Impossibility

DDA Course

week 3

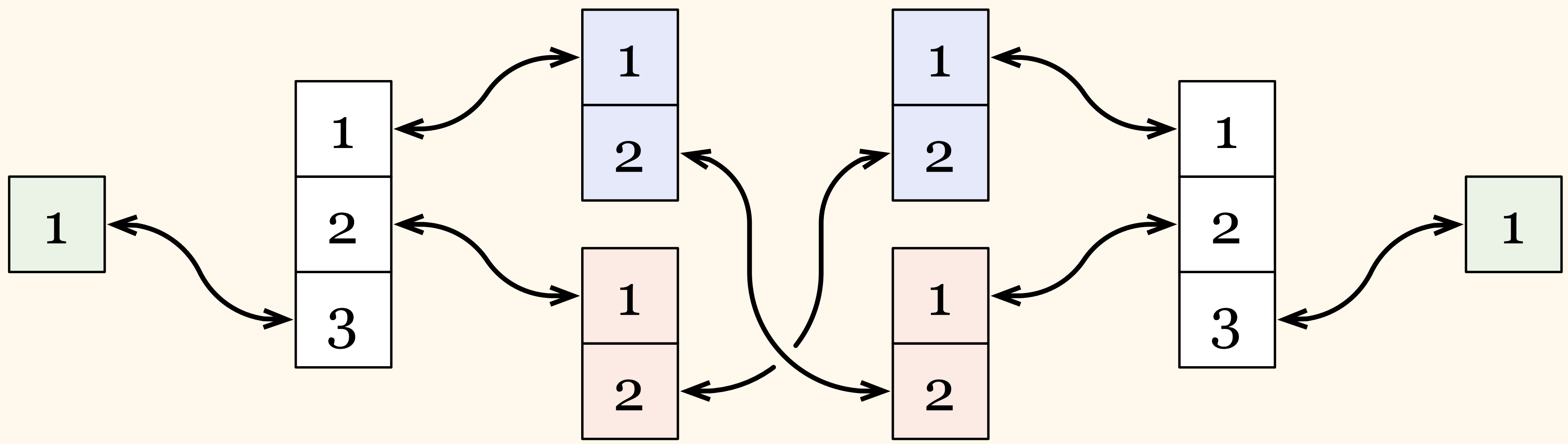
Proof Techniques

- Covering maps
  - problems that cannot solved at all
- Isomorphic local neighbourhoods
  - problems that cannot be solved quickly

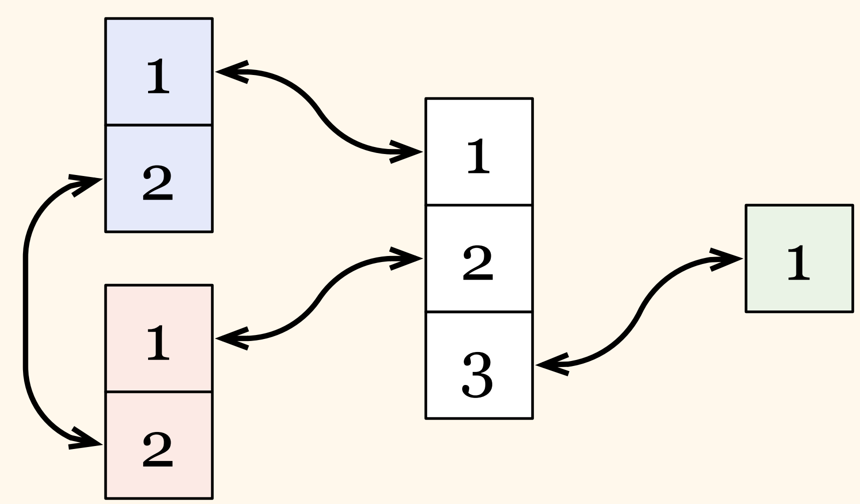
Covering Map

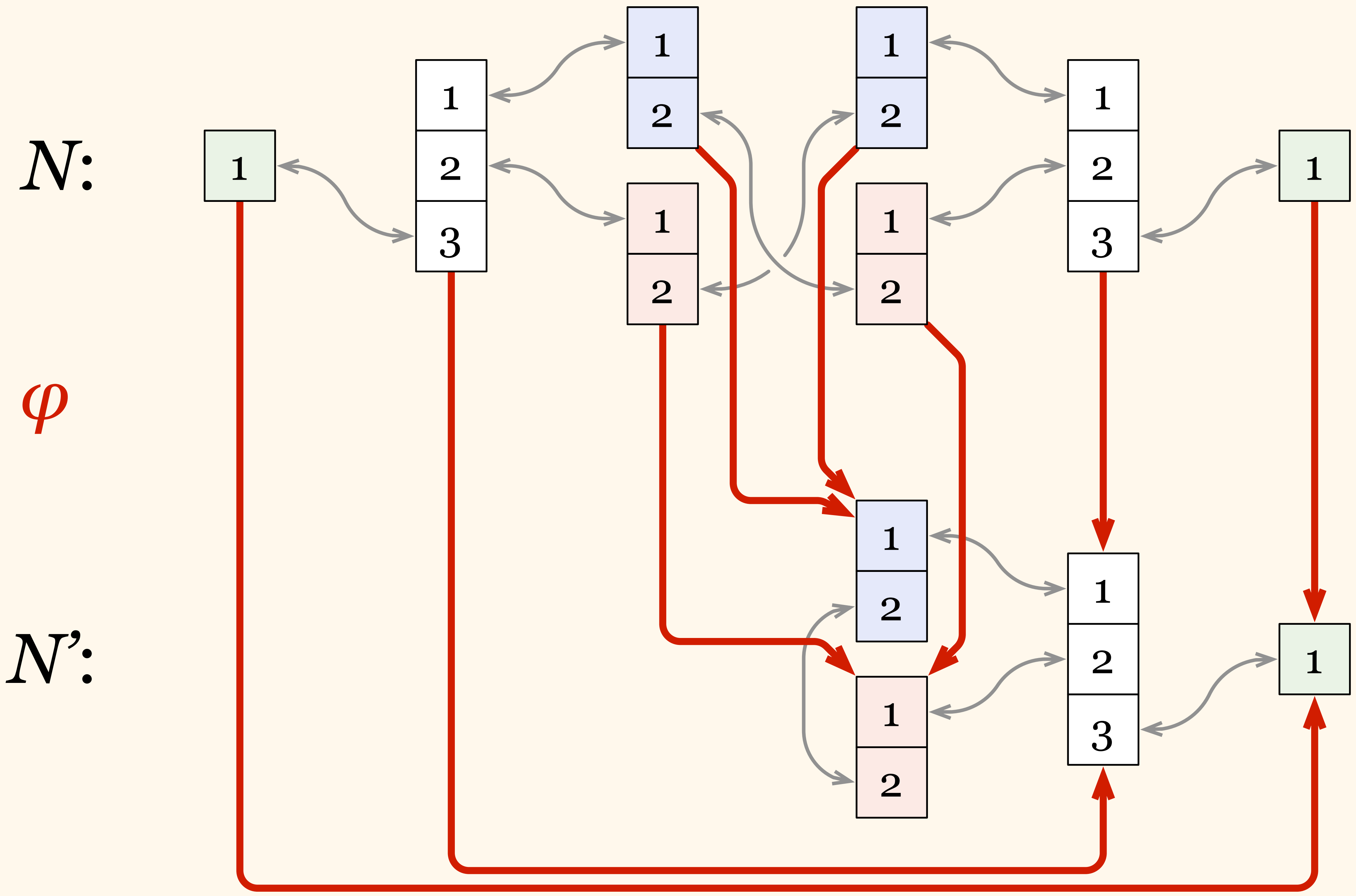
- Networks $N = (V, P, p)$ and $N' = (V', P', p')$
- Surjection $\varphi: V \rightarrow V'$ that *preserves inputs, degrees, connections, and port numbers*

N:

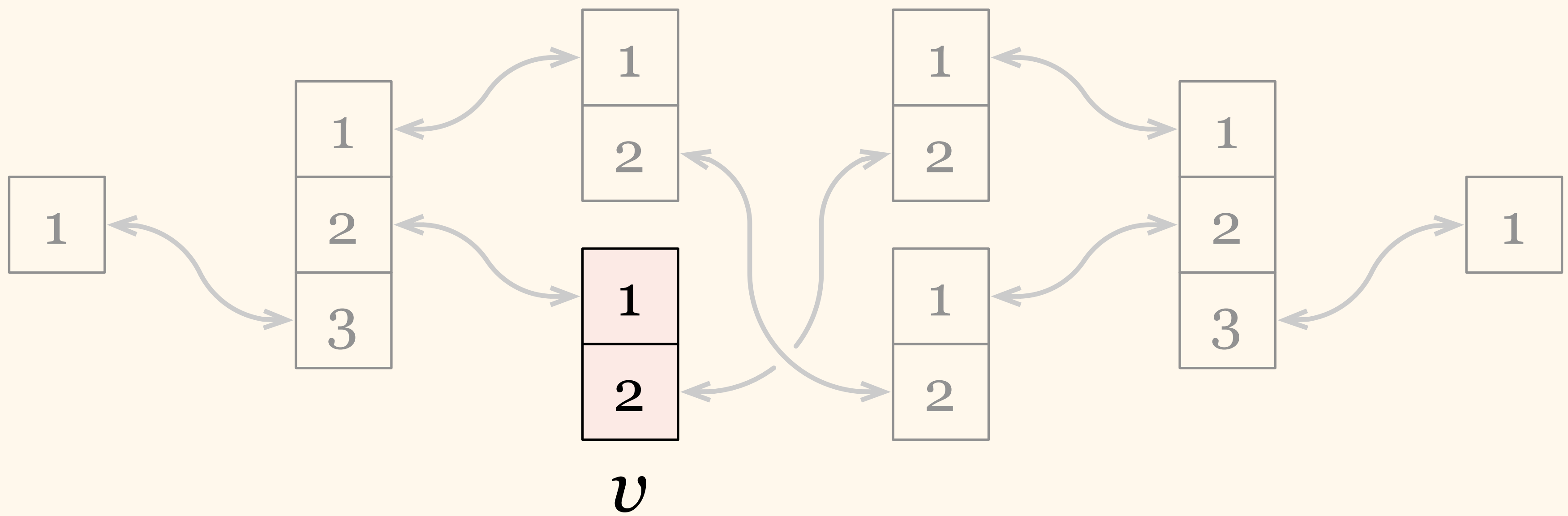


N':



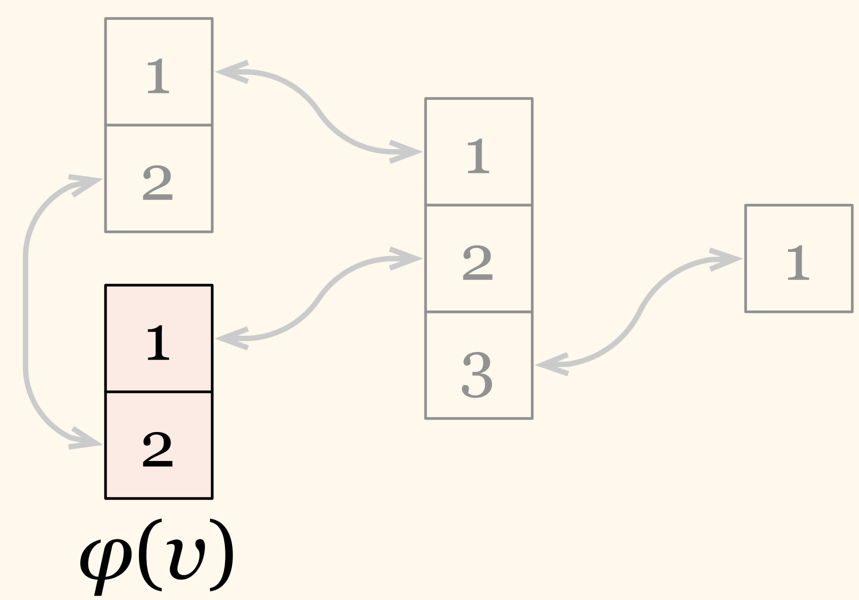

N:
φ
N':

N:



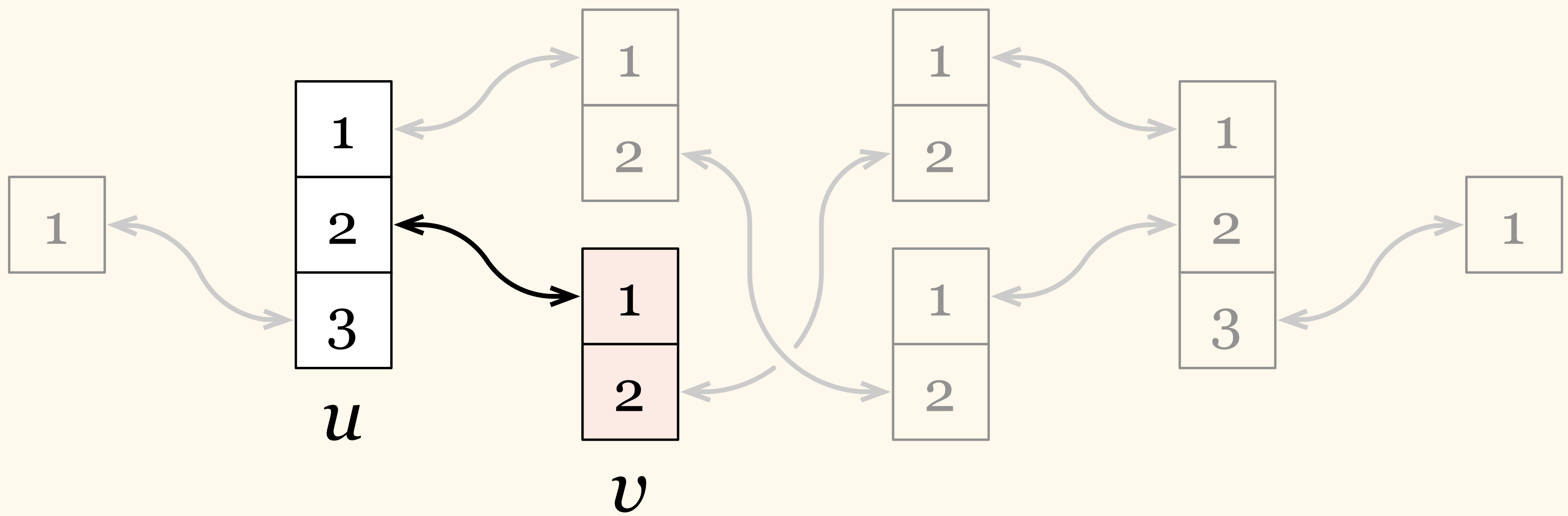
φ

N':



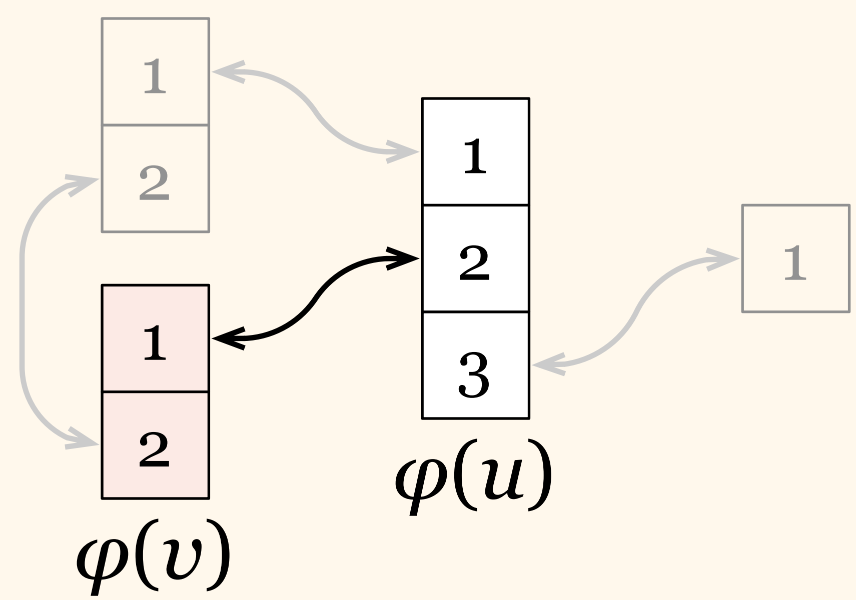
Degrees agree

N:



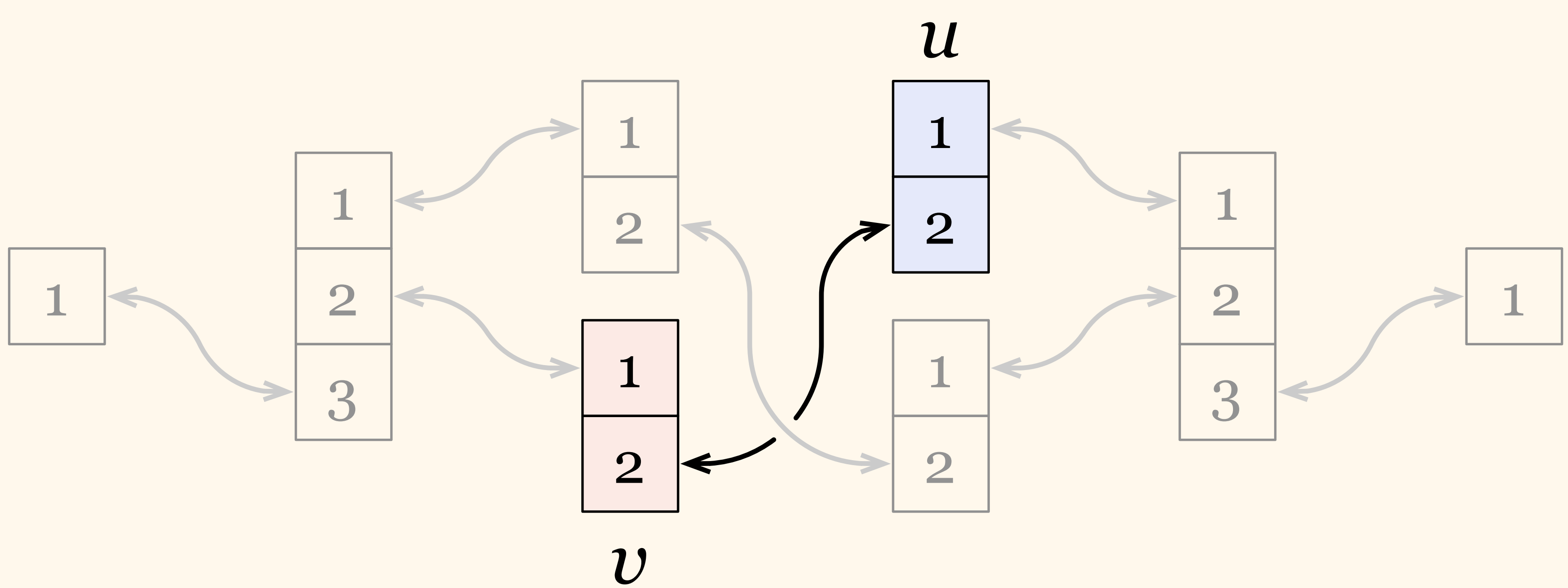
φ

N':



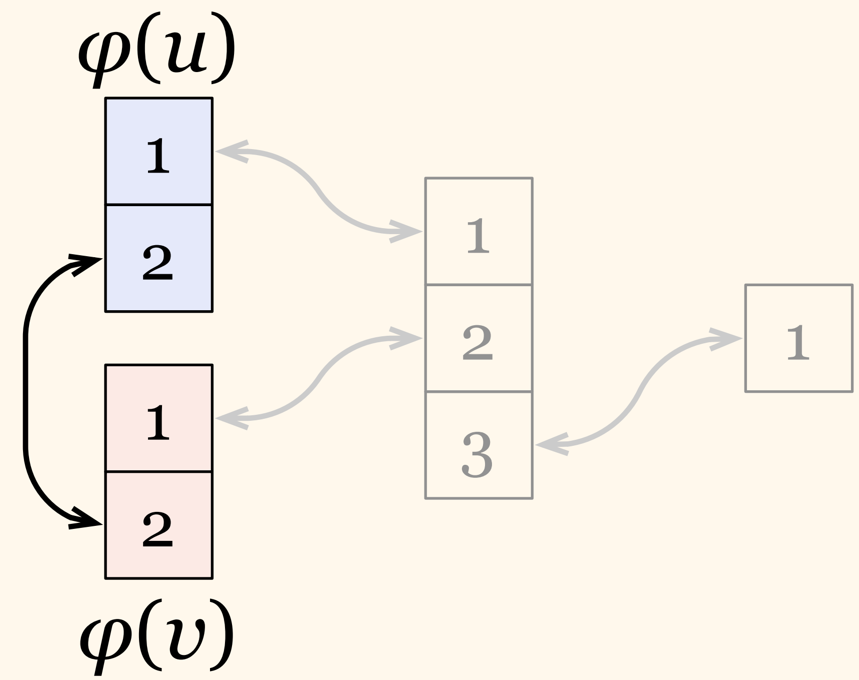
Neighbours in port 1 agree

N:



φ

N':

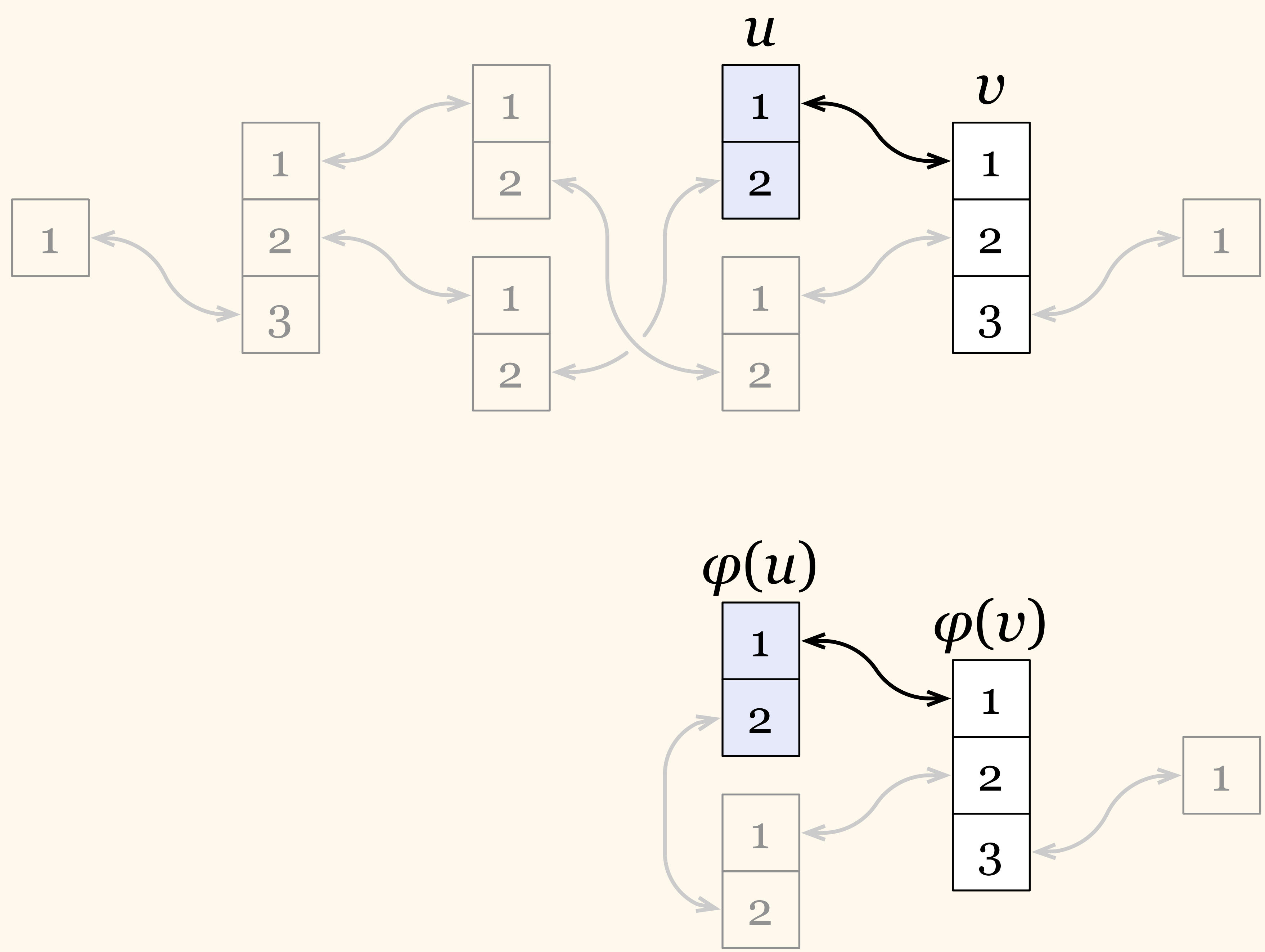


Neighbours in port 2 agree

N:

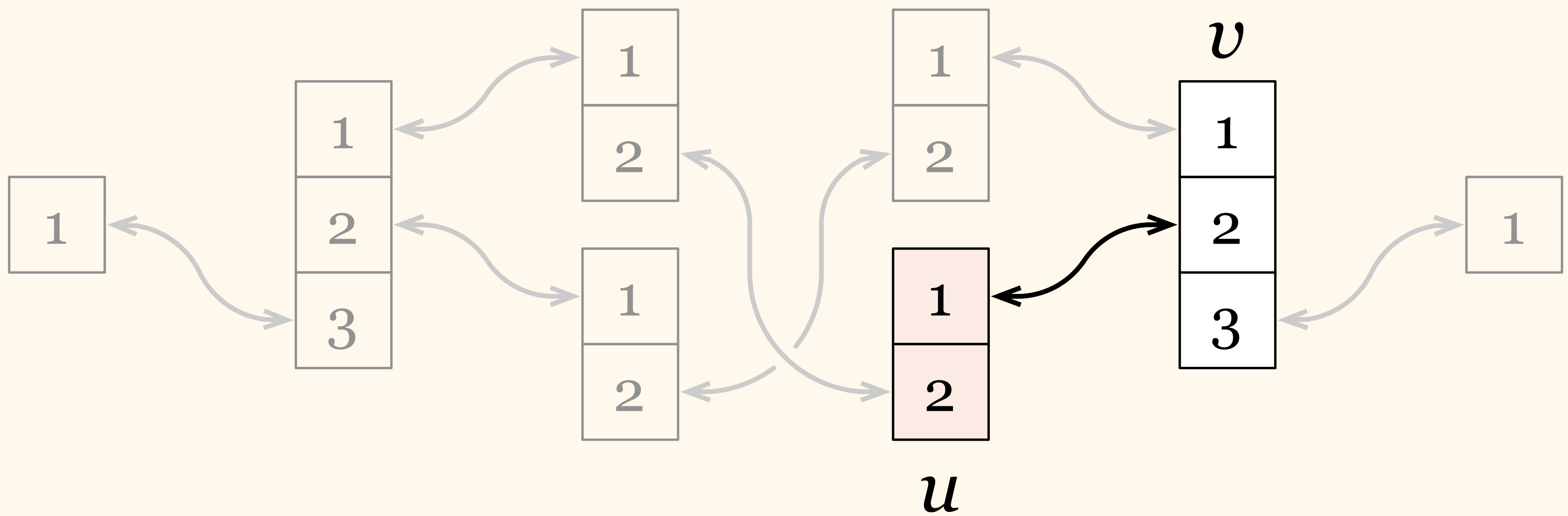
φ

N':



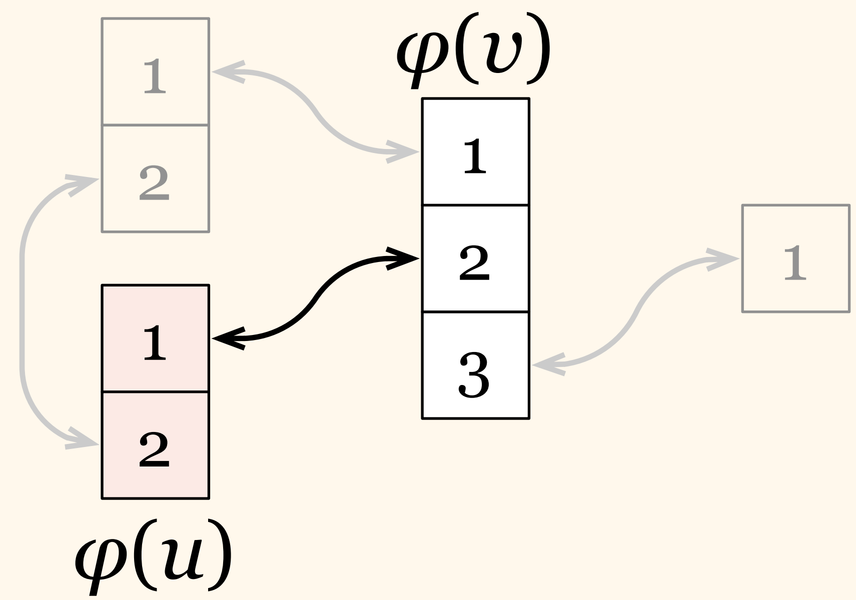
Holds for any pair of nodes

N:



φ

N':

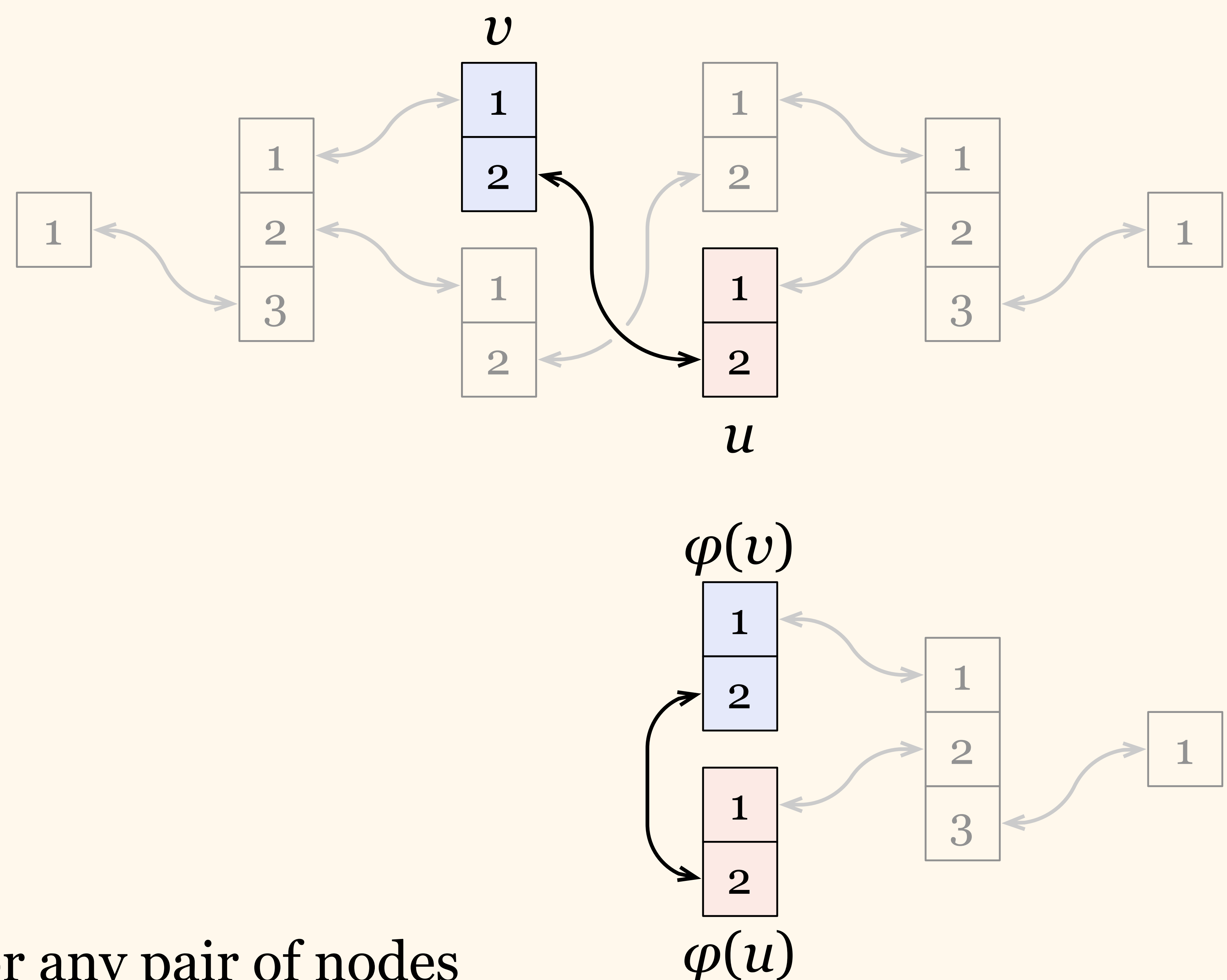


Holds for any pair of nodes

N:

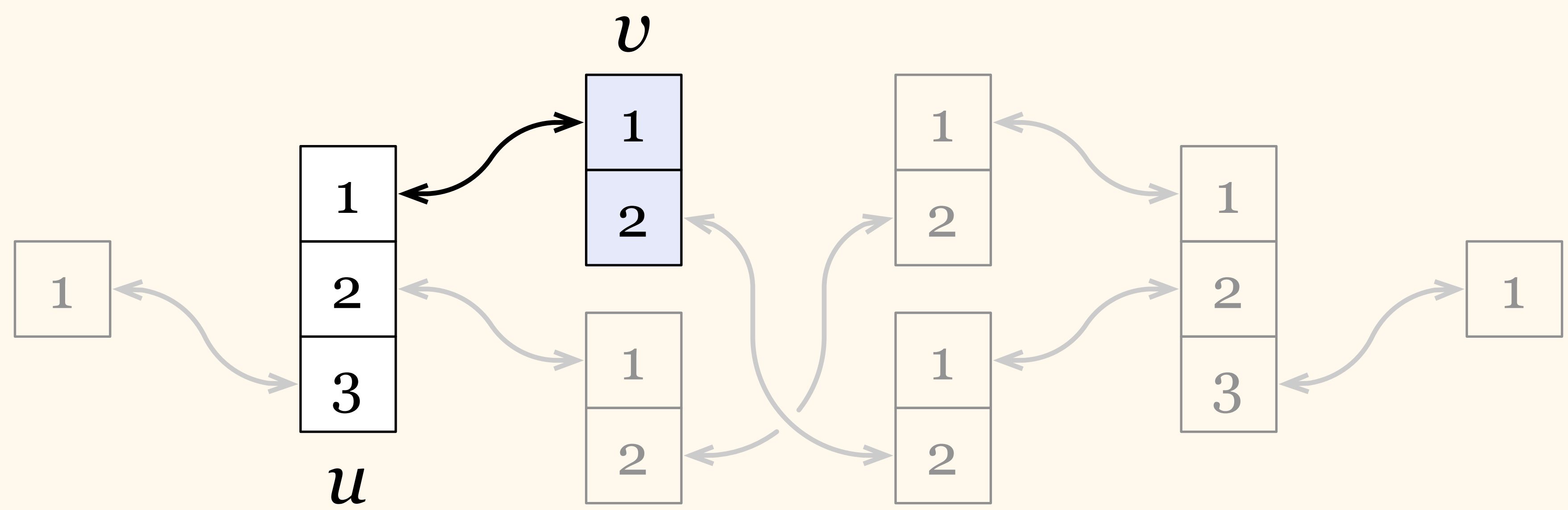
φ

N':



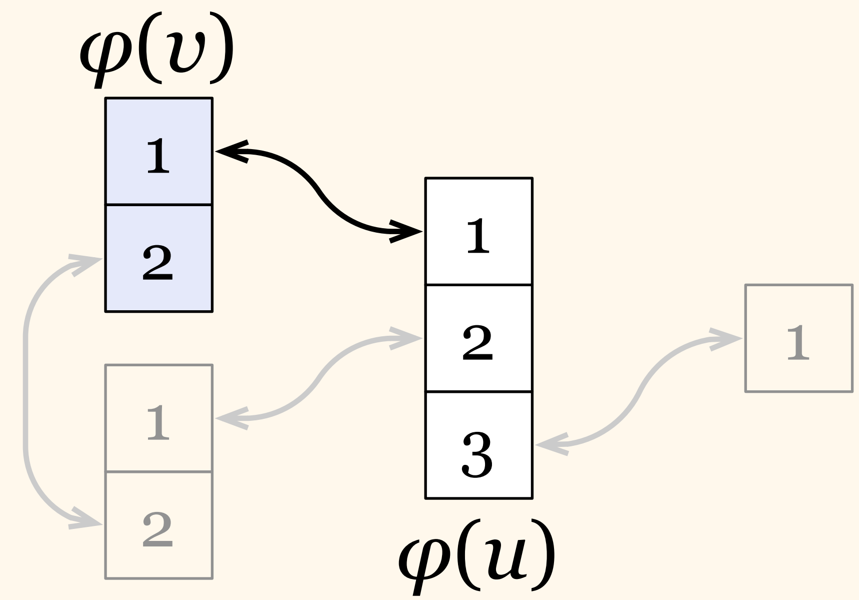
Holds for any pair of nodes

N:



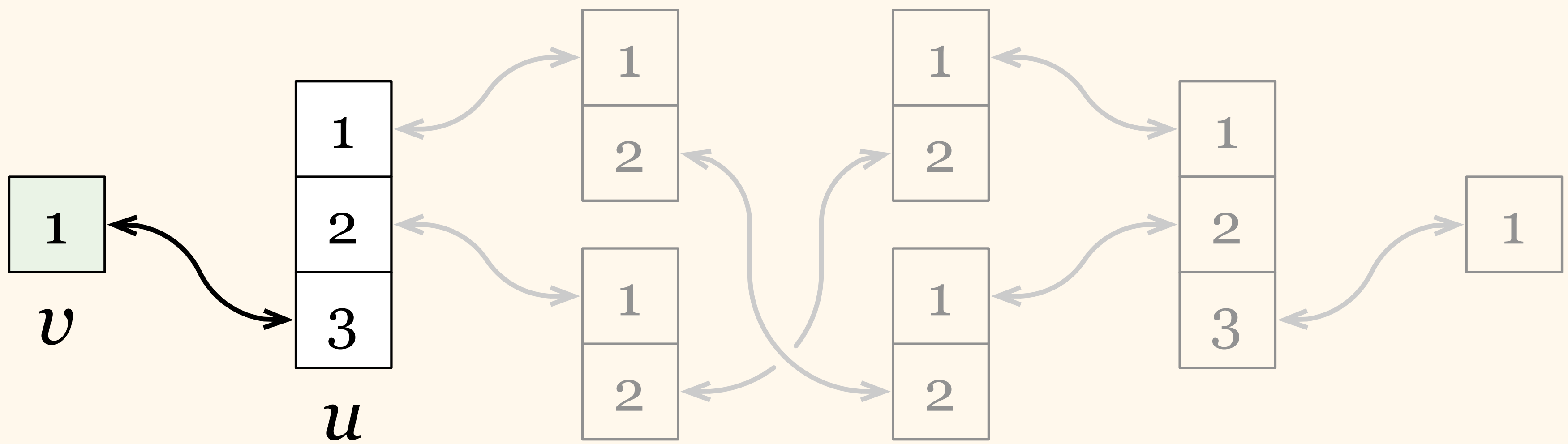
φ

N':



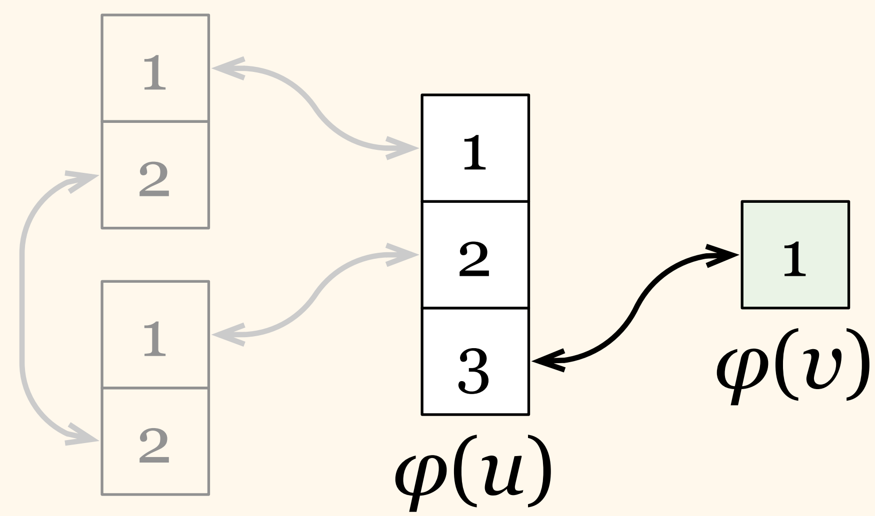
Holds for any pair of nodes

N:



φ

N':



Holds for any pair of nodes

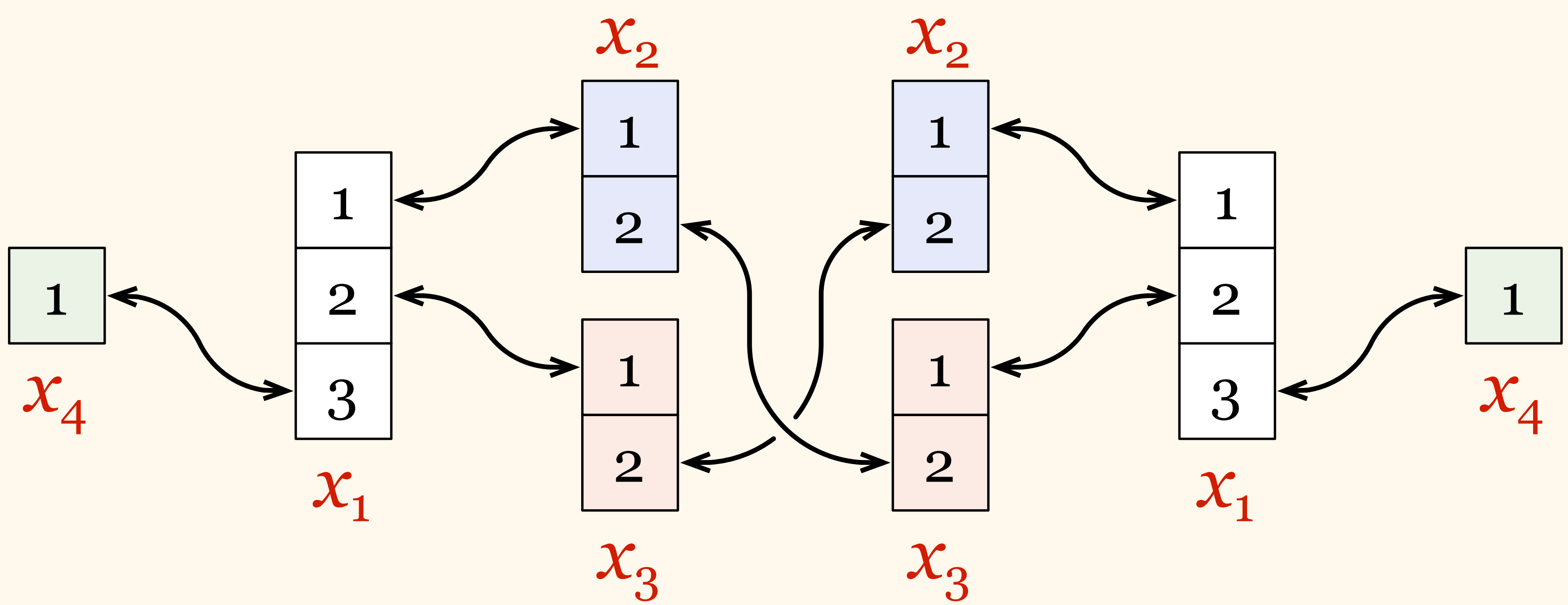
Covering Map

- Networks $N = (V, P, p)$ and $N' = (V', P', p')$
- Surjection $\varphi\colon V \to V'$ that preserves inputs, degrees, connections, and port numbers
- **Theorem**: If we run an algorithm A in N and N', then nodes v and $\varphi(v)$ are *always in the same state*

Covering Map

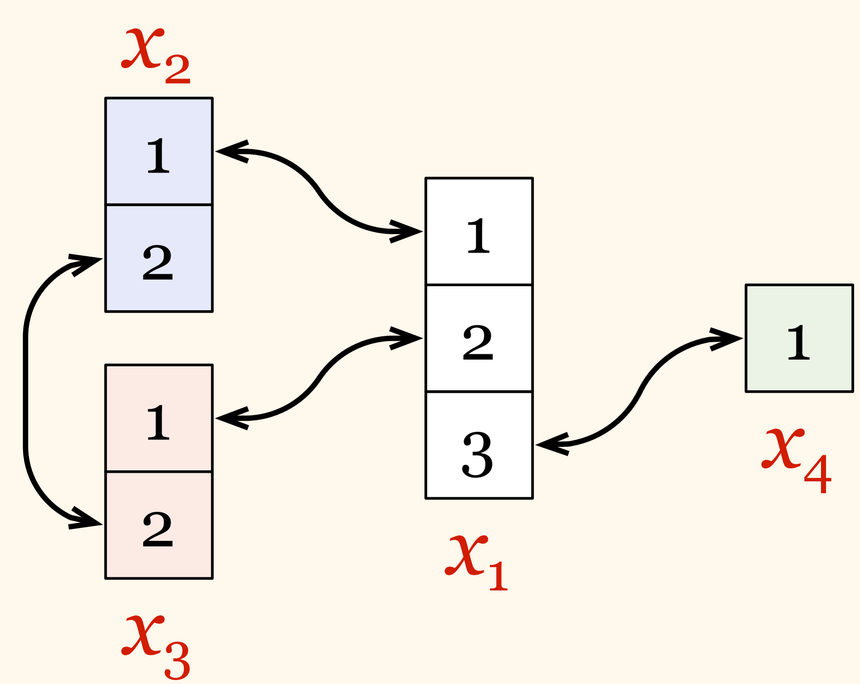
- **Theorem**: If we run an algorithm A in N and N', then nodes v and $\varphi(v)$ are *always in the same state*
- **Proof**: By induction
  - before round i: map φ preserves local states
  - during round i: map φ preserves messages
  - after round i: map φ preserves local states

N:



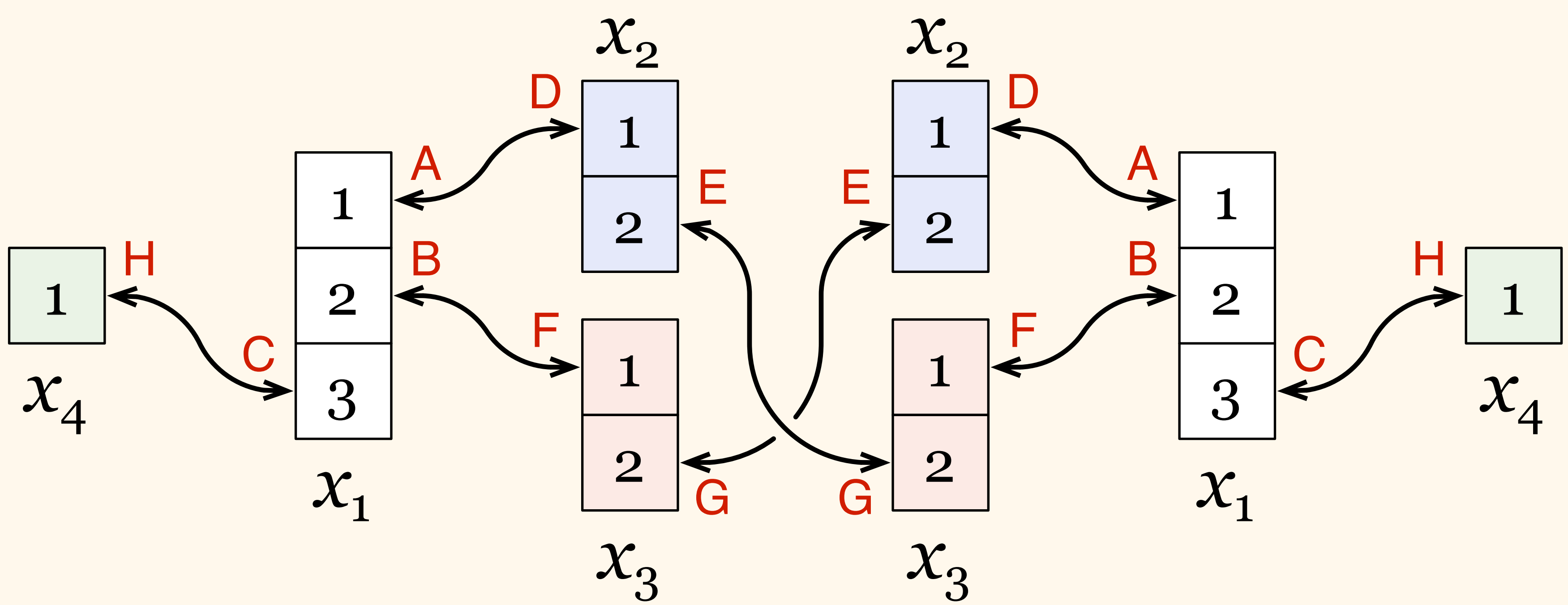
φ

N':



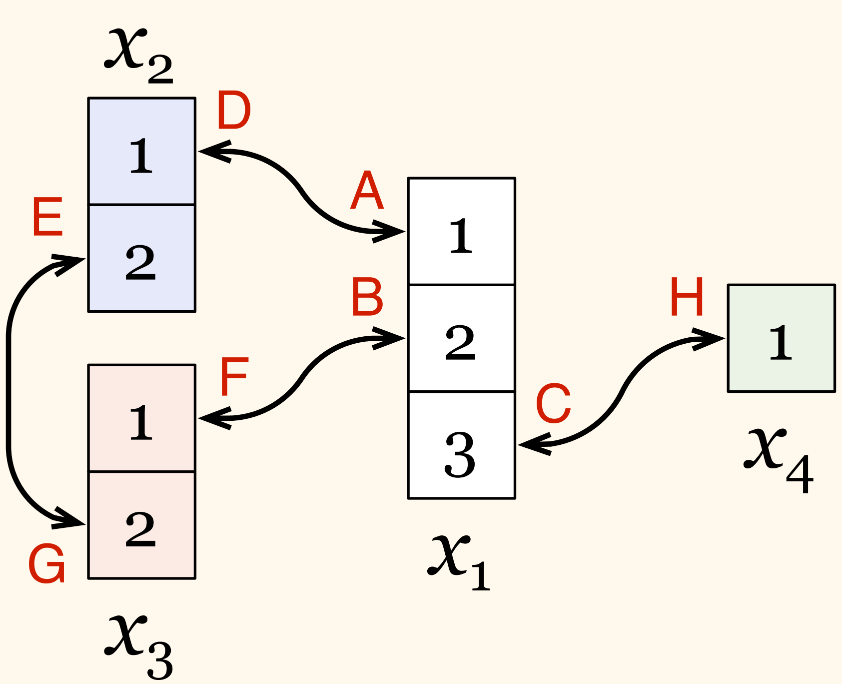
Initially, *local states* agree

N:



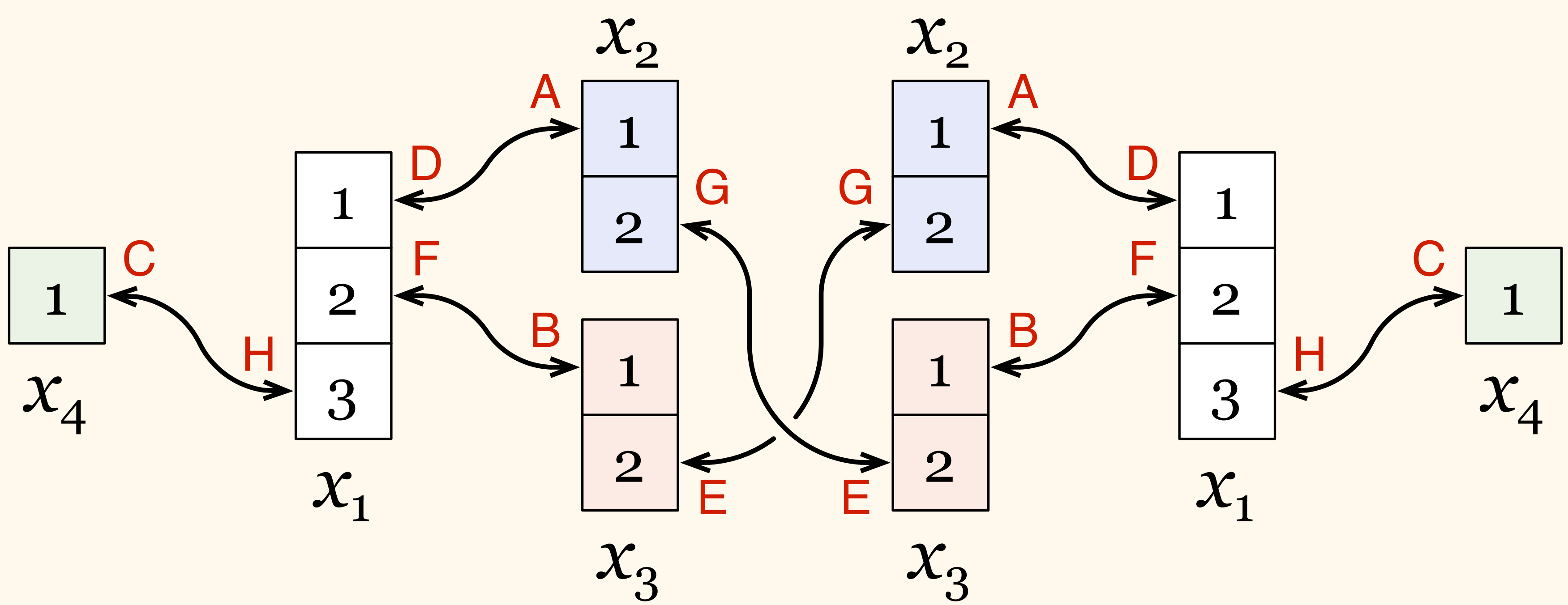
φ

N':



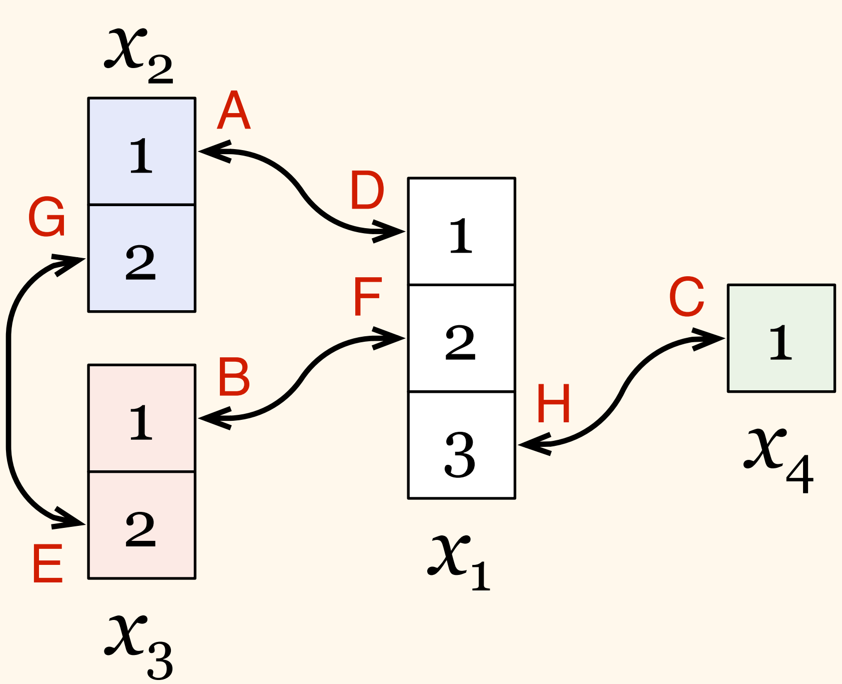
Thus *outgoing messages* agree

N:



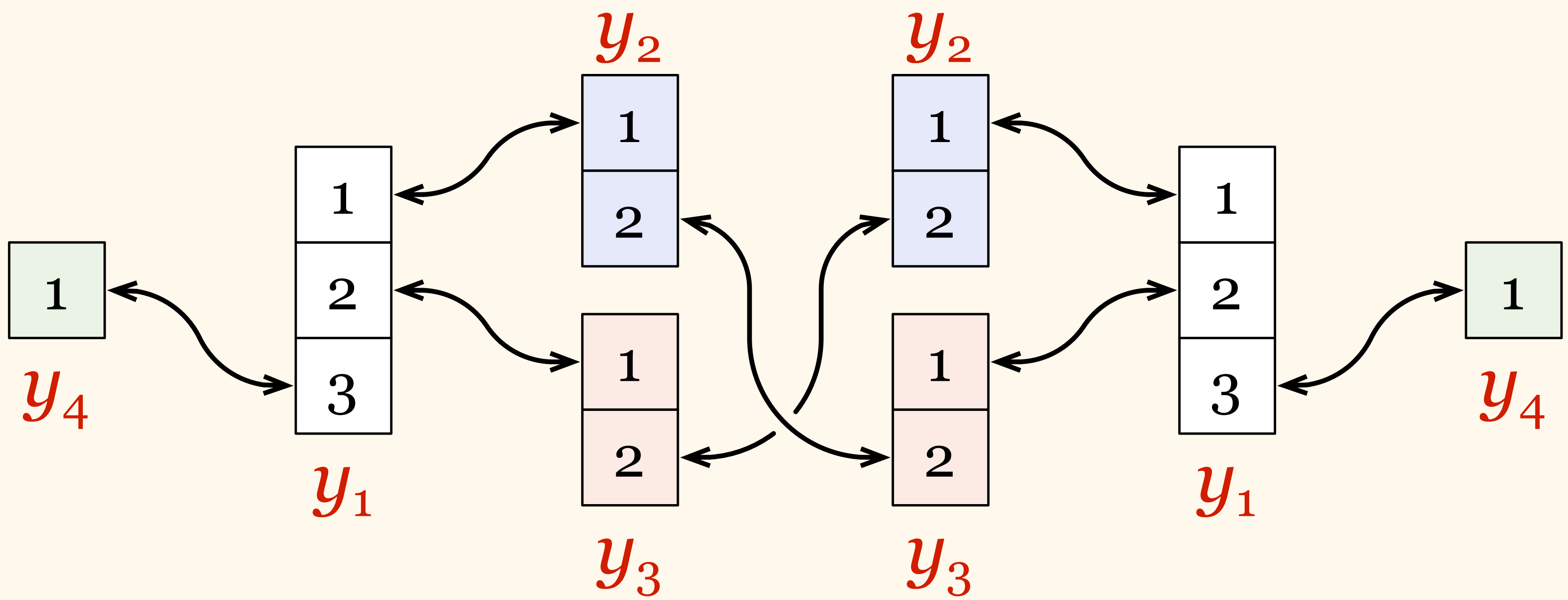
φ

N':



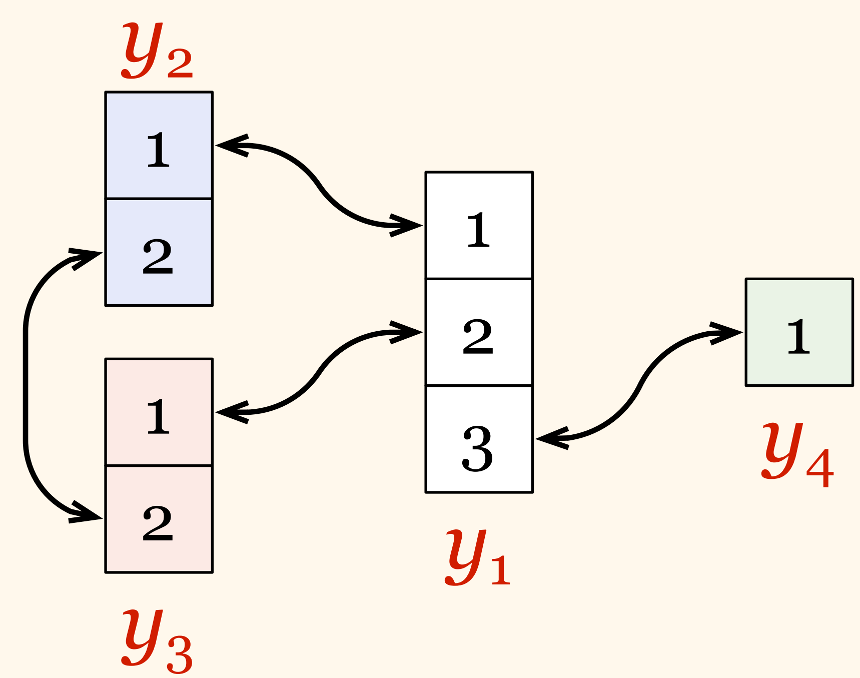
Thus *incoming messages* agree

N:



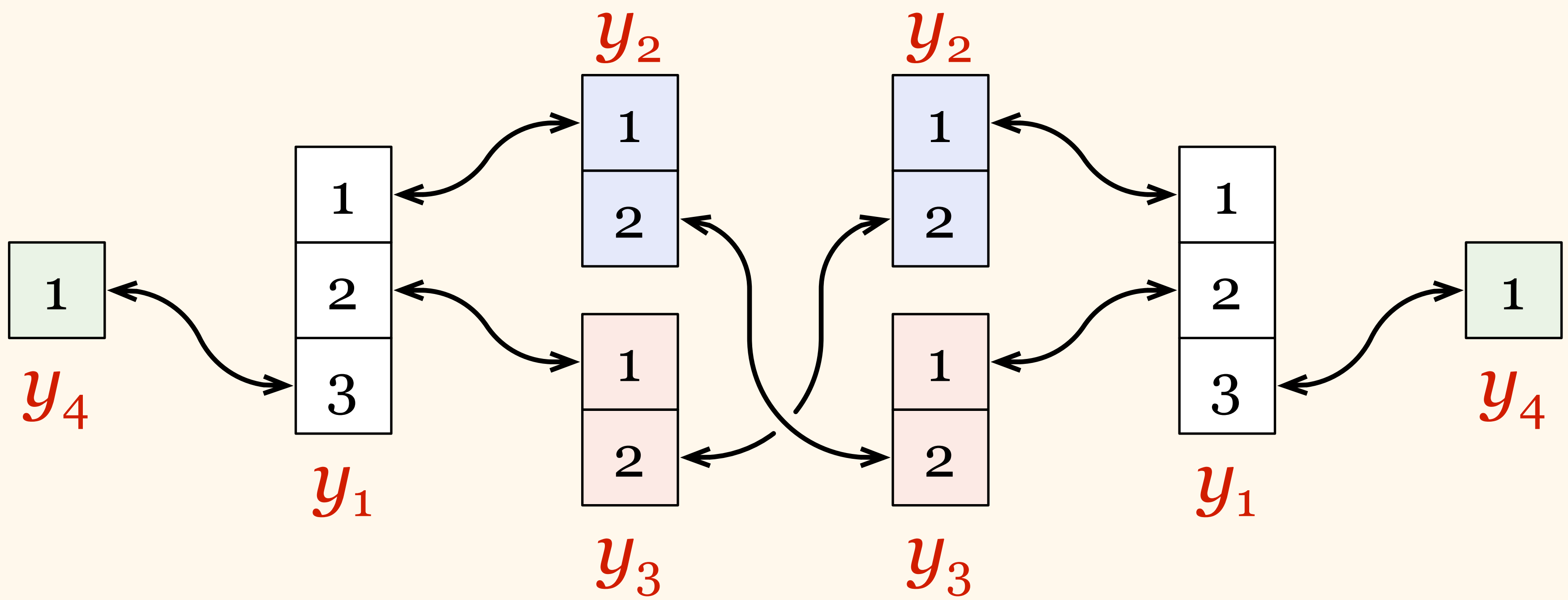
φ

N':



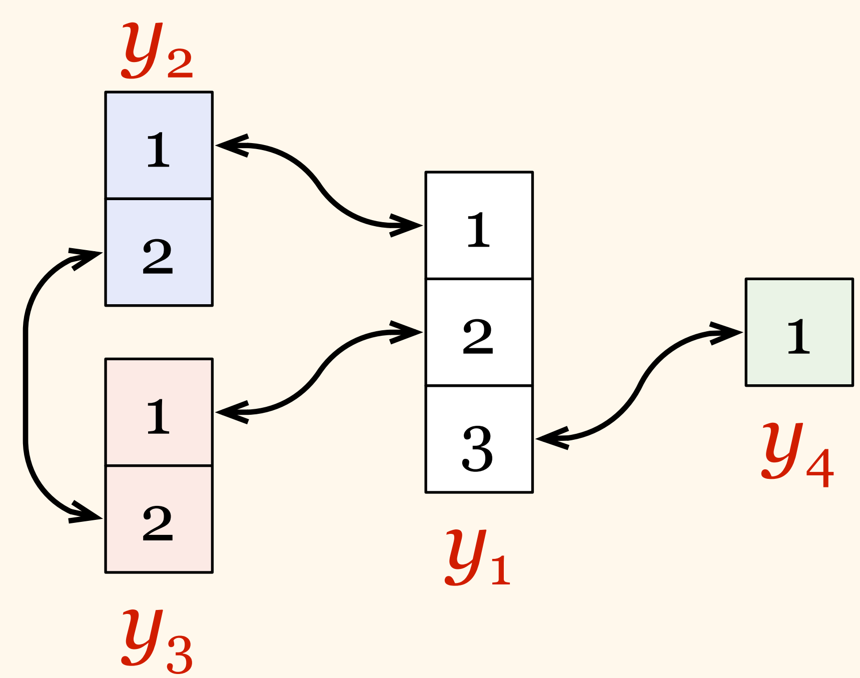
Thus *new local states* agree

N:



φ

N':

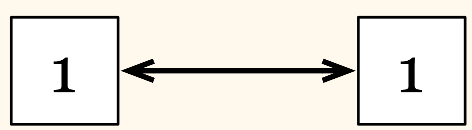


By induction, *local outputs* agree

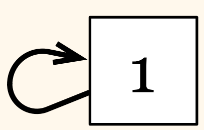
Covering Map

- Application: symmetry breaking in a path graph

N:



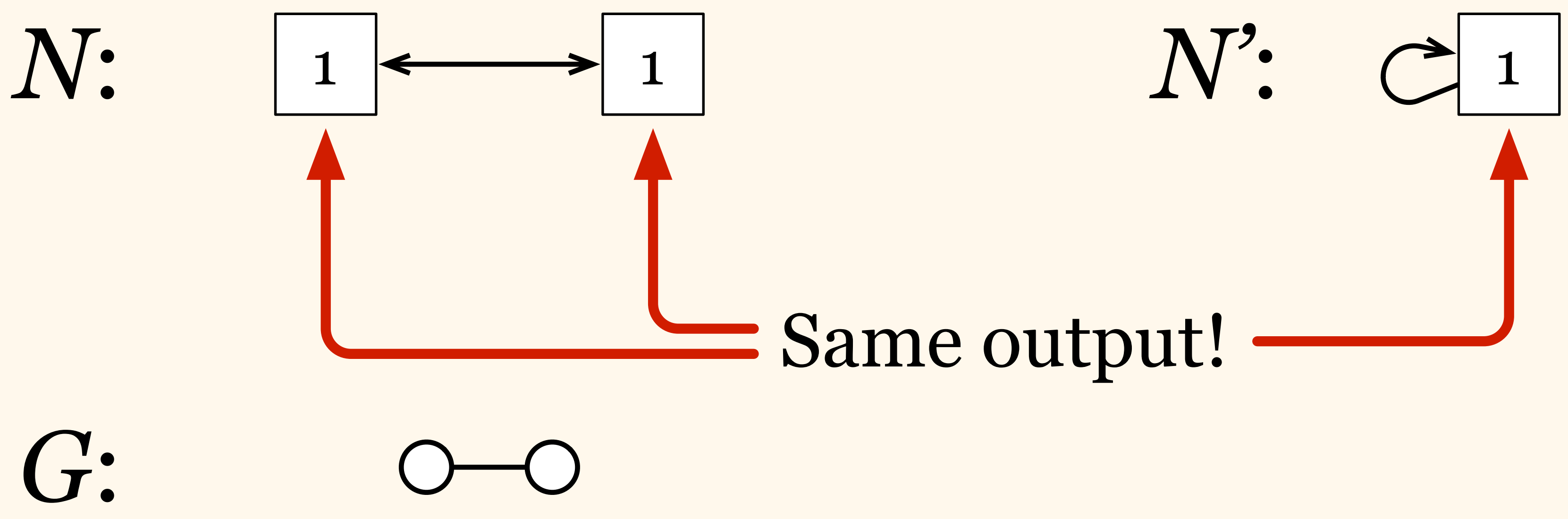
N’:



G:

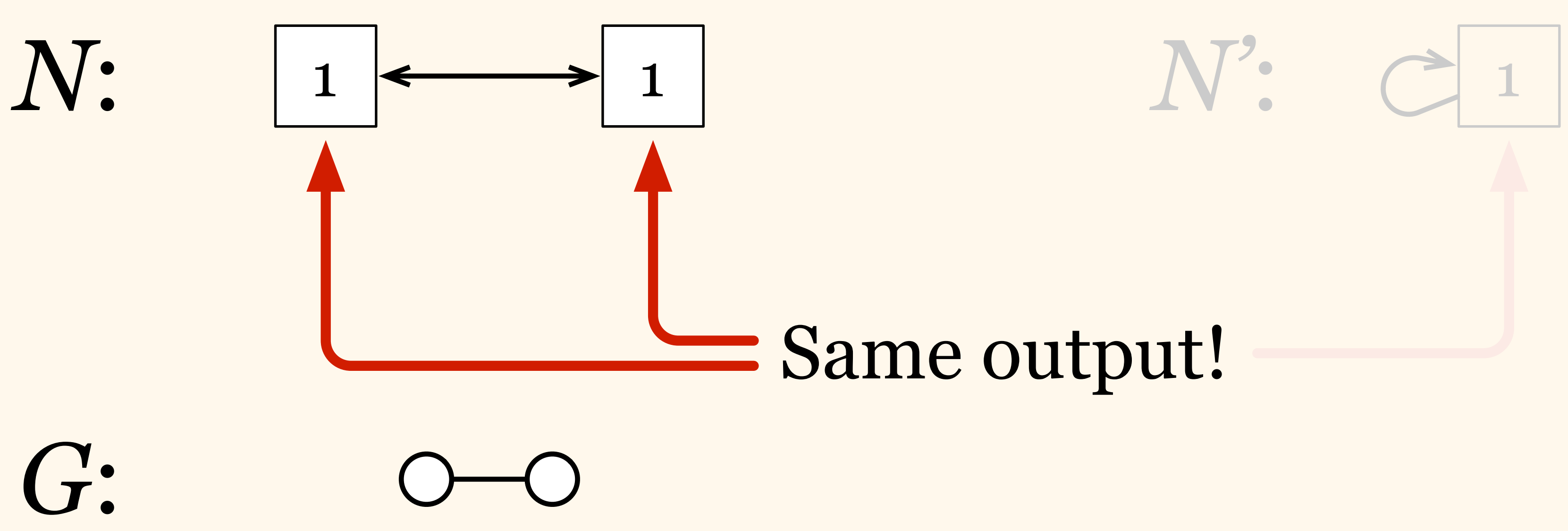
Covering Map

- Application: symmetry breaking in a path graph



Covering Map

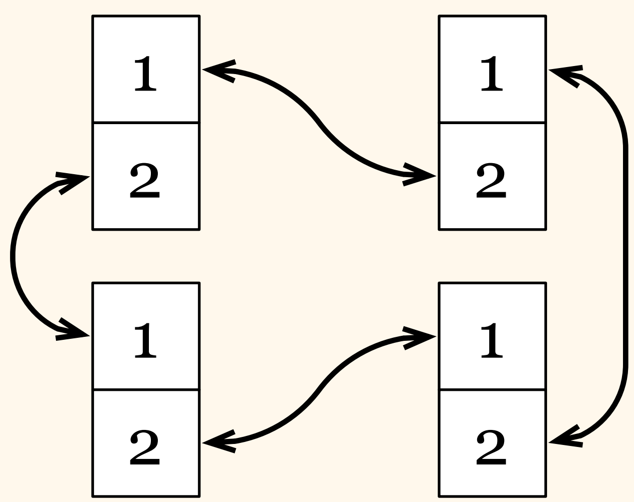
- Application: symmetry breaking in a path graph



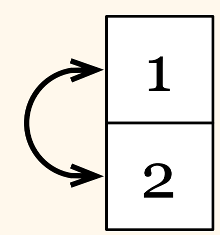
Covering Map

- Application: symmetry breaking in a cycle

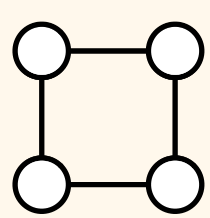
N:



N':



G:

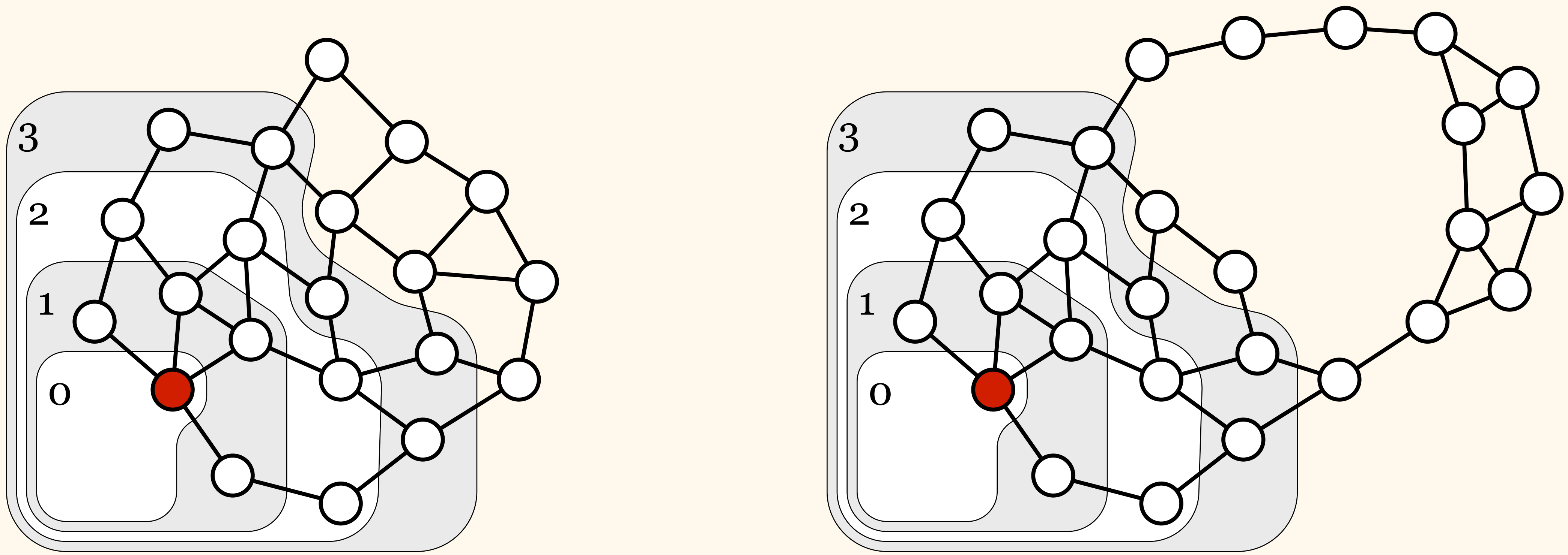


Local Neighbourhoods

- Local neighbourhoods of nodes u and v "look identical" up to distance r
  - isomorphism between radius-r neighbourhood of u and radius-r neighbourhood of v
  - preserves *inputs, degrees, connections, and port numbers*

Local Neighbourhoods

- Local neighbourhoods of nodes u and v "look identical" up to distance r

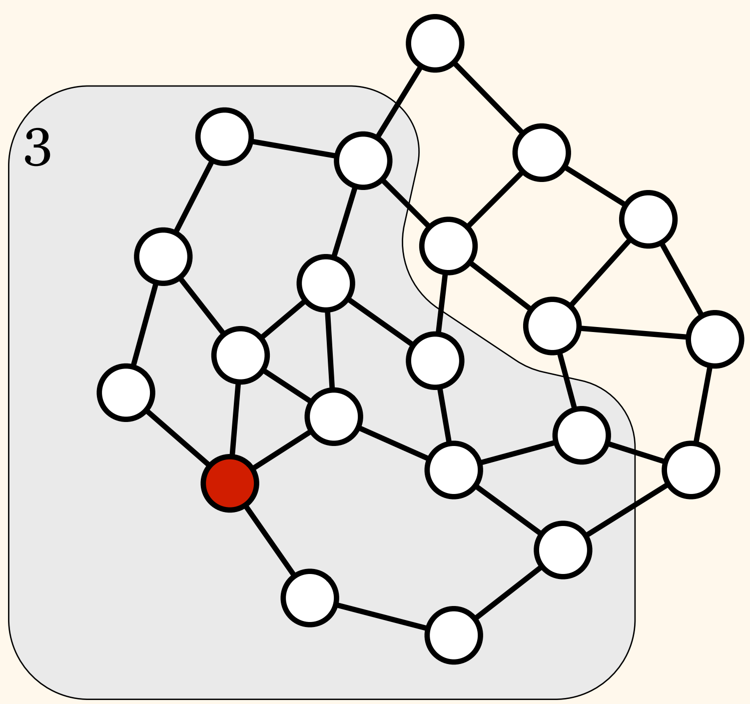


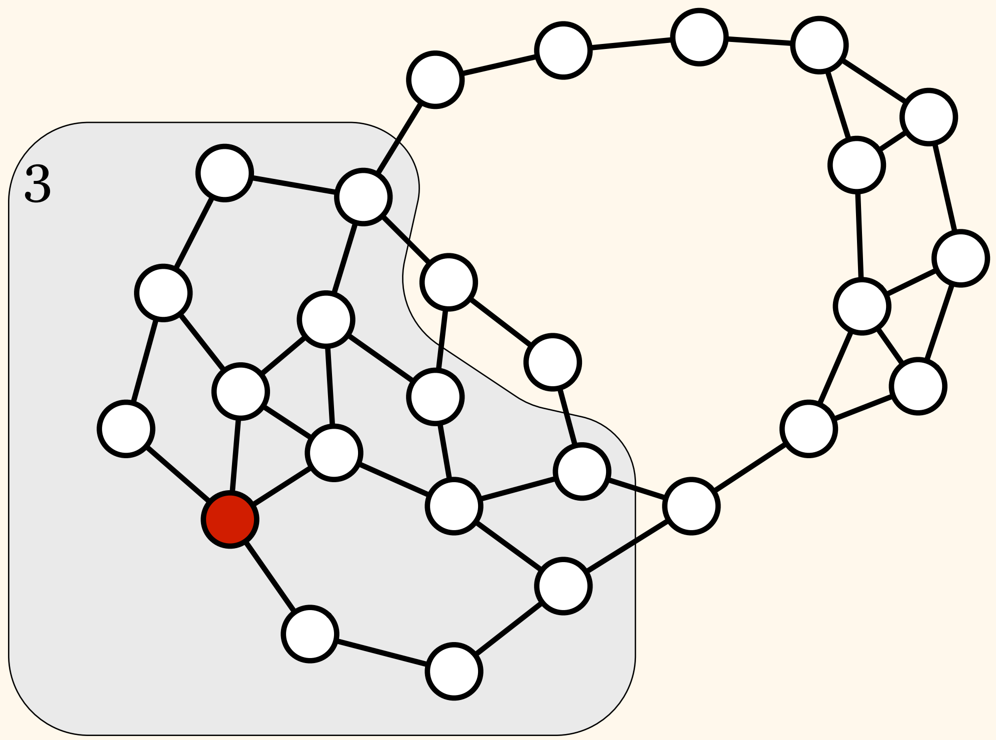
Local Neighbourhoods

- Local neighbourhoods of nodes u and v "look identical" up to distance r
- **Theorem**: In any algorithm, up to time r, the local states of u and v are identical
- *Informal proof*: time ≈ distance
- *Formal proof*: by induction on time

Local Neighbourhoods

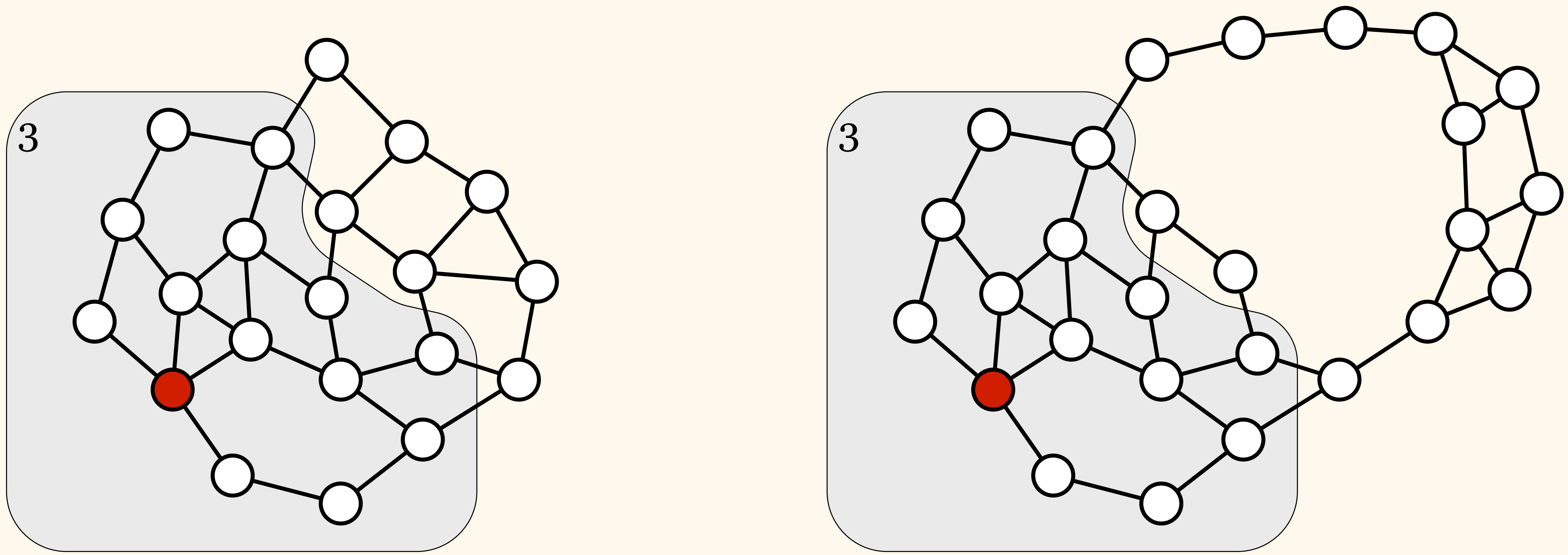
- Time 0: identical *local states* in radius-r neighbourhoods





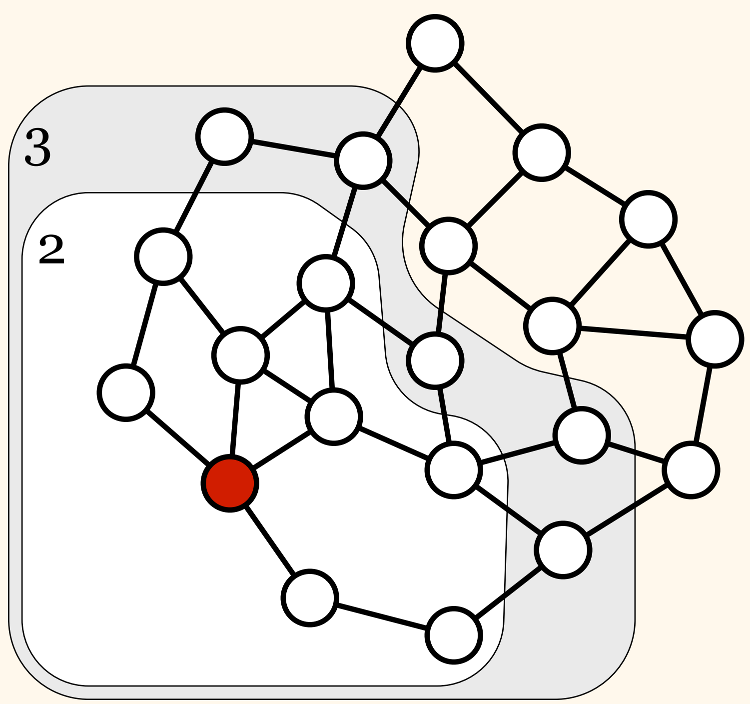
Local Neighbourhoods

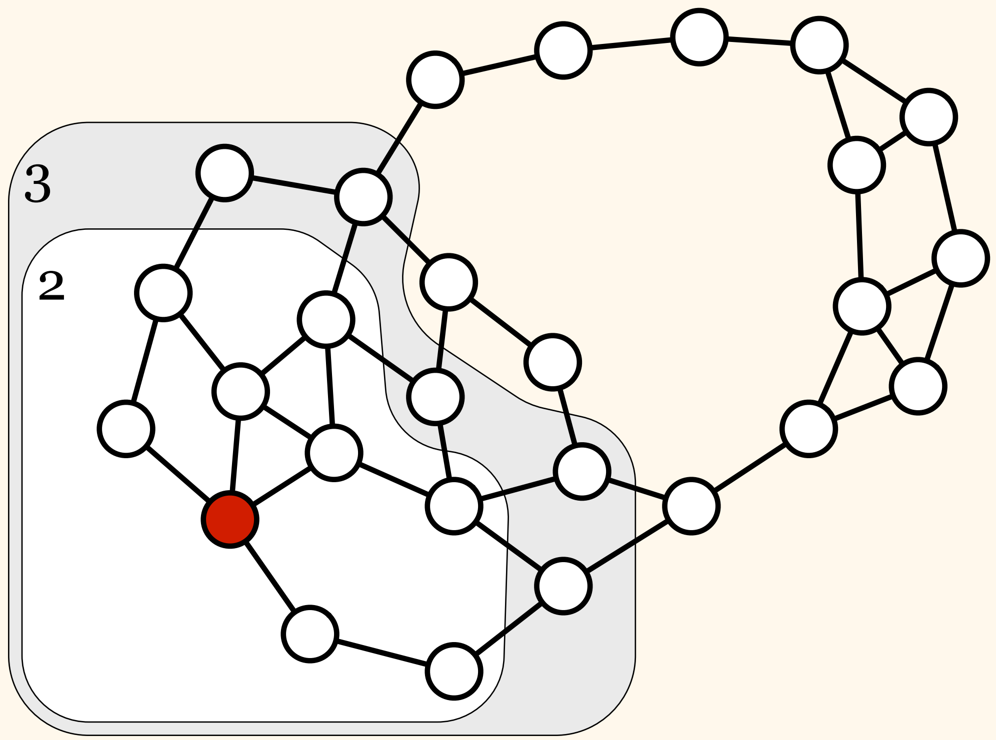
- Time 1: identical *outgoing messages* in radius-r neighbourhoods



Local Neighbourhoods

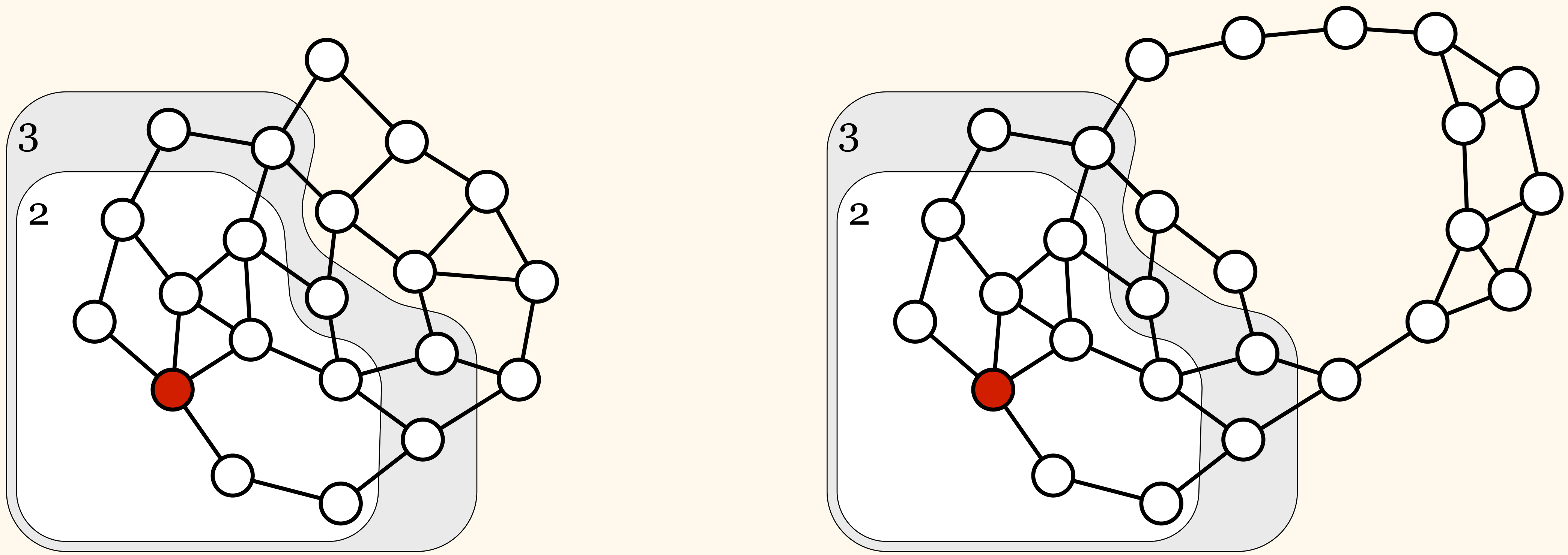
- Time 1: identical *incoming messages* in radius-(r–1) neighbourhoods





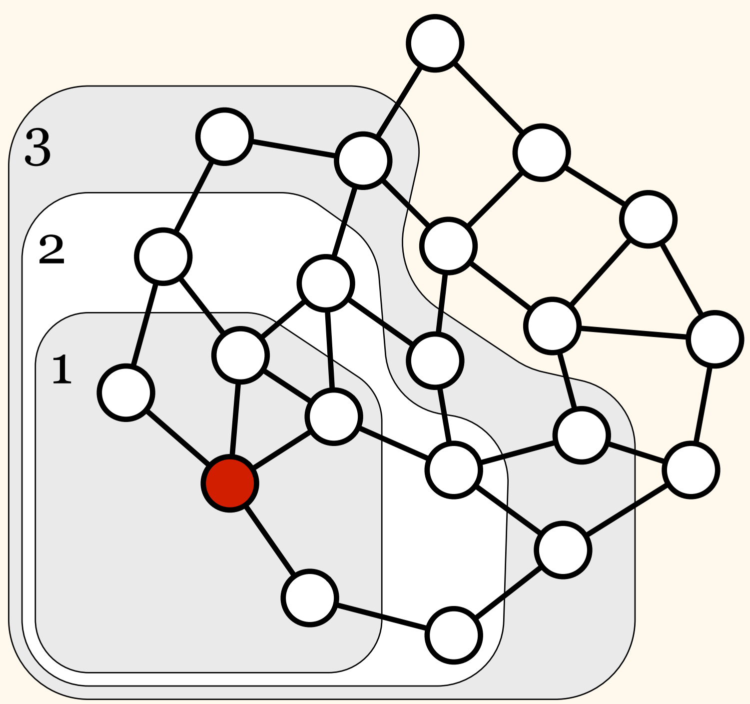
Local Neighbourhoods

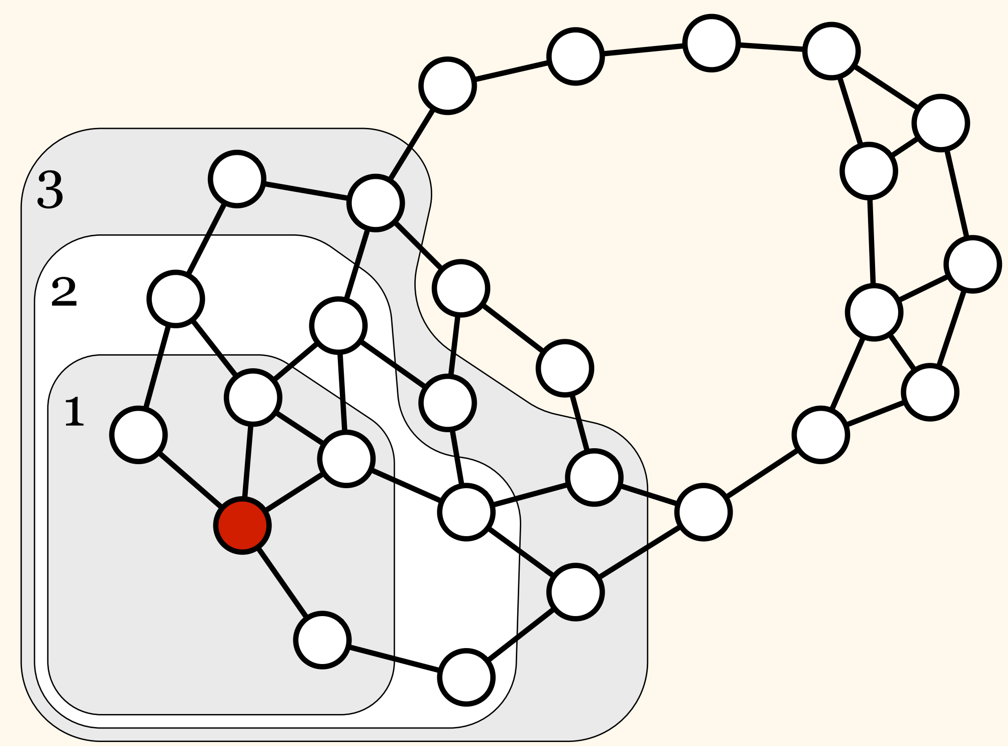
- Time 1: identical *local states* in radius-(r–1) neighbourhoods



Local Neighbourhoods

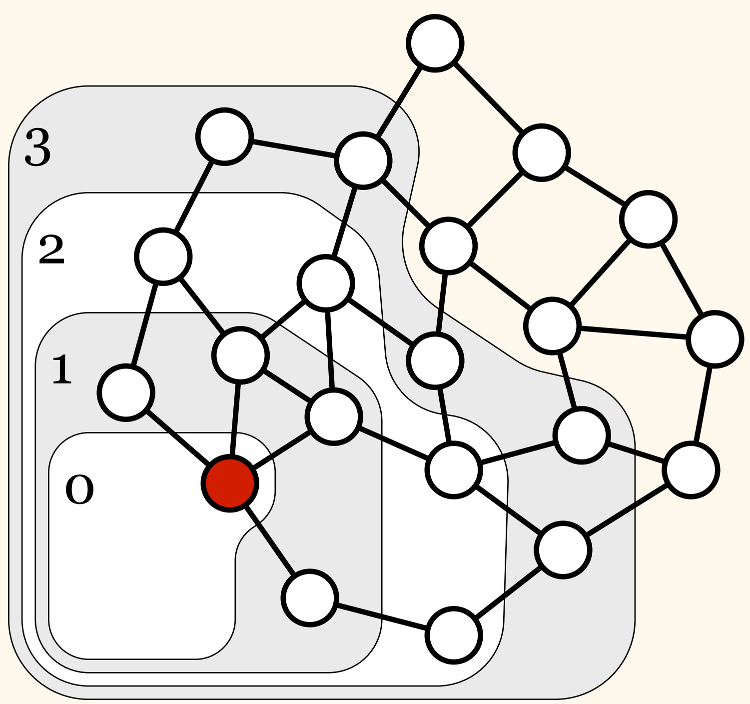
- Time *t*: identical *local states* in radius-(r–t) neighbourhoods

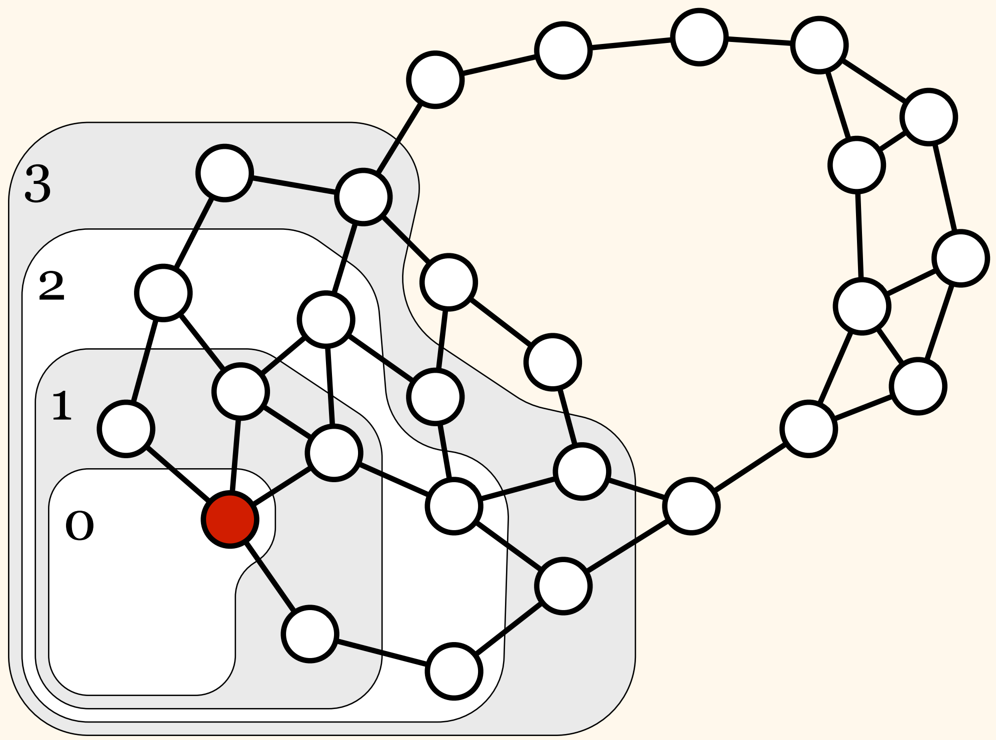




Local Neighbourhoods

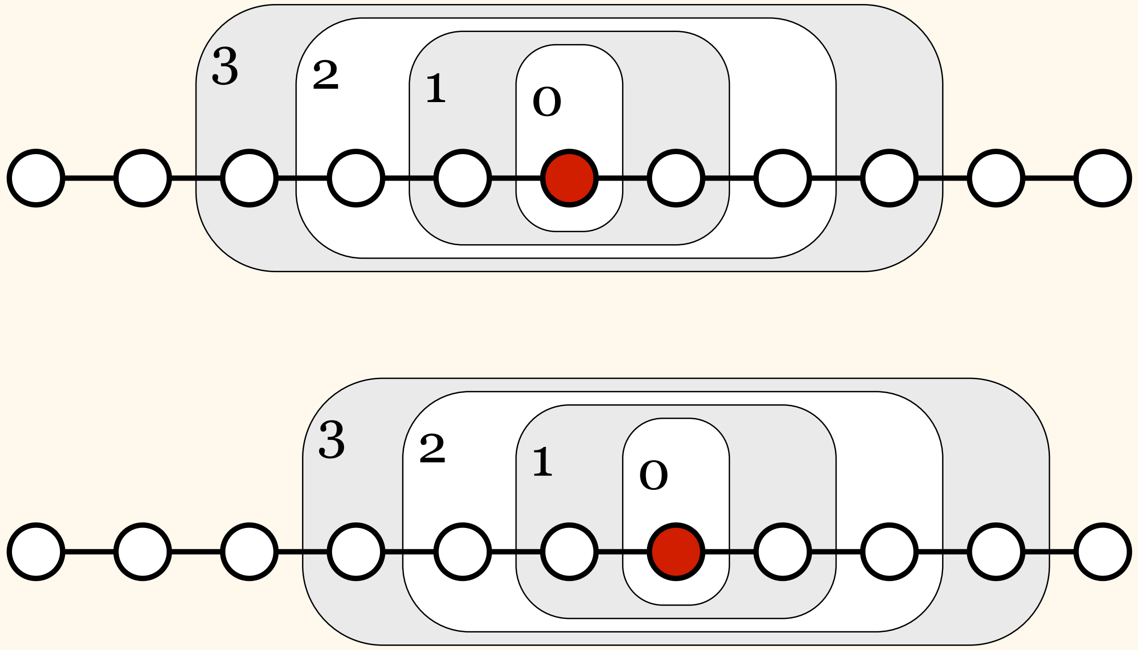
- Time *r*: identical *local states* in radius-0 neighbourhoods





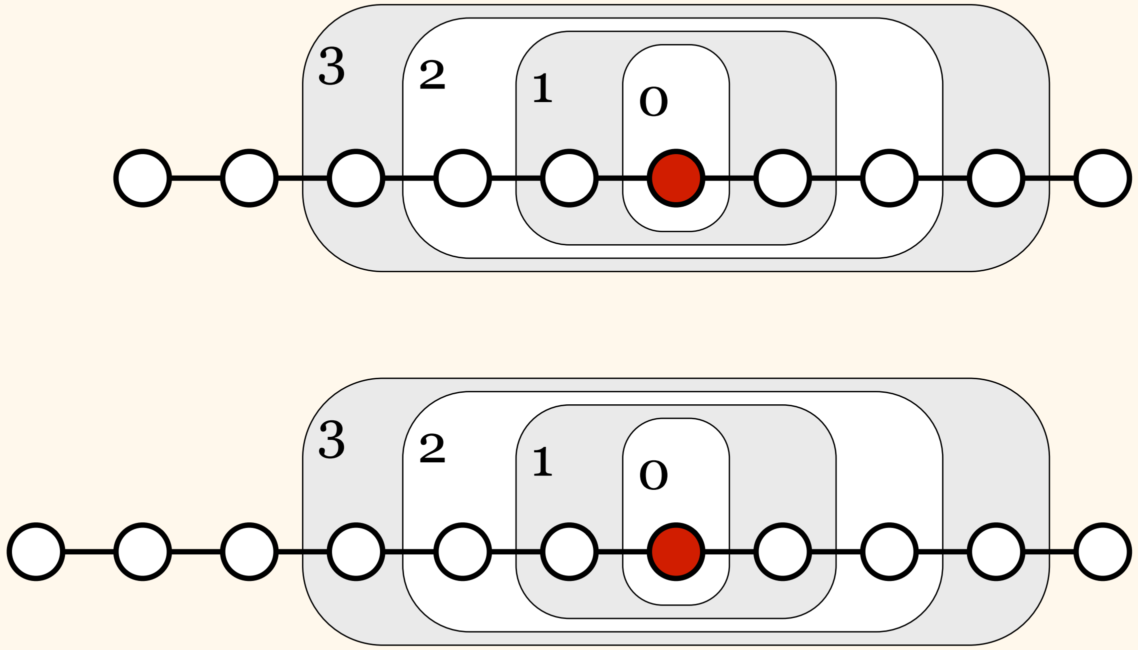
Local Neighbourhoods

- Application: finding midpoint of a path requires $\Omega(n)$ rounds



Local Neighbourhoods

- Application: counting the number of nodes requires $\Omega(n)$ rounds



Proof Techniques

- Covering maps
  - problems that cannot solved at all
- Isomorphic local neighbourhoods
  - problems that cannot be solved quickly
- Plenty of exercises…