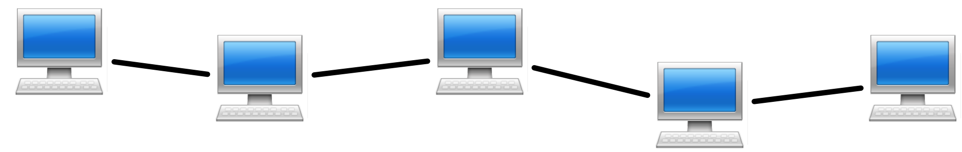
- **Weeks 1–2: informal introduction**
  - network = path 
- **Week 3: graph theory**
- **Weeks 4–7: models of computing**
  - what can be computed (efficiently)?
- **Weeks 8–11: lower bounds**
  - what cannot be computed (efficiently)?
- **Week 12: recap**

Week 8

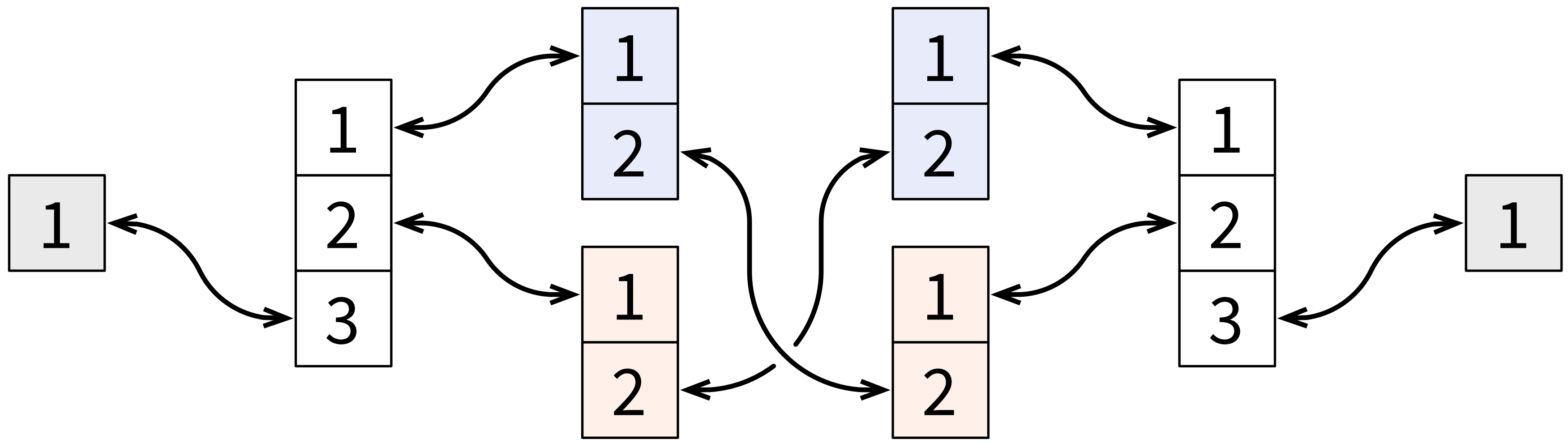
– Covering maps

Covering map

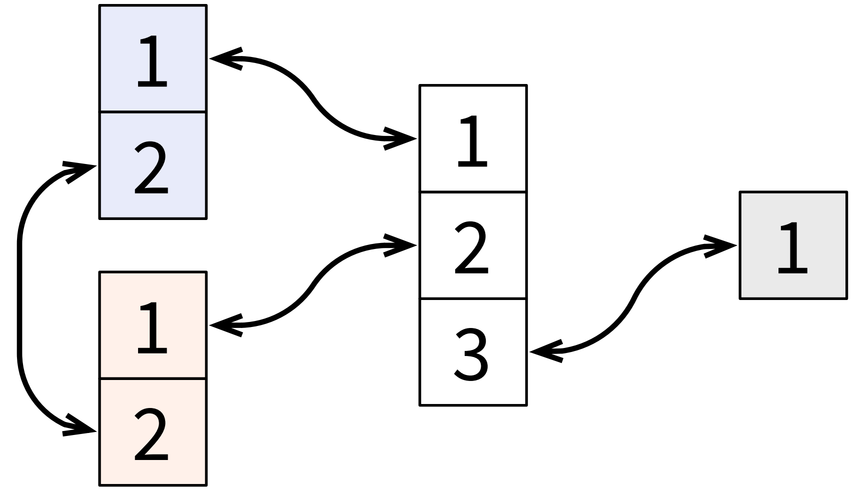
- **Networks $N = (V, P, p)$ and $N' = (V', P', p')$**
- **Surjection $\varphi: V \to V'$ that preserves inputs, degrees, connections, port numbers**
- **“Fools” any deterministic PN-algorithm: cannot distinguish between N and N'**

Networks *N* and *N'*

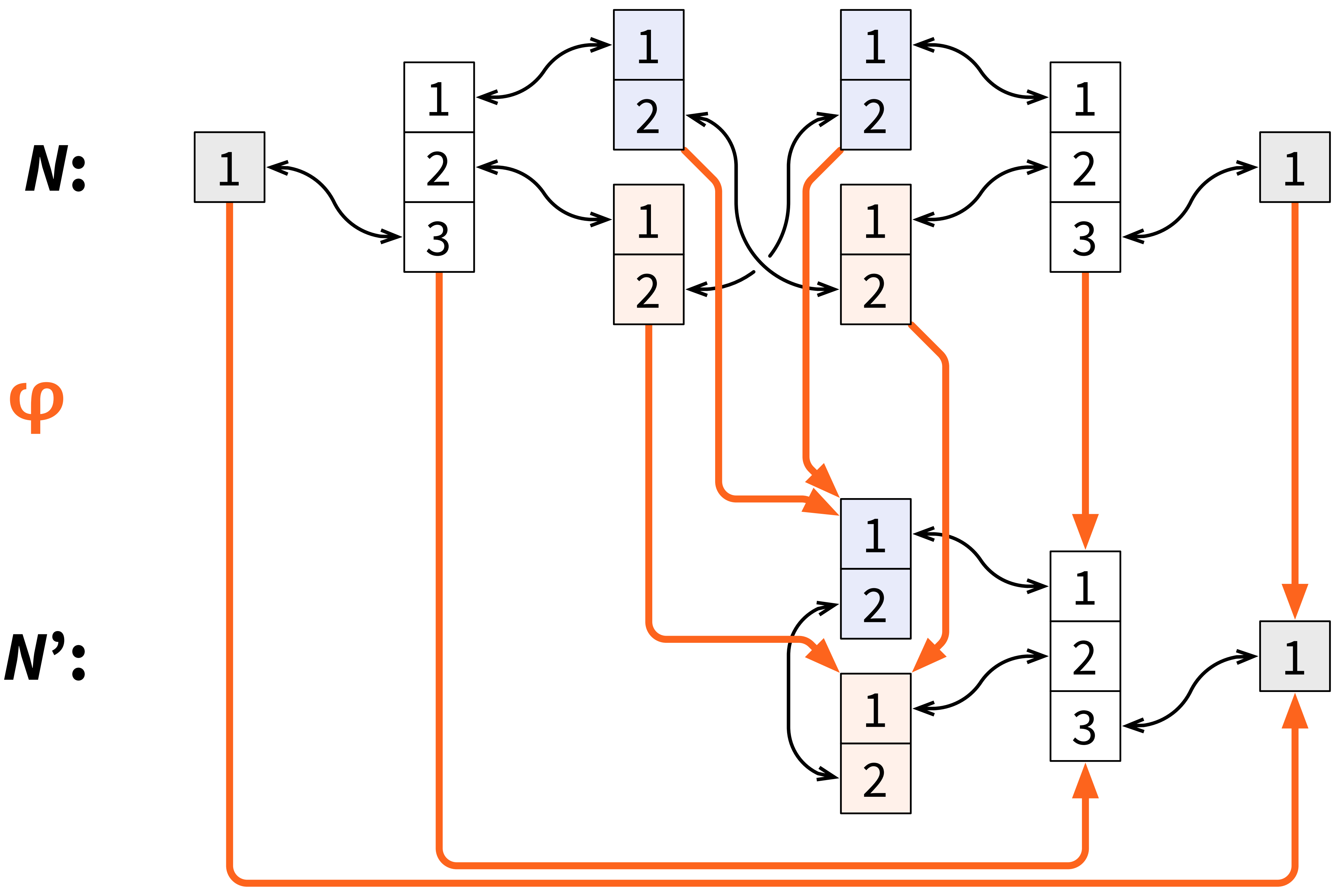
***N*:**



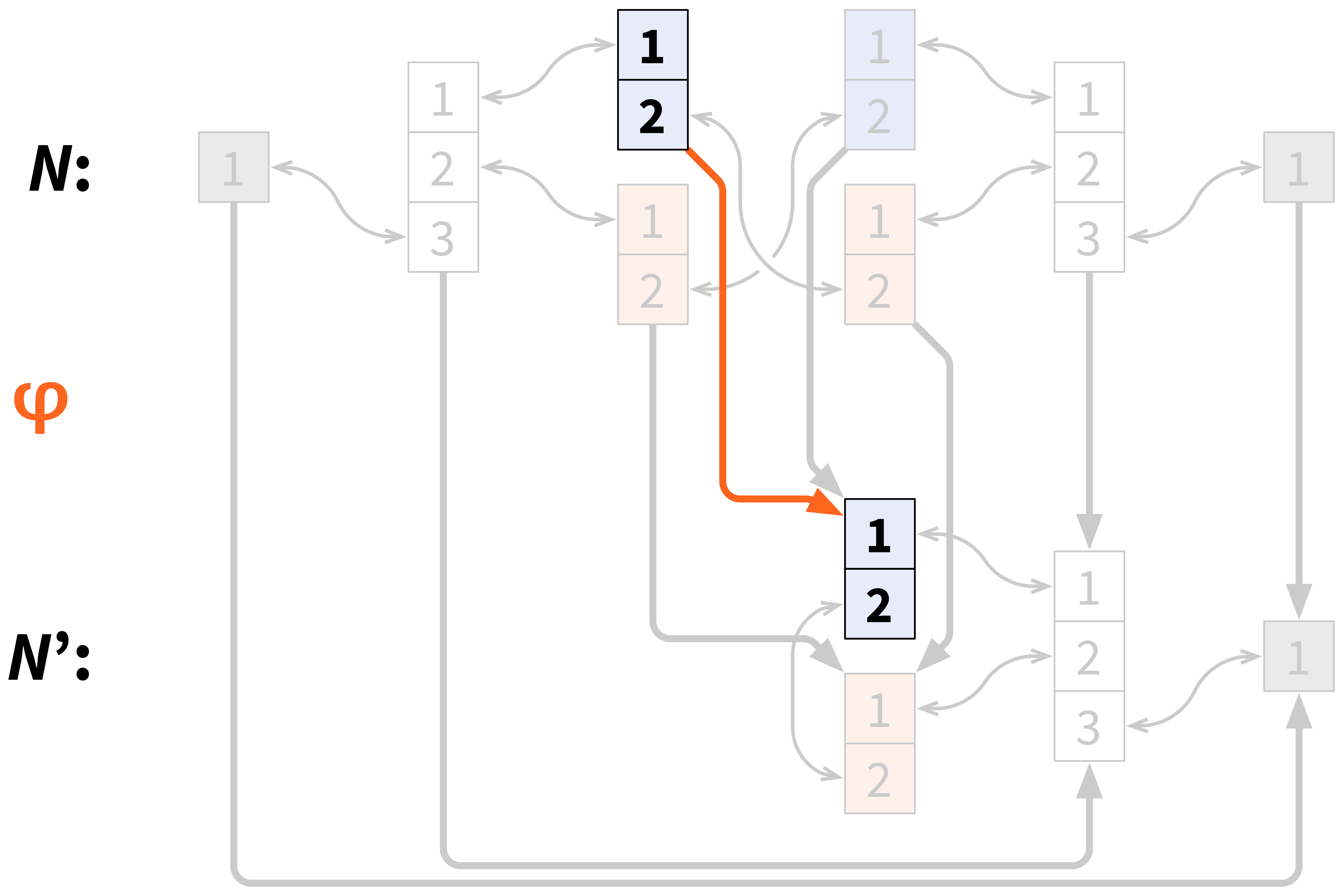
***N'*:**



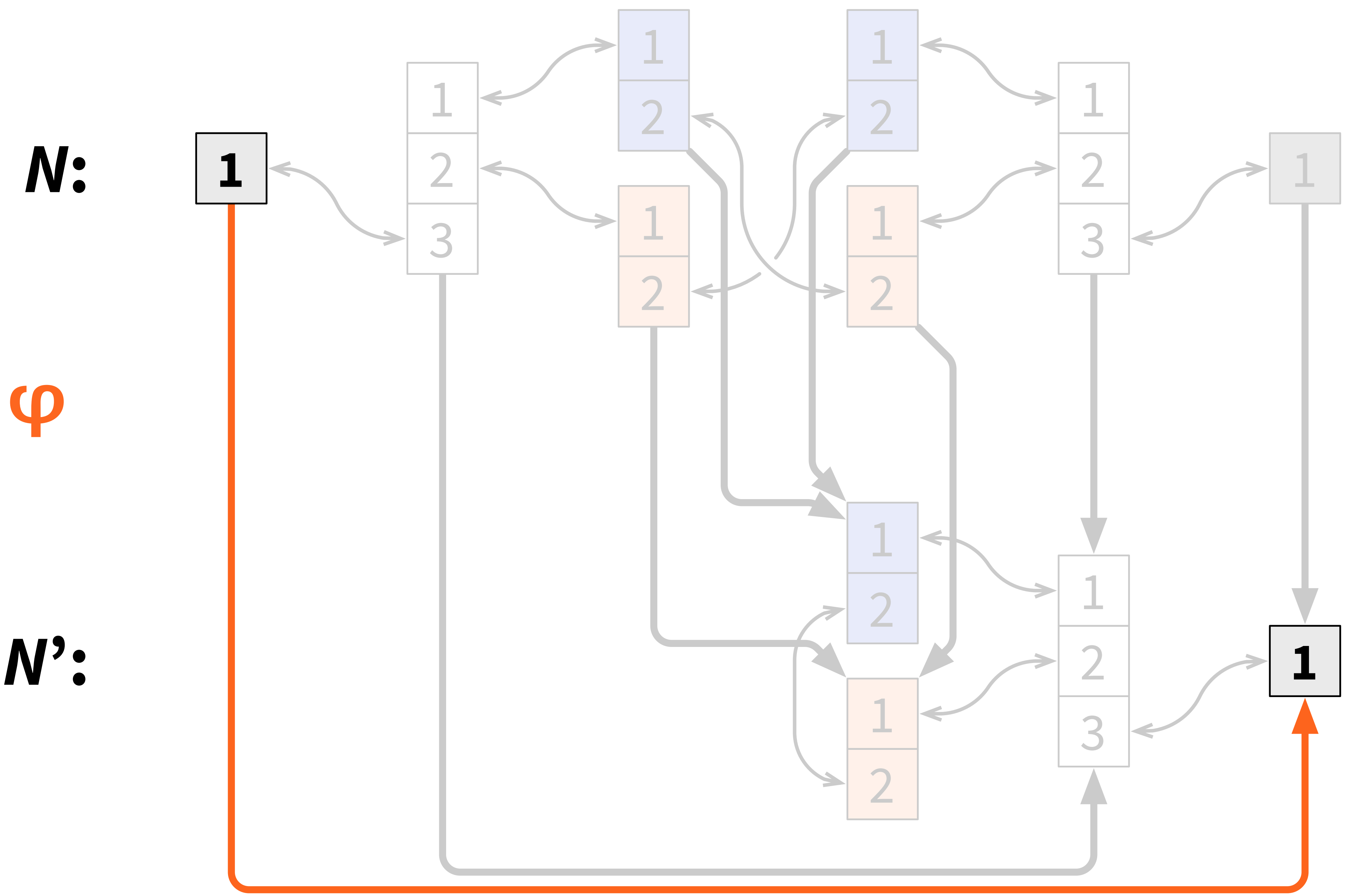
Covering map φ: *V* → *V'*



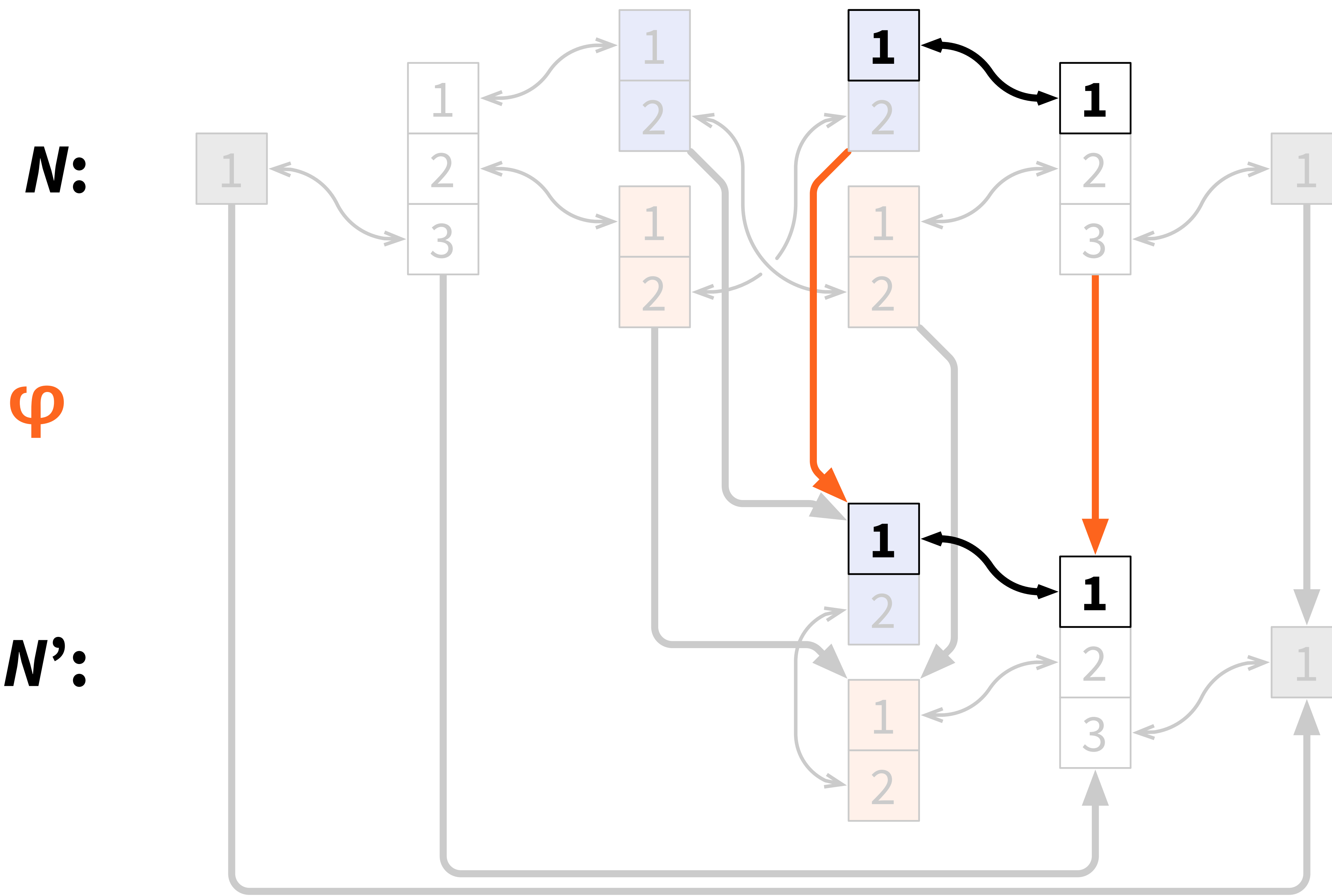
Preserves degrees



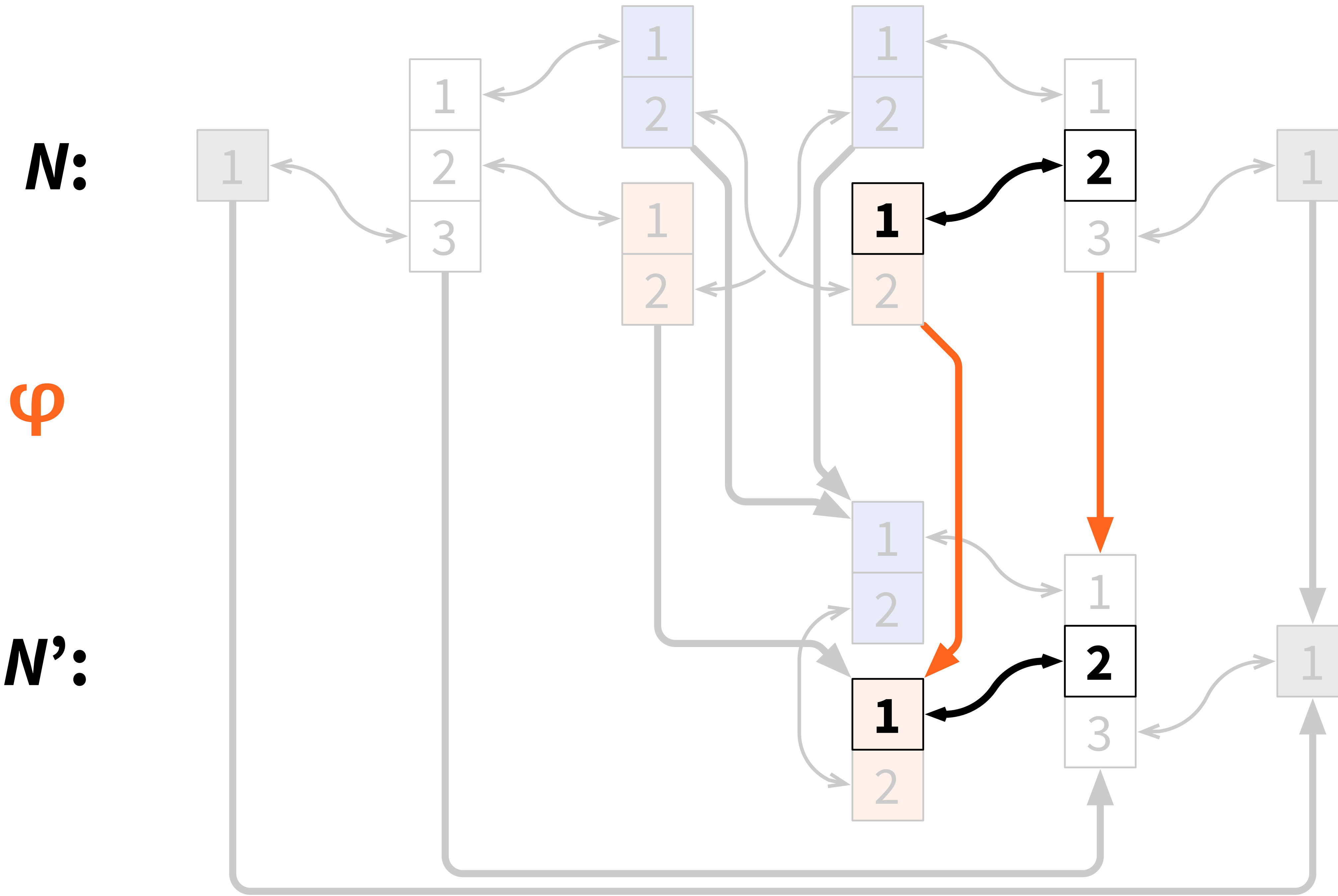
Preserves degrees



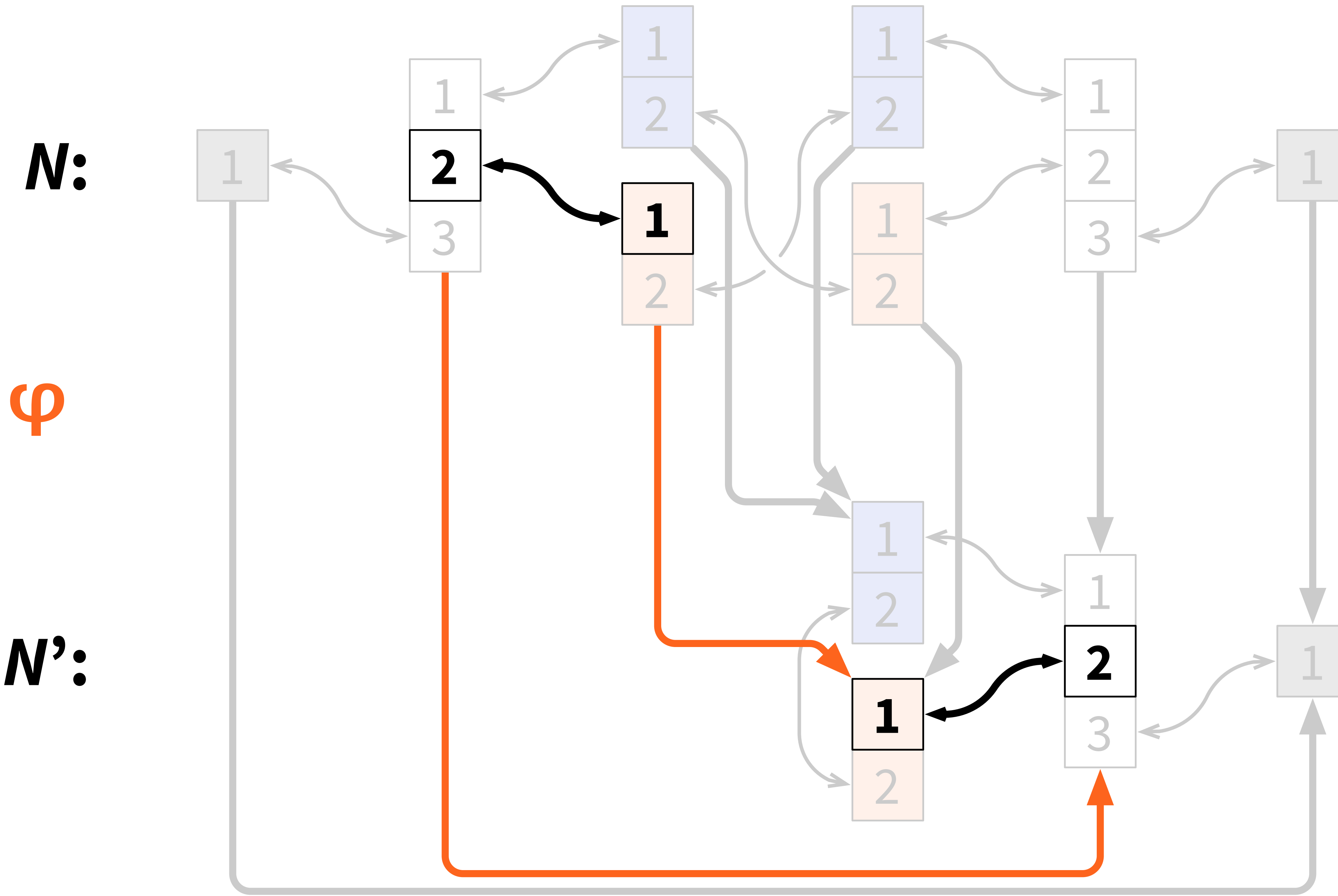
Preserves connections & port numbers



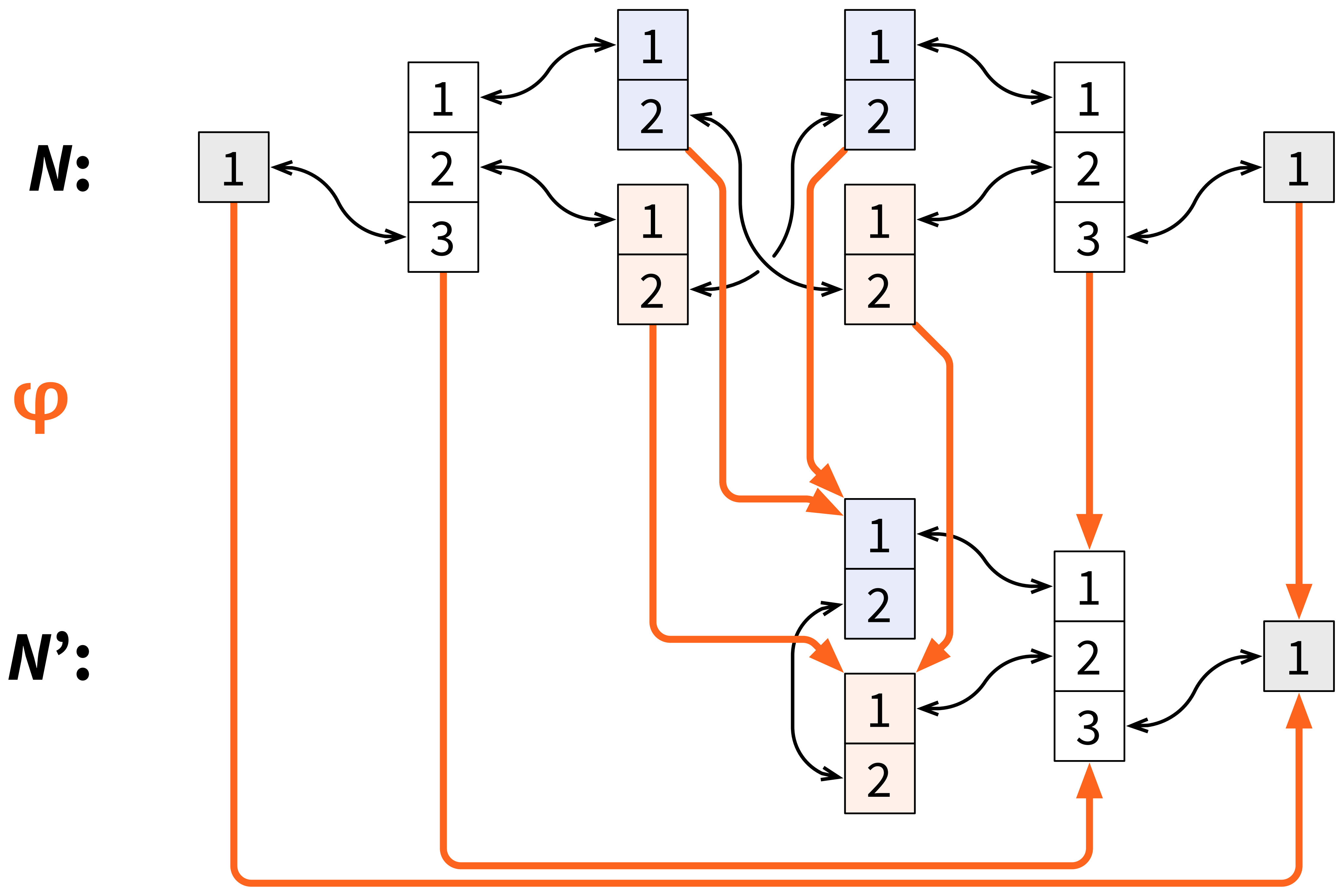
Preserves connections & port numbers



Preserves connections & port numbers



Theorem: preserves outputs!



Covering map

- φ covering map from N to N',
  A deterministic PN-algorithm

- Run A on N and N'

- Theorem: v and $\varphi(v)$ always in the same state

Covering map

- **Theorem:** ***v*** **and $\varphi(v)$ always in the same state**
- **Proof: by induction**
  - before round 1: map φ preserves local states
  - during round 1: map φ preserves messages
  - after round 1: map φ preserves local states

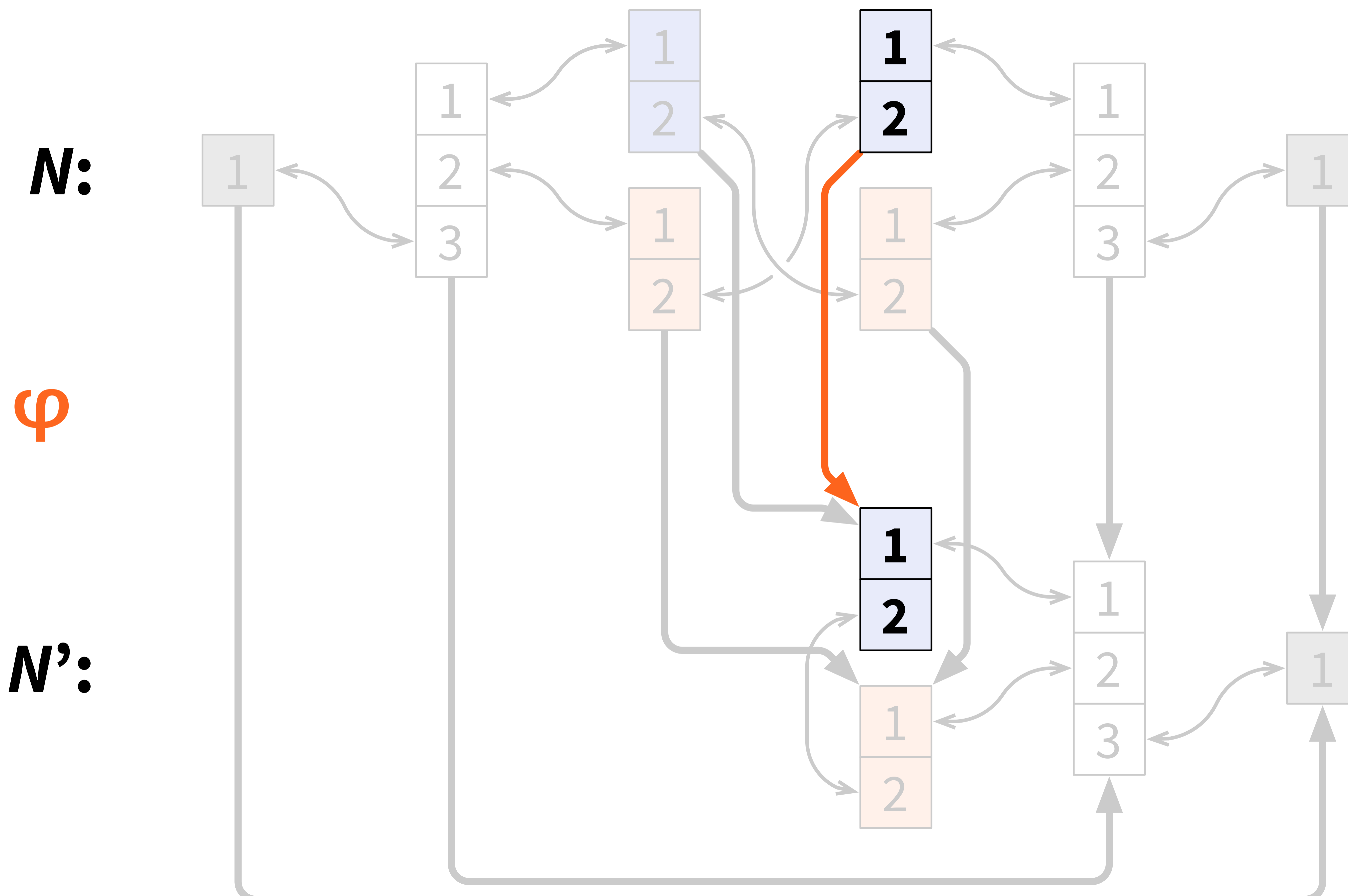
Covering map

- **Theorem:** **v and $\varphi(v)$ always in the same state**
- **Proof: by induction**
  - before round 2: map φ preserves local states
  - during round 2: map φ preserves messages
  - after round 2: map φ preserves local states

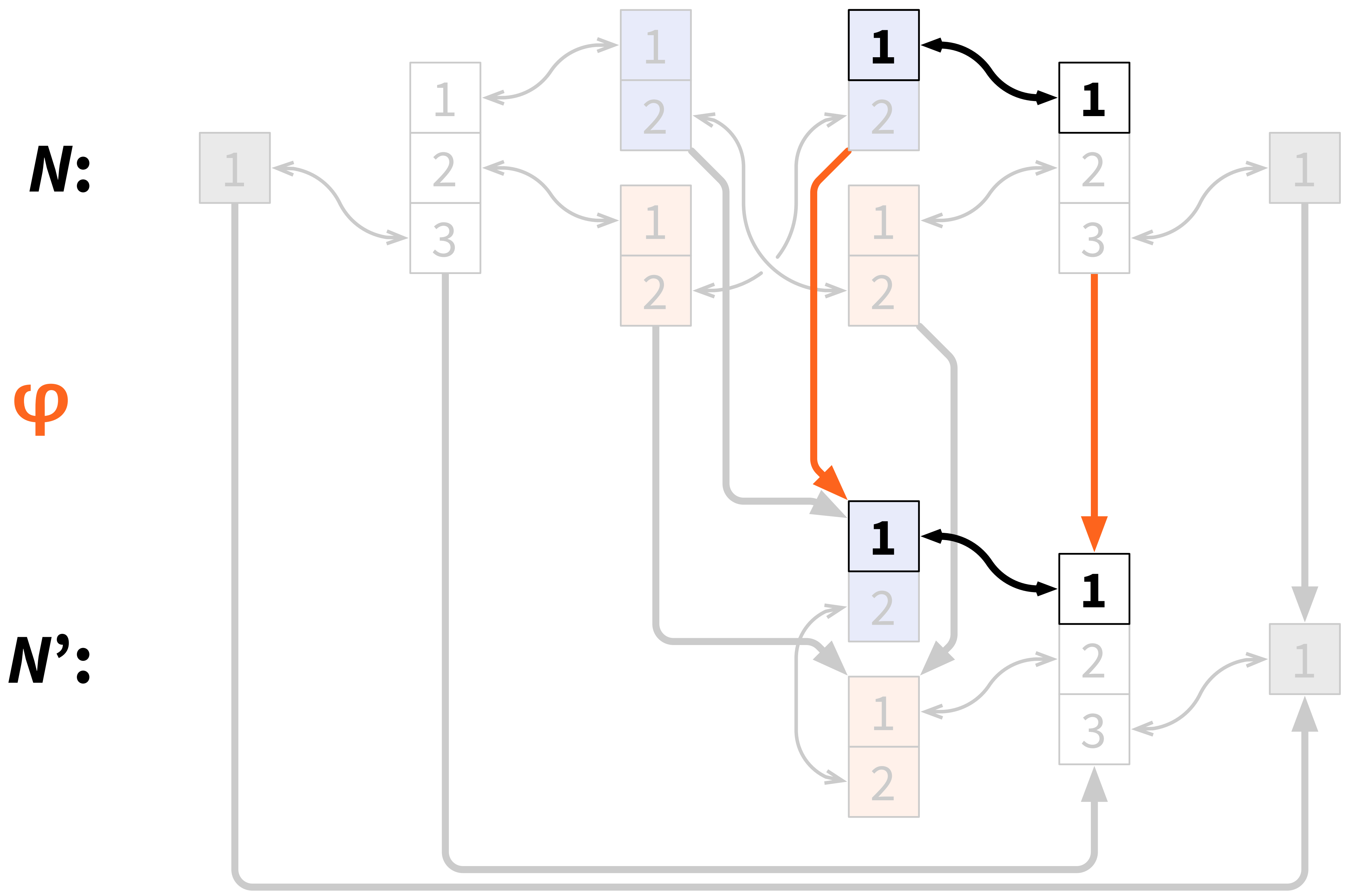
Covering map

- **Theorem:** ***v*** **and $\varphi(v)$ always in the same state**
- **Proof: by induction**
  - before round t: map φ preserves local states
  - during round t: map φ preserves messages
  - after round t: map φ preserves local states

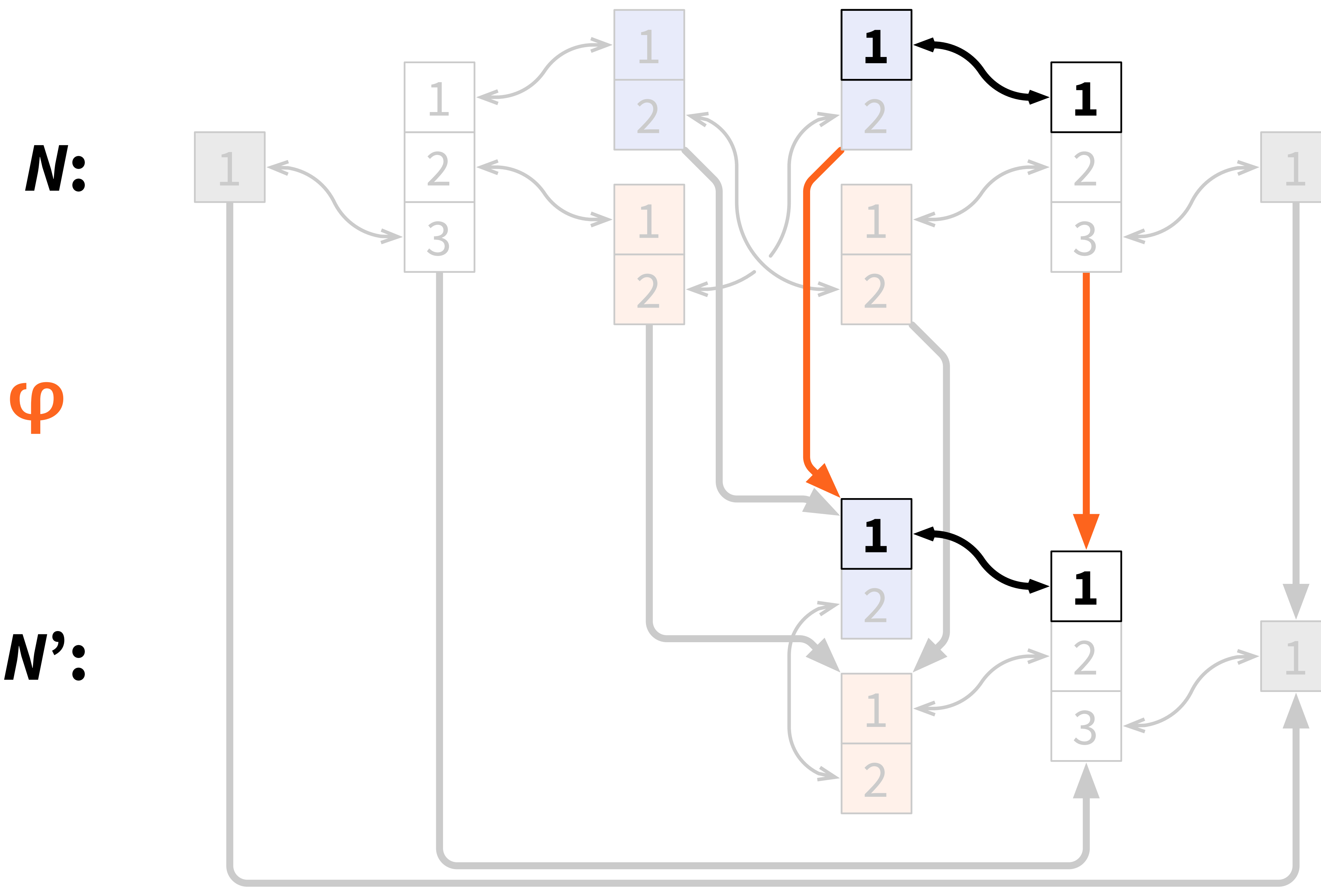
Before round *t*: local states agree



During round *t*: outgoing messages agree



During round *t*: incoming messages agree

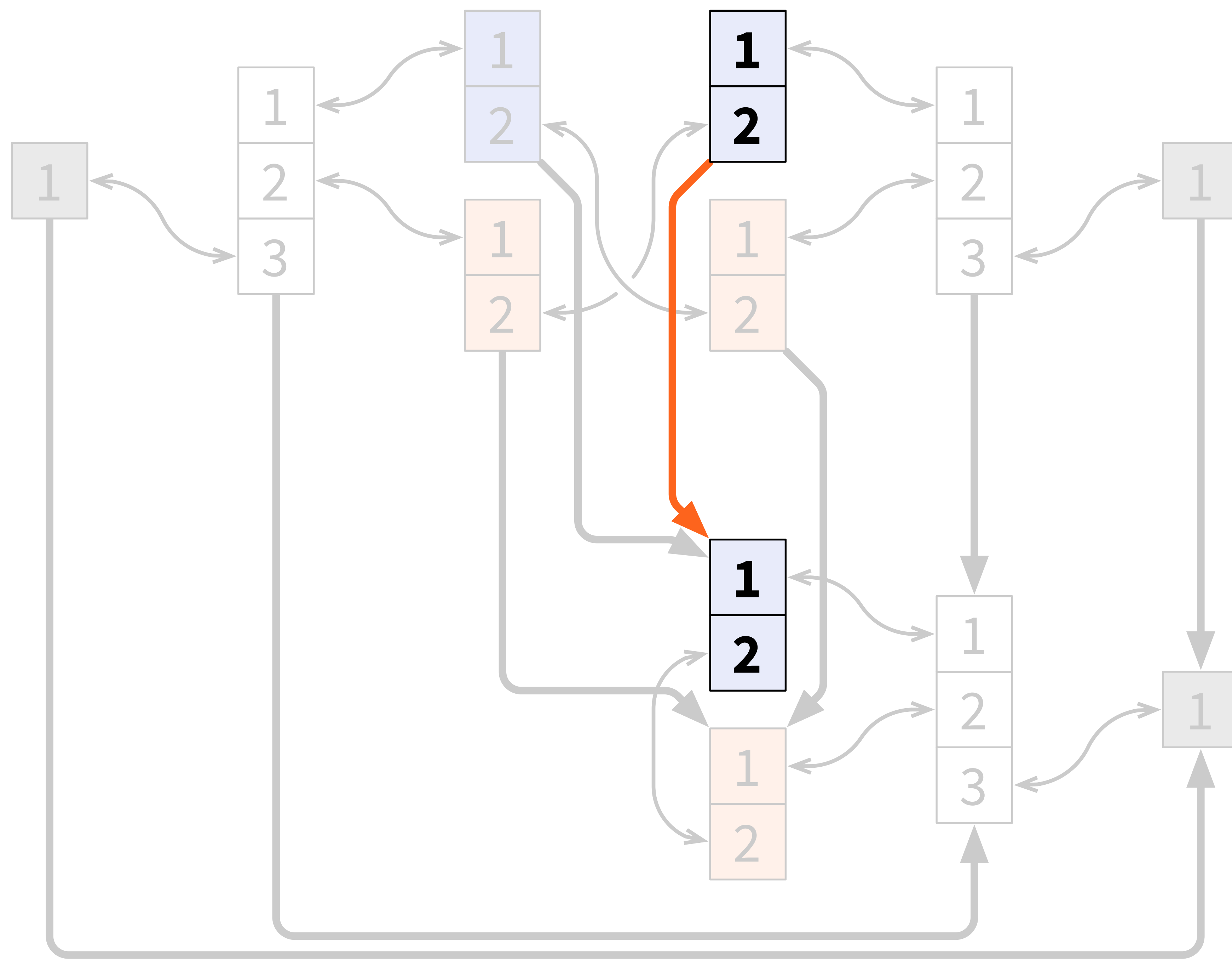


After round *t*: local states agree

***N*:**

φ

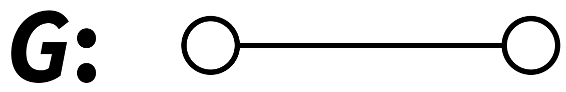
***N'*:**



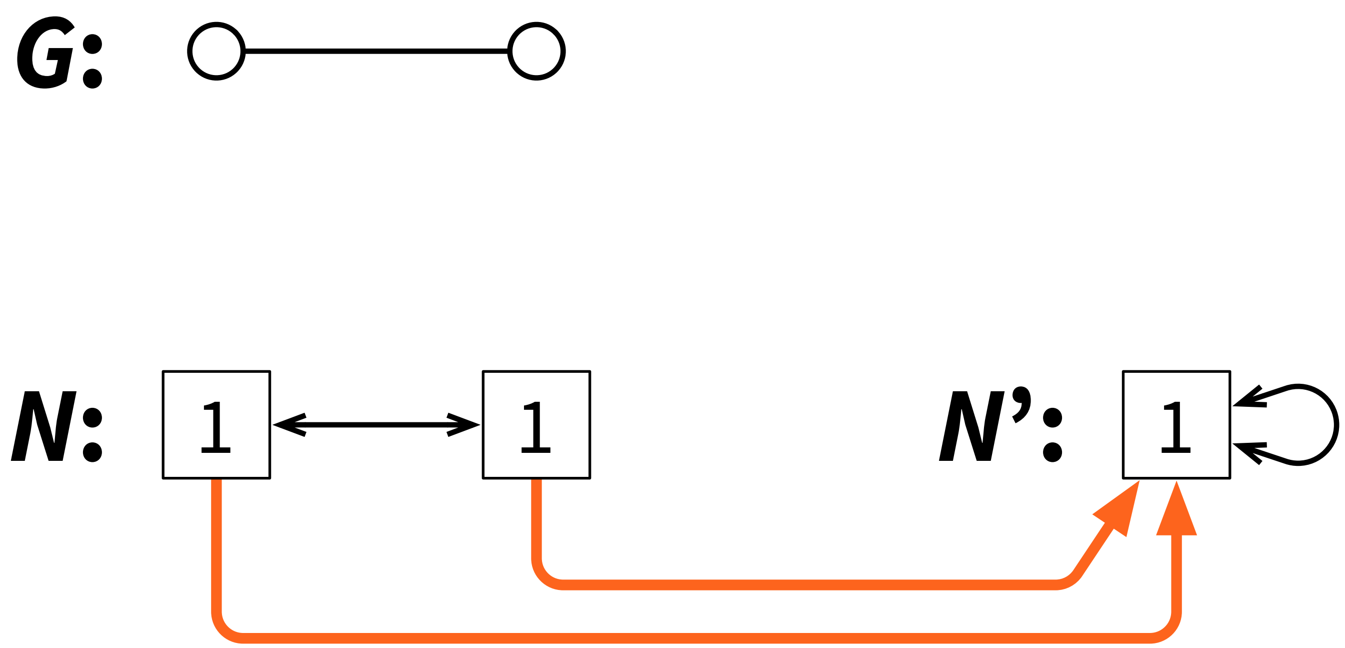
Covering map

- φ covering map from *N* to *N'*,
  A deterministic PN-algorithm
- Run *A* on *N* and *N'*
- Theorem: *v* and φ(*v*) always in the same state
- Corollary: *v* and φ(*v*) have the same output

Application:
Path graphs



Application:
Path graphs

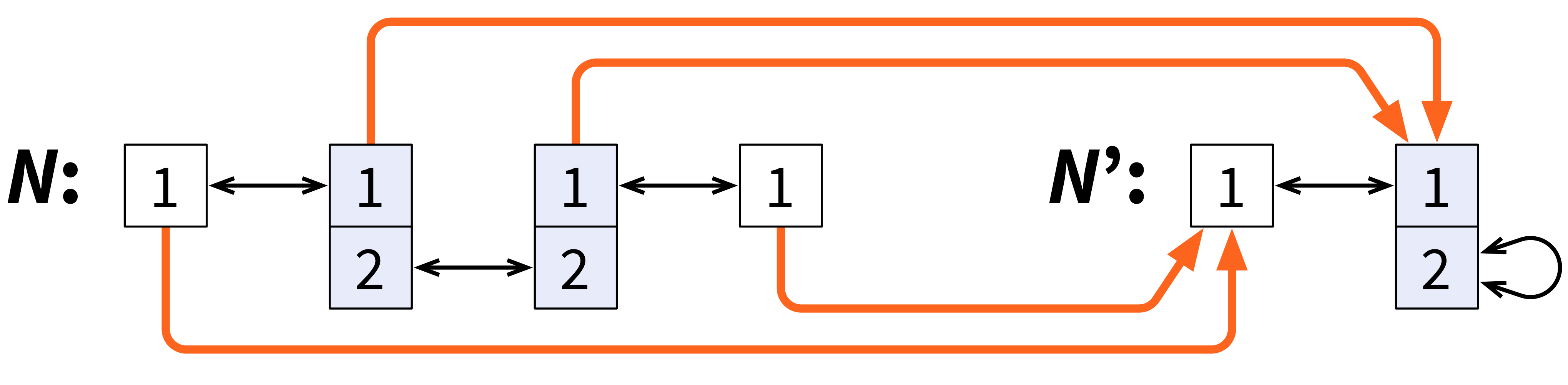


Application:

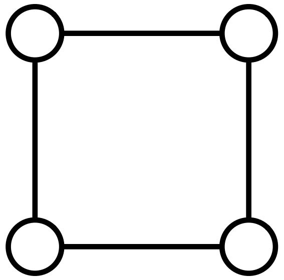
Path graphs

G:

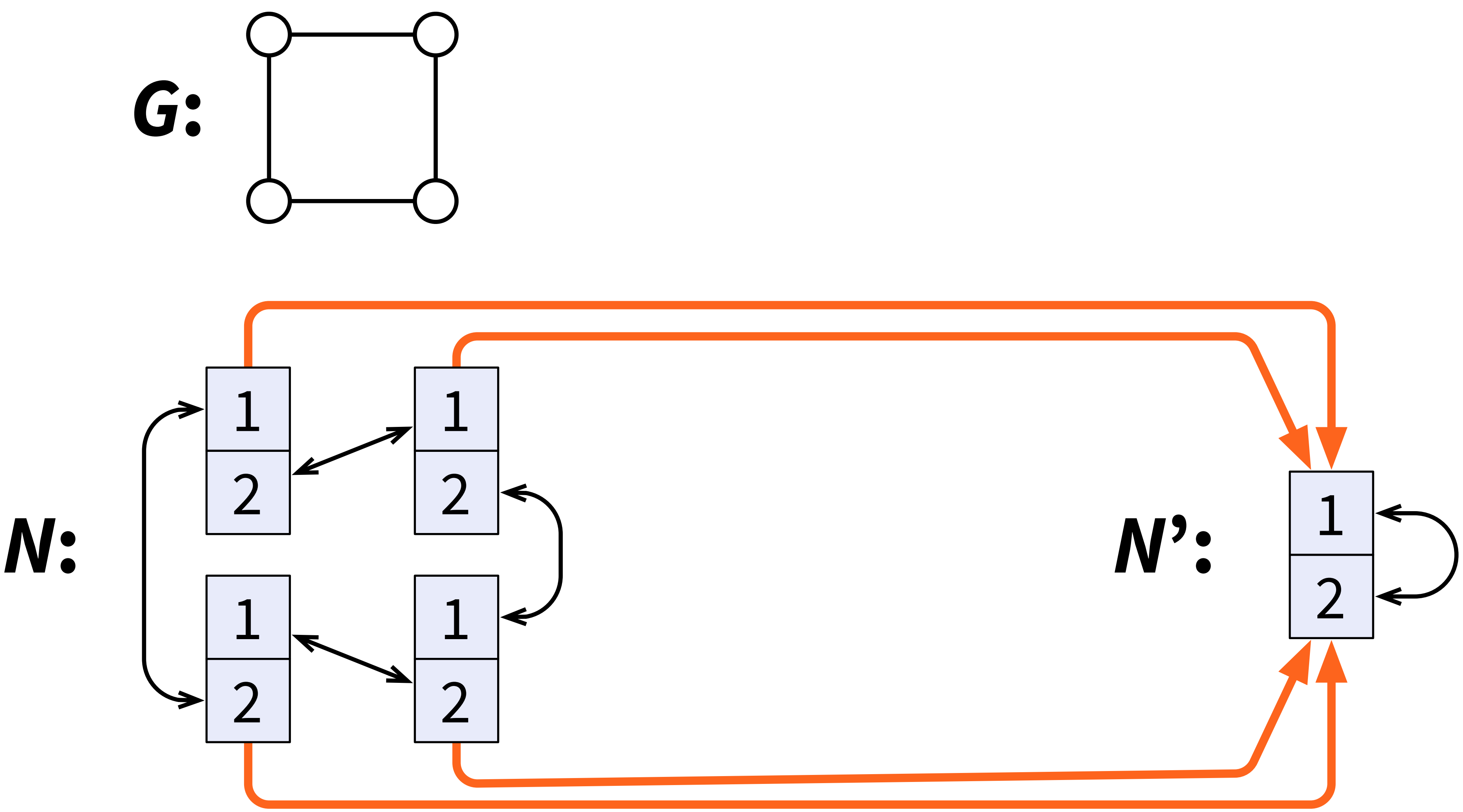
Application:
Path graphs



Application:
Cycle graphs

***G*:** 

Application:
Cycle graphs

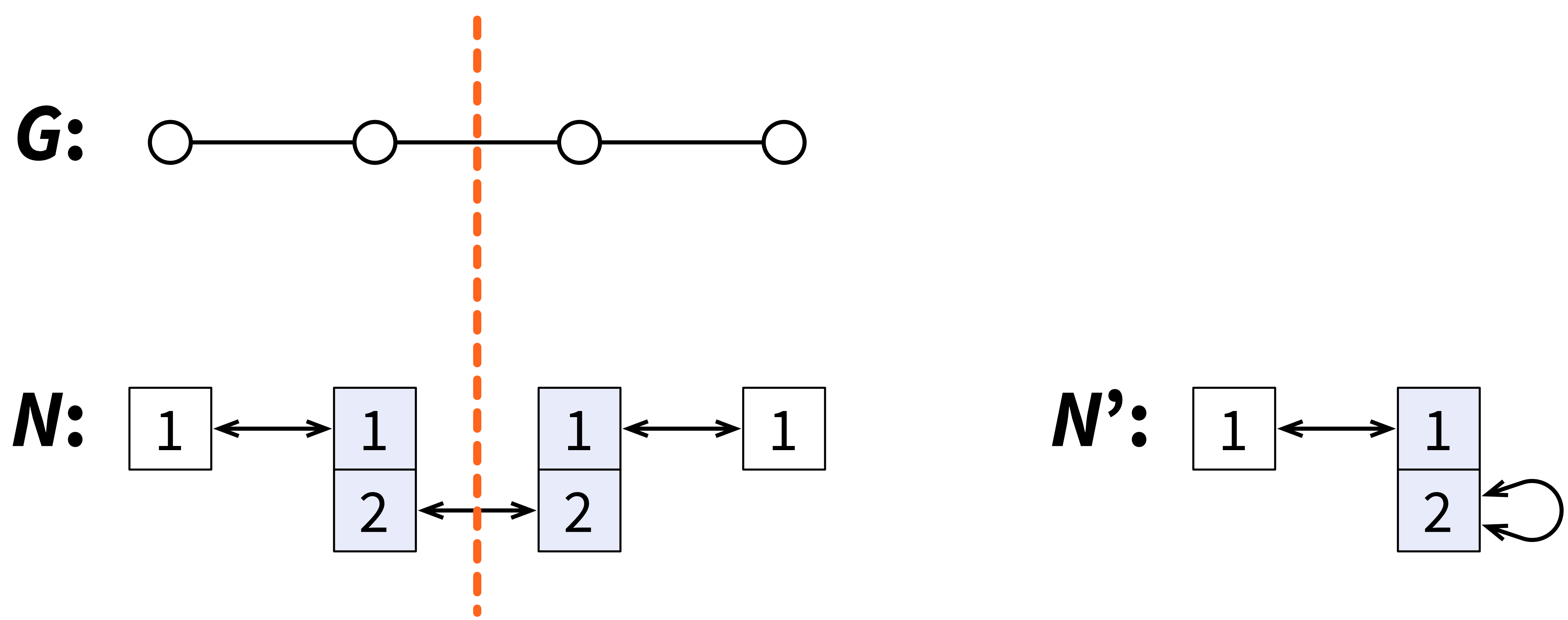


Application:
Cycle graphs

- **Cannot break symmetry in cycles**
- **Deterministic PN algorithms cannot find:**
  - vertex colouring, edge colouring
  - maximal independent set, maximal matching
  - 1.99-approximation of minimum vertex cover

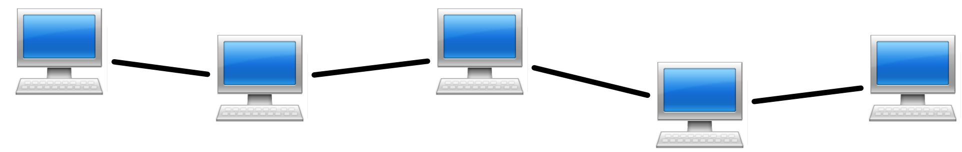
  …

Covering maps and symmetry



Summary

- Covering map: preserves inputs, degrees, connections, port numbers
- Fools any deterministic PN-algorithm
- Can be used to prove that many problems cannot be solved at all in the PN model

- **Weeks 1–2: informal introduction**
  - network = path 
- **Week 3: graph theory**
- **Weeks 4–7: models of computing**
  - what can be computed (efficiently)?
- **Weeks 8–11: lower bounds**
  - what cannot be computed (efficiently)?
- **Week 12: recap**