer.</p> 		

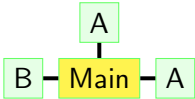
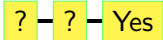
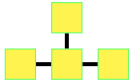
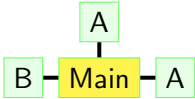
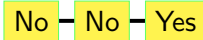
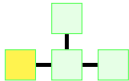
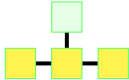
# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p><b>Non-Counting:</b> <math>\{A, B\}</math>.</p> 	<p><b>Halting:</b> Nodes cannot change answer.</p> 	<p><b>Synchronous:</b></p> 	
<p><b>Counting:</b> <math>\{\{A, A, B\}\}</math>.</p> 	<p><b>Stable Consensus:</b> Nodes can change their answer.</p> 	<p><b>Exclusive:</b></p> 	

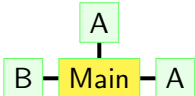
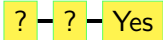
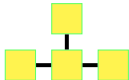
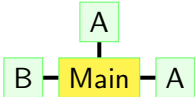
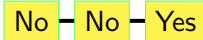
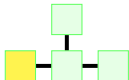

# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p><b>Non-Counting:</b> <math>\{A, B\}</math>.</p> 	<p><b>Halting:</b> Nodes cannot change answer.</p> 	<p><b>Synchronous:</b></p> 	
<p><b>Counting:</b> <math>\{\{A, A, B\}\}</math>.</p> 	<p><b>Stable Consensus:</b> Nodes can change their answer.</p> 	<p><b>Exclusive:</b></p> 	
		<p><b>Liberal:</b></p> 	

# The Four Distinguishing Aspects of the Models

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

# The Four Distinguishing Aspects of the Models

Detection	Acceptance	Selection	Fairness
<p><b>Non-Counting:</b> <math>\{A, B\}</math>.</p> 	<p><b>Halting:</b> Nodes cannot change answer.</p> 	<p><b>Synchronous:</b></p> 	<p><b>Adversarial Scheduling:</b> Every node <math>v</math> is selected infinitely often.</p>
<p><b>Counting:</b> <math>\{\{A, A, B\}\}</math>.</p> 	<p><b>Stable Consensus:</b> Nodes can change their answer.</p> 	<p><b>Exclusive:</b></p> 	<p><b>Pseudo-Stochastic:</b> Every finite sequence of selections occurs infinitely often.</p>
		<p><b>Liberal:</b></p> 	



# Classification

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

# Classification

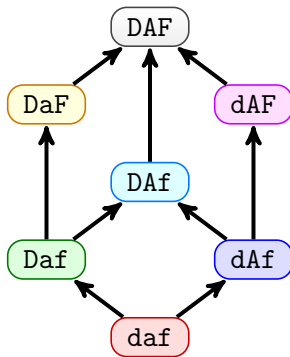
Prior Research: Choice in **Selection Aspect** does **not** influence **decision power**.

<i>Detection</i>	<i>Acceptance</i>	<i>Fairness</i>
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**.

<i>Detection</i>	<i>Acceptance</i>	<i>Fairness</i>
d: non-counting D: counting	a: halting A: stable consensus	f: adversarial scheduling 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  $\varphi$  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  $\varphi$  is called a **labeling property** if it only depends on the **color counts**.

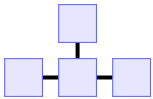
<b>Non-Example</b>	The graph is a <b>star</b> .


# 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  $\varphi$  is called a **labeling property** if it only depends on the **color counts**.

<b>Non-Example</b>	The graph is a <b>star</b> .
<b>Accepted</b>	

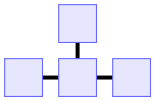




# Labeling Properties

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  $\varphi$  is called a **labeling property** if it only depends on the **color counts**.

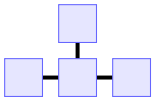

<b>Non-Example</b>	The graph is a <b>star</b> .
<b>Accepted</b>	
<b>Rejected</b>	


# 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  $\varphi$  is called a **labeling property** if it only depends on the **color counts**.

<b>Non-Example</b>	The graph is a <b>star</b> .
<b>Accepted</b>	
<b>Rejected</b>	

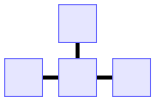

<b>Example</b>	There exists a <b>blue</b> node.

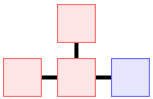
# Labeling Properties

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  $\varphi$  is called a **labeling property** if it only depends on the **color counts**.

Non-Example	The graph is a <b>star</b> .
Accepted	
Rejected	

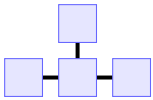

Example	There exists a <b>blue</b> node.
Accepted	

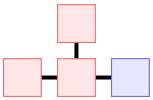
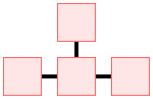
# Labeling Properties

We determine the **decision power** for **labeling properties**.

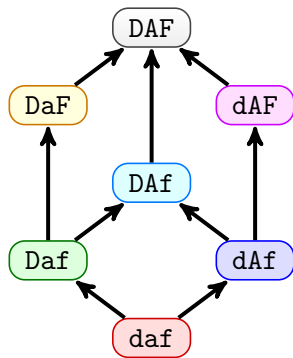
For every colored graph: **color count** = function assigning to a color the **number of nodes**.

A property  $\varphi$  is called a **labeling property** if it only depends on the **color counts**.

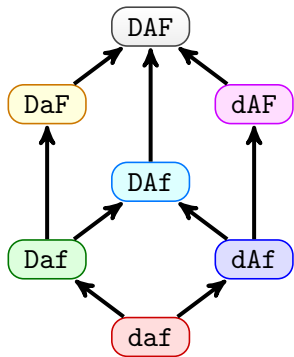
Non-Example	The graph is a <b>star</b> .
Accepted	
Rejected	

Example	There exists a <b>blue</b> node.
Accepted	
Rejected	

# Our Results/Unrestricted Set of Graphs

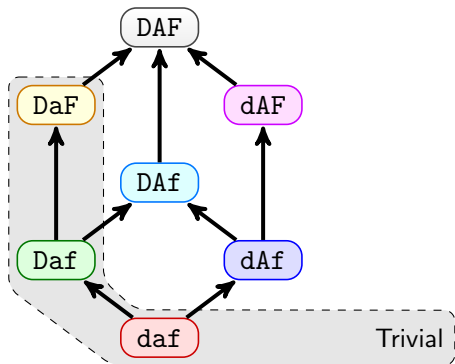


# Our Results/Unrestricted Set of Graphs



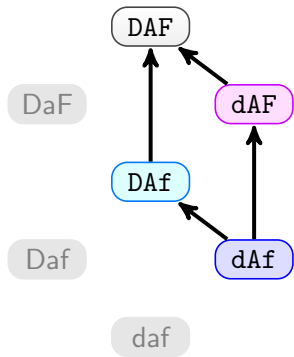
Property	There exists <b>one blue</b> node.
Accepted	
Rejected	

# Our Results/Unrestricted Set of Graphs



Class	Trivial
includes	True, False

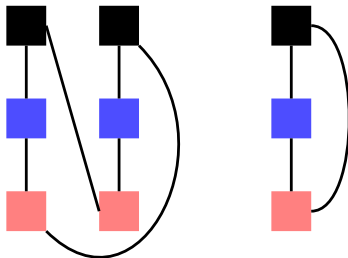
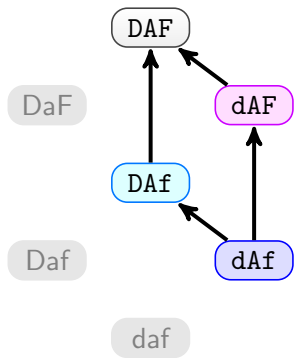
# Our Results/Unrestricted Set of Graphs



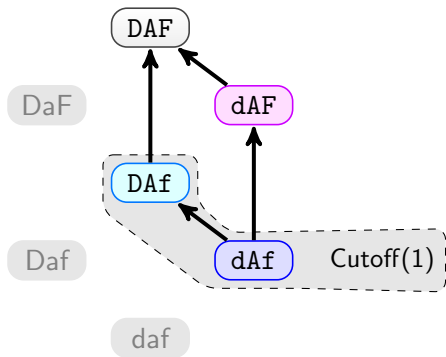
Property	There exist <b>two blue</b> nodes.
Accepted	
Rejected	



# Our Results/Unrestricted Set of Graphs

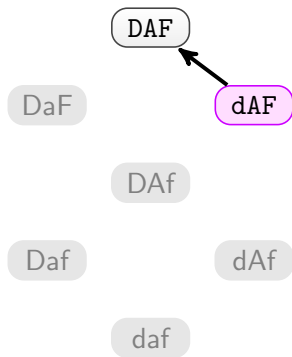


# Our Results/Unrestricted Set of Graphs



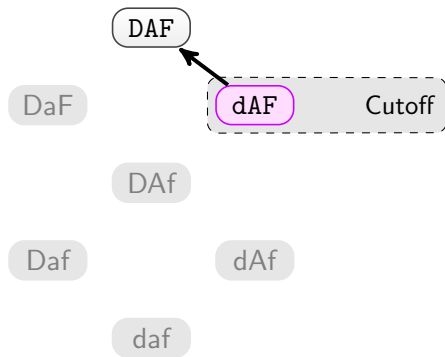
Class	Cutoff(1)
included	There exists both <b>one red</b> and <b>one blue</b> node.
not included	There exist <b>two blue</b> nodes.

# Our Results/Unrestricted Set of Graphs



Property	There exist <b>more blue</b> nodes than <b>red</b> nodes.
Accepted	
Rejected	

# Our Results/Unrestricted Set of Graphs

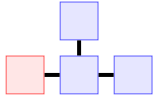
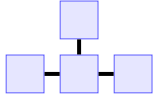


Class	Cutoff
included	There exist <b>three blue</b> nodes and there exist <b>two red</b> nodes.
not included	There exist <b>more blue</b> nodes than <b>red</b> nodes.

# Our Results/Unrestricted Set of Graphs



Where is the **Limit**? What about **PRIMES**?

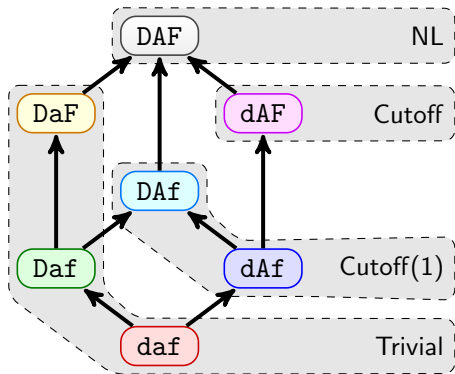
Property	The number of <b>blue</b> nodes is a <b>prime number</b> .
Accepted	
Rejected	

# Our Results/Unrestricted Set of Graphs



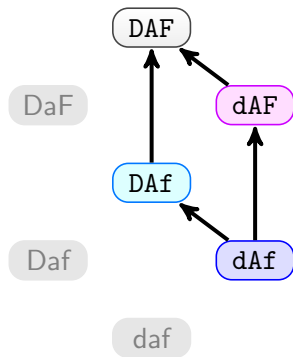
Class	$NL = NSPACE(\log n)$
inputs	$n$ blue nodes means input size $\Theta(n)$ , i.e. input in unary!
includes	There exist more blue nodes than red nodes. The number of blue nodes is a prime number.

# Our Results/Unrestricted Set of Graphs



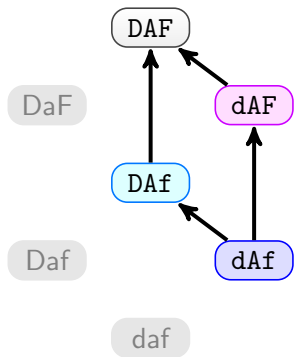
Class	Distinguishing Property
Cutoff(1)	There exists <b>one blue</b> node.
Cutoff	There exist <b>two blue</b> nodes.
NL	There exist <b>more blue</b> nodes than <b>red</b> nodes.

# Our Results/ $k$ -Degree-Bounded Graphs



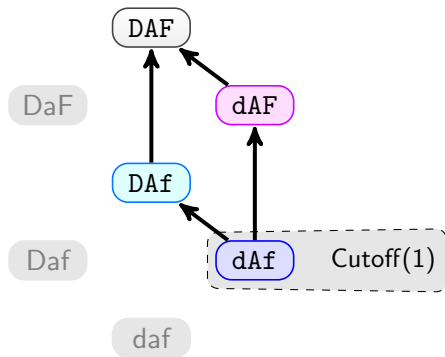


# Our Results/ $k$ -Degree-Bounded Graphs



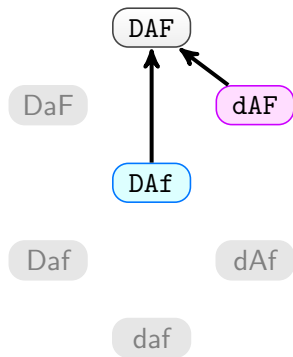
Property	There exist <b>more blue</b> nodes than <b>red</b> nodes.
Accepted	<p>A graph with one red node at the top and three blue nodes below it, connected in a T-shape.</p>
Rejected	<p>A graph with one red node at the top, one blue node in the middle, and two red nodes at the bottom, connected in a T-shape.</p>

# Our Results/ $k$ -Degree-Bounded Graphs



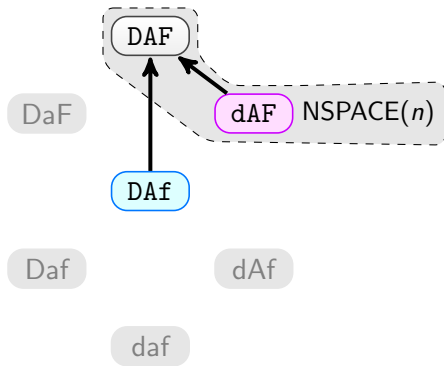
Class	Cutoff(1)
included	There exists both <b>one red</b> and <b>one blue</b> node.
not included	There are <b>two blue</b> nodes. There are <b>more blue</b> nodes <b>than red</b> nodes.

# Our Results/ $k$ -Degree-Bounded Graphs



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

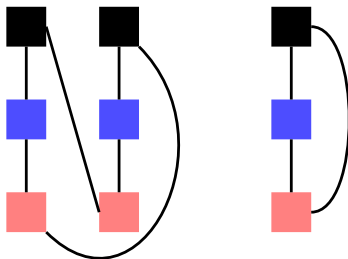
# Our Results/ $k$ -Degree-Bounded Graphs



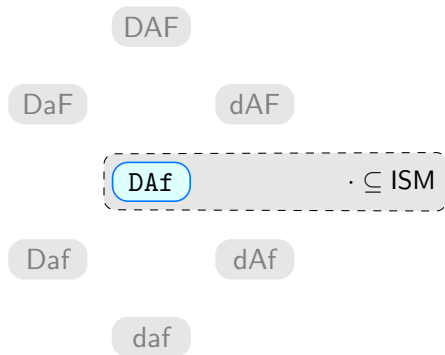
Class	$\text{NSPACE}(n)$
How huge?	gigantic. theoretical maximum.

# 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	$\varphi \in \text{ISM} \Leftrightarrow \forall \lambda \in \mathbb{N}_{>0} : \varphi(L) = \varphi(\lambda \cdot L).$
included	There exist <b>more blue</b> nodes <b>than red</b> nodes.
not included	There exist <b>two blue</b> nodes.

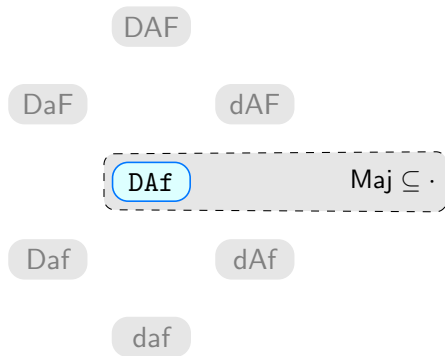
# Our Results/ $k$ -Degree-Bounded Graphs



Priority: Can we decide Majority with DAF?

Property	There exist more blue nodes than red nodes.
Accepted	<pre>graph TD; R[Red] --- B1[Blue]; B1 --- B2[Blue]; B1 --- B3[Blue];</pre>
Rejected	<pre>graph TD; R1[Red] --- B[Blue]; B --- R2[Red]; B --- R3[Red];</pre>

# Our Results/ $k$ -Degree-Bounded Graphs

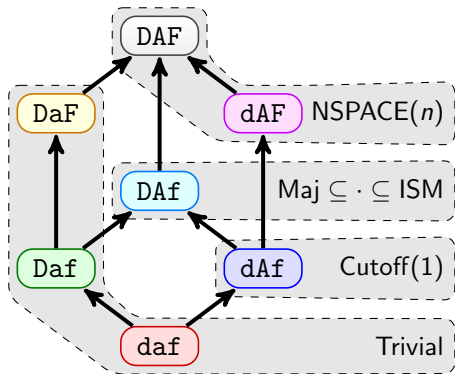


Yes!

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 <b>one blue</b> node.
ISM	There exist <b>more blue</b> nodes <b>than red</b> nodes.
$NSPACE(n)$	The number of <b>blue</b> nodes is a <b>prime number</b> .

Thank you for your Attention!

