Parameterized Analysis of Immediate Observation Petri Nets

Chana Weil-Kennedy

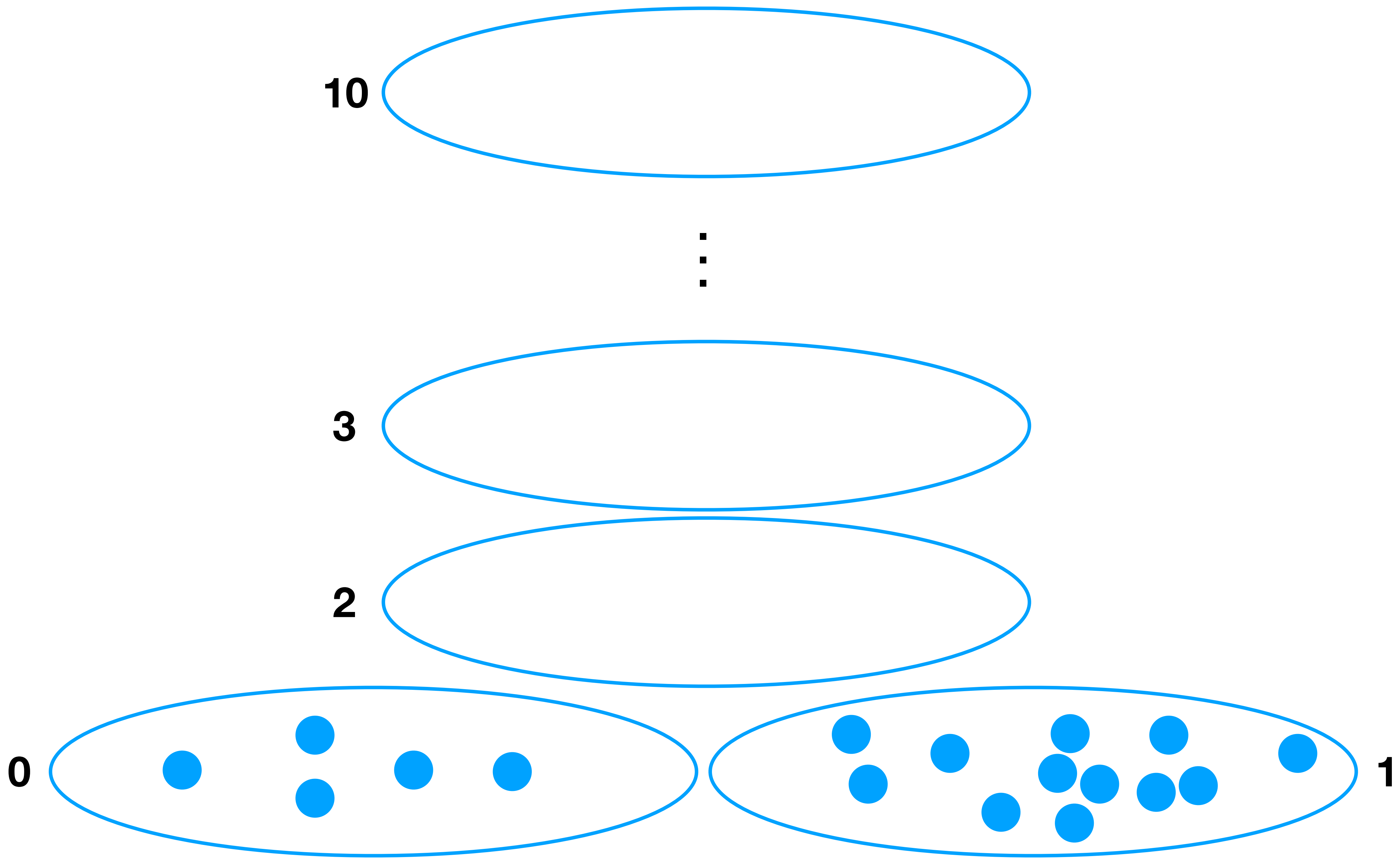
joint work with Javier Esparza, Mikhail Raskin

The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 787367

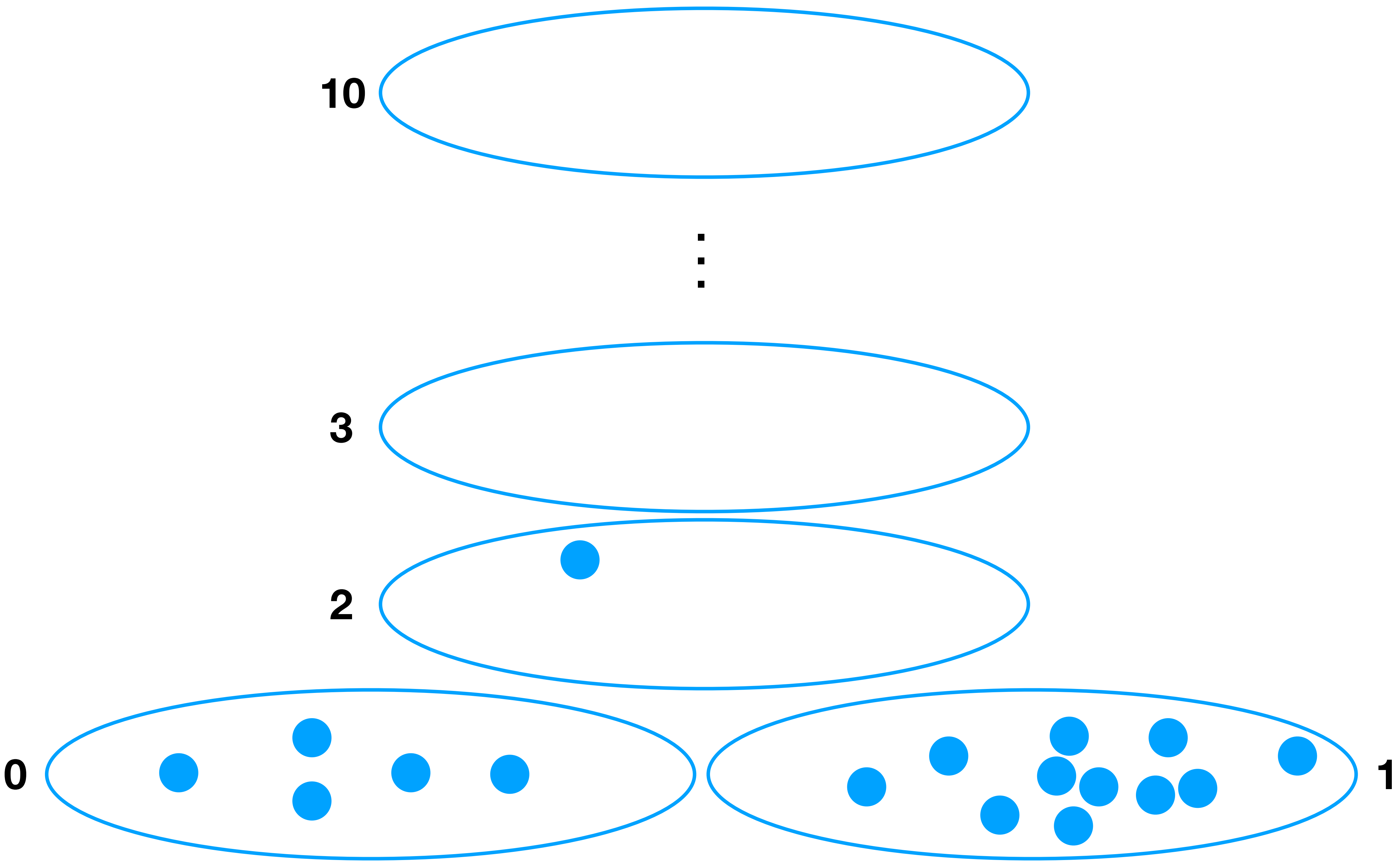
European Research Council

Established by the European Commission

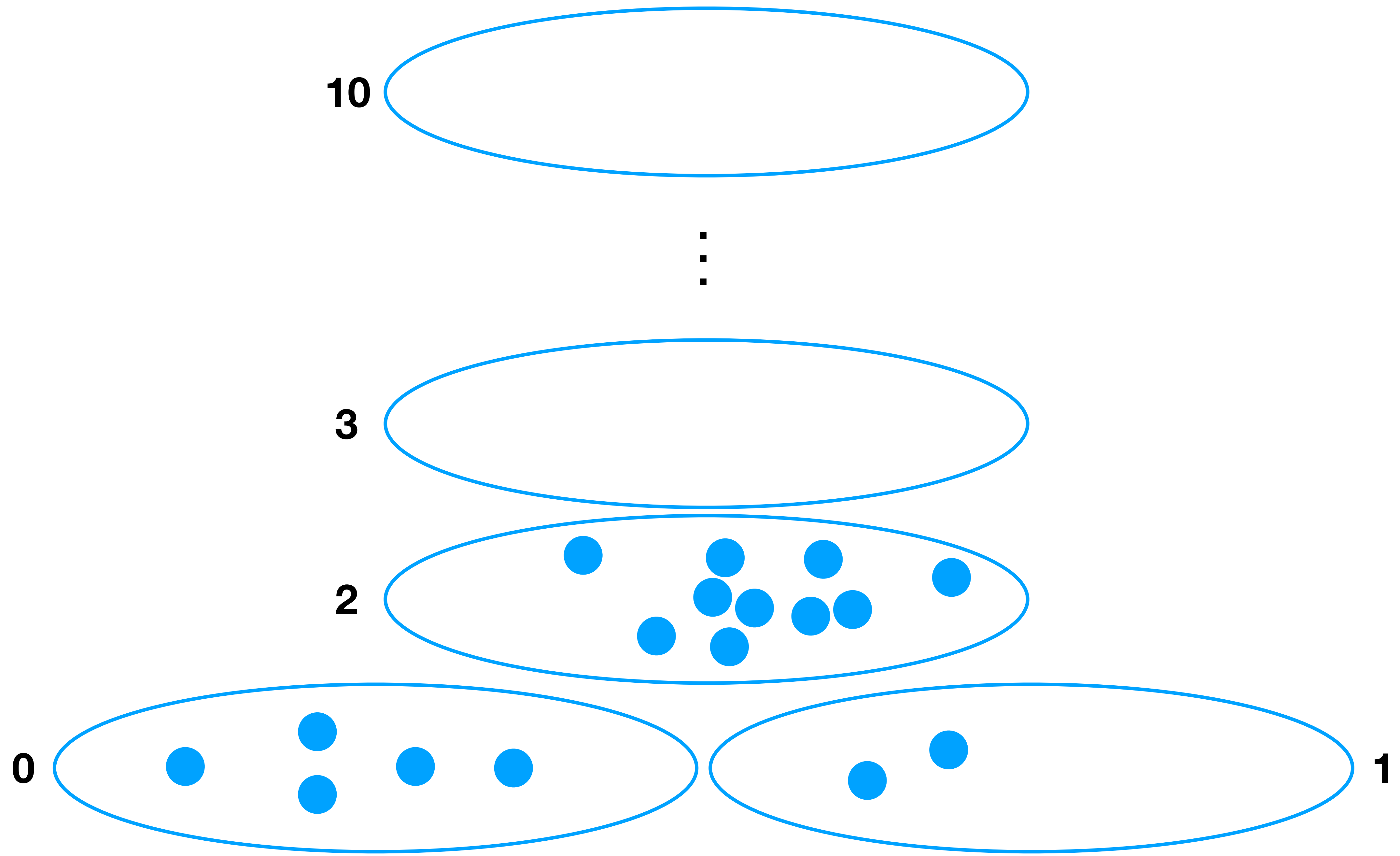
Counting Researchers Protocol



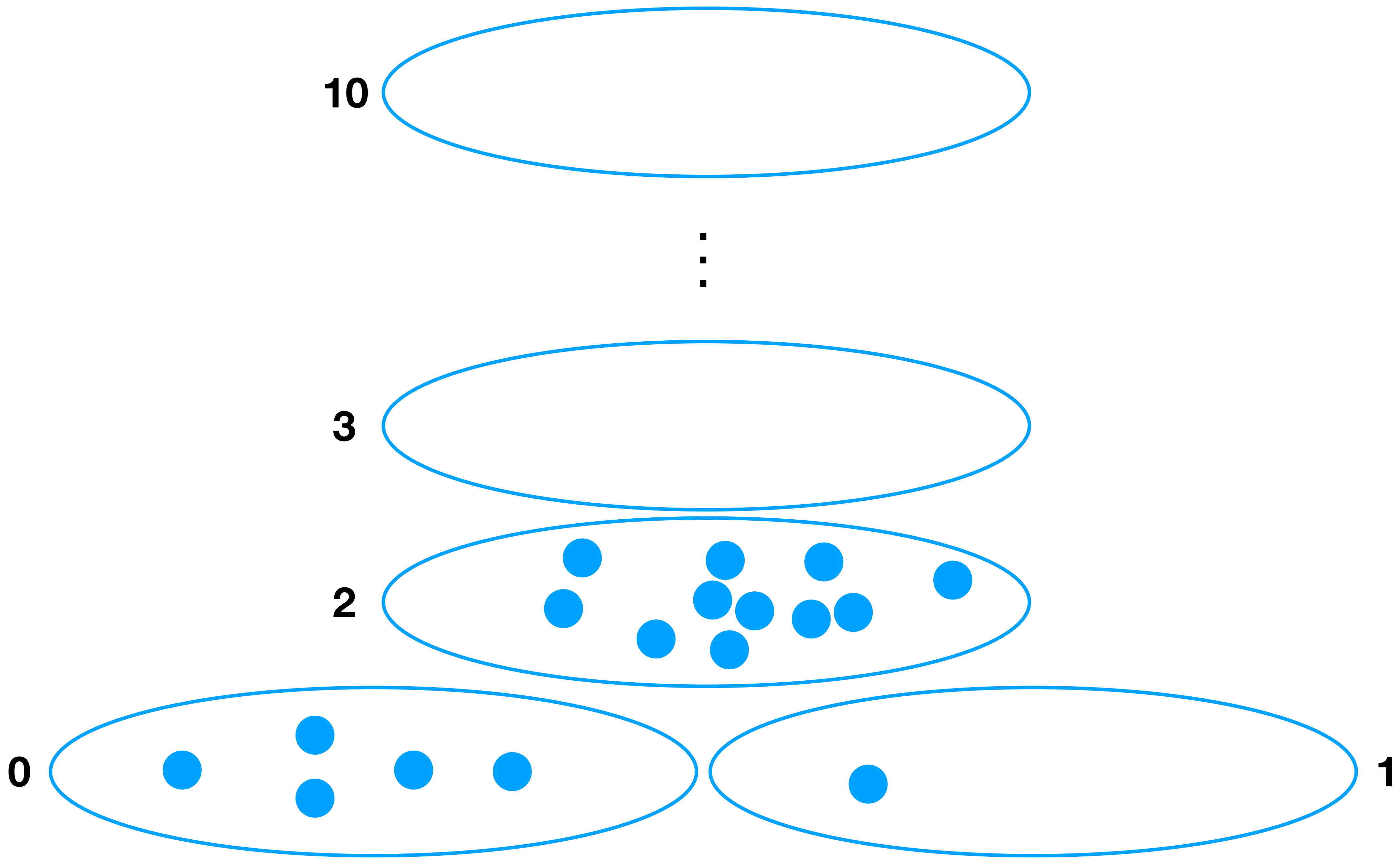
Counting Researchers Protocol



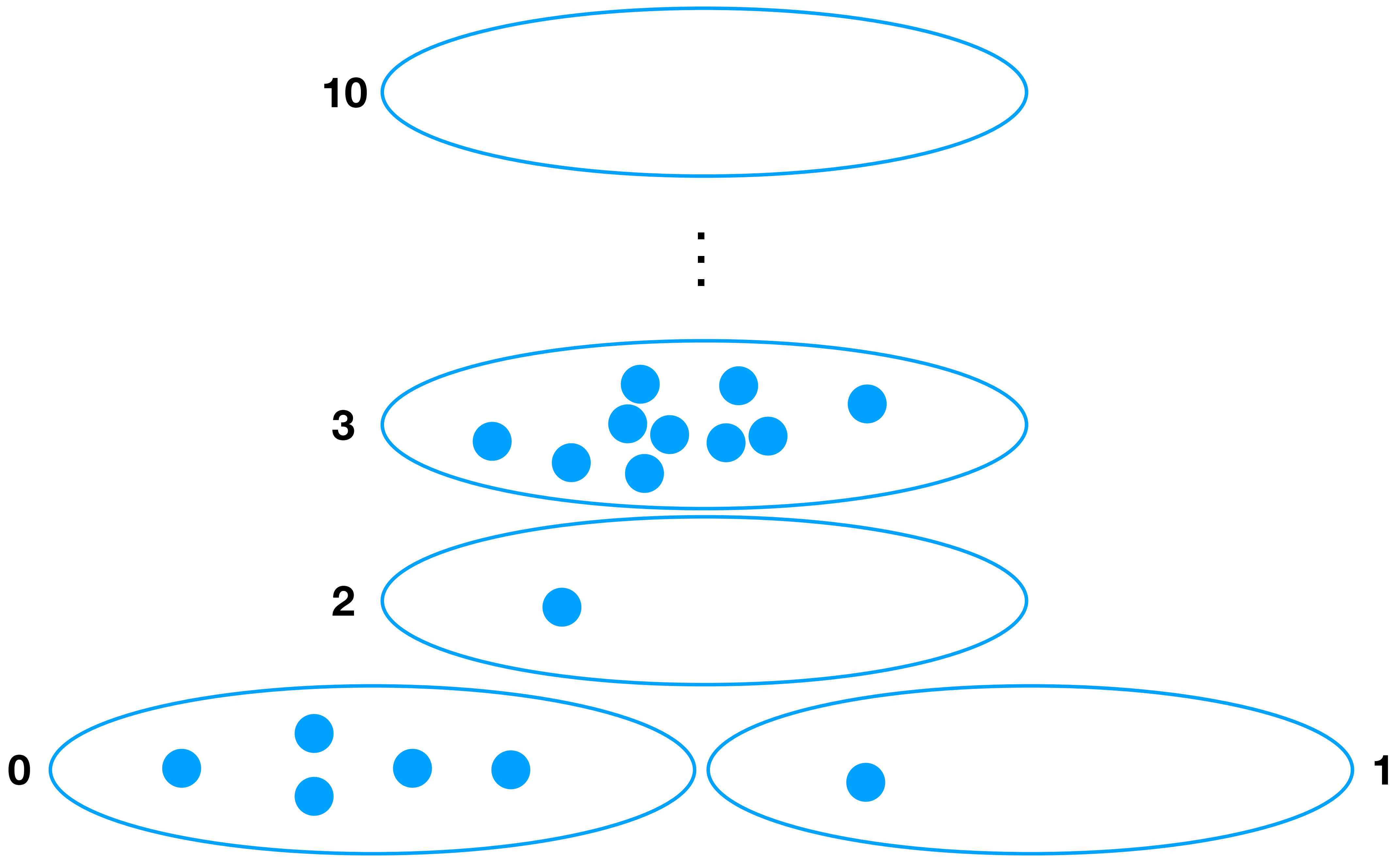
Counting Researchers Protocol



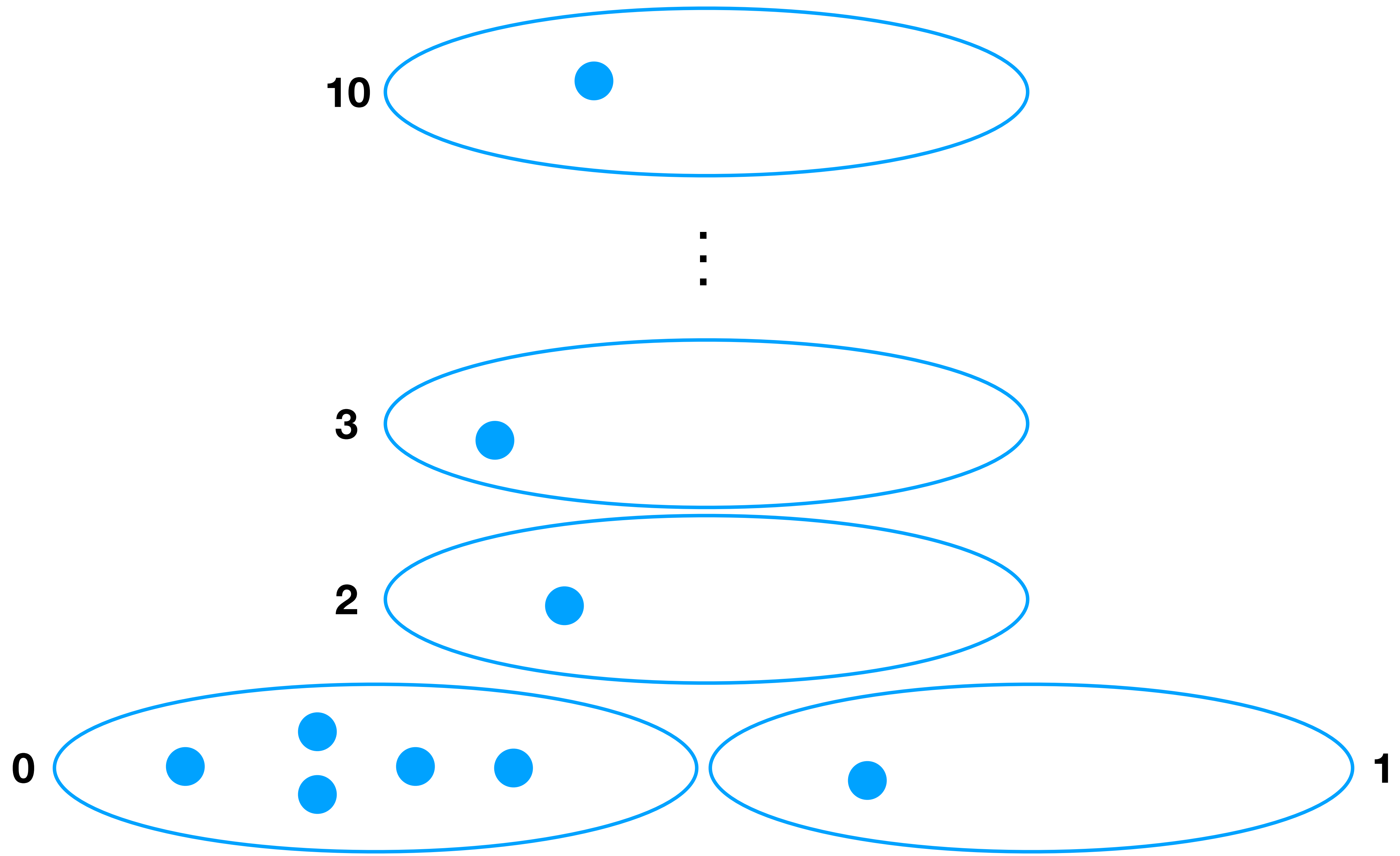
Counting Researchers Protocol



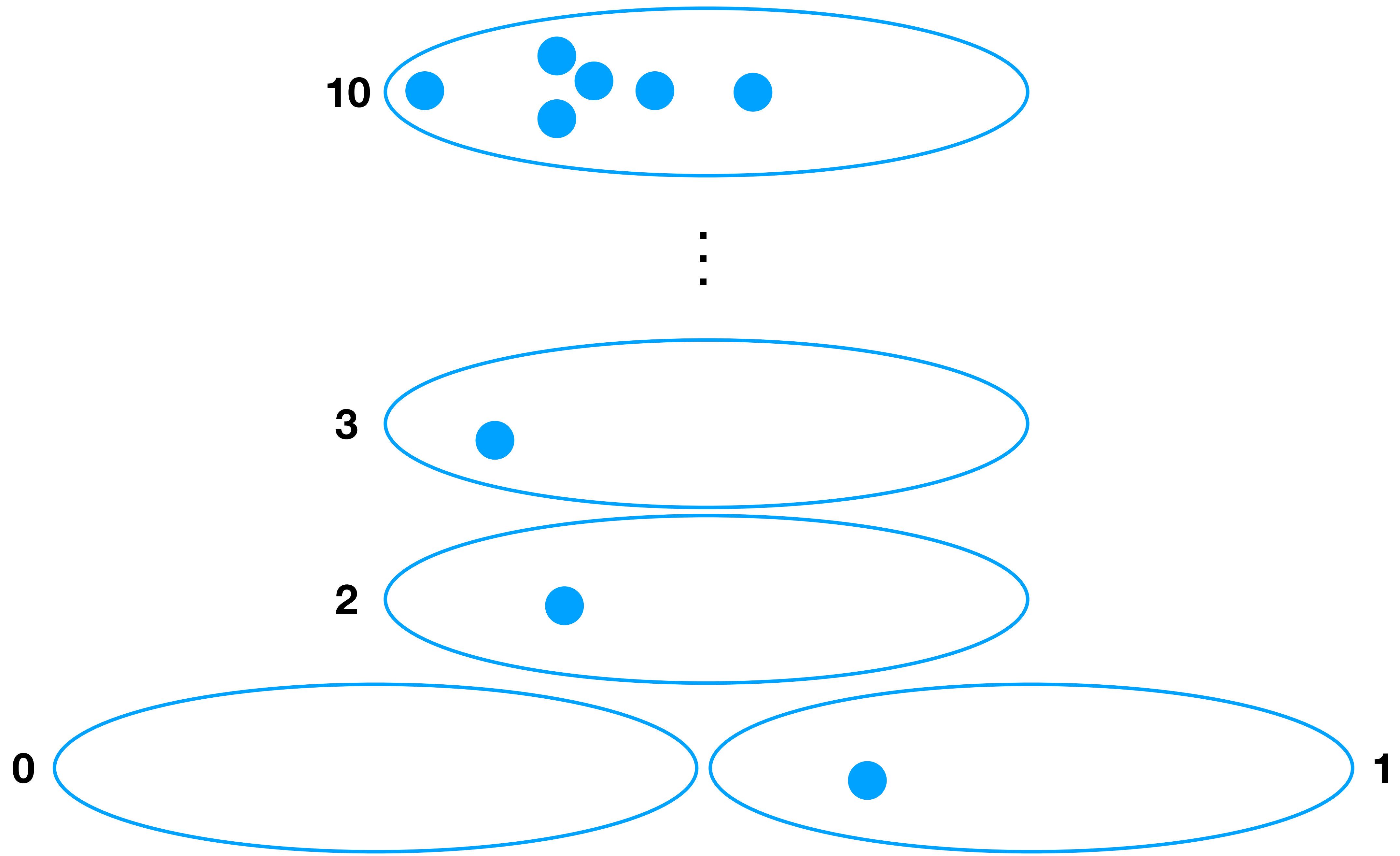
Counting Researchers Protocol



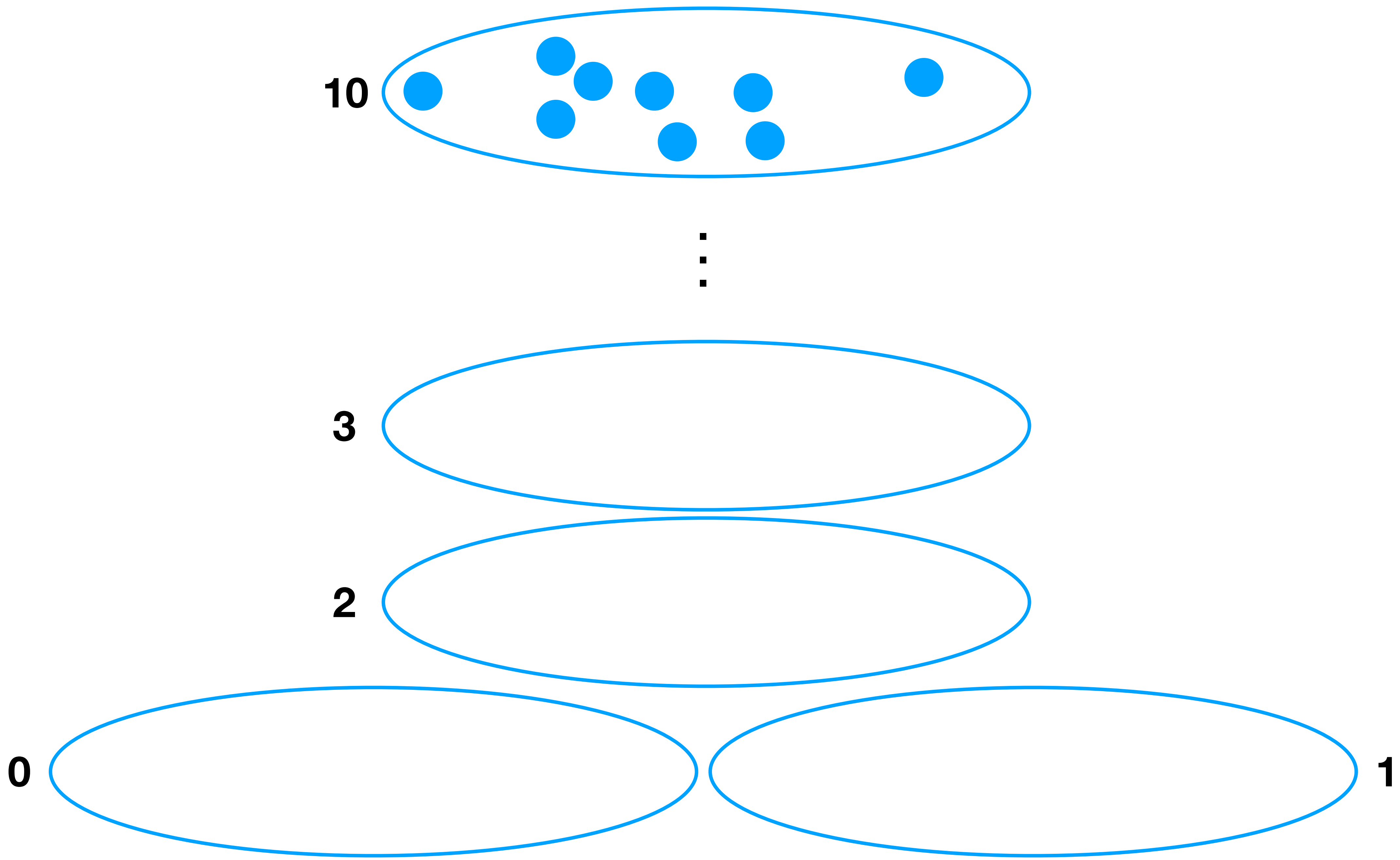
Counting Researchers Protocol



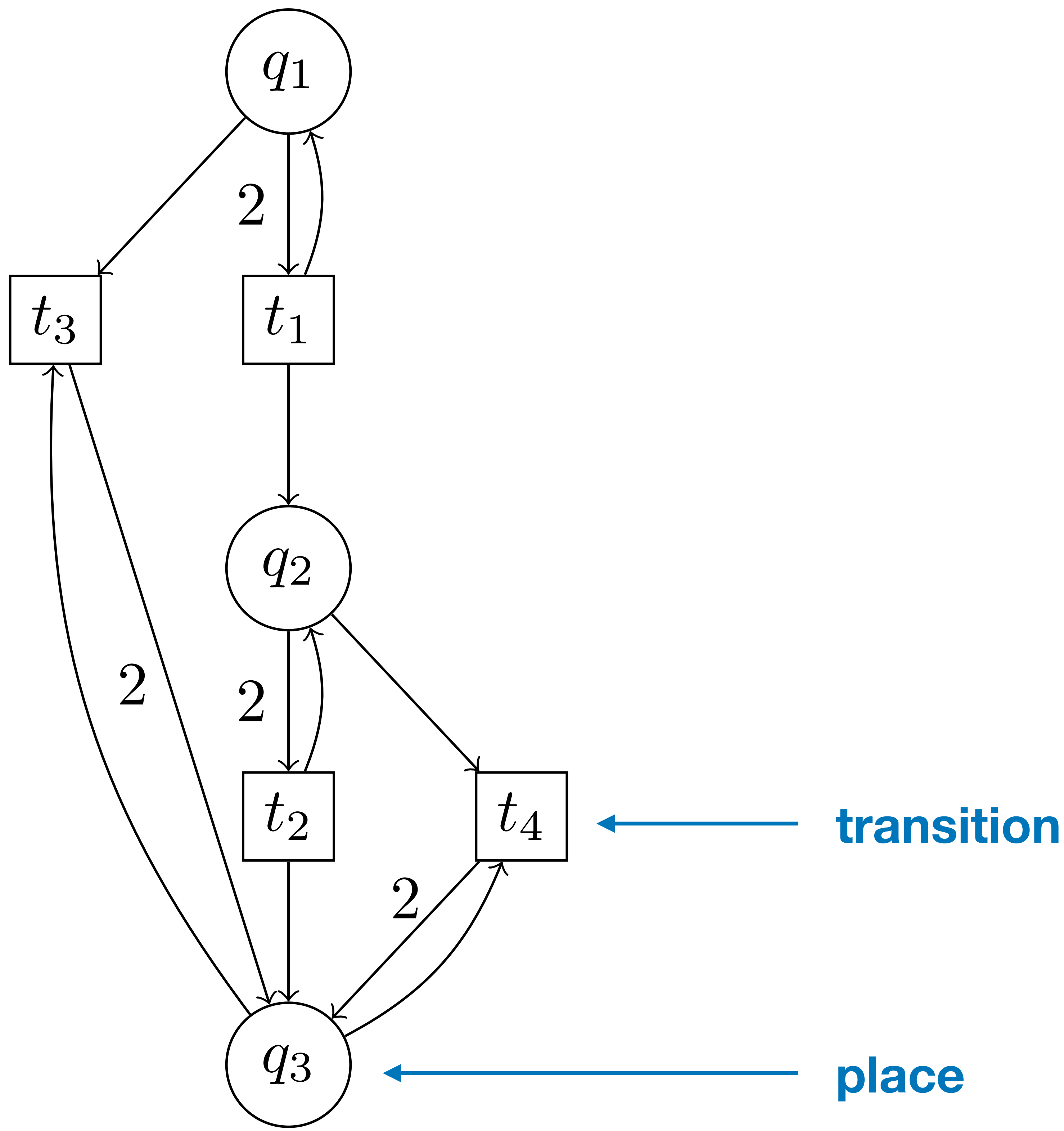
Counting Researchers Protocol



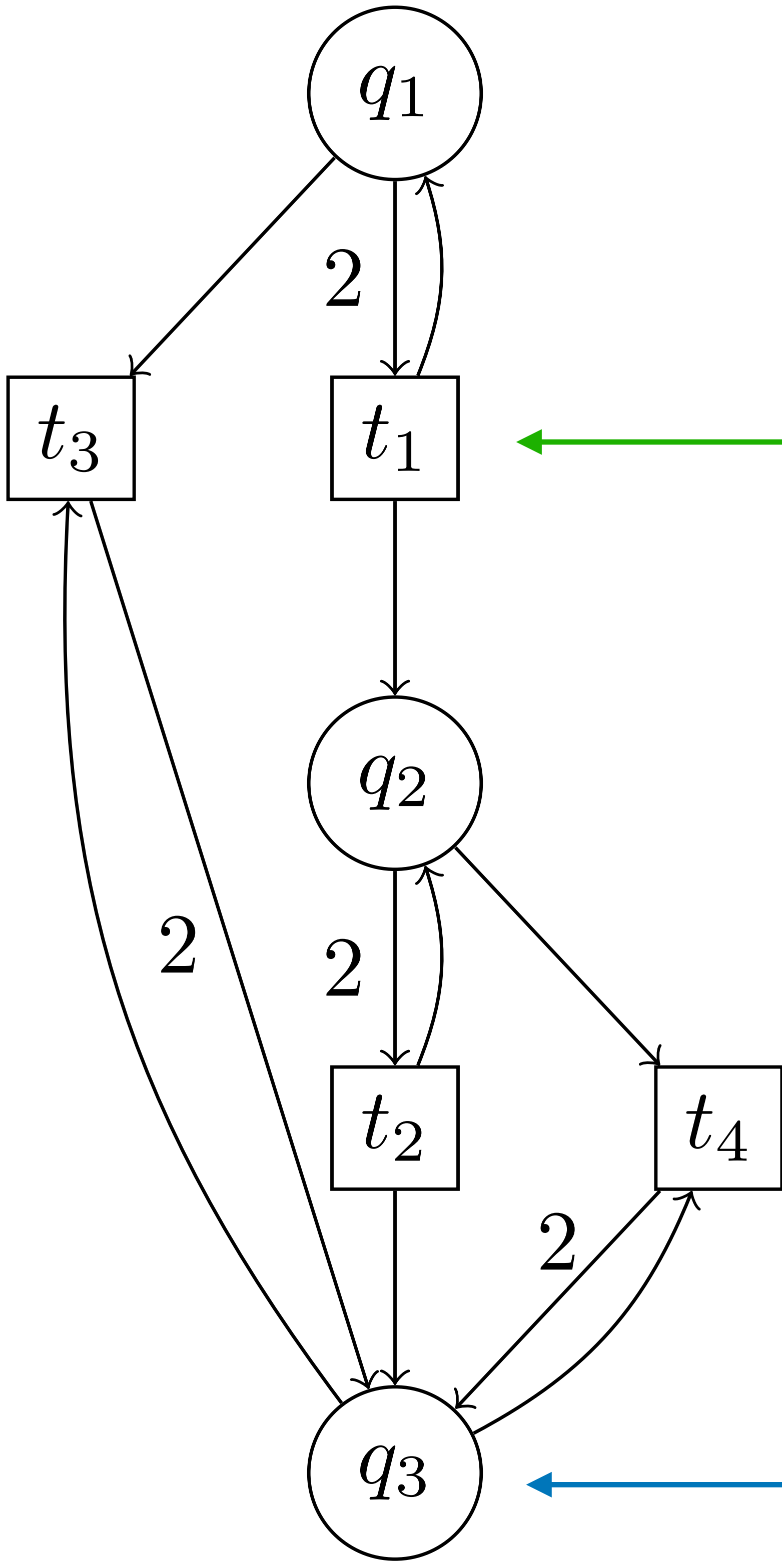
Counting Researchers Protocol



Immediate Observation Nets



Immediate Observation Nets



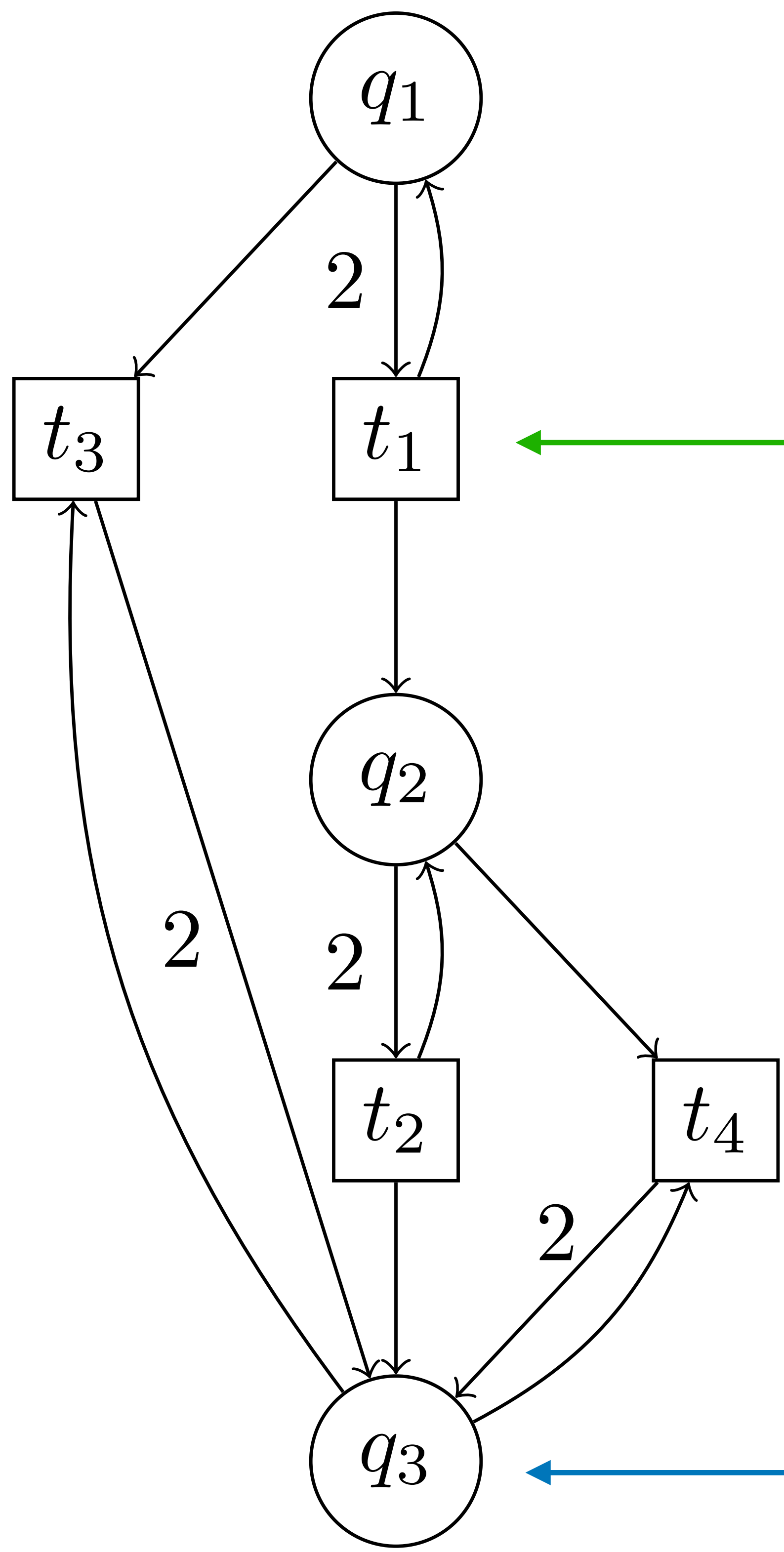
example:

$(q_1, q_1) \mapsto (q_1, q_2)$

transition

place

Immediate Observation Nets



example:

$(q_1, q_1) \mapsto (q_1, q_2)$

A Petri net is an **IO net** if all its transitions are of the form

$(p_s\ , p_o) \mapsto (p_d\ , p_o)$

transition

place

Immediate Observation Nets

- we call **marking** a configuration of the IO net
- a **run** is a sequence of valid transitions
- each **immediate observation** population protocol has an underlying Petri net - this net is an IO net

Parameterized Problems

We are interested in **parameterized** problems in the number of agents (population protocols) or molecules (enzymatic chemical networks)

Parameterized Problems

We are interested in **parameterized** problems in the number of agents (population protocols) or molecules (enzymatic chemical networks)

does an infinite set of Petri nets differing only in their initial markings satisfy a given property ?

Parameterized Problems

We are interested in **parameterized** problems in the number of agents (population protocols) or molecules (enzymatic chemical networks)

does an infinite set of Petri nets differing only in their initial markings satisfy a given property ?

We concentrate on infinite sets defined by counting constraints.

Results

- Cube-reachability, cube-coverability and cube-liveness in IO nets are **PSPACE-complete** … even though they are already PSPACE-hard for singleton sets of markings

Results

- Cube-reachability, cube-coverability and cube-liveness in IO nets are **PSPACE-complete** … even though they are already PSPACE-hard for singleton sets of markings

This is in *strong contrast* with more general Petri net classes like 1-conservative nets: these problems are PSPACE for singletons, but as hard as for general Petri nets for cubes.

Results

- Cube-reachability, cube-coverability and cube-liveness in IO nets are **PSPACE-complete** … even though they are already PSPACE-hard for singleton sets of markings

This is in *strong contrast* with more general Petri net classes like 1-conservative nets: these problems are PSPACE for singletons, but as hard as for general Petri nets for cubes.

- Application: the correctness problem for IO population protocols is **PSPACE-complete**

How

We use

- a fundamental technique: pruning
- a useful representation: counting constraints

The Pruning Theorem

For any run

$$M \xrightarrow{*} M' \geq M''$$

there exists a run

$$S \xrightarrow{*} S' \geq M''$$

The Pruning Theorem

For any run

$$M \xrightarrow{*} M' \geq M''$$

$$\geq \qquad\qquad \geq$$

there exists a run

$$S \xrightarrow{*} S' \geq M''$$

The Pruning Theorem

For any run

$$M \xrightarrow{*} M' \geq M''$$

$$\geq \qquad\qquad \geq$$

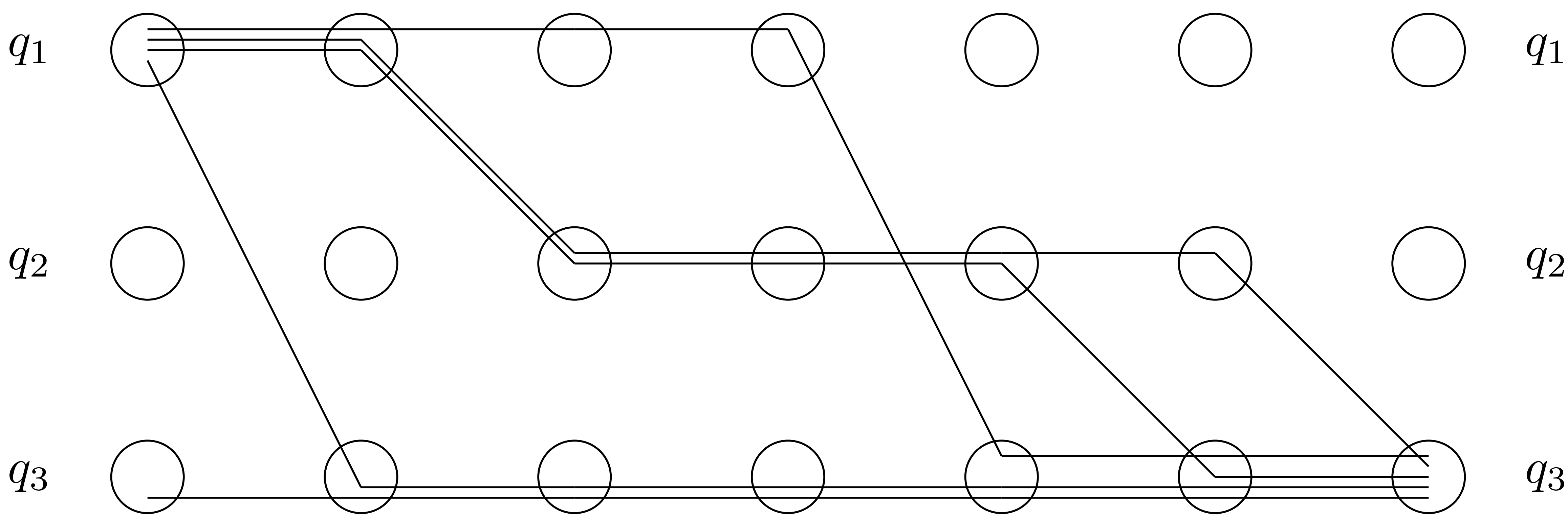
there exists a run

$$S \xrightarrow{*} S' \geq M''$$

such that

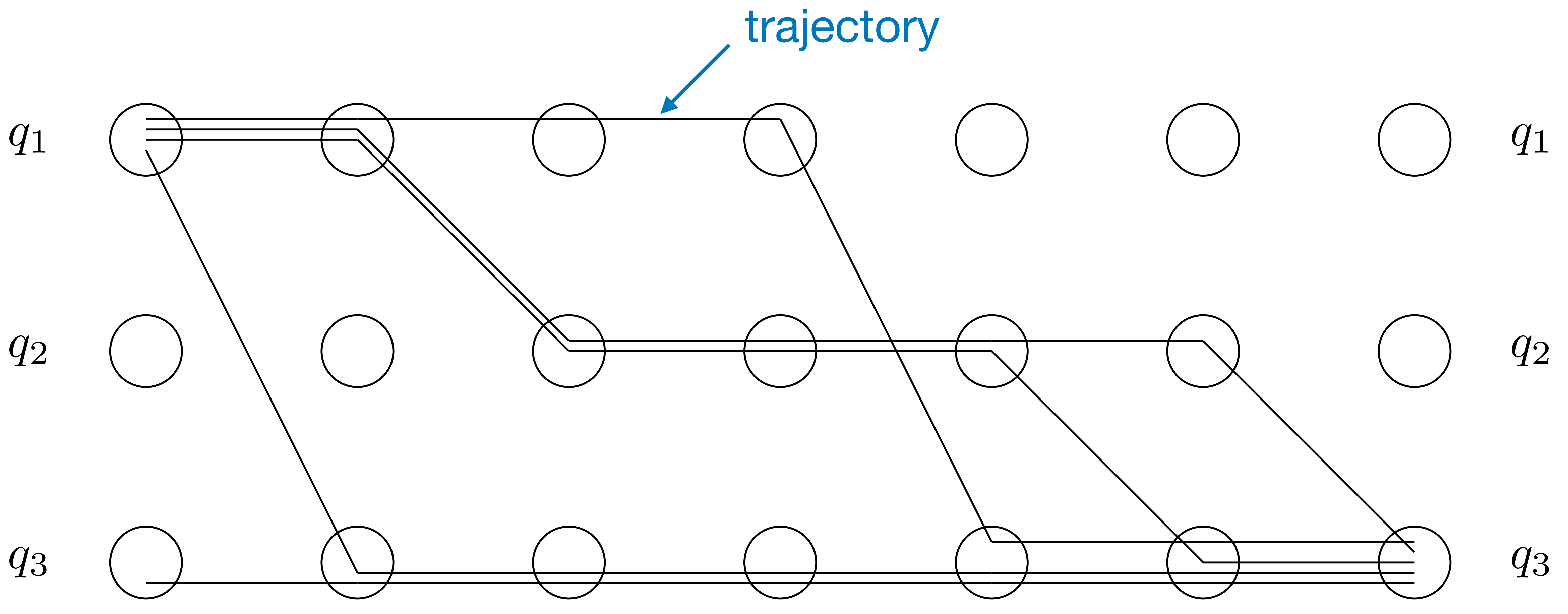
$$|S| \leq |M''| + n^3$$

The Pruning Theorem



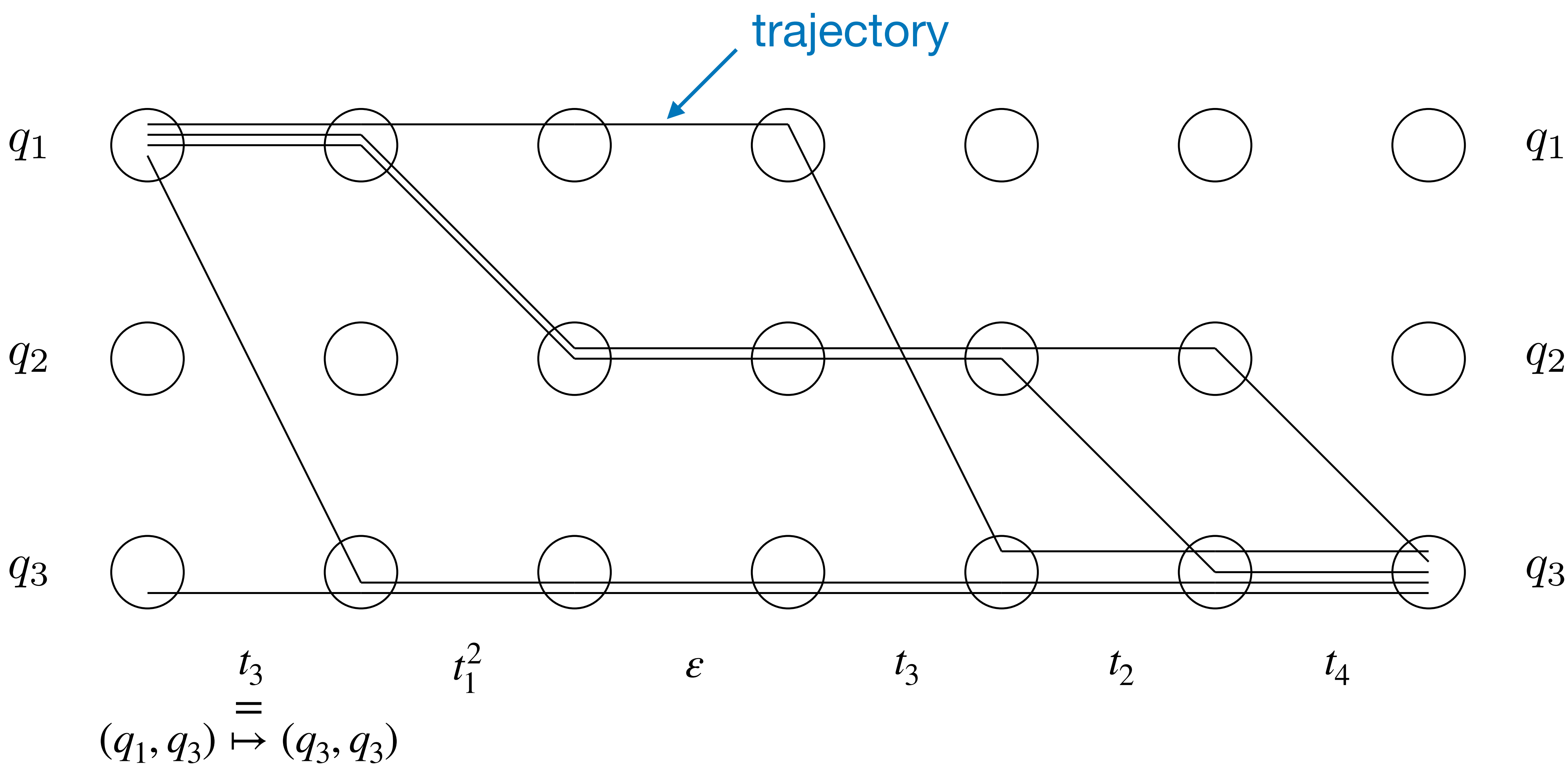
History = the trajectories of the different tokens in a run

The Pruning Theorem



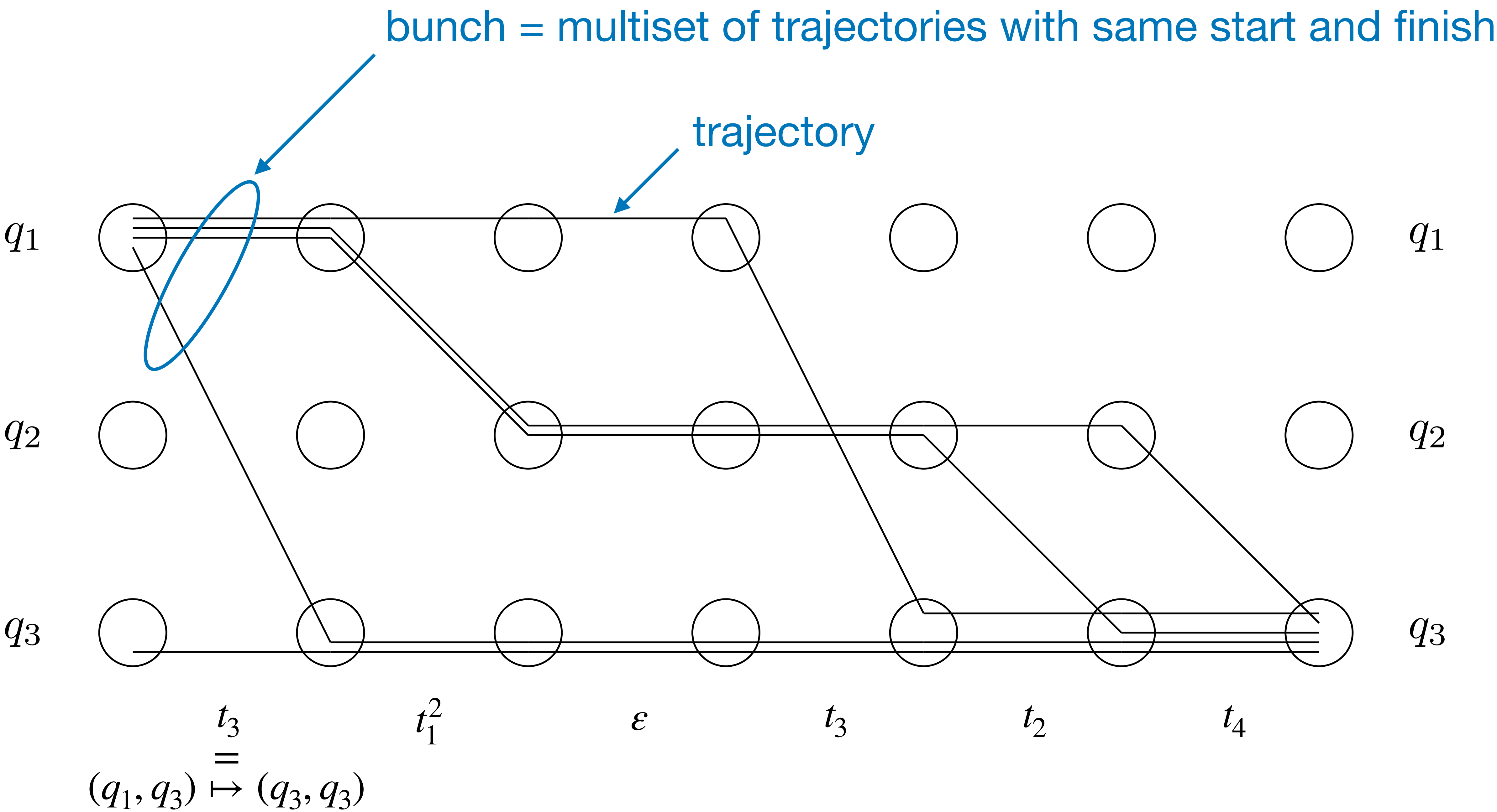
History = the trajectories of the different tokens in a run

The Pruning Theorem



History = the trajectories of the different tokens in a run

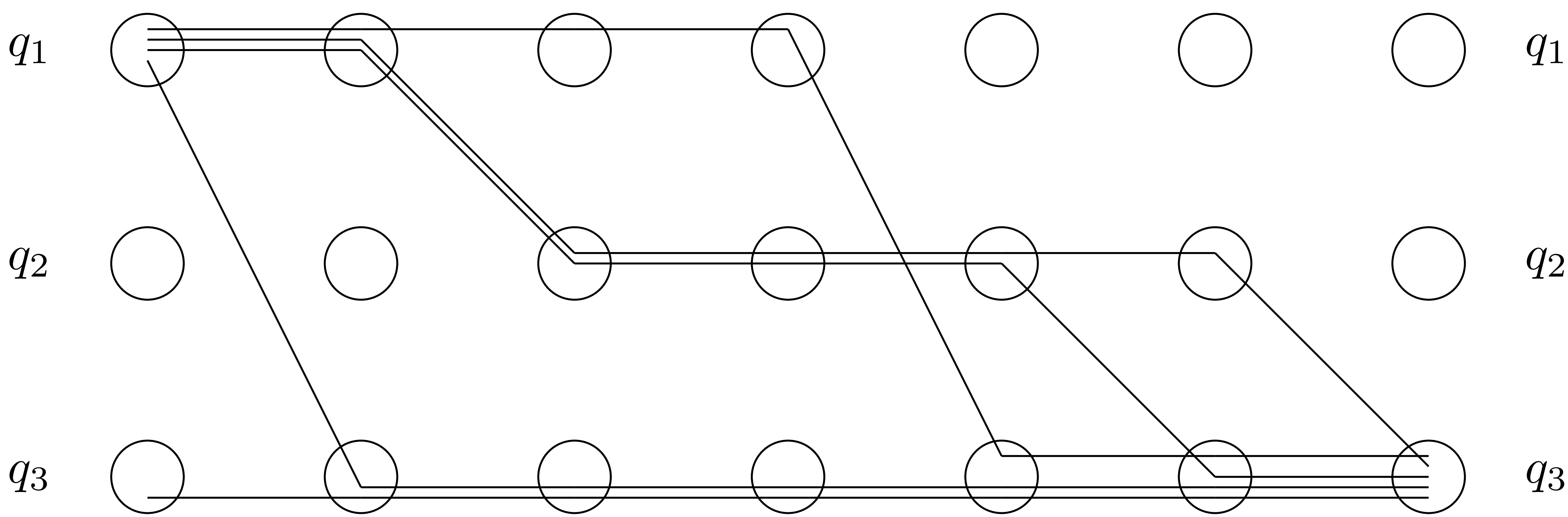
The Pruning Theorem



History = the trajectories of the different tokens in a run

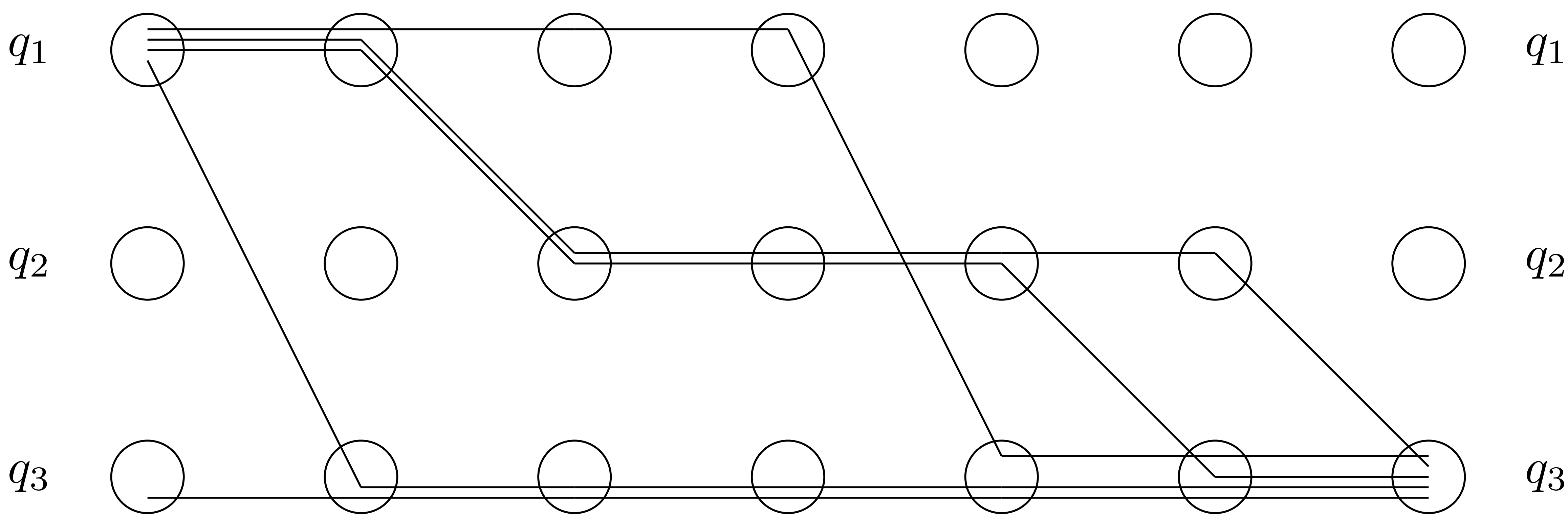
The Pruning Theorem

Transitions are enabled as soon as there is one token in the observed place



The Pruning Theorem

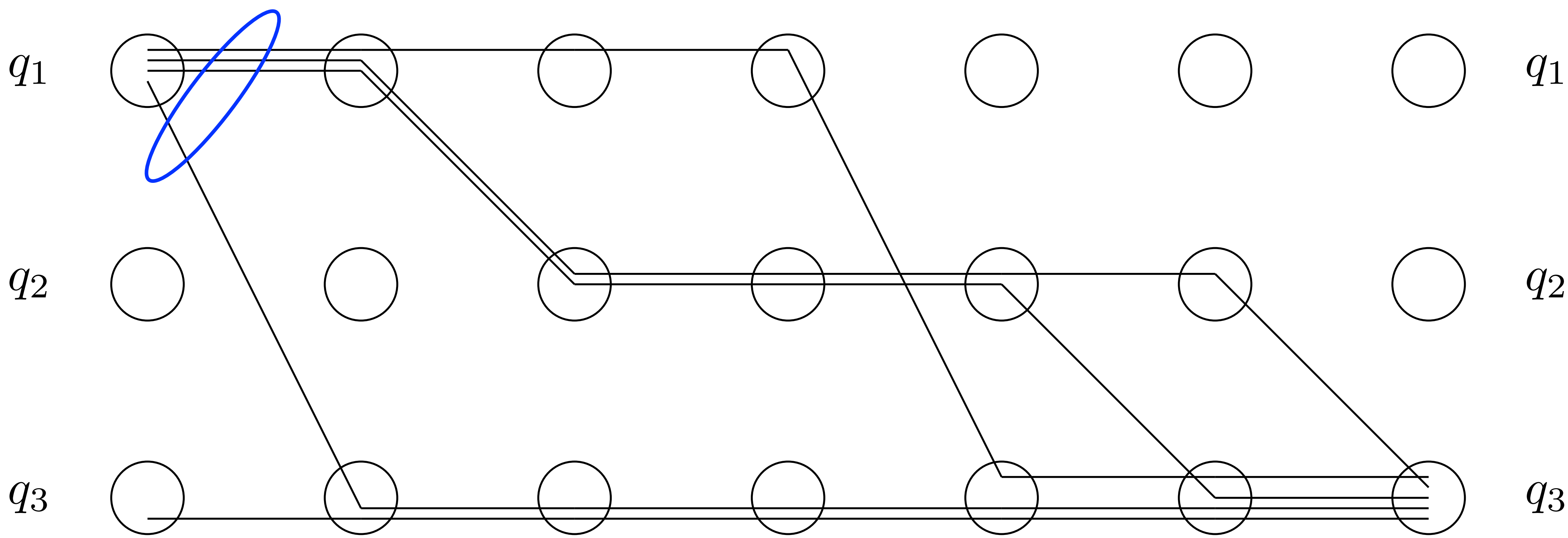
Transitions are enabled as soon as there is one token in the observed place



Idea: place a token in each observed place before its **first** “observation” and leave it till its **last** “observation”

The Pruning Theorem

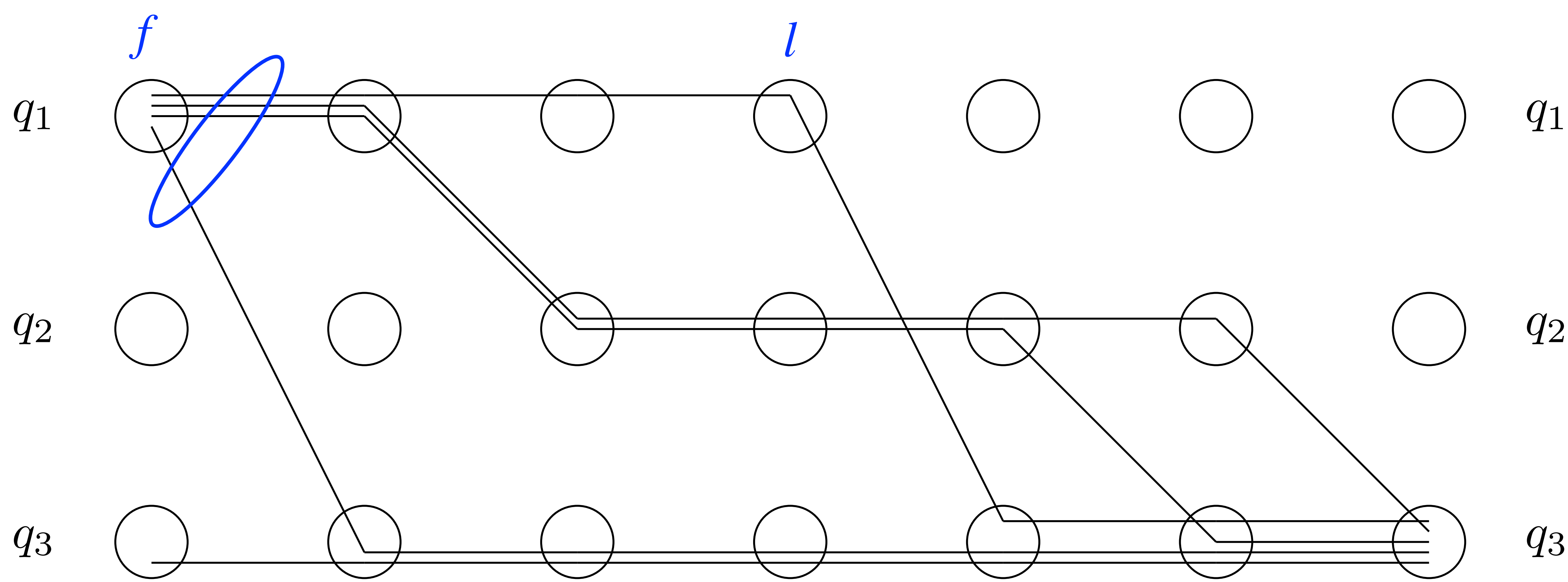
Transitions are enabled as soon as there is one token in the observed place



Idea: place a token in each observed place before its **first** “observation” and leave it till its **last** “observation”

The Pruning Theorem

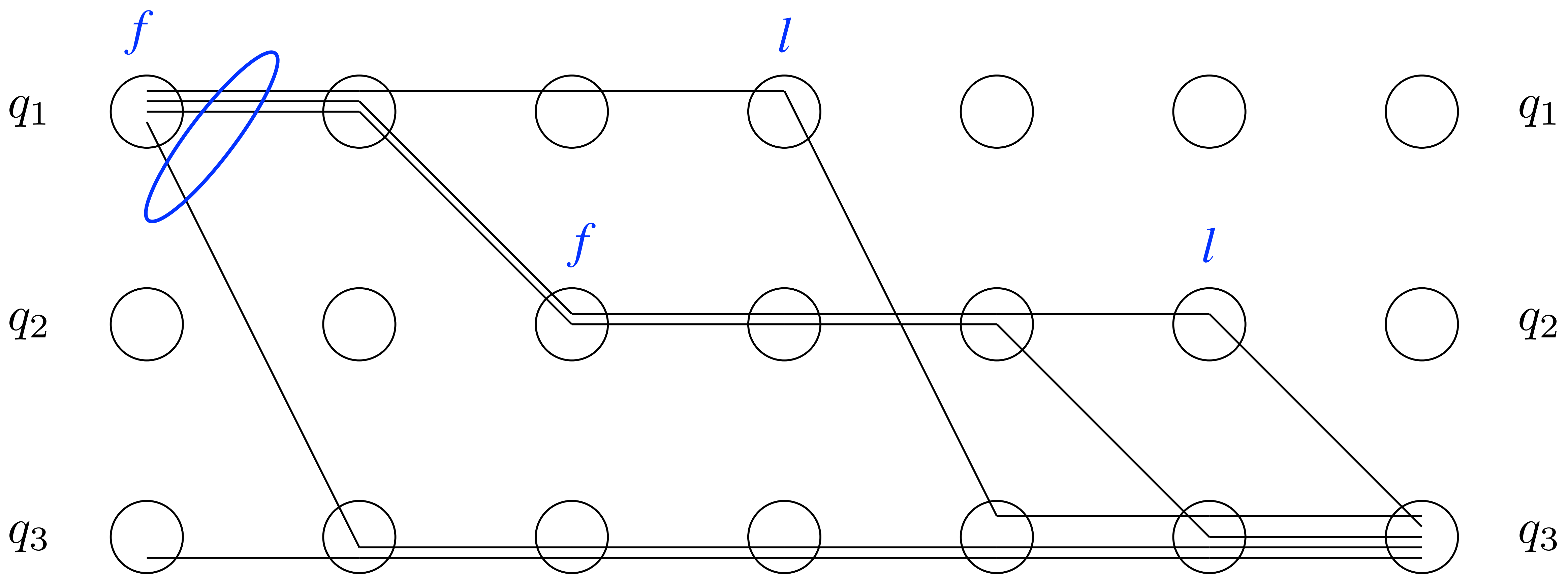
Transitions are enabled as soon as there is one token in the observed place



Idea: place a token in each observed place before its **first** “observation”
and leave it till its **last** “observation”

The Pruning Theorem

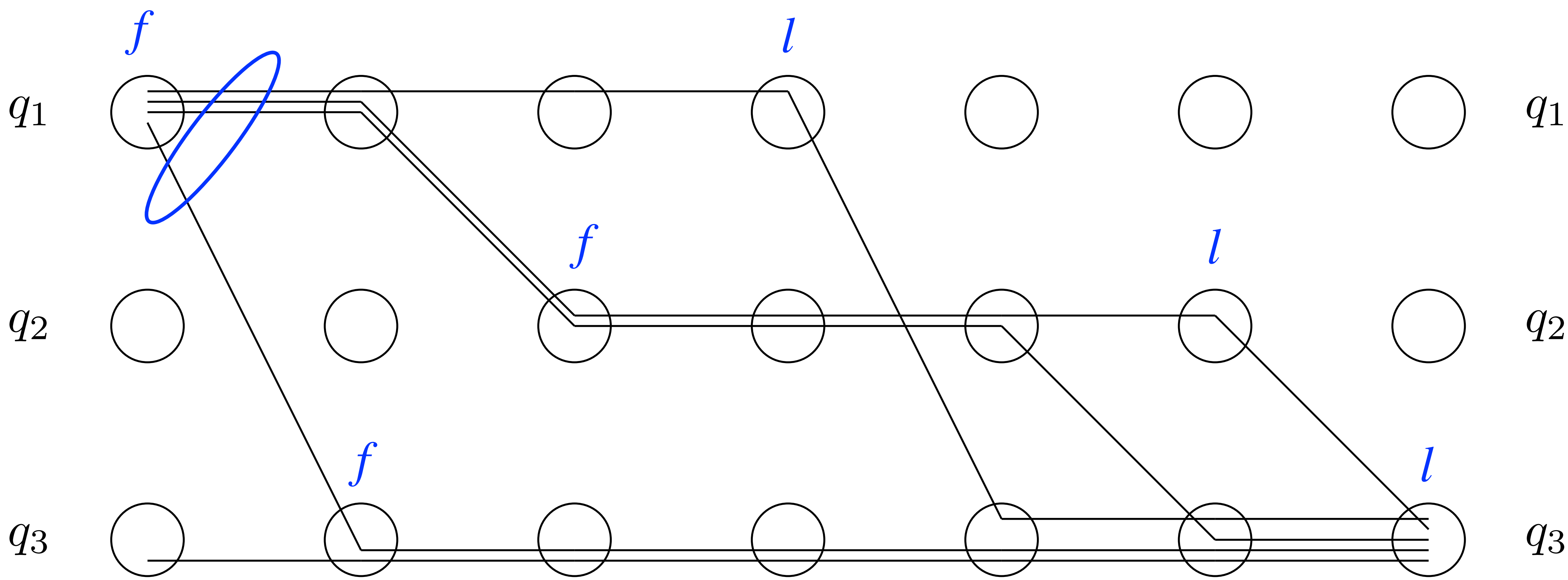
Transitions are enabled as soon as there is one token in the observed place



Idea: place a token in each observed place before its **first** "observation" and leave it till its **last** "observation"

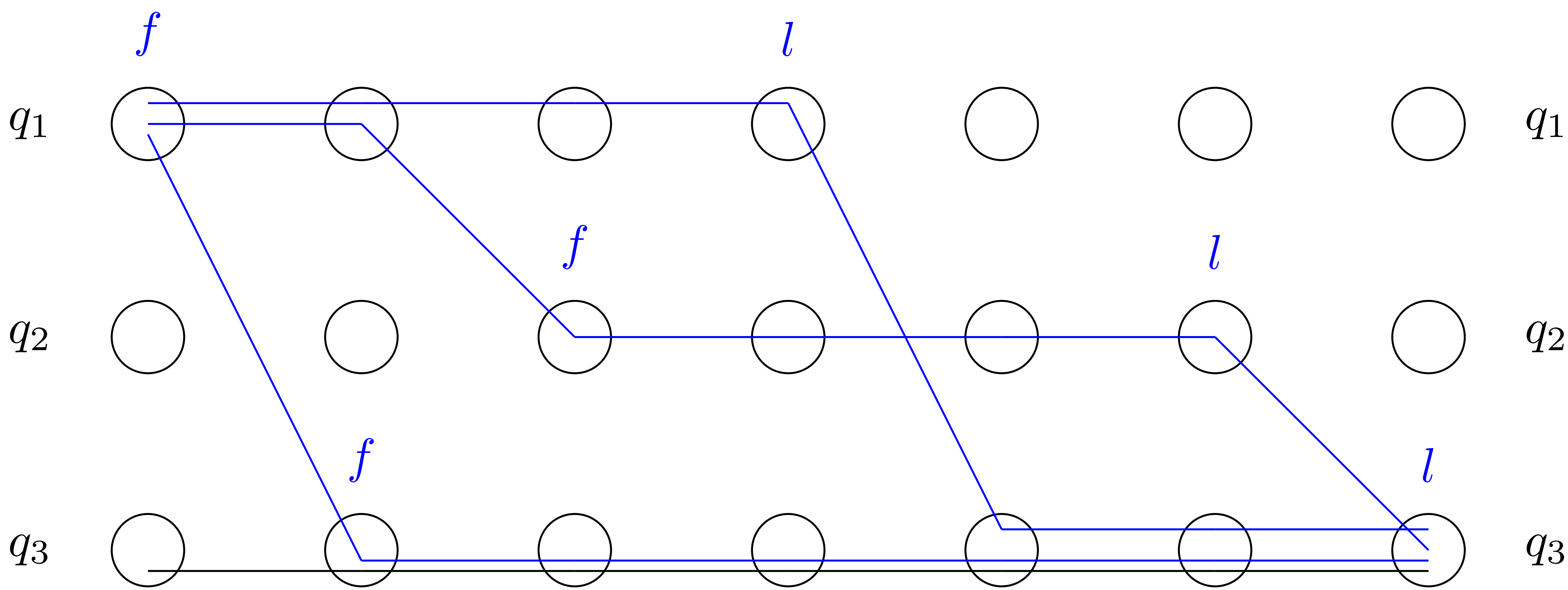
The Pruning Theorem

Transitions are enabled as soon as there is one token in the observed place



Idea: place a token in each observed place before its **first** “observation” and leave it till its **last** “observation”

The Pruning Theorem



The Pruning Theorem

For any run

$$M \xrightarrow{*} M' \geq M''$$

$$\geq \qquad \geq$$

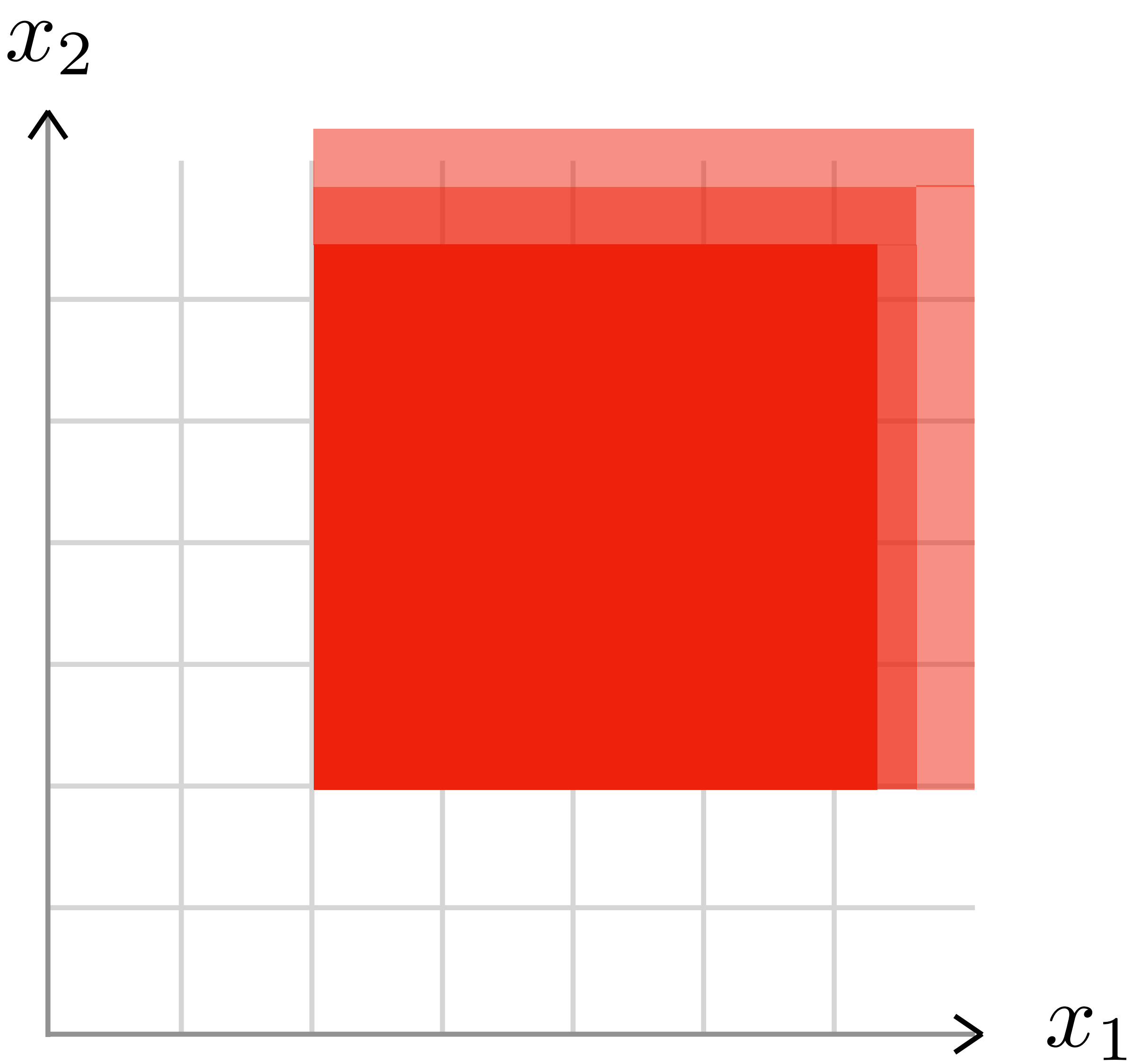
there exists a run

$$S \xrightarrow{*} S' \geq M''$$

such that

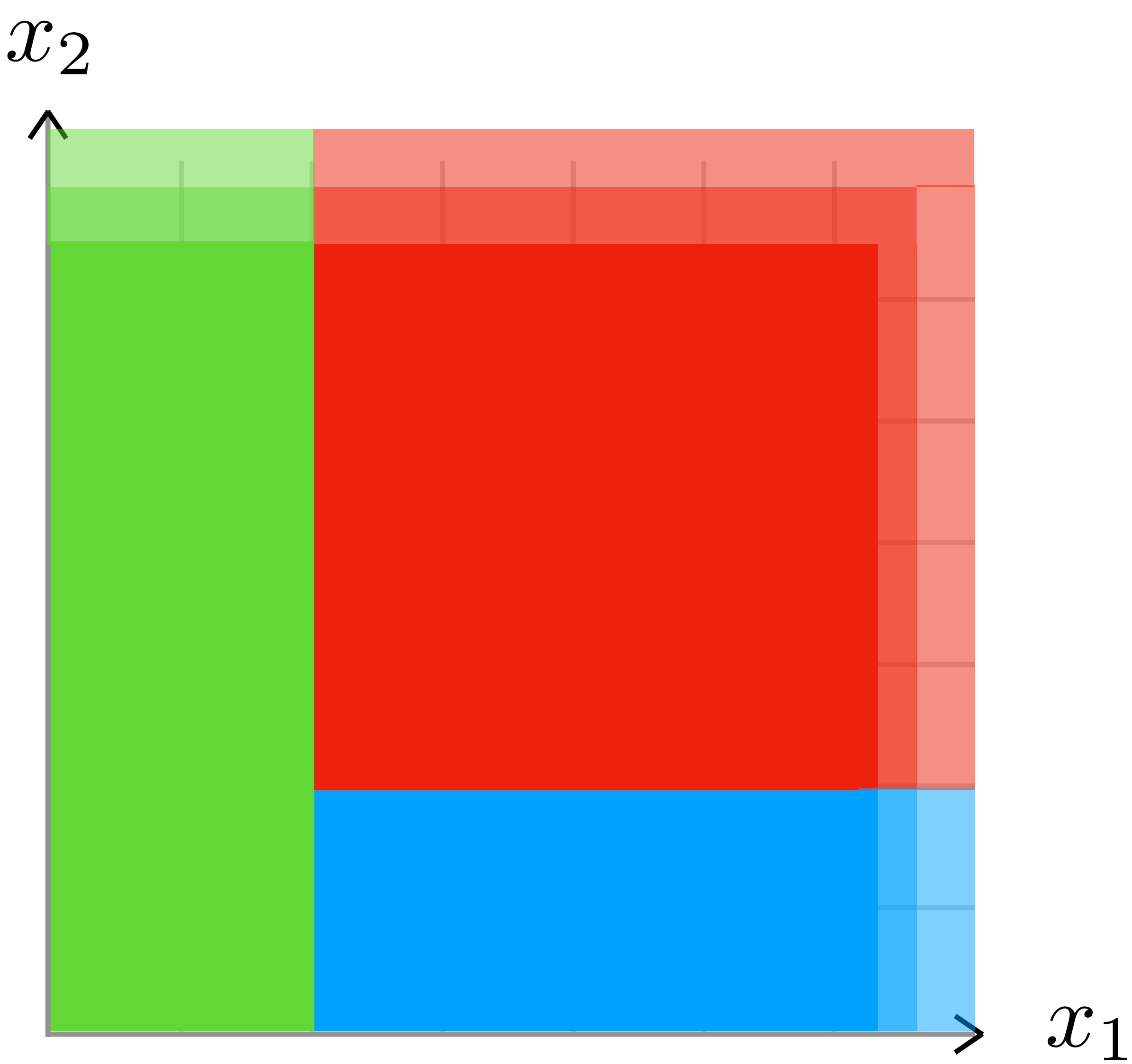
$$|S| \leq |M''| + n^3$$

Counting Constraints



$$2 \leq x_1 \leq \infty$$
$$2 \leq x_2 \leq \infty$$

Counting Constraints



$$\neg \begin{array}{c} 2 \leq x_1 \leq \infty \\ 2 \leq x_2 \leq \infty \end{array}$$

$$\Downarrow$$

$$\begin{array}{c} 2 \leq x_1 \leq \infty \\ 0 \leq x_2 \leq 2 \end{array} \vee \begin{array}{c} 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq \infty \end{array}$$

Bringing it together

Bound on the size of the reachability sets

For *N* an IO net with *n* places, for *S* a counting set, there exists counting constraints representing *pre*(S)* and *post*(S)* whose size is bound by

$$\|pre^*(S)\| \leq \|S\| + n^3$$

$$\|post^*(S)\| \leq \|S\| + n^3$$

Bringing it together

Cube-reachability can be solved in PSPACE

Algorithm idea: Let *S* and *S'* two counting sets.

If S' is reachable from *S,* then $S \cap pre^*(S')$ is not empty.

By the previous theorem (and other results), there exists a "small" marking in $S \cap pre^*(S')$.

We pick such a marking in *S* and such a marking in *S',* and then guess a path from one to the other.

Conclusion

- This entails the results for cube-coverability and cube-liveness in IO nets.
- The PSPACE algorithm is also induced by the cube-reachability algorithm and the fact that protocol correctness can be expressed a formula over counting sets of markings in the corresponding IO net.
- The PSPACE-hardness results come from reductions from the halting problem for bounded-tape Turing Machines.

Conclusion

- This entails the results for cube-coverability and cube-liveness in IO nets.
- The PSPACE algorithm is also induced by the cube-reachability algorithm and the fact that protocol correctness can be expressed a formula over counting sets of markings in the corresponding IO net.
- The PSPACE-hardness results come from reductions from the halting problem for bounded-tape Turing Machines.

Thank you !