Stable Dinner Party Seating Arrangements

Damien Berriaud
Andrei Constantinescu
Roger Wattenhofer



Distributed Computing Group
ETH zürich

1.

A chaotic dinner

When the round table wreaks havoc

Setting the scene

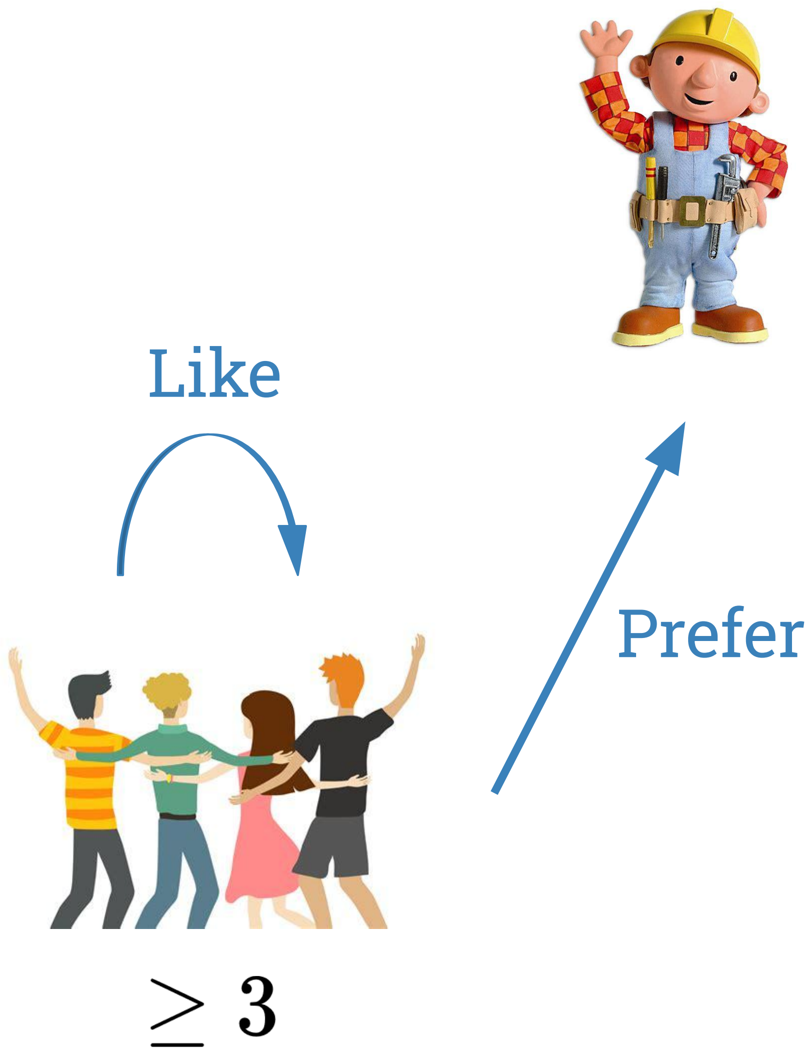
≥ 3

Setting the scene

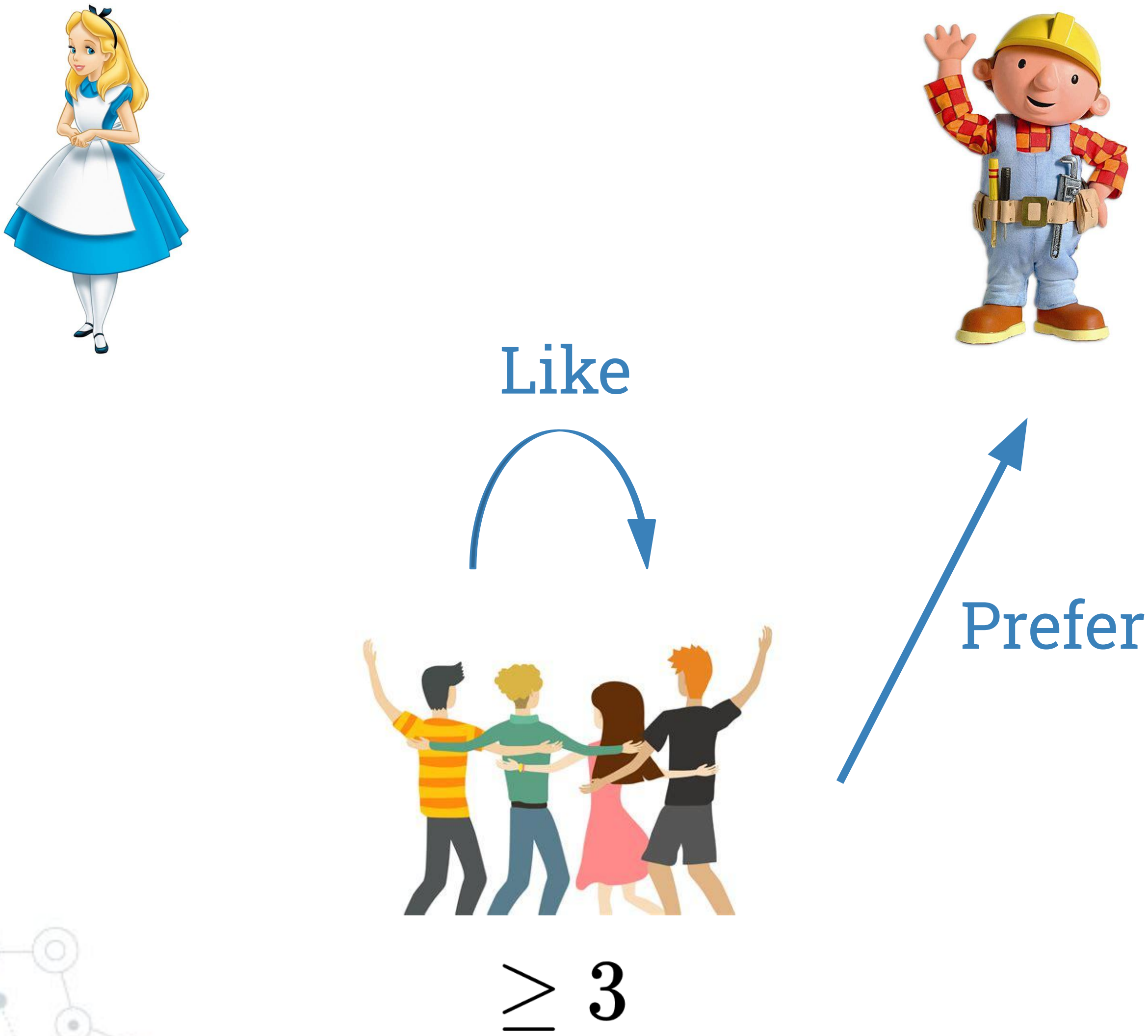
Like

$$\geq 3$$

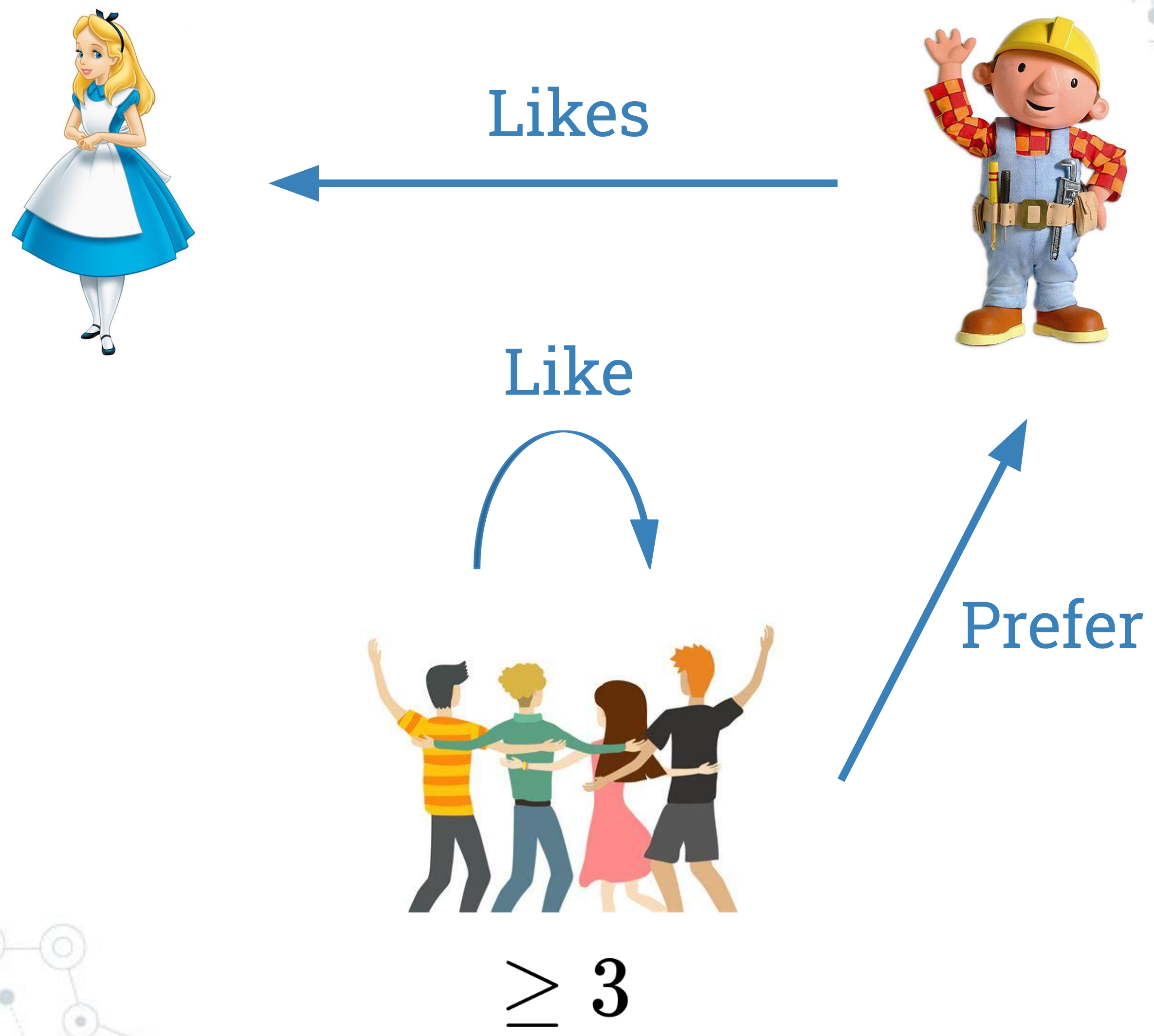
Setting the scene



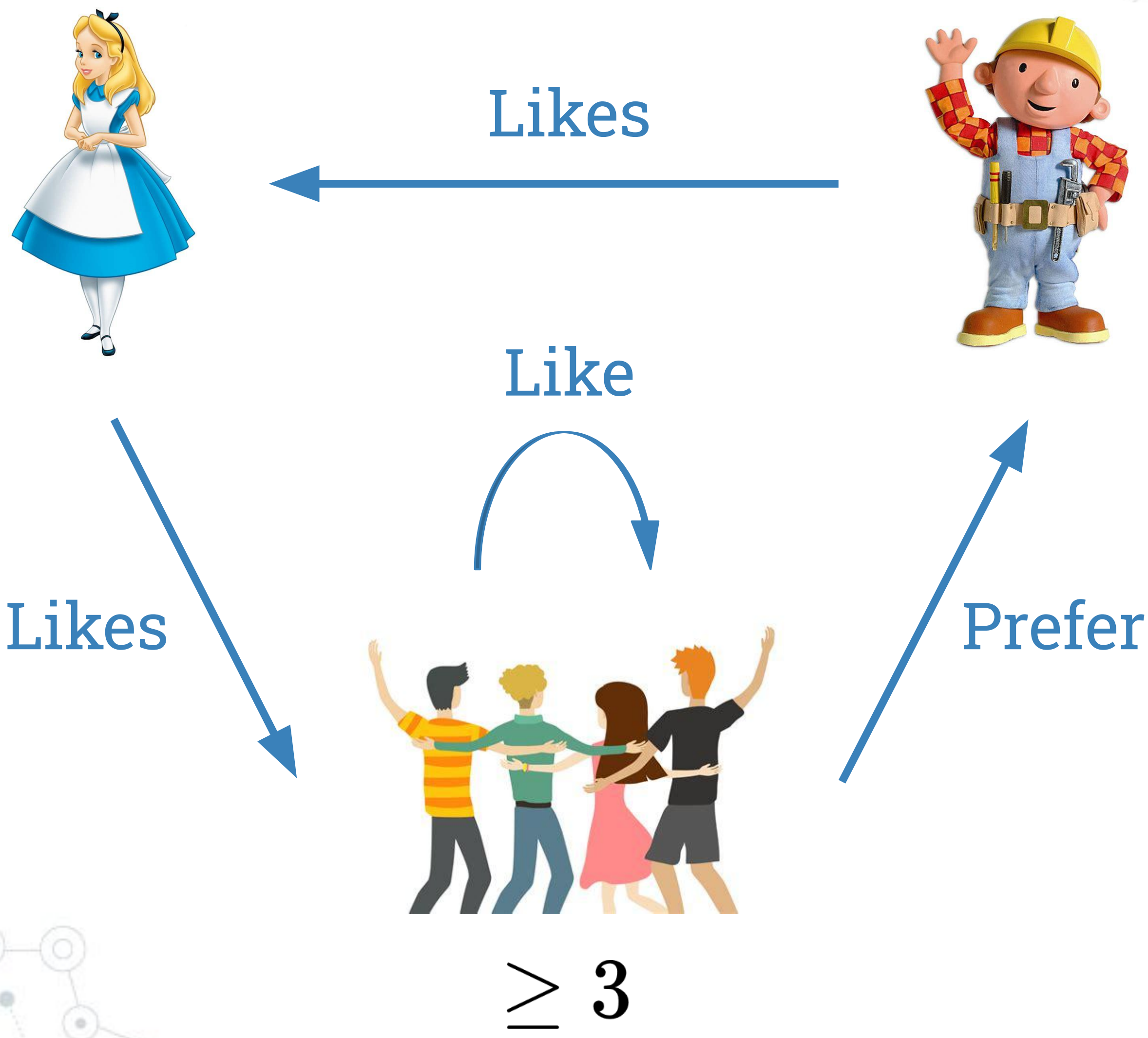
Setting the scene



Setting the scene

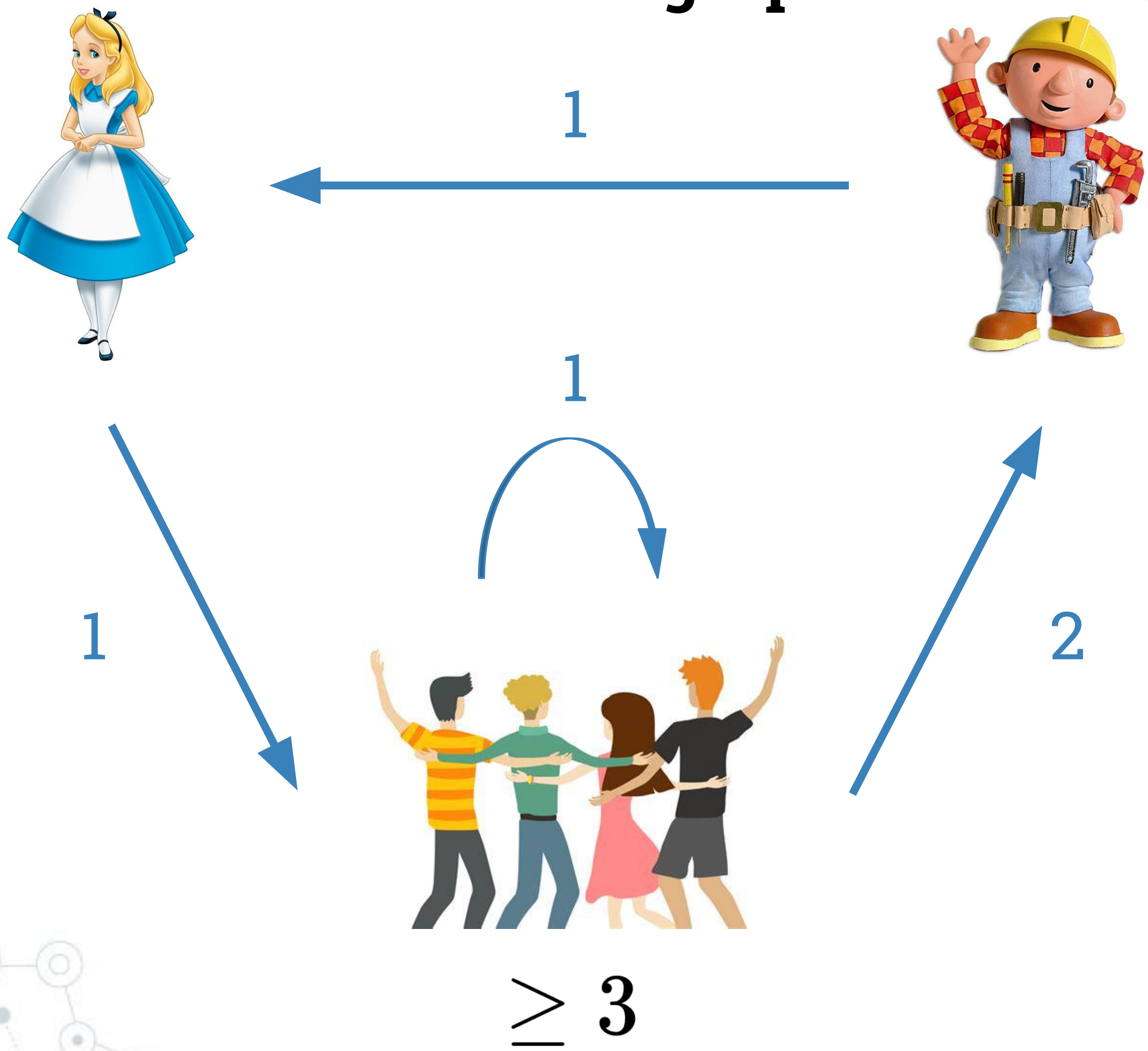


Setting the scene



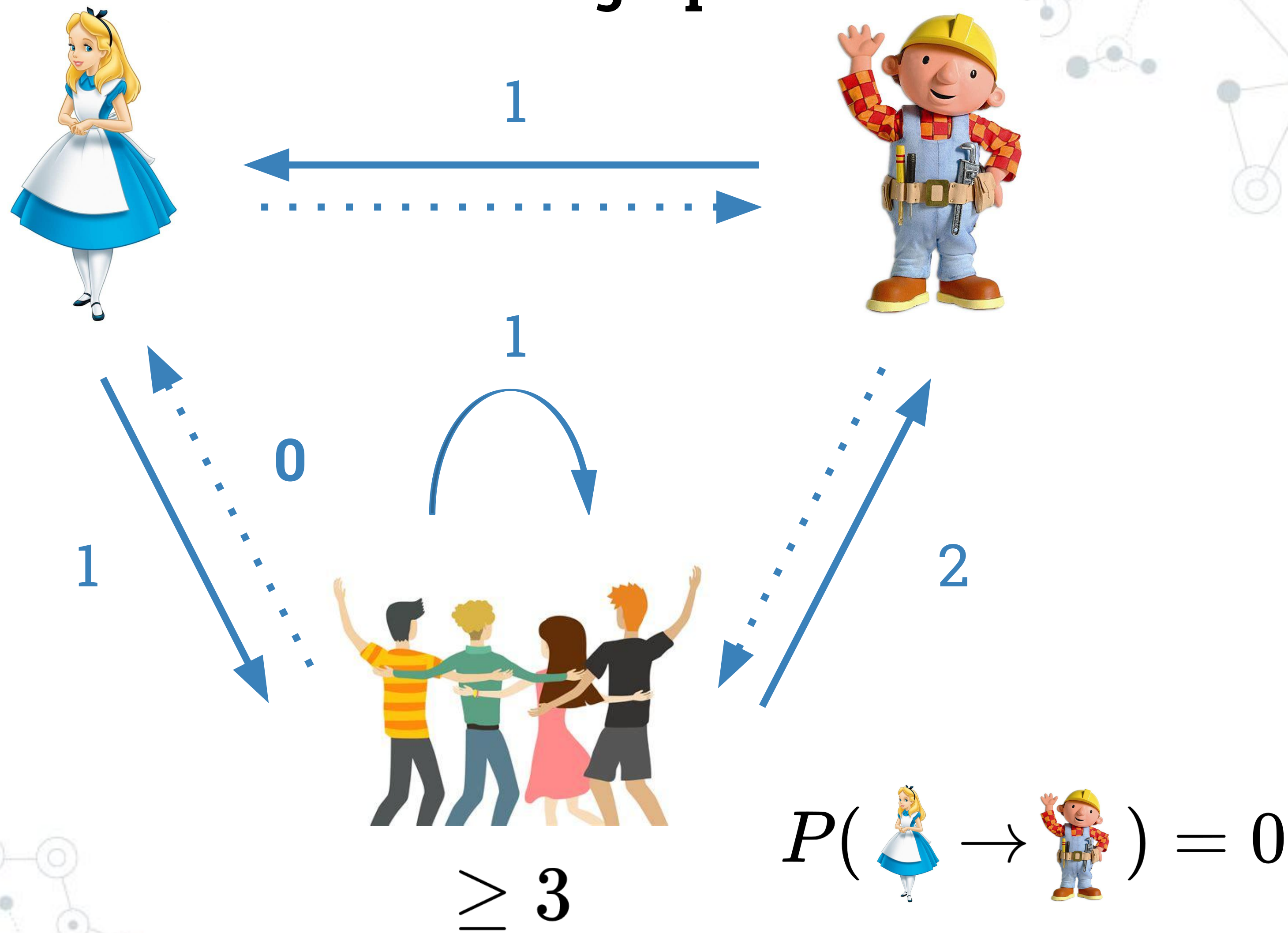
Setting the scene

Preference graph



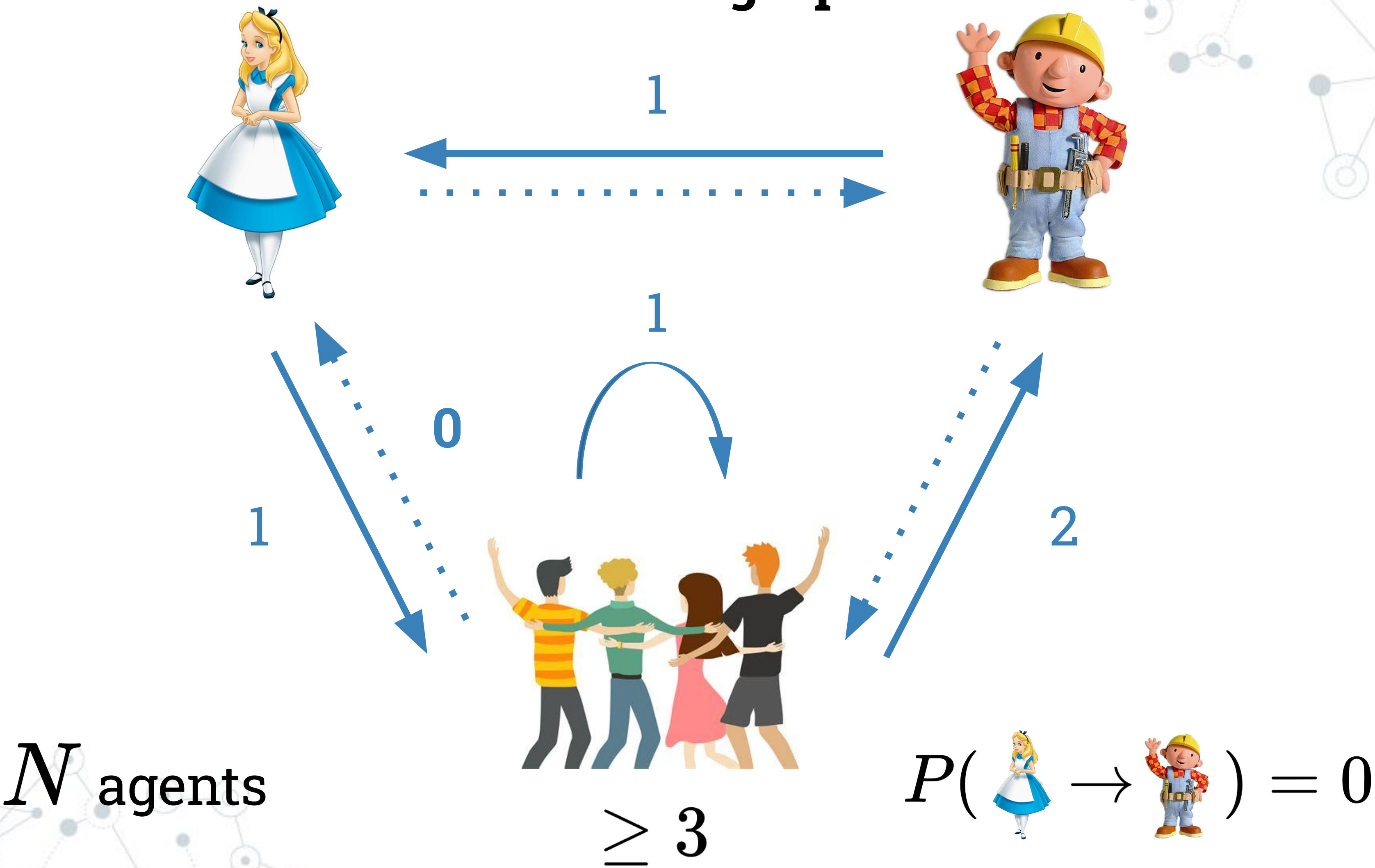
Setting the scene

Preference graph

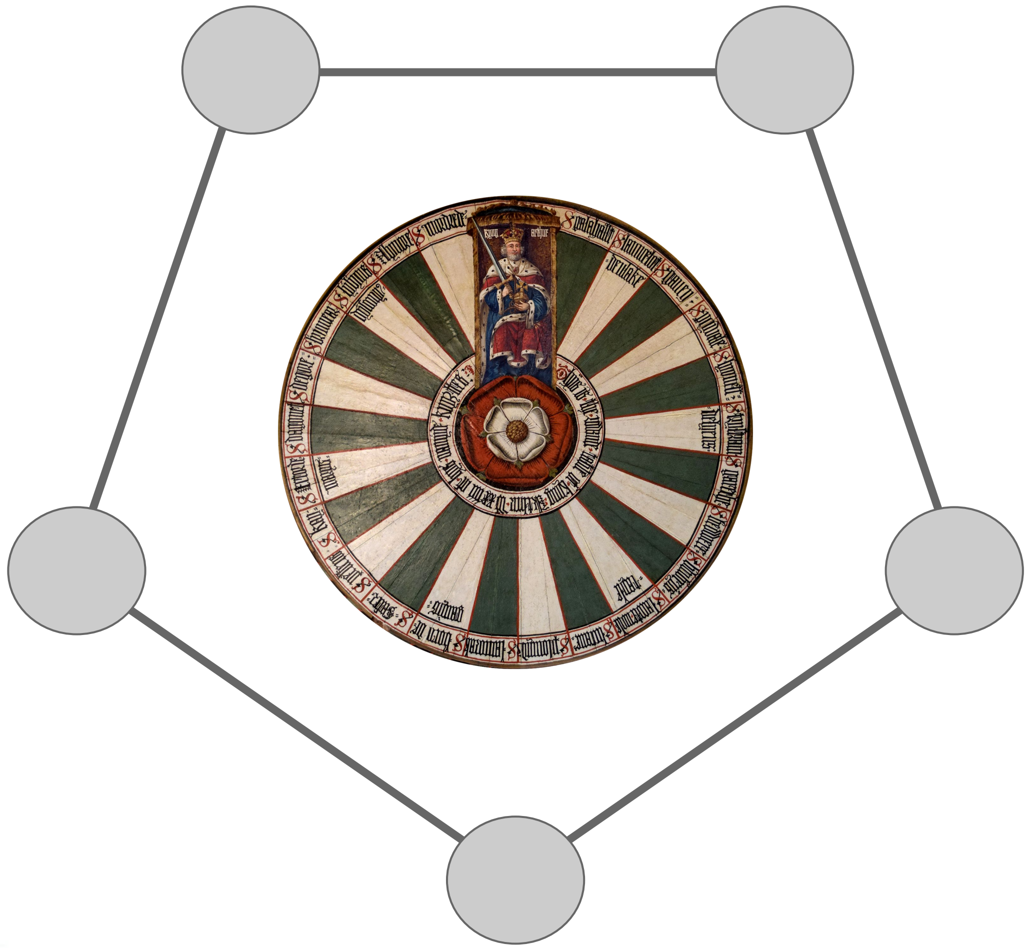


Setting the scene

Preference graph



Setting the scene



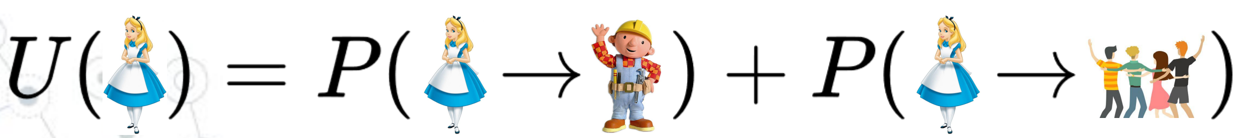
N vertices **seating graph**

Setting the scene

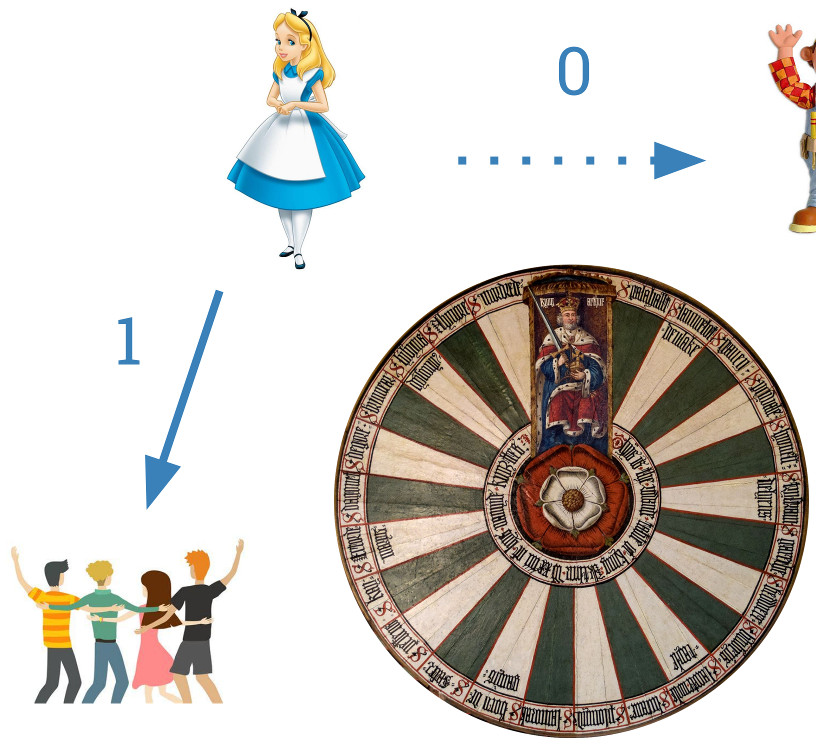
Utility: Additive on neighbours

Setting the scene

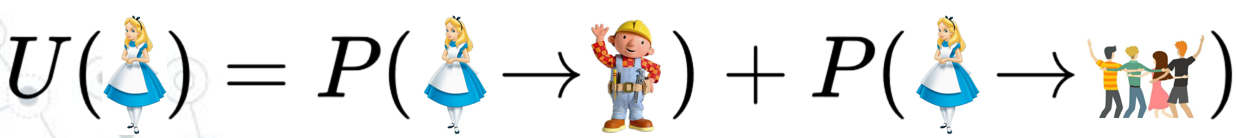
Utility: Additive on neighbours



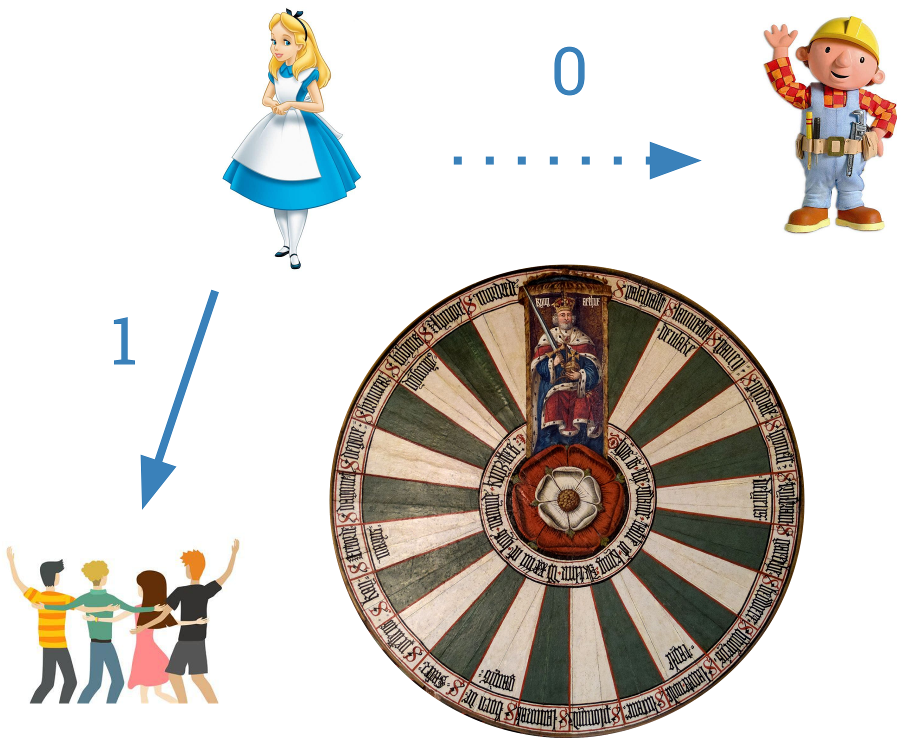
Setting the scene



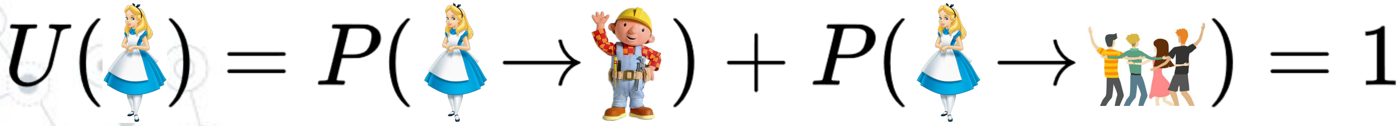
Utility: Additive on neighbours



Setting the scene



Utility: Additive on neighbours



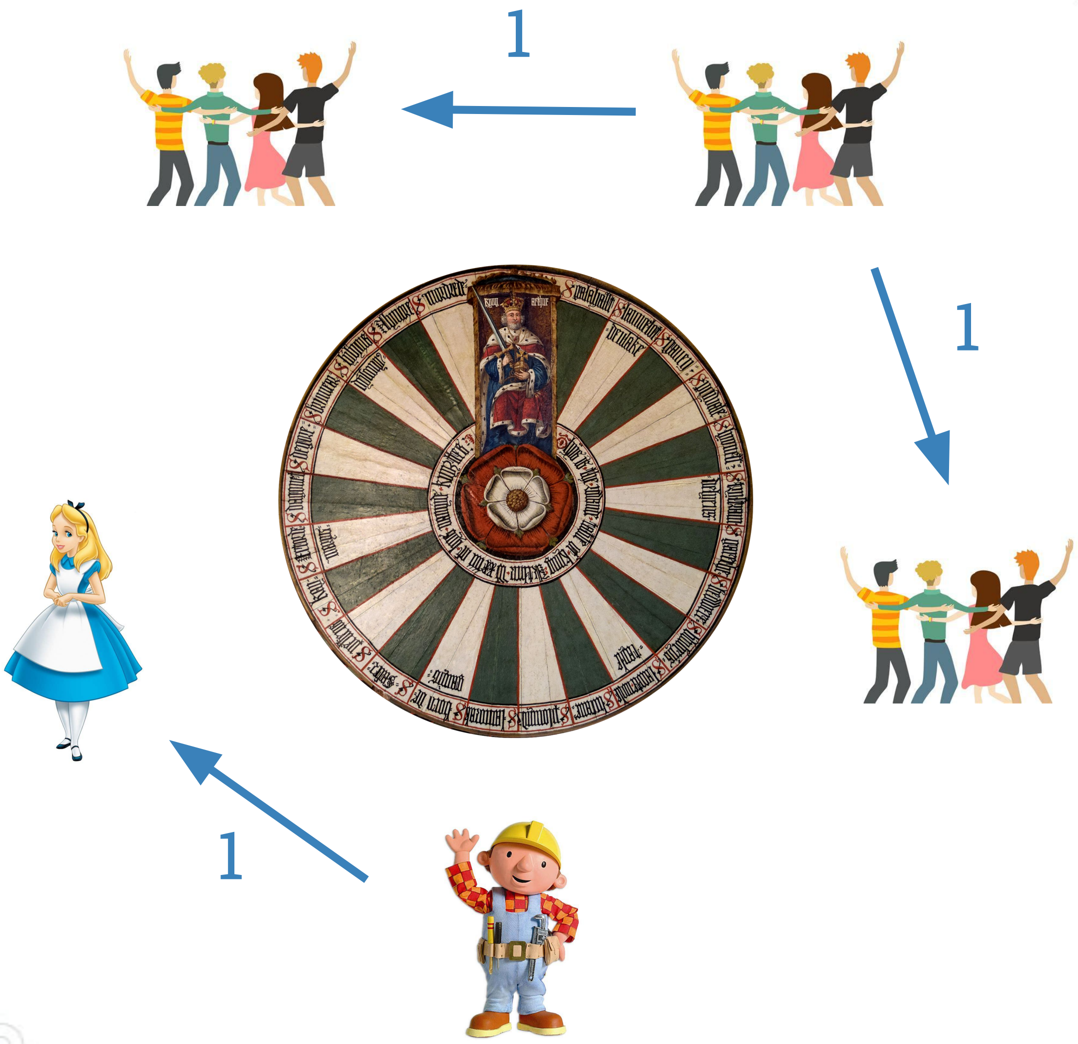
Chaos? Run...

Chaos? Run...

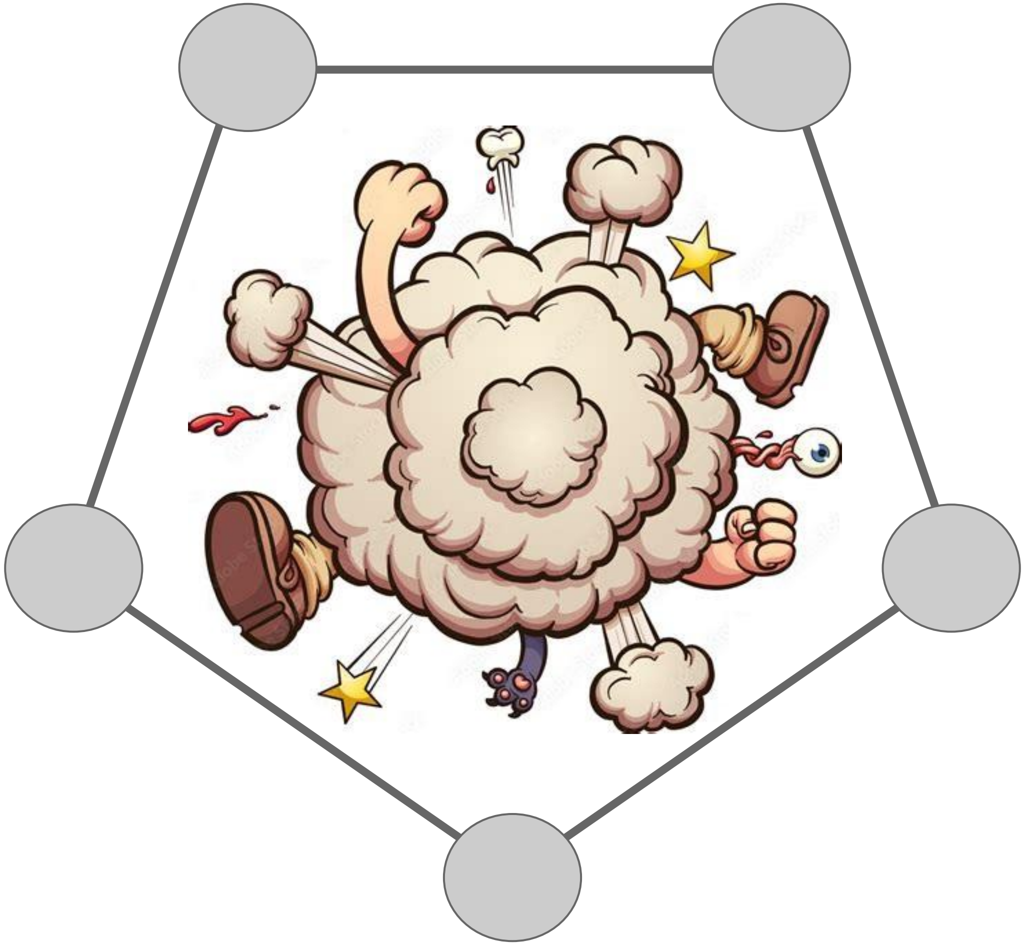
Both **strictly** improve!

Chaos? Run & Chase!

Chaos? Run & Chase!



Chaos? Run & Chase!



Unstable preferences = There is no stable arrangement!

A few definitions

***k*-valued**

$$\iff$$

Preferences use **at most**
k values.

A few definitions

***k*-valued**

$$\Longleftrightarrow$$

Preferences use **at most *k*** values.

i.e.

$$\forall x, y, P(x \to y) \in O$$
$$|O| \leq k$$

A few definitions

***k*-valued**

$\iff$

Preferences use **at most** ***k*** values.

i.e.

$$\forall x, y, P(x \to y) \in O$$
$$|O| \leq k$$

***k*-class**

$\iff$

Preferences use **at most** ***k*** classes.

A few definitions

k-valued

$$\Longleftrightarrow$$

Preferences use **at most** ***k*** values.

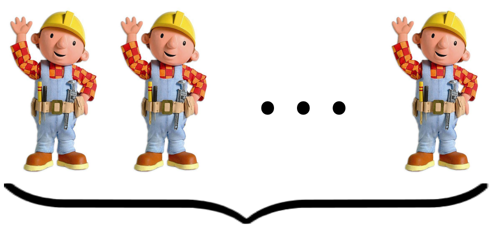
i.e.

$$\forall x, y, P(x \to y) \in O$$
$$|O| \leq k$$

k-class

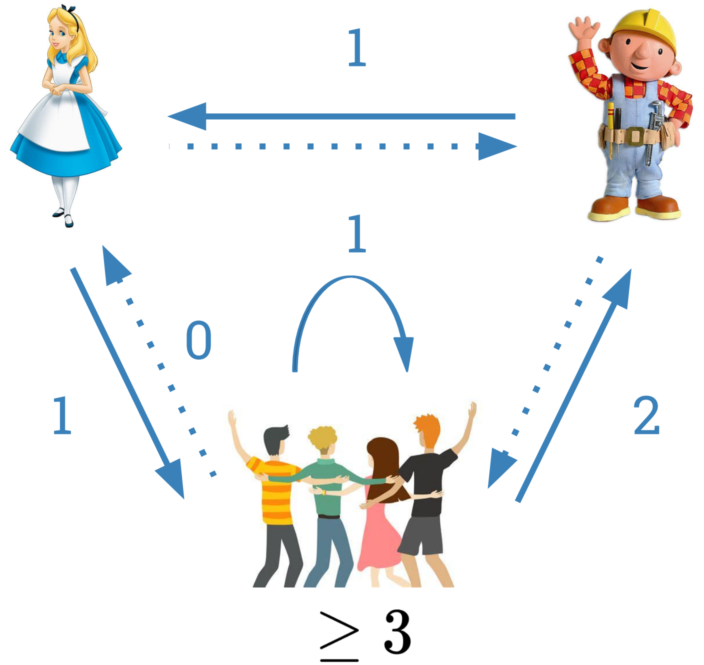
$$\Longleftrightarrow$$

Preferences use **at most** ***k*** classes.



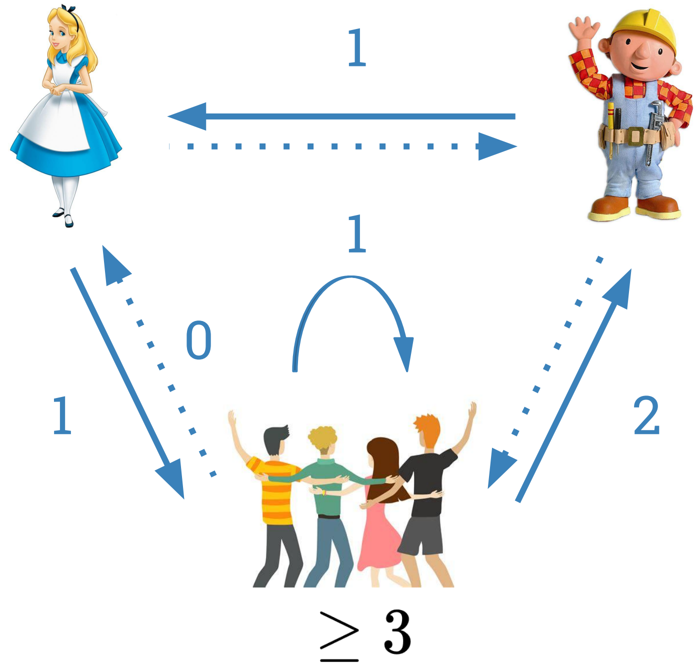
Indistinguishable agents
=
One class

Example



$$\forall x, y, P(x \to y) \in \{0, 1, 2\}$$

Example

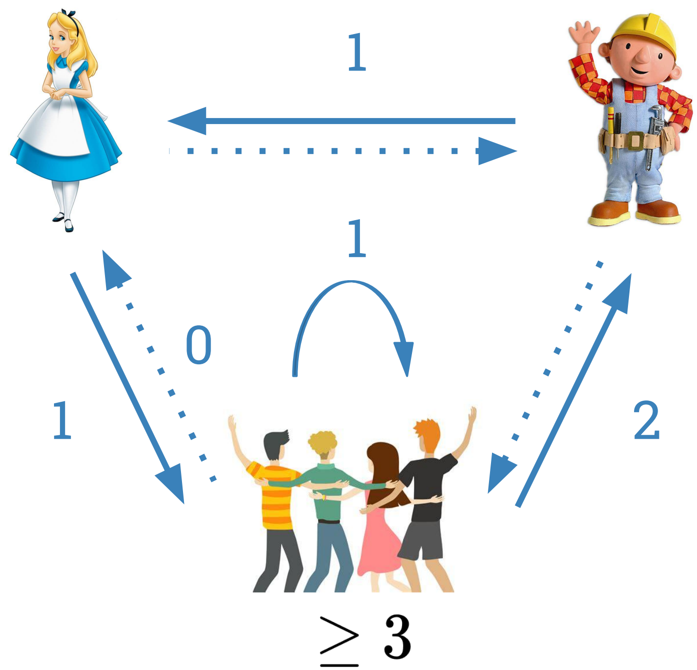


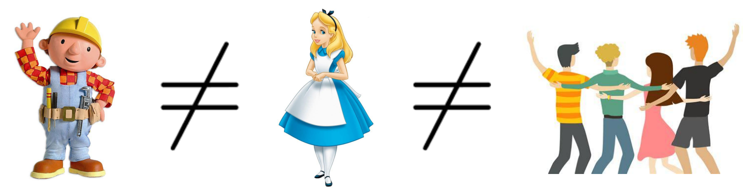
$$\forall x, y, P(x \to y) \in \{0, 1, 2\}$$

$$\Downarrow$$

3-valued

Example



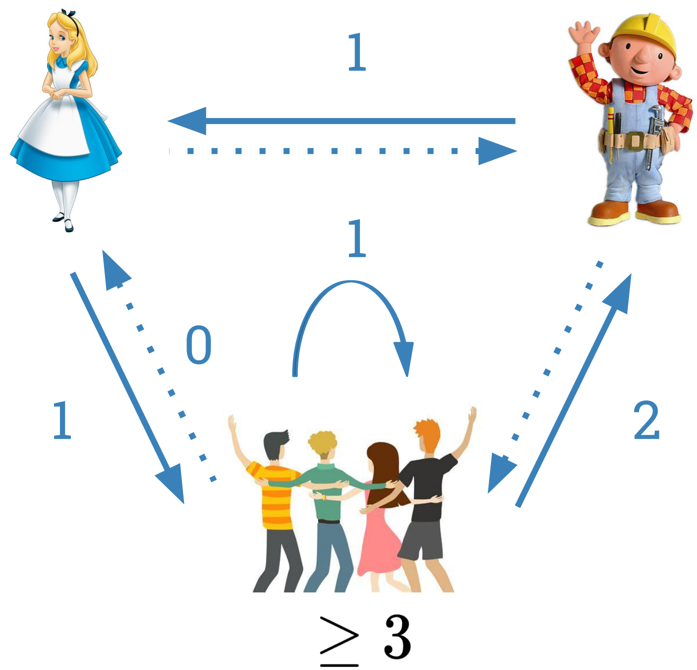


$$\forall x, y, P(x \to y) \in \{0, 1, 2\}$$

$$\Downarrow$$

3-valued

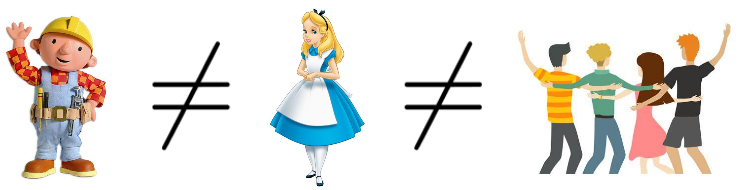
Example



$$\forall x, y, P(x \to y) \in \{0, 1, 2\}$$

$$\Downarrow$$

3-valued



$$\Downarrow$$

3-class

Results

Does a **stable** arrangement always exist?

# Classes / # Values	≤ 2	3	≥ 4
2			
≥ 3		**No**	**No**

On **cycles**

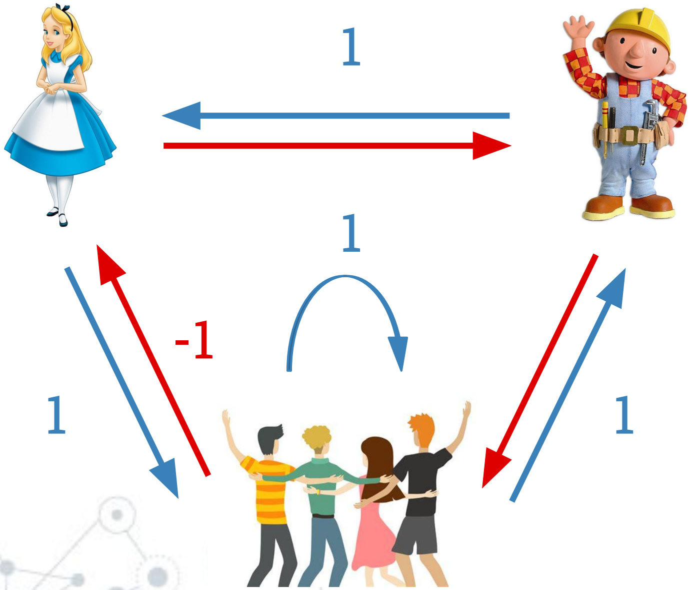
And on a path?

Irregularity at the extremities:

And on a path?

Irregularity at the extremities:

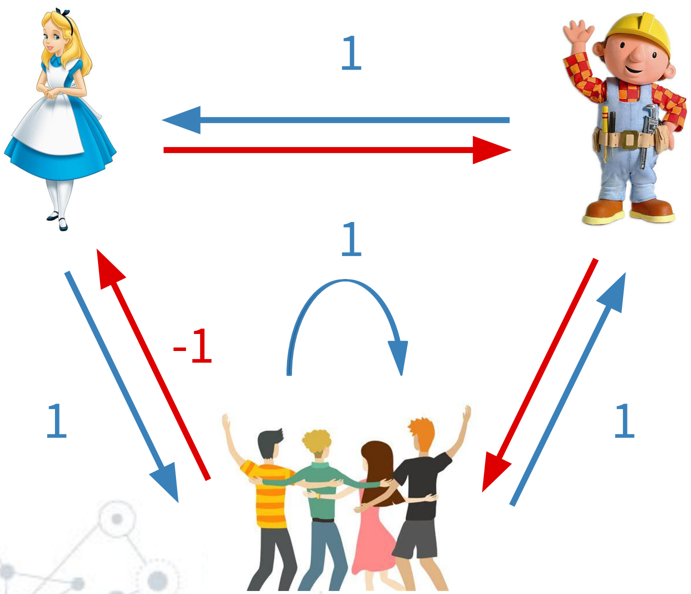
Either use **negative** preferences,



And on a path?

Irregularity at the extremities:

Either use **negative** preferences,



Or consider **more** agents.

$\times 3$

≥ 8

Results

Does a **stable** arrangement always exist?

# Values \ **# Classes**	≤ 2	3	≥ 4
2			
≥ 3		**No**	**No**

On **cycles**

# Values \ **# Classes**	≤ 2	≥3
2		
≥ 3		

On **paths**

Results

Does a **stable** arrangement always exist?

# Values \ # Classes	≤ 2	3	≥ 4
2			
≥ 3		**No**	**No**

On **cycles**

# Values \ # Classes	≤ 2	≥3
2		**No**[1]
≥ 3		**No**

On **paths**

[1] With negative preferences

2.

Stability of 2-class preferences

A story in Red & Blue

2.

Stability of 2-class preferences

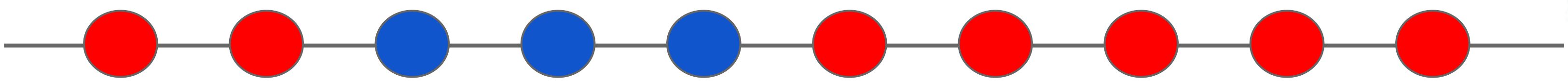
A story in Red & Blue

# Classes / # Values	≤ 2	3	≥ 4
2			
≥ 3		**No**	**No**

On **cycles**

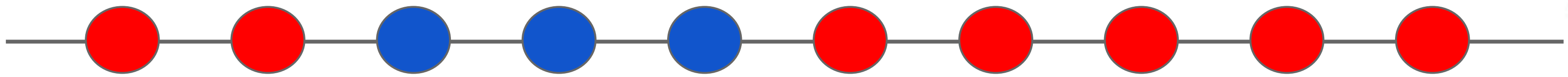
Case analysis

If there is a **self-prefering** class:

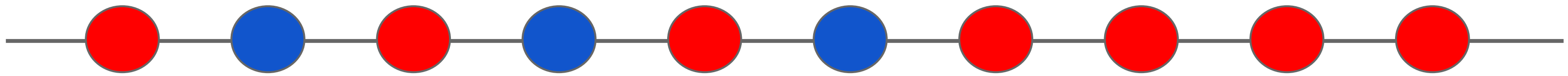


Case analysis

If there is a **self-prefering** class:



Otherwise:



Results

Does a **stable** arrangement always exist?

# Values \ # Classes	≤ 2	3	≥ 4
2			
≥ 3		**No**	**No**

On **cycles**

# Values \ # Classes	≤ 2	≥3
2		**No**[1]
≥ 3		**No**

On **paths**

[1] With negative preferences

Results

Does a **stable** arrangement always exist?

# Values \ # Classes	≤ 2	3	≥ 4
2	Yes		
≥ 3	Yes	No	No

On **cycles**

# Values \ # Classes	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

3.

Adding one class?

Headaches incoming...

3.

Adding one class?

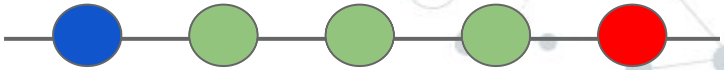
Headaches incoming...

# Classes / # Values	≤ 2	3	≥ 4
2	Yes		
≥ 3	Yes	No	No

On **cycles**

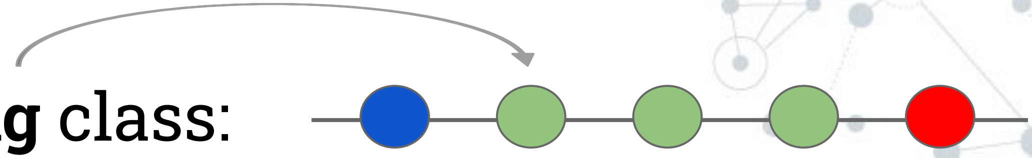
Case analysis

- If there is a **self-liking** class:



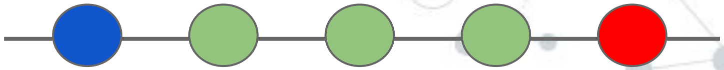
Case analysis

- If there is a **self-liking** class:

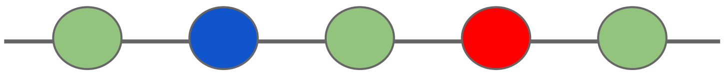


Case analysis

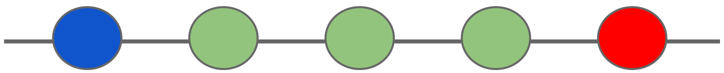
- If there is a **self-liking** class:



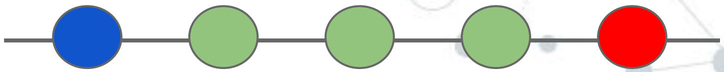
- Else if a class **likes / is disliked by** all others:



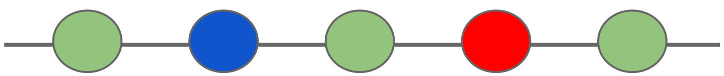
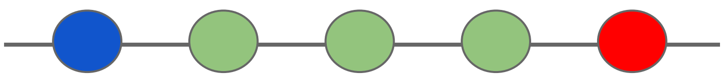
or

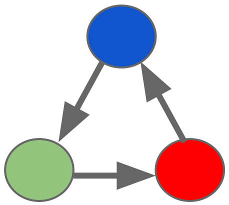


Case analysis

- If there is a **self-liking** class: 

- Else if a class **likes / is disliked by** all others:

 or 

- Else, preferences are 

$\Longrightarrow$ Potential argument.

Results

Does a **stable** arrangement always exist?

# Values \ **# Classes**	≤ 2	3	≥ 4
2	Yes		
≥ 3	Yes	No	No

On **cycles**

# Values \ **# Classes**	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

Results

Does a **stable** arrangement always exist?

# Values \ # Classes	≤ 2	3	≥ 4
2	Yes	Yes	
≥ 3	Yes	No	No

On **cycles**

# Values \ # Classes	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

4.

Unstable binary preferences

Exhausted? We're getting there...

4.

Unstable binary preferences

Exhausted? We're getting there…

# Classes / # Values	≤ 2	3	≥ 4
2	Yes	Yes	
≥ 3	Yes	No	No

On **cycles**

Exhaustion results

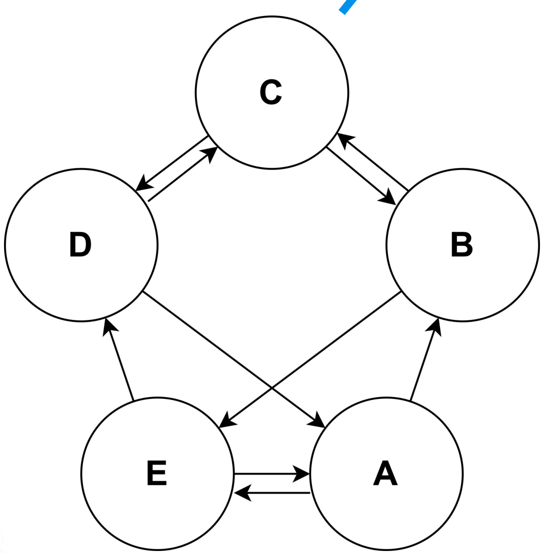
Number of **unstable** binary preference graphs

N	3	4	5	6	7
Cycle	0	0	**1**	**0**	**3**
Path	0	0	0	0	0

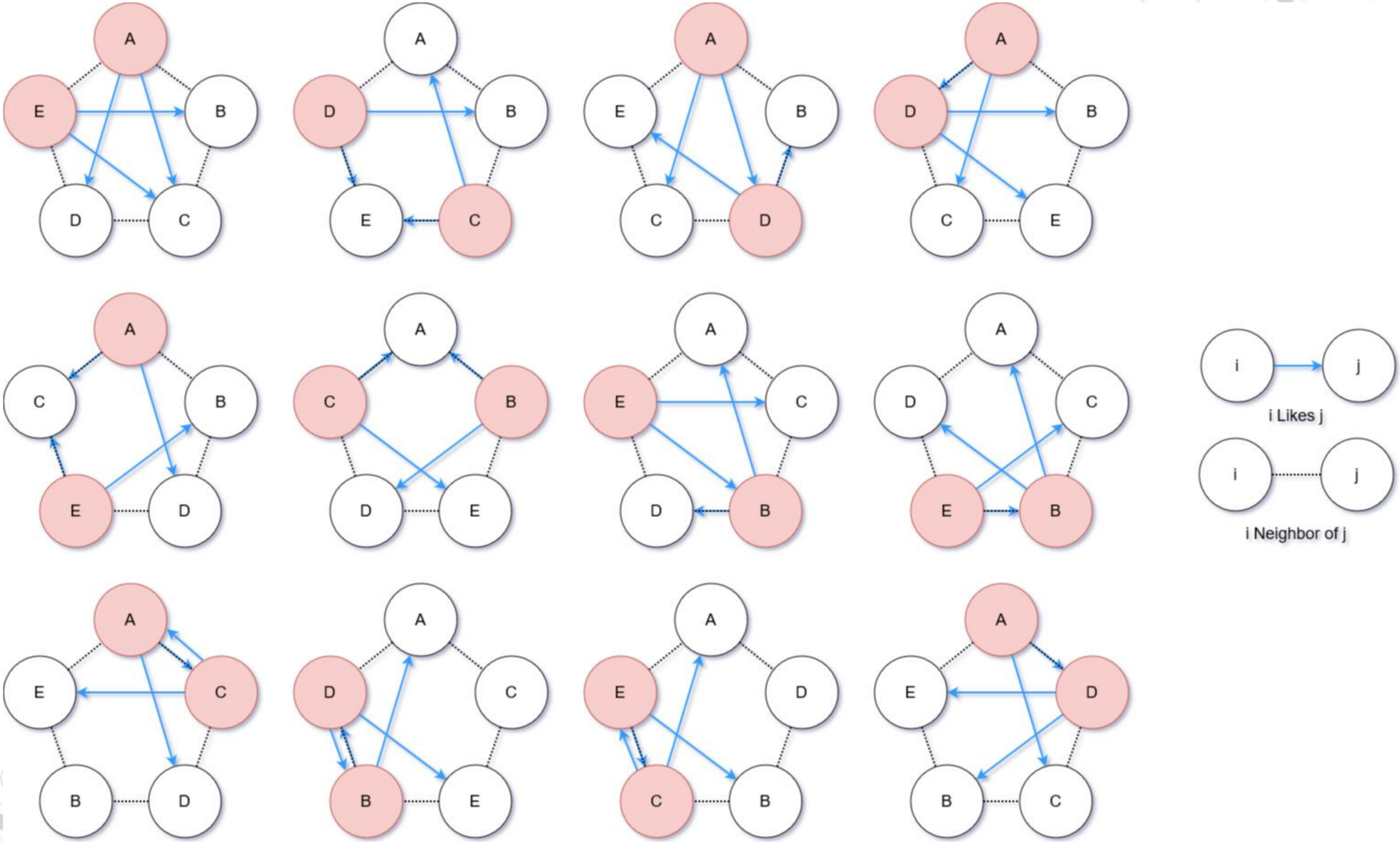
Exhaustion results

Number of **unstable** binary preference graphs

N	3	4	5	6	7
Cycle	0	0	**1**	**0**	**3**
Path	0	0	0	0	0



Why is it unstable?

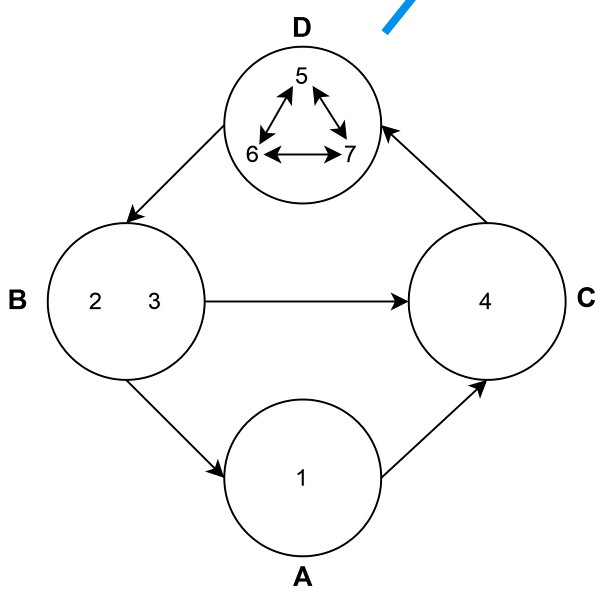


Exhaustion results

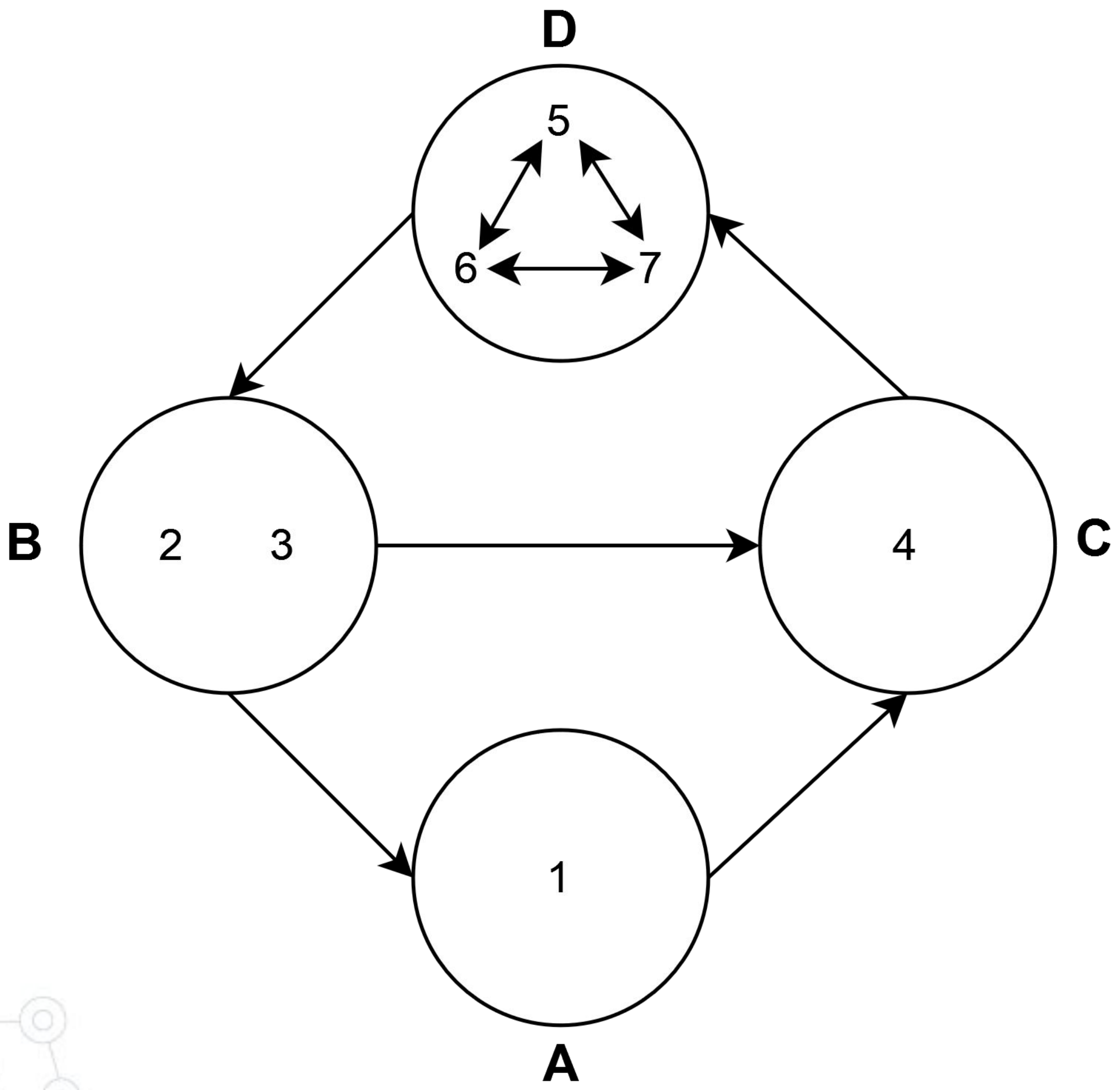
Number of **unstable** binary preference graphs

N	3	4	5	6	7
Cycle	0	0	**1**	**0**	**3**
Path	0	0	0	0	0

E.g.

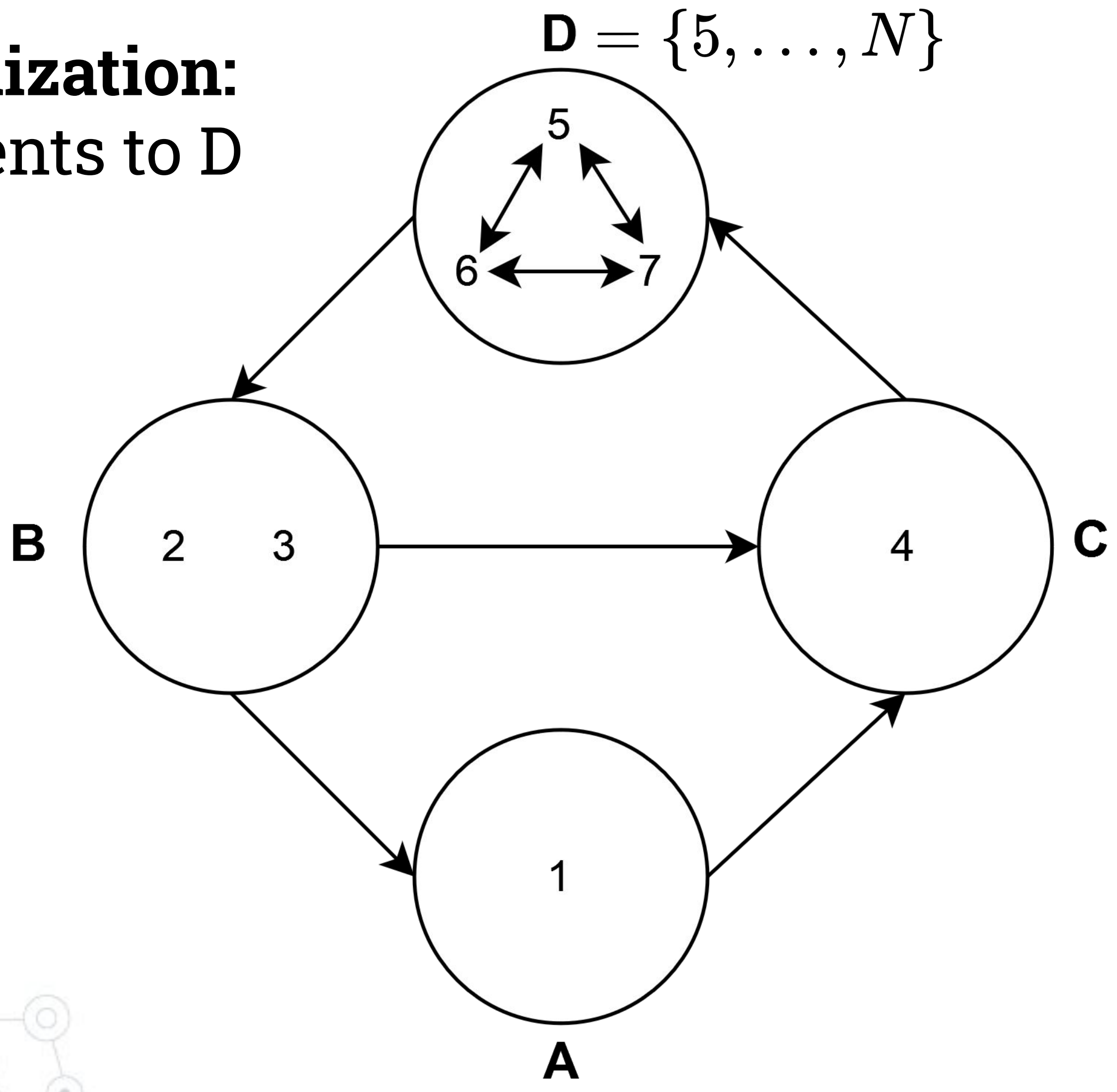


A more interesting example



A more interesting example

Generalization:
Add agents to D



Results

Does a **stable** arrangement always exist?

# Classes / # Values	≤ 2	3	≥ 4
2	Yes	Yes	
≥ 3	Yes	No	No

On **cycles**

# Classes / # Values	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

Results

Does a **stable** arrangement always exist?

# Values \ # Classes	≤ 2	3	≥ 4
2	Yes	Yes	No
≥ 3	Yes	No	No

On **cycles**

# Values \ # Classes	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

5.

A quick word on complexity

Hopefully in poly-time...

Bounded number of classes

Keep track of:

- Occurrences of triplets
- Occurrences of classes
- Three last agents

$\Big\}$ $O(1)$ variables

$\Longrightarrow$ **Non-deterministic** algorithm that guesses a arrangement and checks its stability in $O(\log N)$ space.

$\Longrightarrow$ $\exists$ **deterministic** algorithm in poly-time.

Complexity results

# Classes	Bounded	Unbounded
Cycles	**Poly-time**	
Paths	**Poly-time**	

Complexity results

# Classes	Bounded	Unbounded
Cycles	**Poly-time**	**NP-hard**[2]
Paths	**Poly-time**	**NP-hard**[3]

Proved in **[1]**:

[2] With 4 non-negative values

[3] With 6 values, including negatives

[1] Ceylan, E., Chen, J., Roy, S.: Optimal seat arrangement: What are the hard and easy cases? In: Elkind, E. (ed.) IJCAI'23. pp. 2563–2571 (8 2023)

Conjectures

# Classes	Bounded	Unbounded	Unbounded **with binary values**
Cycles	**Poly-time**	**NP-hard**[2]	
Paths	**Poly-time**	**NP-hard**[3]	

Conjectures

# Classes	Bounded	Unbounded	Unbounded **with binary values**
Cycles	**Poly-time**	**NP-hard**[2]	**NP-hard?**
Paths	**Poly-time**	**NP-hard**[3]	**Poly-time?**

6.

Summary

For those who fell asleep

Summary

Does a **stable** arrangement always exist?

# Values \ # Classes	≤ 2	3	≥ 4
2	Yes	Yes	No
≥ 3	Yes	No	No

On **cycles**

# Values \ # Classes	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

Complexity

# Classes	Bounded	Unbounded
Cycles	**Poly-time**	**NP-hard[2]**
Paths	**Poly-time**	**NP-hard[3]**

[2] With 4 non-negative values

[3] With 6 values, including negatives

Thank you!

Summary

Damien Berriaud

Does a **stable** arrangement always exist?

# Classes / # Values	≤ 2	3	≥ 4
2	Yes	Yes	No
≥ 3	Yes	No	No

On **cycles**

# Classes / # Values	≤ 2	≥3
2	Yes	No[1]
≥ 3	Yes	No

On **paths**

[1] With negative preferences

Complexity

# Classes	Bounded	Unbounded
Cycles	**Poly-time**	**NP-hard**[2]
Paths	**Poly-time**	**NP-hard**[3]

[2] With 4 non-negative values

[3] With 6 values, including negatives