universität wien

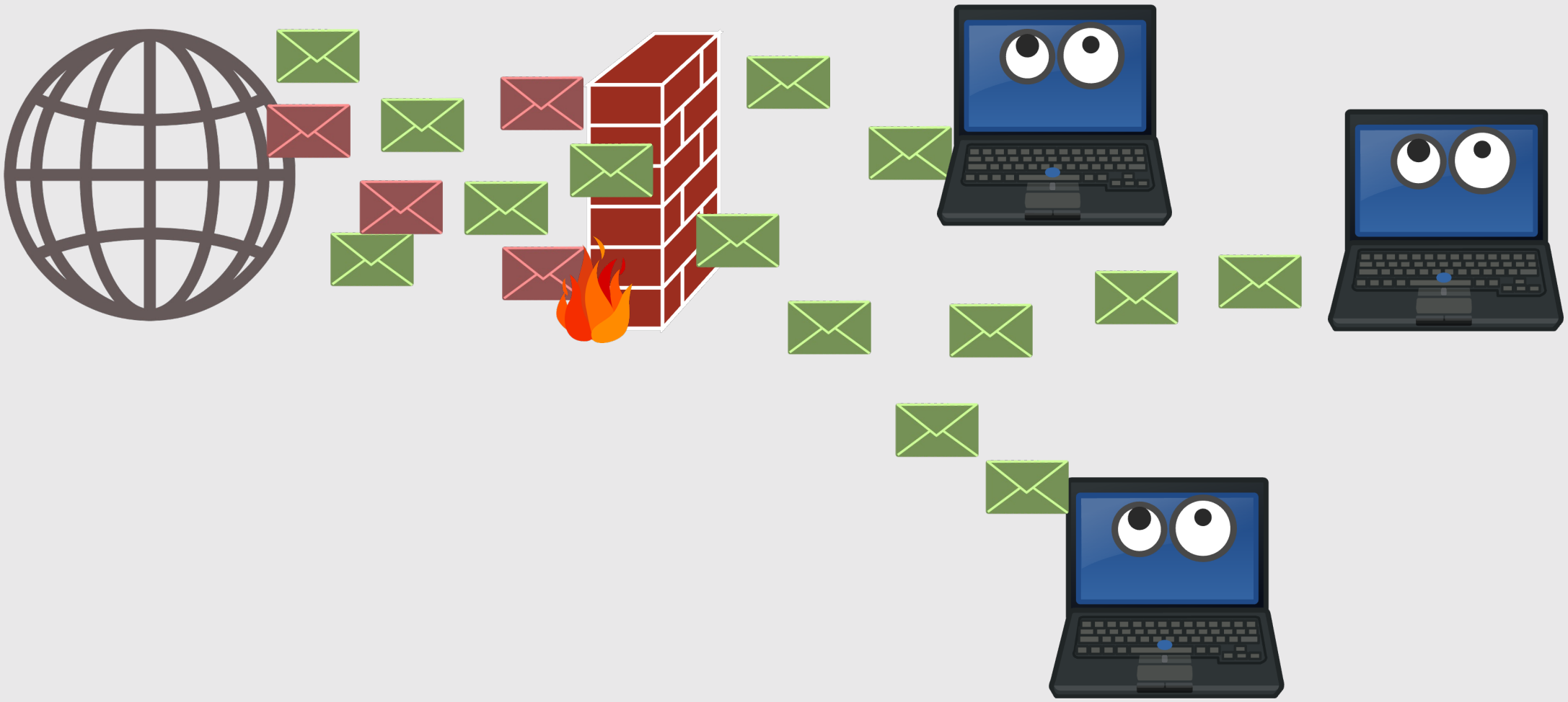
berlin

Self-Adjusting Partially Ordered Lists

Vamsi Addanki, Macej Pacut, Arash Pourdamghani, Gábor Rétvári,
Stefan Schmid and Juan Vanerio

MŰEGYETEM 1782

A Real World Need



A Real World Need

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY

Packet Classification

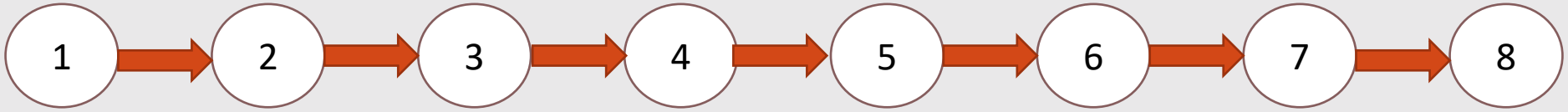
N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY

Packet Classification & Orders

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY

Model: A List

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY



Model: A List

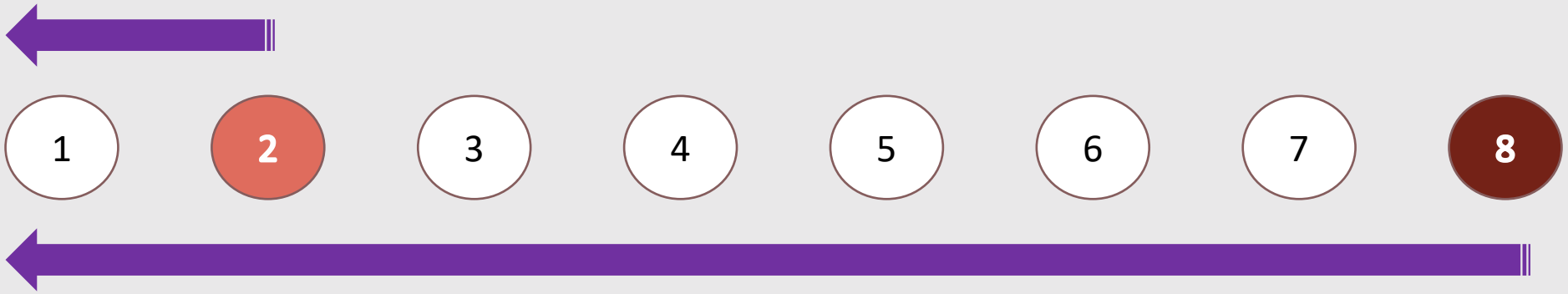
N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY

Temporal structure

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	**IP**	**0.0.0.0/0**	**0.0.0.0/0**			**DENY**

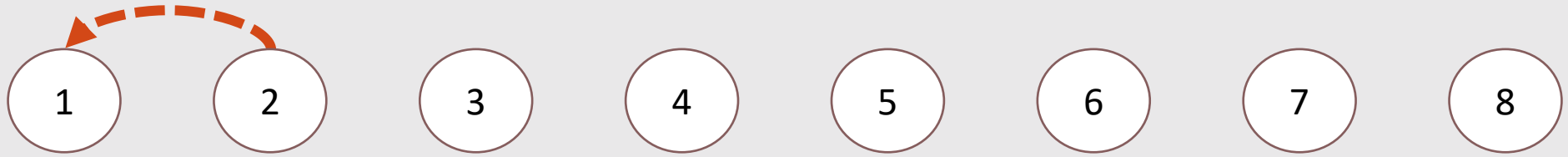
Is Self-adjustment Possible?

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	**IP**	**0.0.0.0/0**	**0.0.0.0/0**			**DENY**



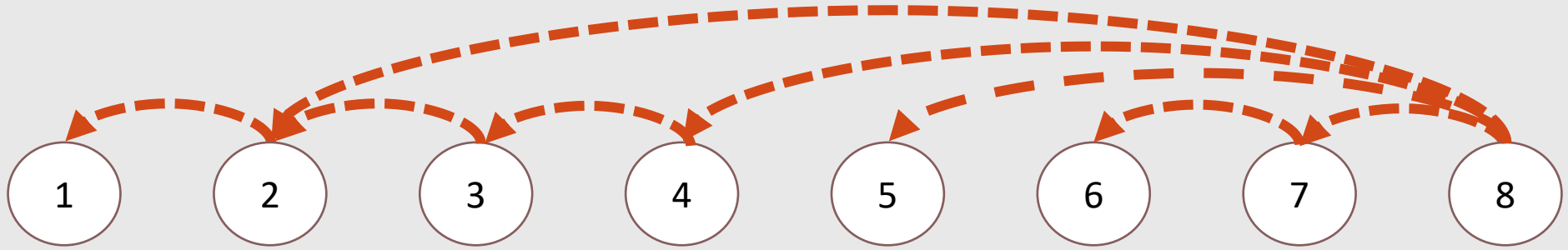
Enforced Orders

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY



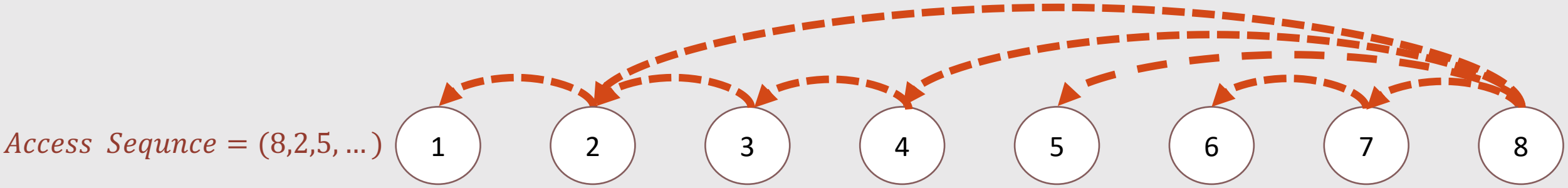
Enforced Orders

N	Protocol	Src. IP	Dst. IP	Src. Port	Dst. Port	Action
1	TCP	10.12.12.0/24	20.0.0.1/32	ANY	80	DENY
2	TCP	0.0.0.0/0	20.0.0.1/32	ANY	80	ACCEPT
3	IP	0.0.0.0/0	20.0.0.1/32			DENY
4	UDP	0.0.0.0/0	0.0.0.0/0	1000-2000	1000-2000	ACCEPT
5	UDP	20.0.0.0/24	10.0.10.0/24	ANY	3306	ACCEPT
6	TCP	10.12.12.0/24	0.0.0.0/0	21	21	DENY
7	IP	10.0.0.0/16	20.0.0.0/20			ACCEPT
8	IP	0.0.0.0/0	0.0.0.0/0			DENY



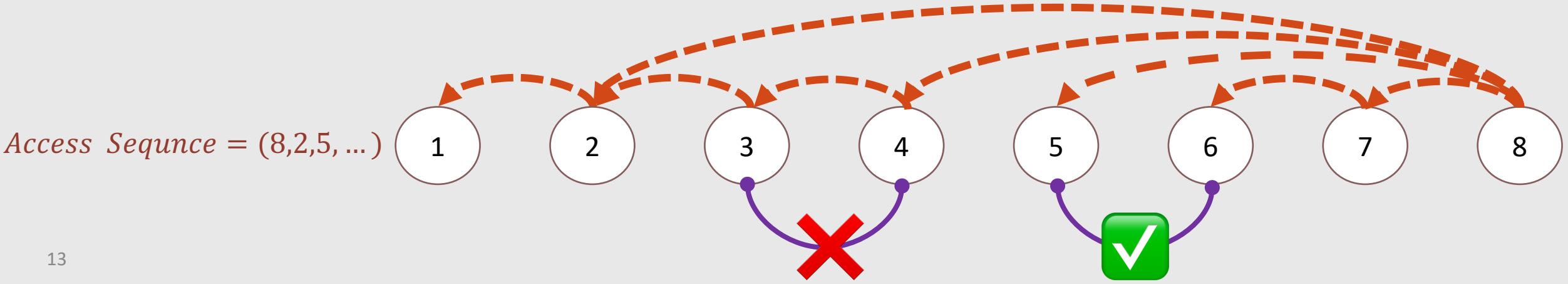
Formal Question

- Inputs:
  - A set of items in a link list
  - A set of “enforced” orderings between items
  - An access sequence revealed over time



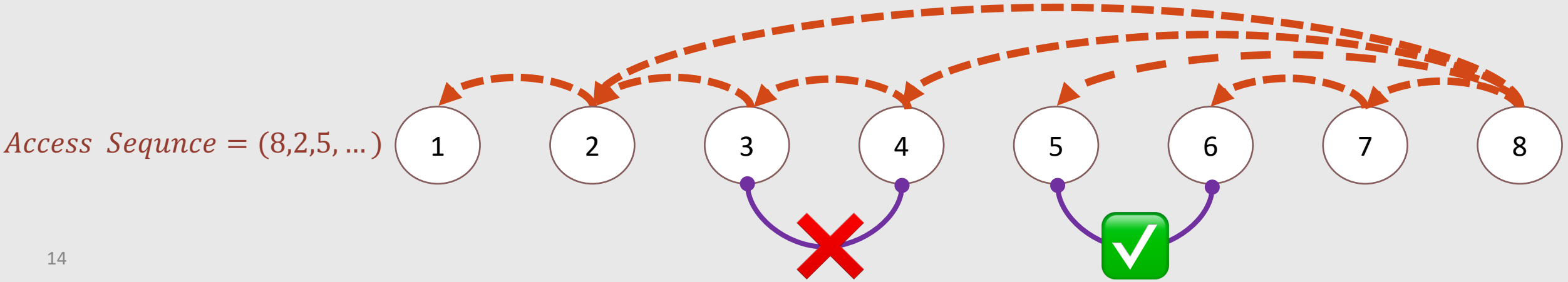
Formal Question

- Inputs:
  - A set of items in a link list
  - A set of “enforced” orderings between items
  - An access sequence revealed over time
- Operations:
  - Swapping position of two items while respecting orders



Formal Question

- Inputs:
  - A set of items in a link list
  - A set of “enforced” orderings between items
  - An access sequence revealed over time
- Operations:
  - Swapping position of two items while respecting orders
- Objective:
  - Minimizing the total (access + reconfiguration) cost while respecting orders



Our Approach

- Going beyond worst-case analysis

Our Approach

- Going beyond worst-case analysis
- Constant competitiveness, given a constant c:

  Total cost of an algorithm $\leq_{\forall\ inputs}\ c \cdot$ Total cost of the optimal offline algorithm

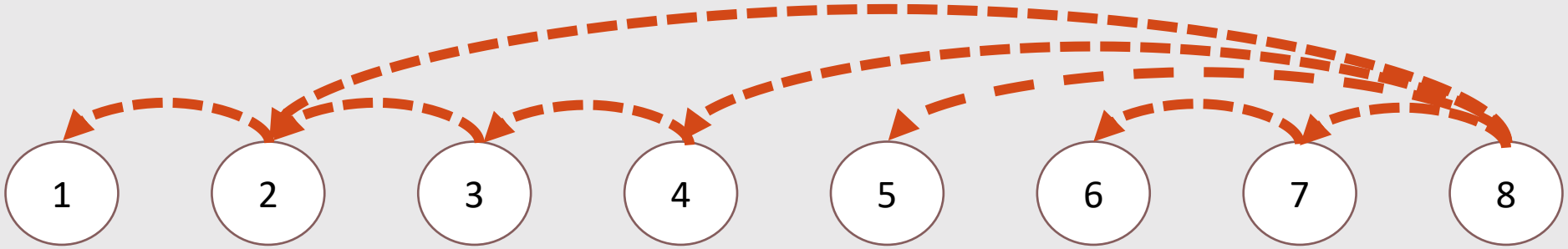
Our Approach

- Going beyond worst-case analysis
- Constant competitiveness, given a constant c:

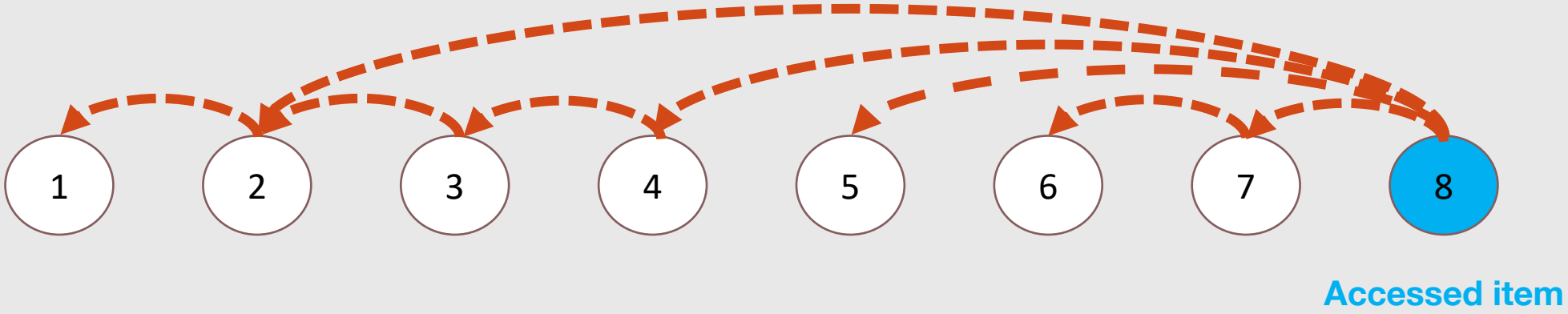
  Total cost of an algorithm $\leq_{\forall\, inputs}\ c \cdot$ Total cost of the optimal offline algorithm
- Our observation:

Local moves ensure constant competitiveness!

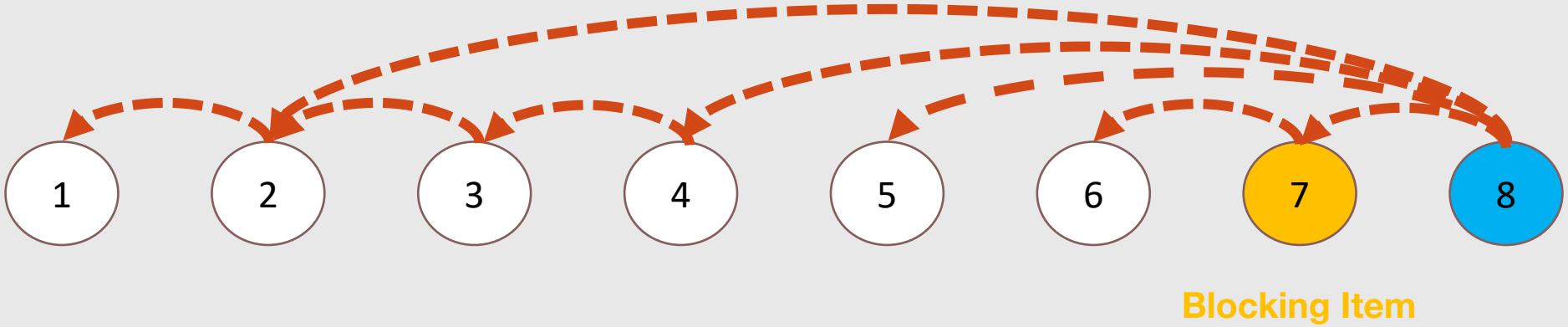
Deterministic Algorithm: Move-Recursively-Forward (MRF)



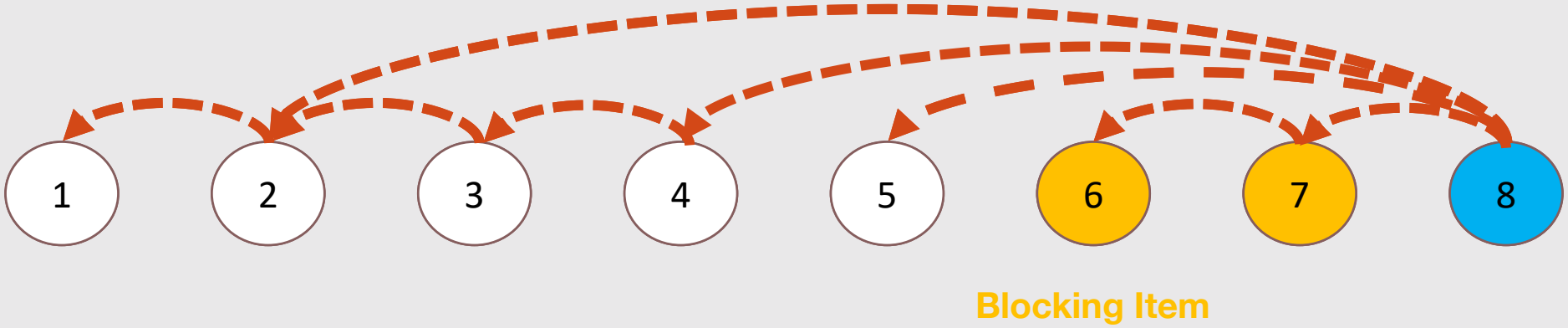
Deterministic Algorithm: Move-Recursively-Forward (MRF)



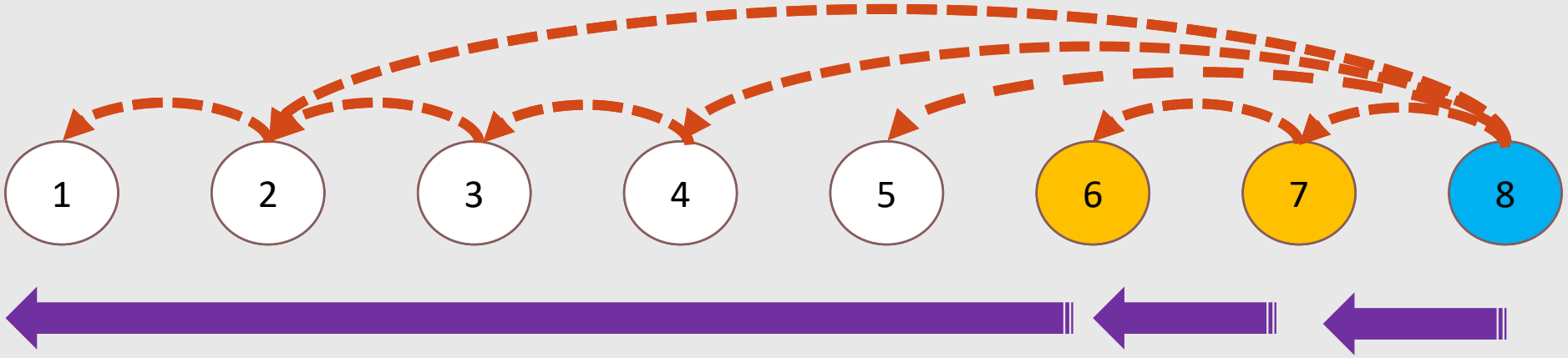
Deterministic Algorithm: Move-Recursively-Forward (MRF)



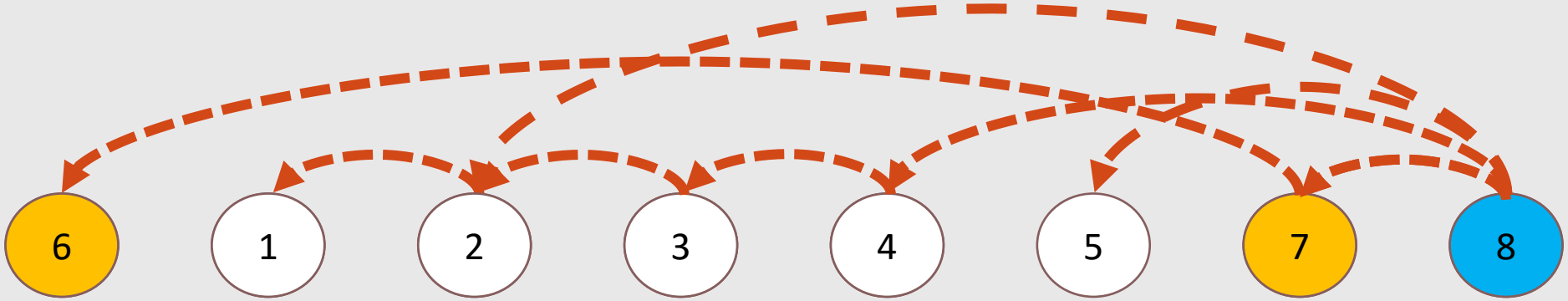
Deterministic Algorithm: Move-Recursively-Forward (MRF)



Deterministic Algorithm: Move-Recursively-Forward (MRF)



Deterministic Algorithm: Move-Recursively-Forward (MRF)



MRF Analysis

- Upper bound:

MRF is strictly 4-competitive.

MRF Analysis

- Upper bound:

  MRF is strictly 4-competitive.

- General proof idea:

  Potential function analysis based on inversions

MRF Analysis

- Upper bound:

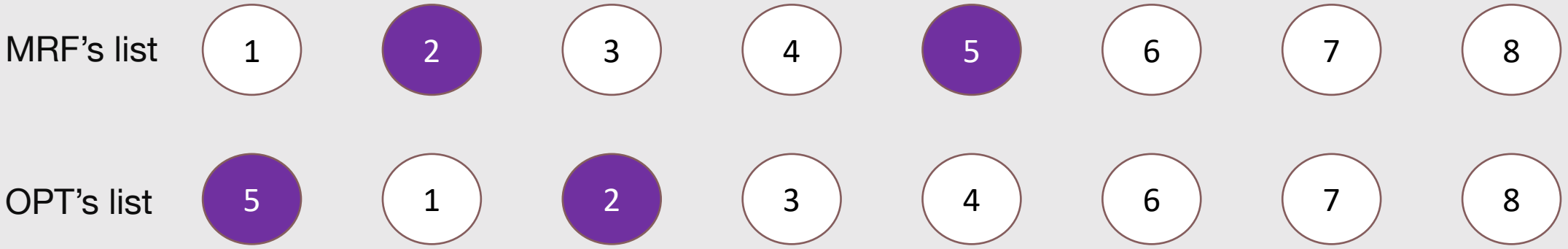
  MRF is strictly 4-competitive.

- General proof idea:

  Potential function analysis based on inversions

- An inversion:

  Mismatch between MRF's list and OPT's list



MRF Analysis

- Upper bound:

  MRF is strictly 4-competitive.

- General proof idea:

  Potential function analysis based on inversions

- An inversion:

  Mismatch between MRF's list and OPT's list

- Challenge:

  Identifying set of inversions after MRF moves

Lower Bound Analysis

- Lower bound:
  - For deterministic case,
  - Given an ϵ based on structure of partial orders

Any online algorithm is $3 - \epsilon$ competitive.

Lower Bound Analysis

- Lower bound:
  - For deterministic case,
  - Given an ϵ based on structure of partial orders

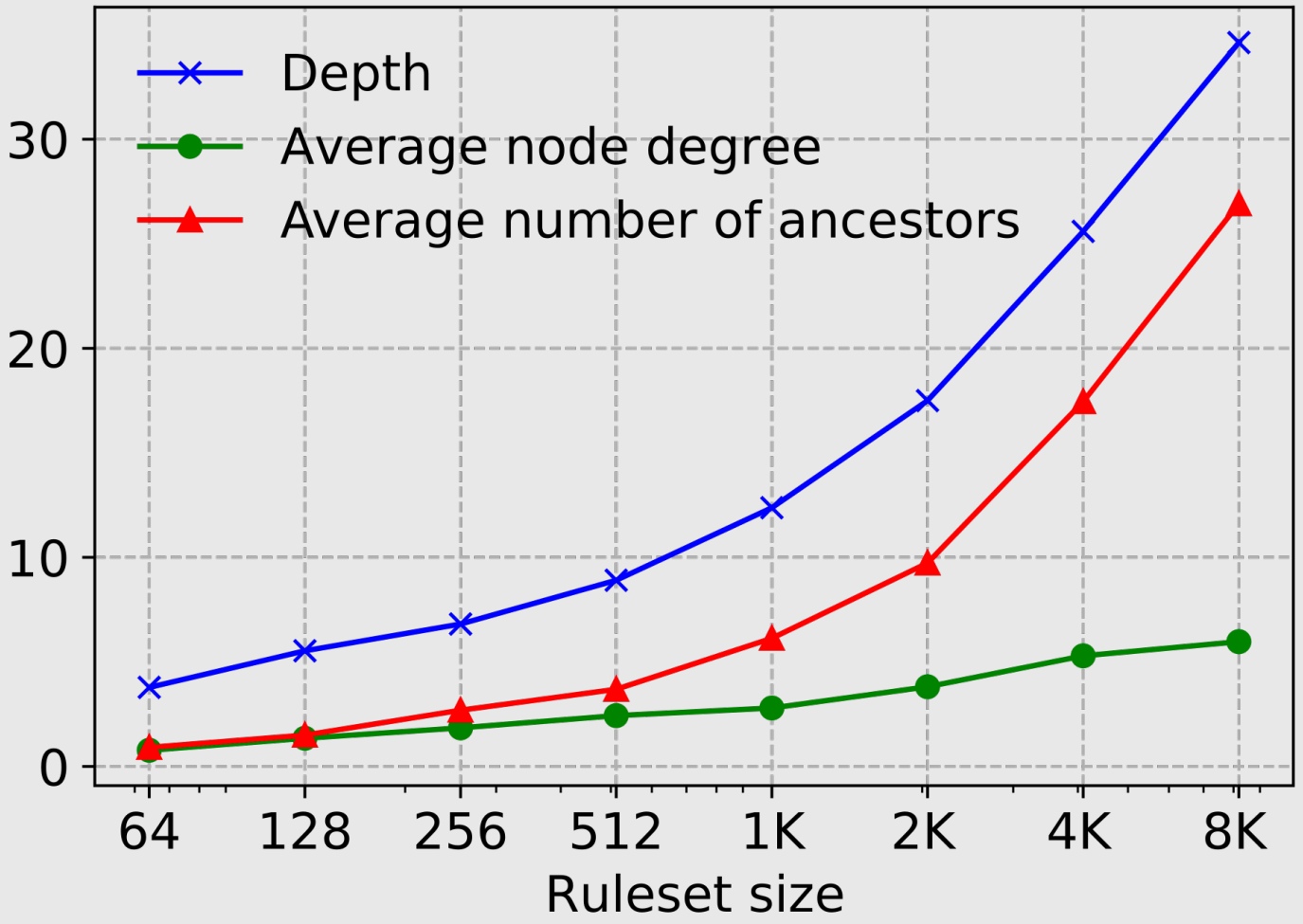
Any online algorithm is $3 - \epsilon$ competitive.

- Proof idea:

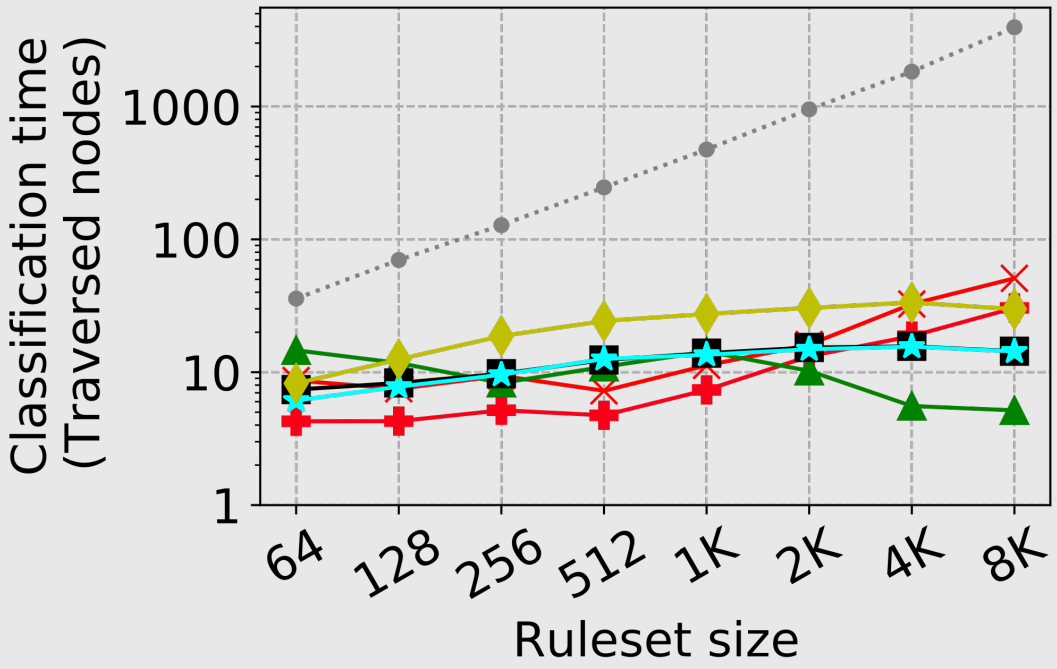
Constructing two special request sequences.

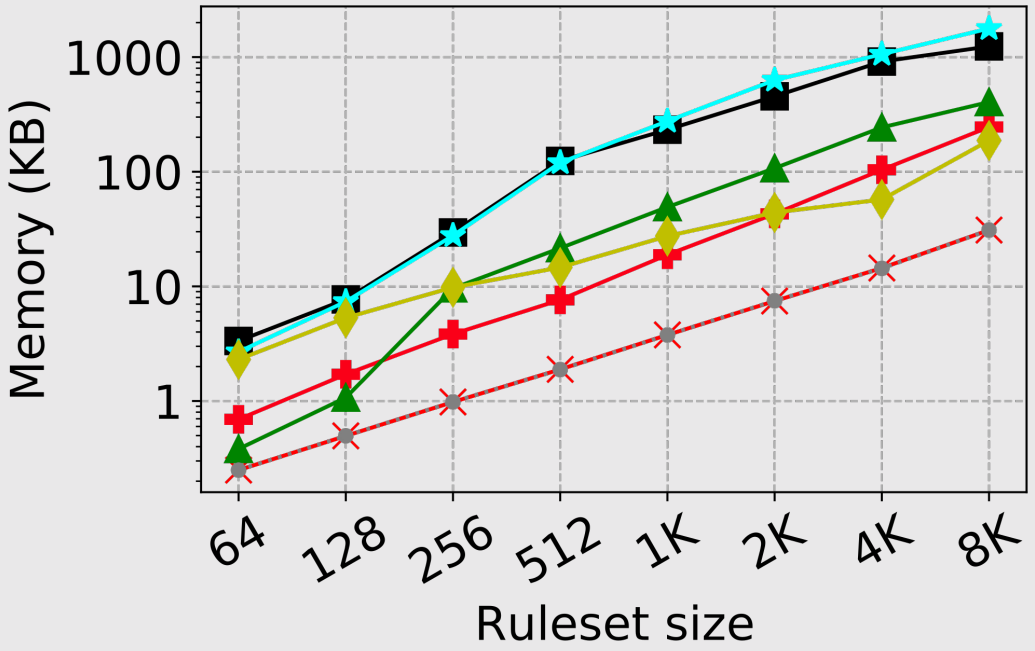
  - One for algorithms with “static” strategy
  - One for algorithms with “dynamic” strategy

Real World Performance: Dataset



Real World Performance: High Locality

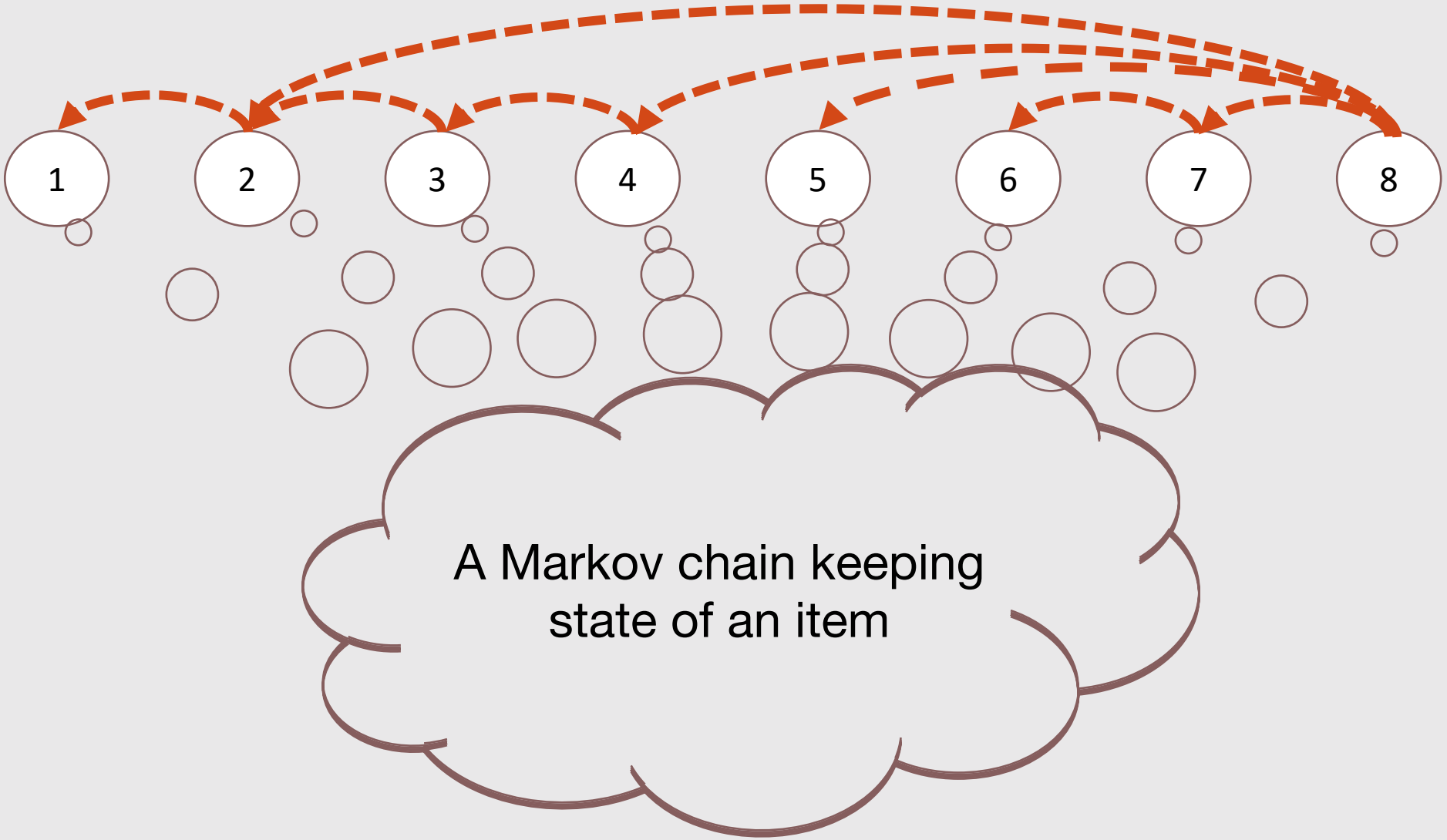




Transforming Traditional List Access Approaches

Algorithm	Randomized	Competitve Factor
Move-to-Front[Sleator & Tarjan, Commun. ACM'85]	❌	4
BIT [Reingold et al. Algorithmica 94]	✅	3
COUNTER [Reingold et al. Algorithmica 94]	✅	2.75
RANDOM-RESET [Reingold et al. Algorithmica 94]	✅	$\sqrt{7}$

Randomized Version



Thank You