Gödel Hashing

matt.might.net
@mattmight

Disclaimer

“simple, fun idea”

“simple, fun idea”

“works well in practice,”

“simple, fun idea”

“works well in practice,”

“but theory says it will not.”

An old problem

An older solution

A big impact

An old problem

“CFA is slow!”

An older solution

Gödel hashing

functional
monotonic
compact
dynamic
incremental
perfect

Inspired by a true theorem.

Word-level parallelism!

Great cache behavior!

A big impact

Minutes of work

2x

2x 5x

2x 5x 8x

2x 5x 8x 100x

Motivation

```
(f x)
```

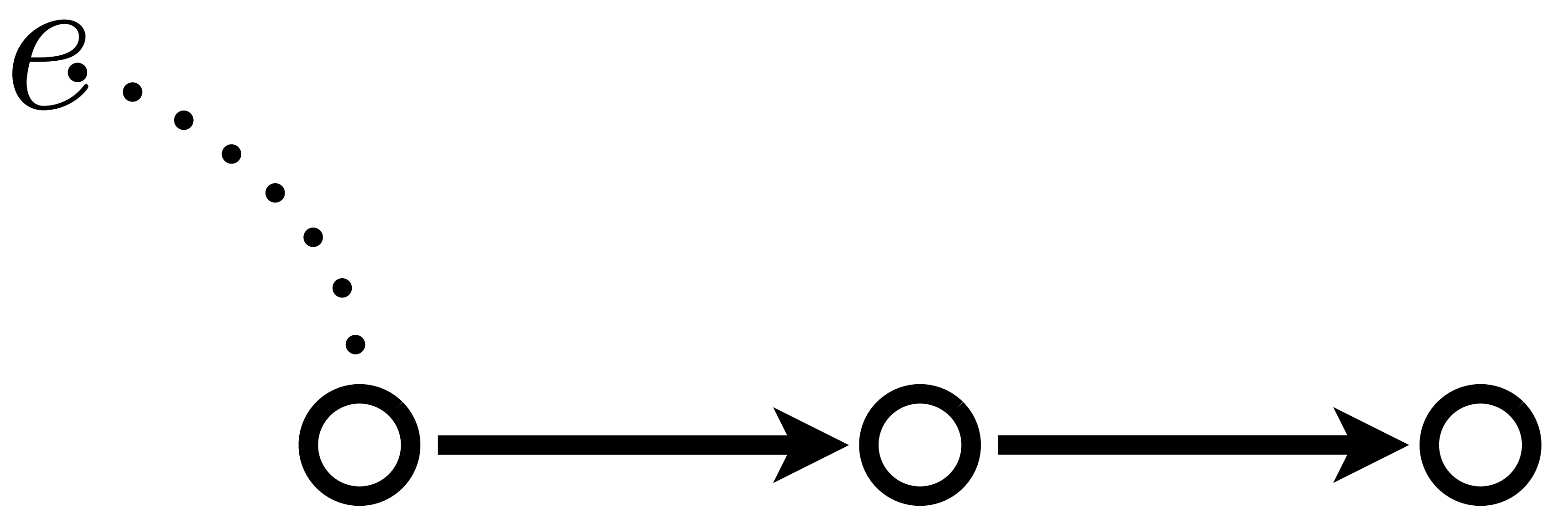
$f(x)$

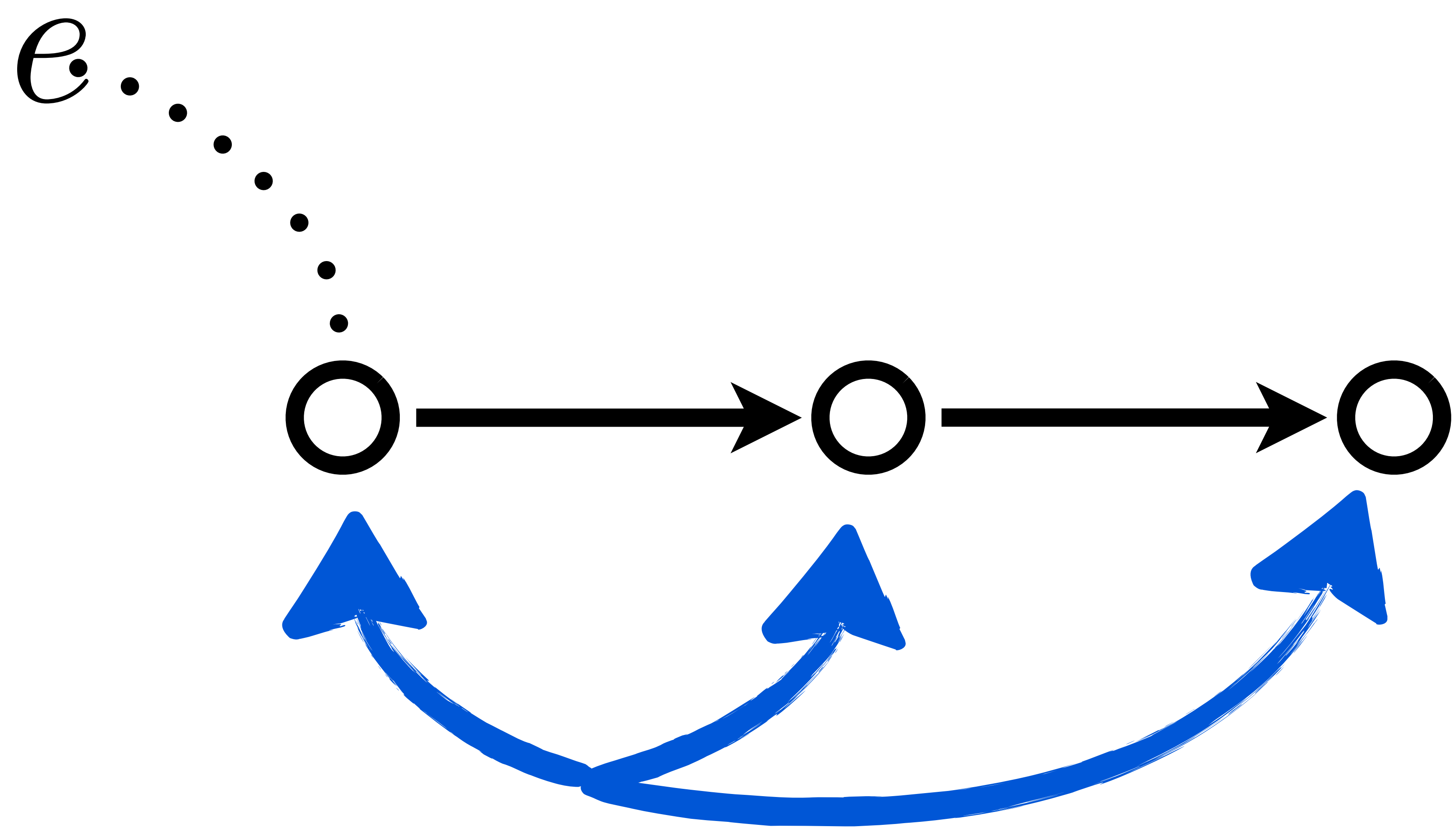
What is `f`?

Why not run the program?

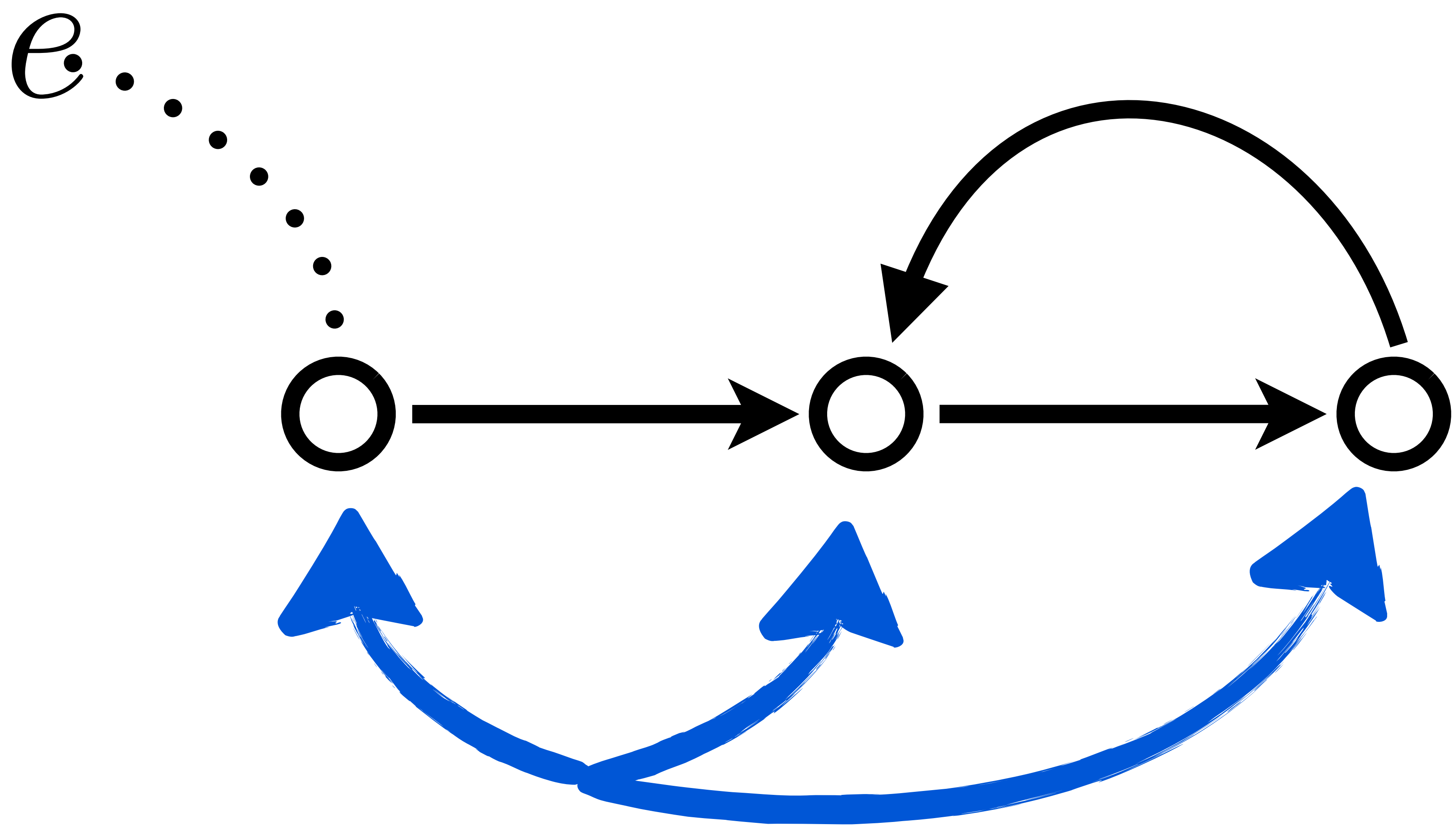
e

e

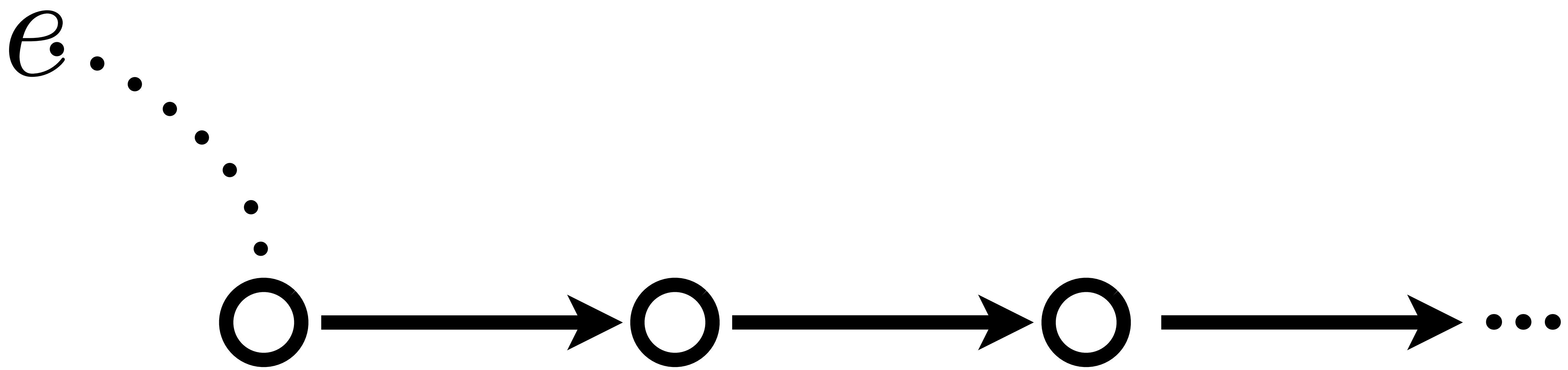

e



What is $\mathtt{f}$, *here*?

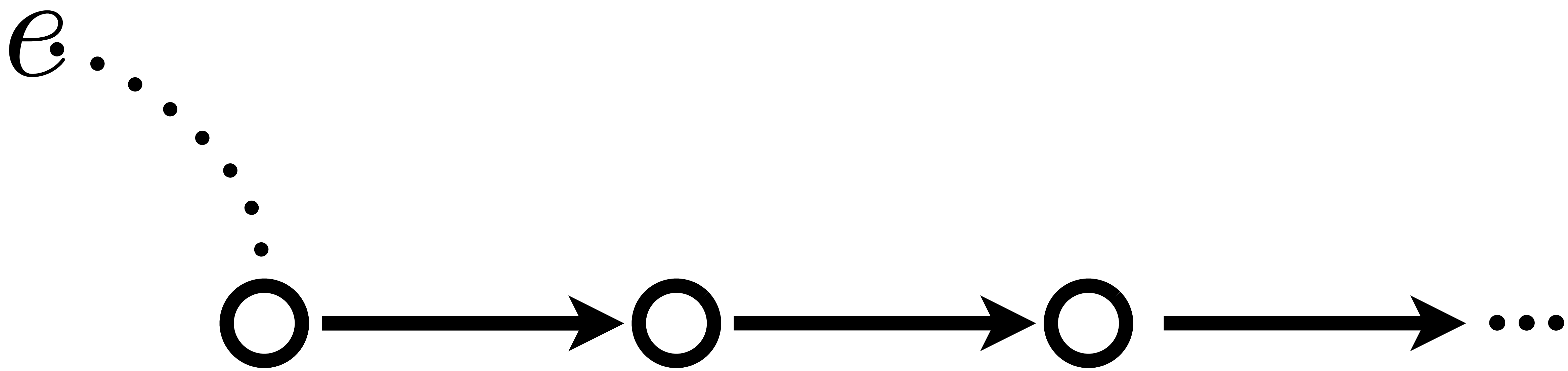


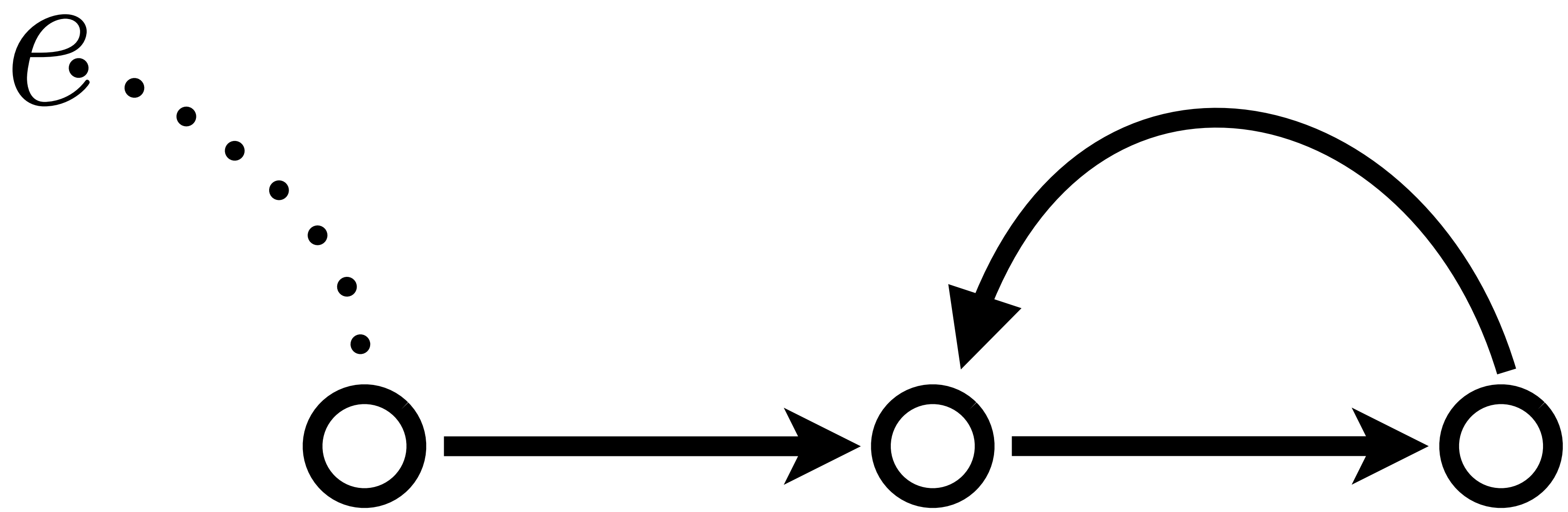
What is f, *here*?

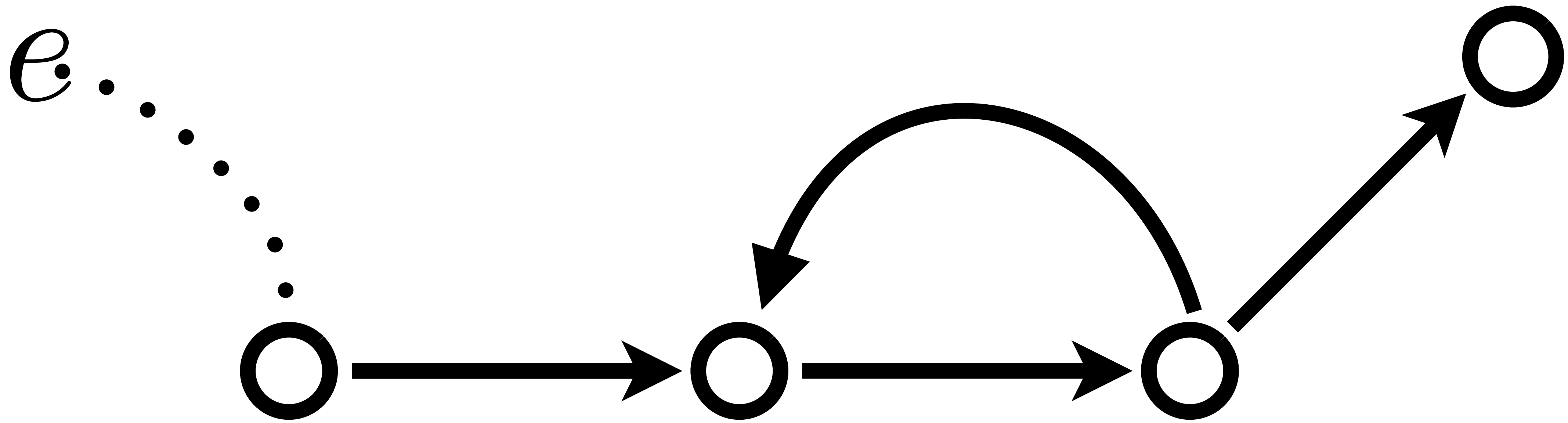


e

e
AAM
...

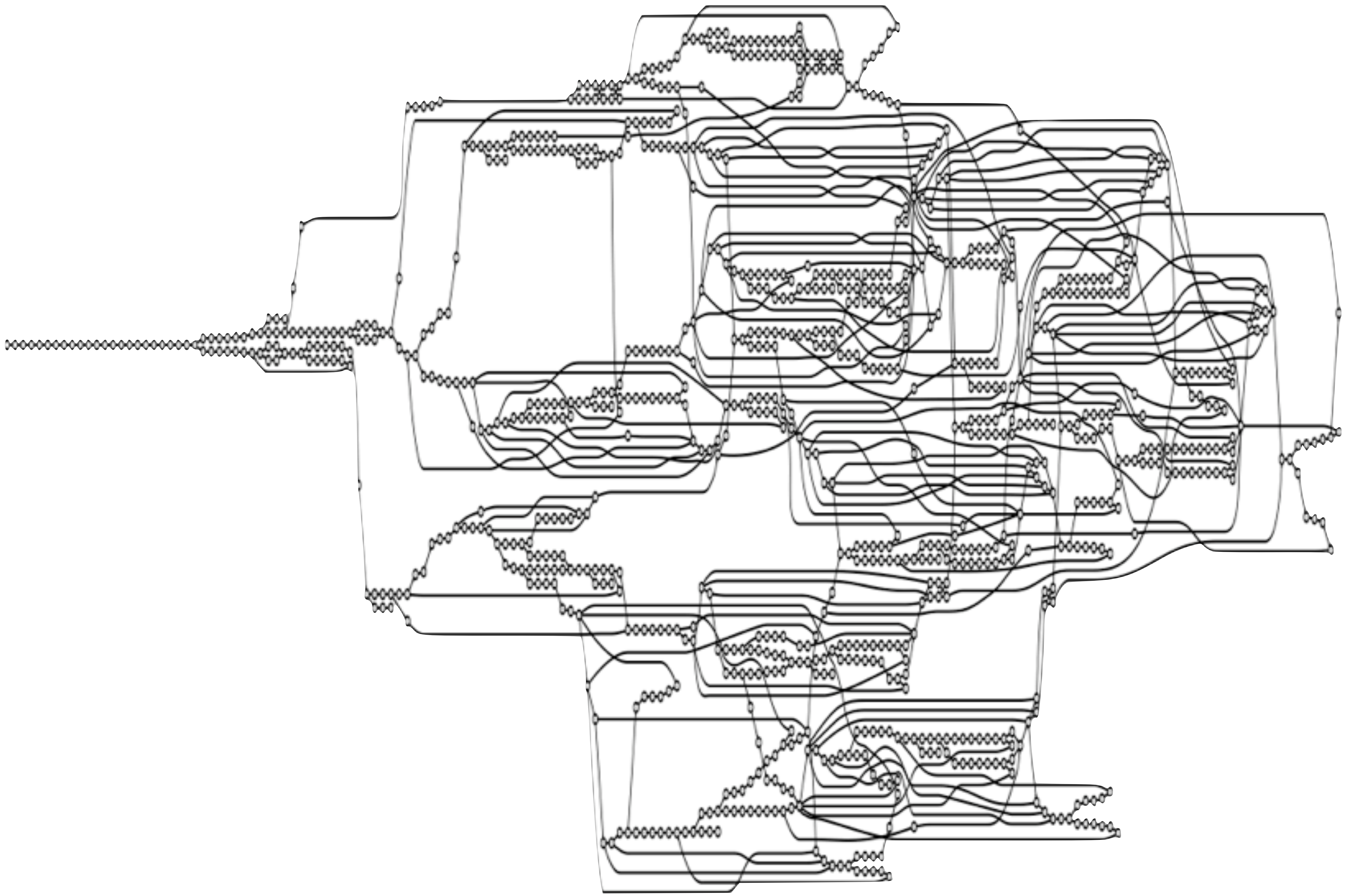

e

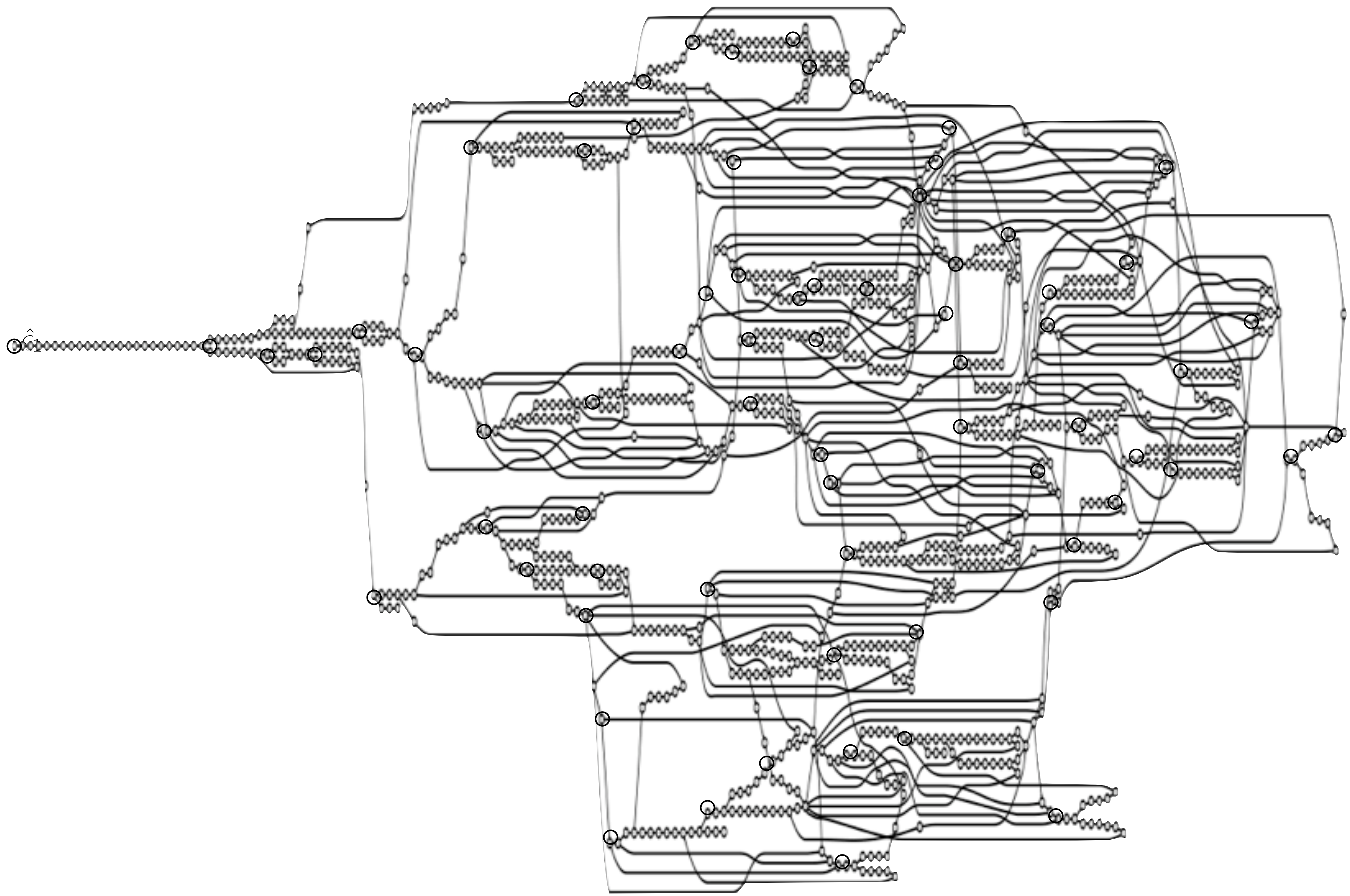

e

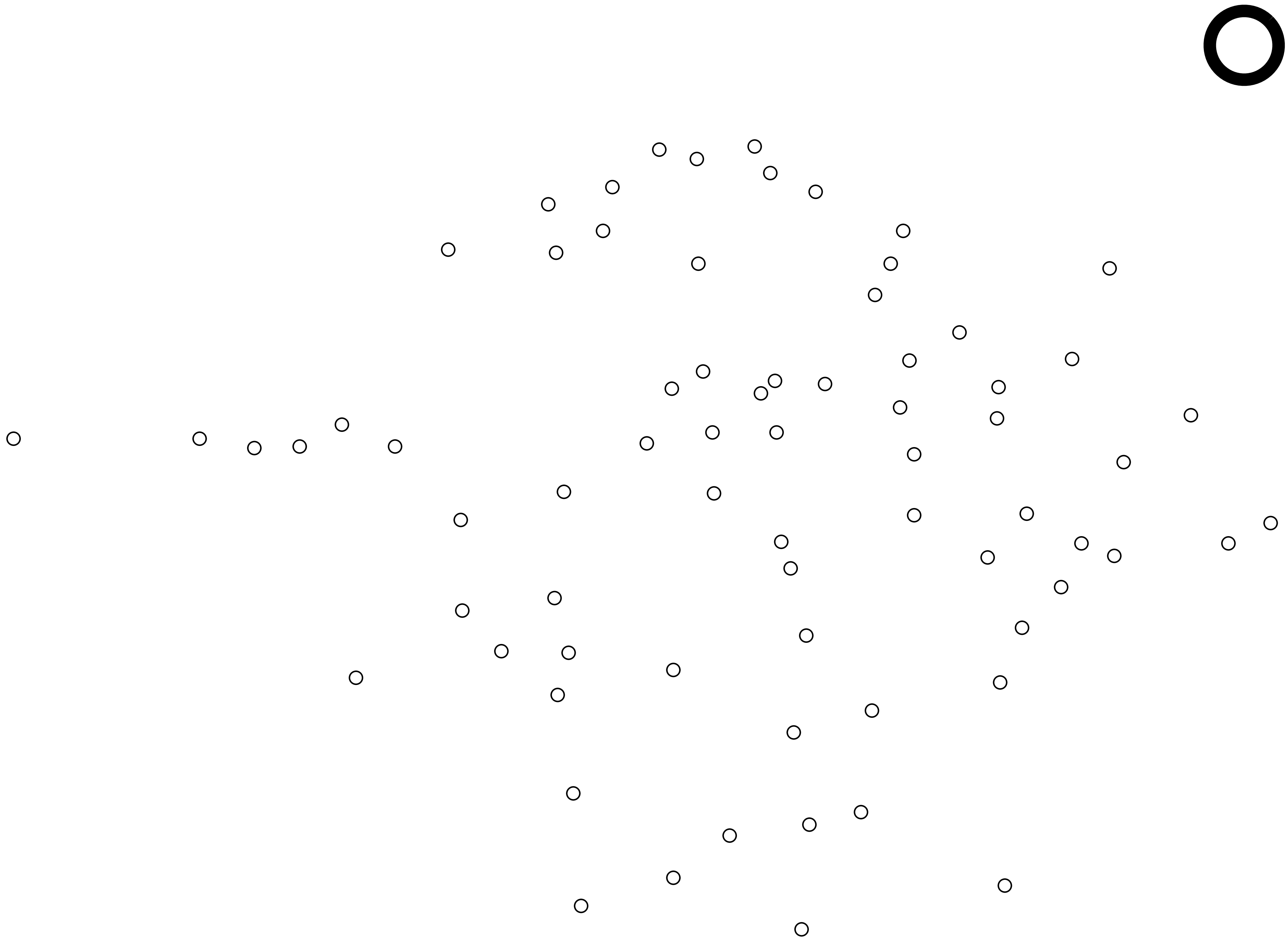


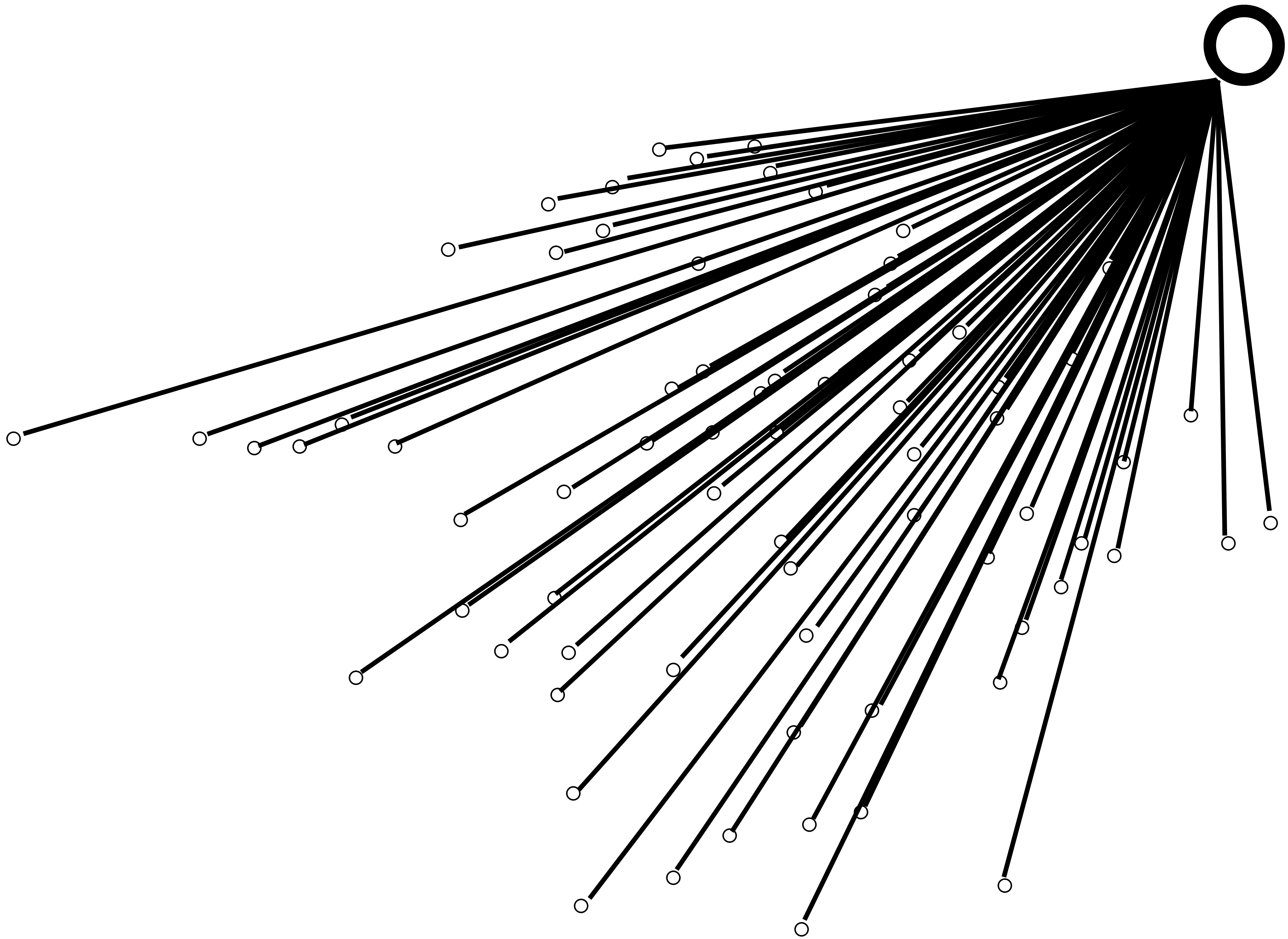
e

Problem









$\sqsubseteq$

$\sqsubseteq$

$$\hat{\varsigma}_1 \sqsubseteq \hat{\varsigma}_2$$

$$\hat{\varsigma}_1 = \left(e, \hat{\rho}, \hat{\sigma}, \hat{\kappa}\right)$$

expression store

$$\hat{\varsigma}_1 = (e, \hat{\rho}, \hat{\sigma}, \hat{\kappa})$$

environment stack

expression store

$$\hat{\varsigma}_1 = (e, \hat{\rho}, \hat{\sigma}, \hat{\kappa})$$

environment stack

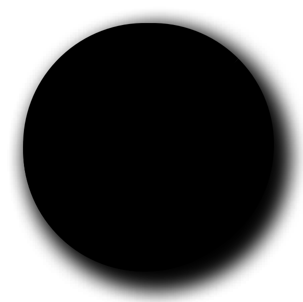
$$\hat{\sigma} : \widehat{Addr} \to \mathcal{P}(\widehat{Value})$$

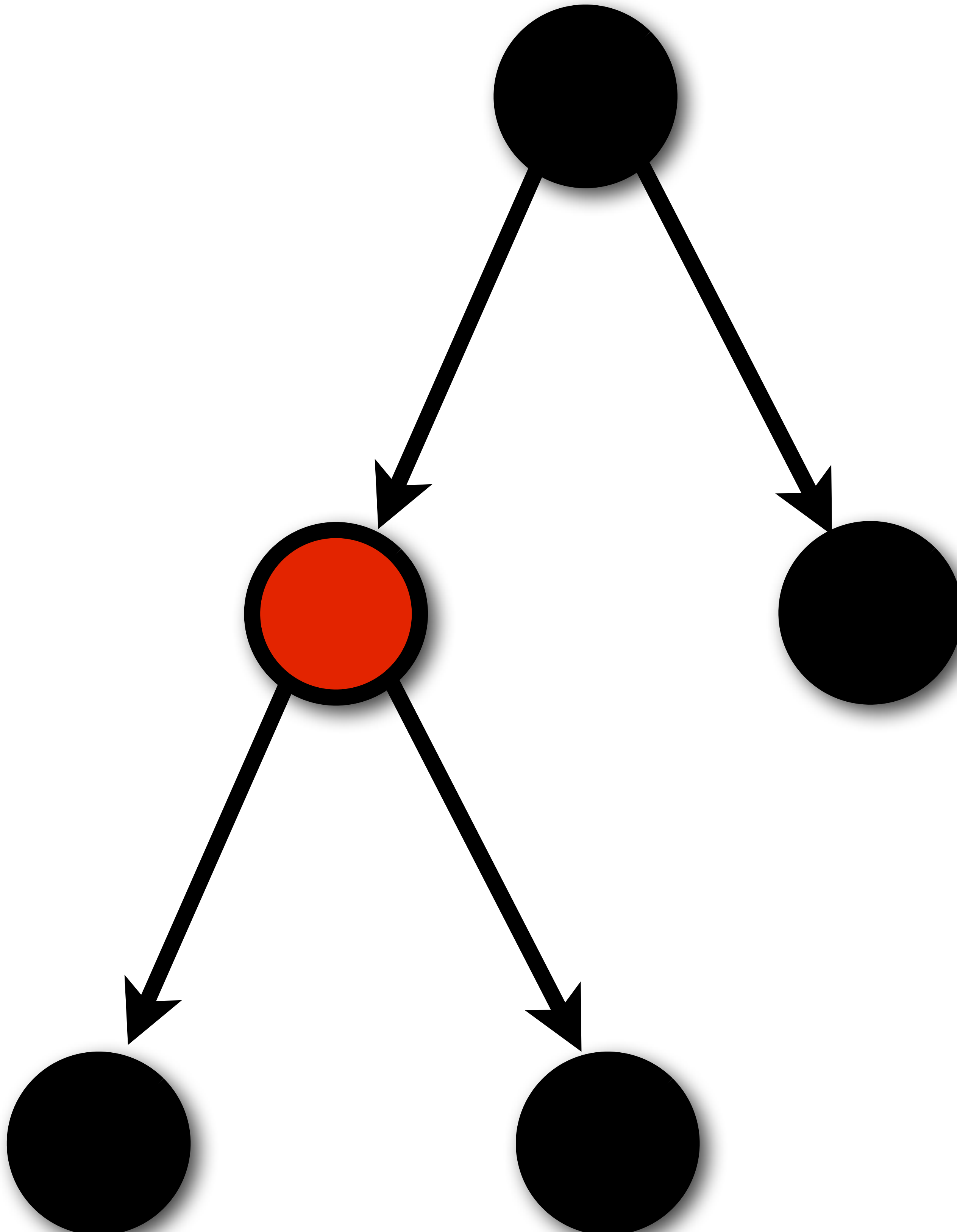
$$\widehat{Value}$$

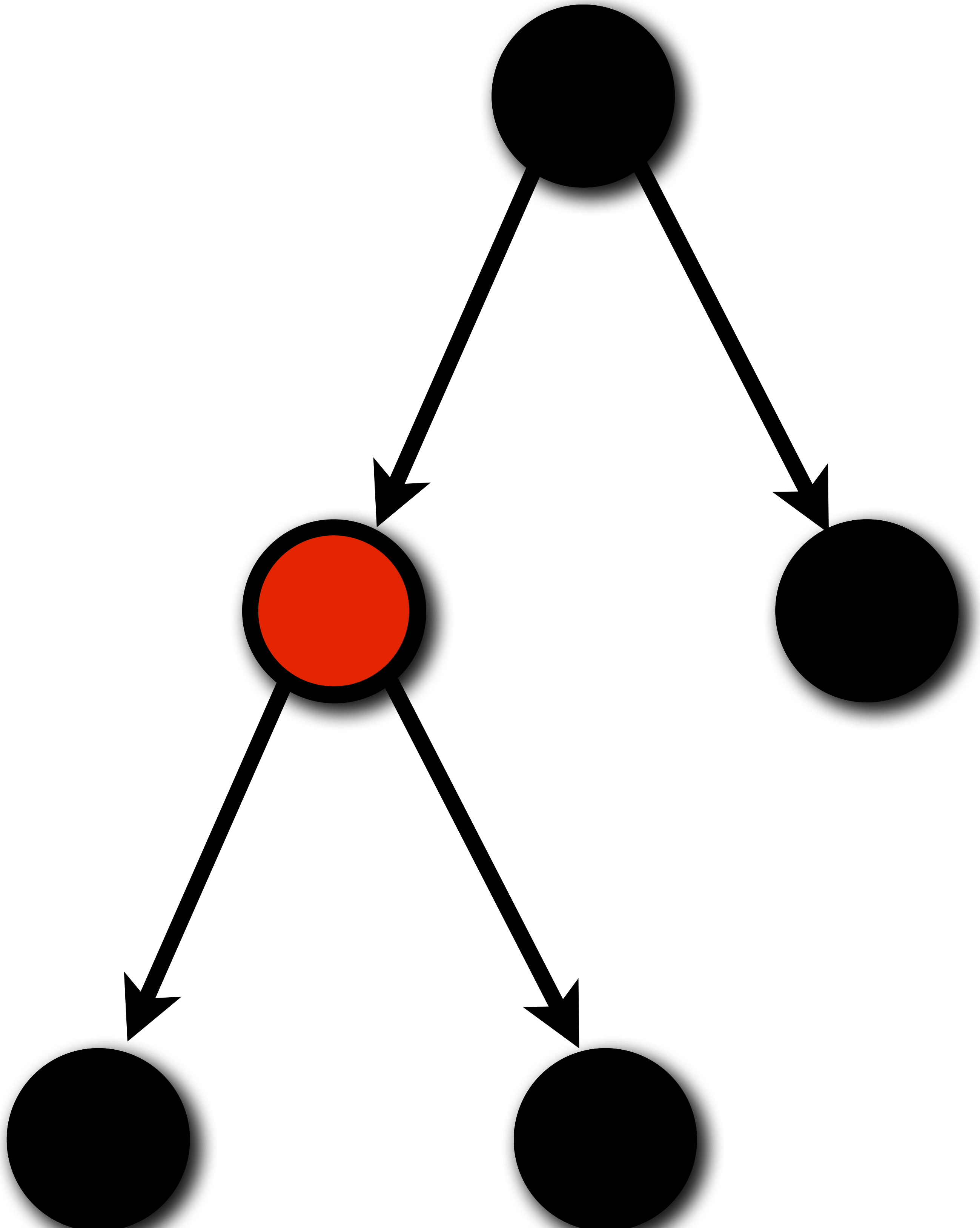
$$\widehat{Addr}$$

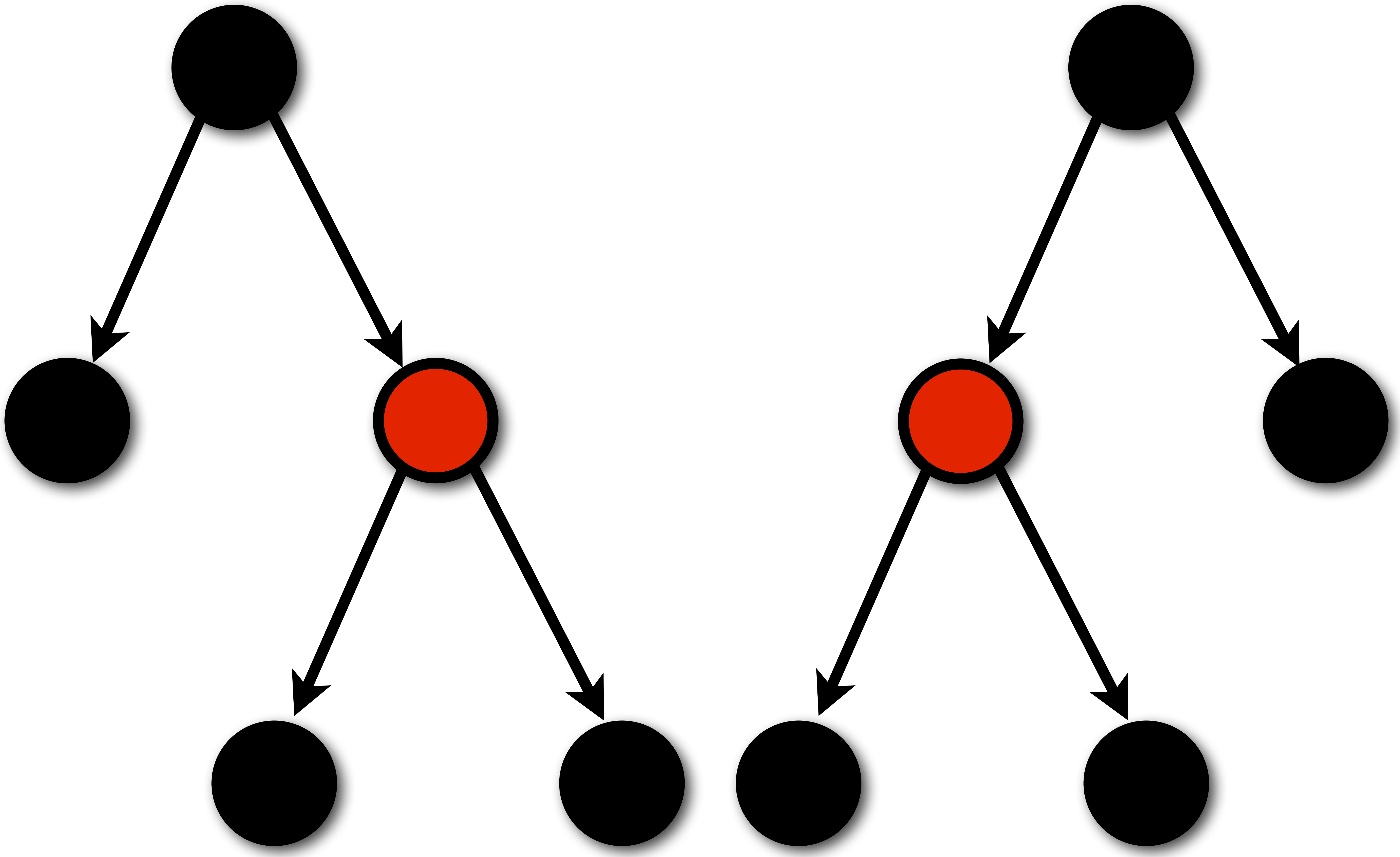
0	1	0	0	0	1	0	1
0	0	0	0	1	1	0	0
0	0	0	1	0	1	0	0
0	1	0	0	0	0	1	0
1	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	0	0	0	1	1	1
0	0	1	0	0	0	1	0

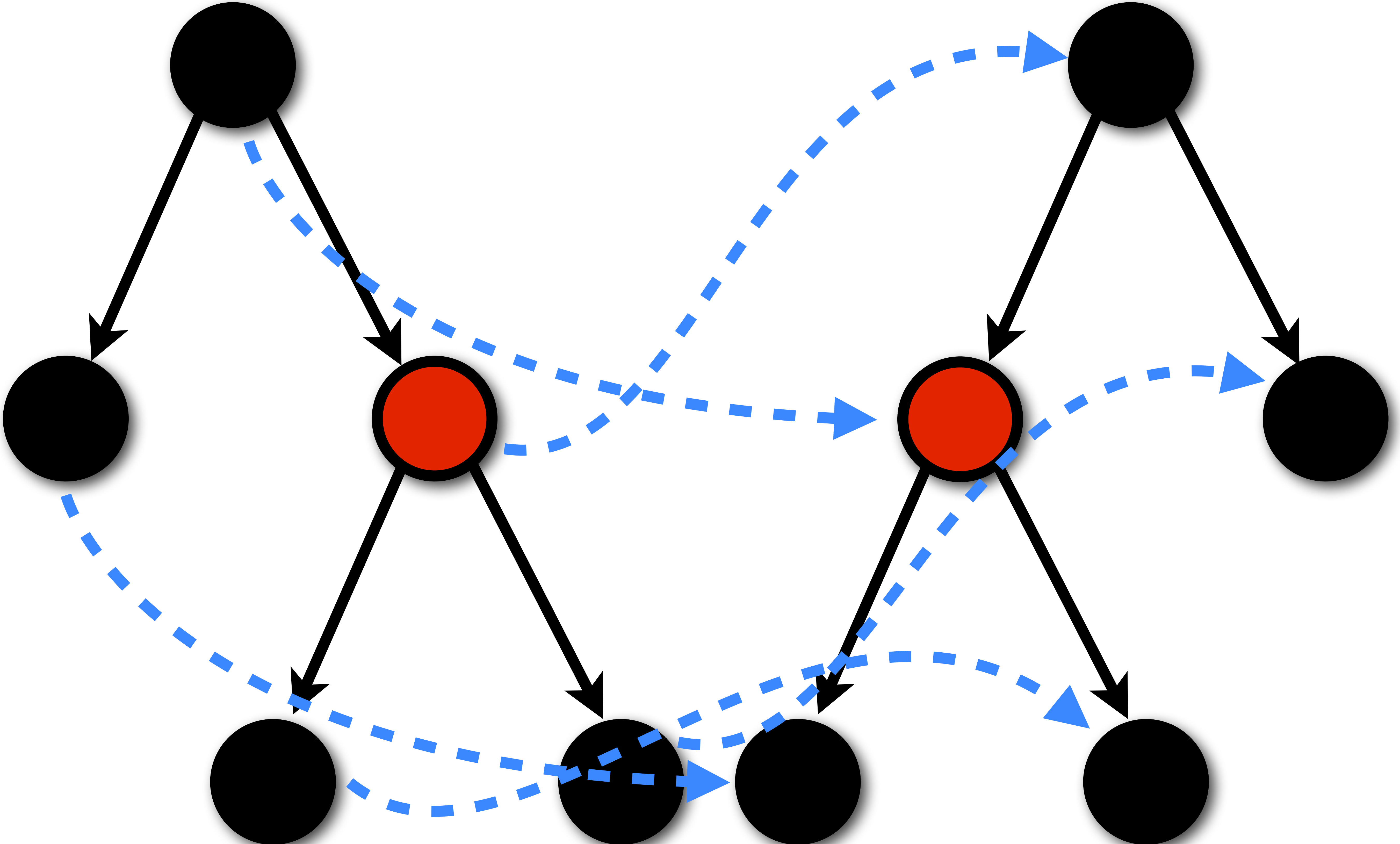
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0
0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0

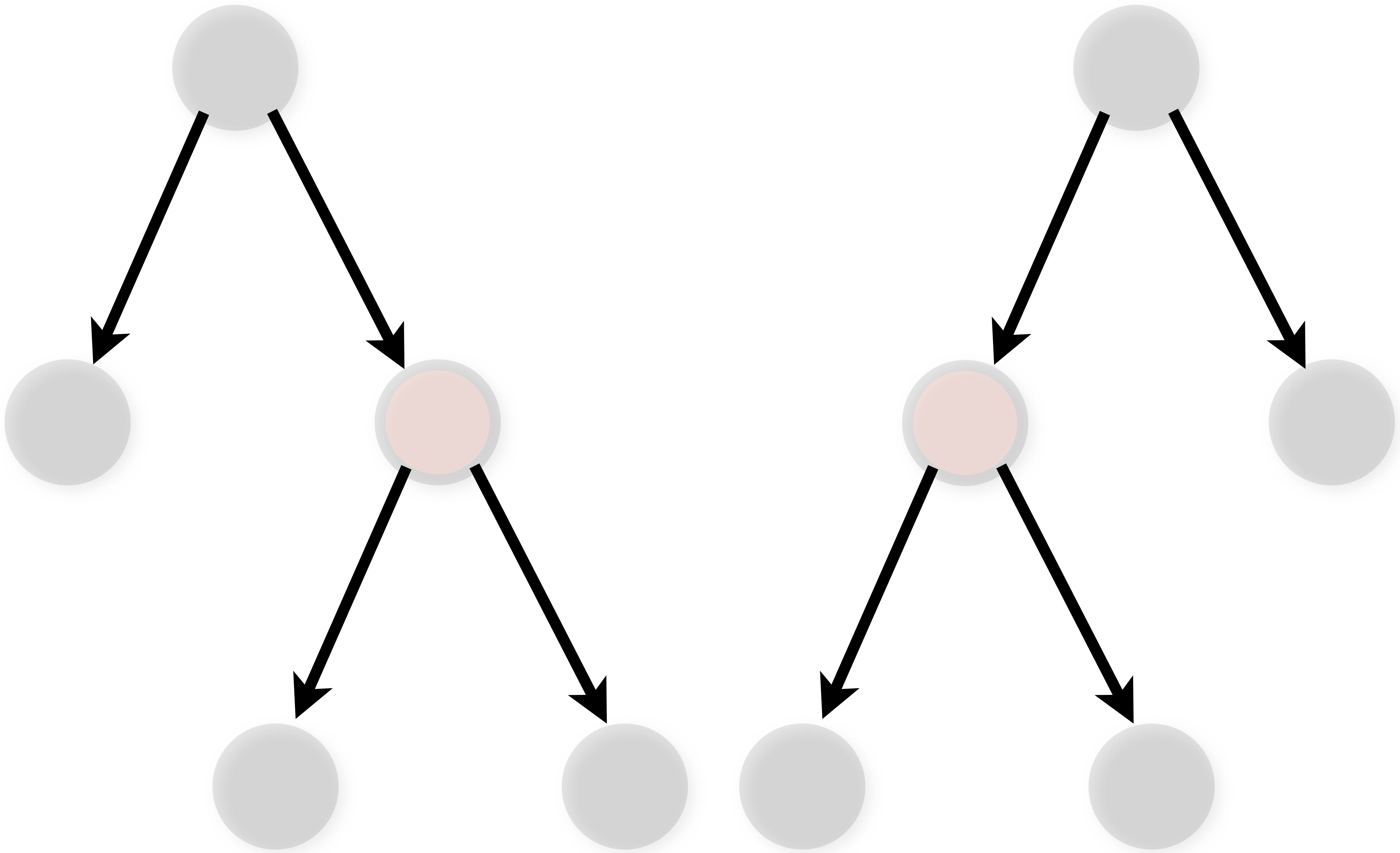


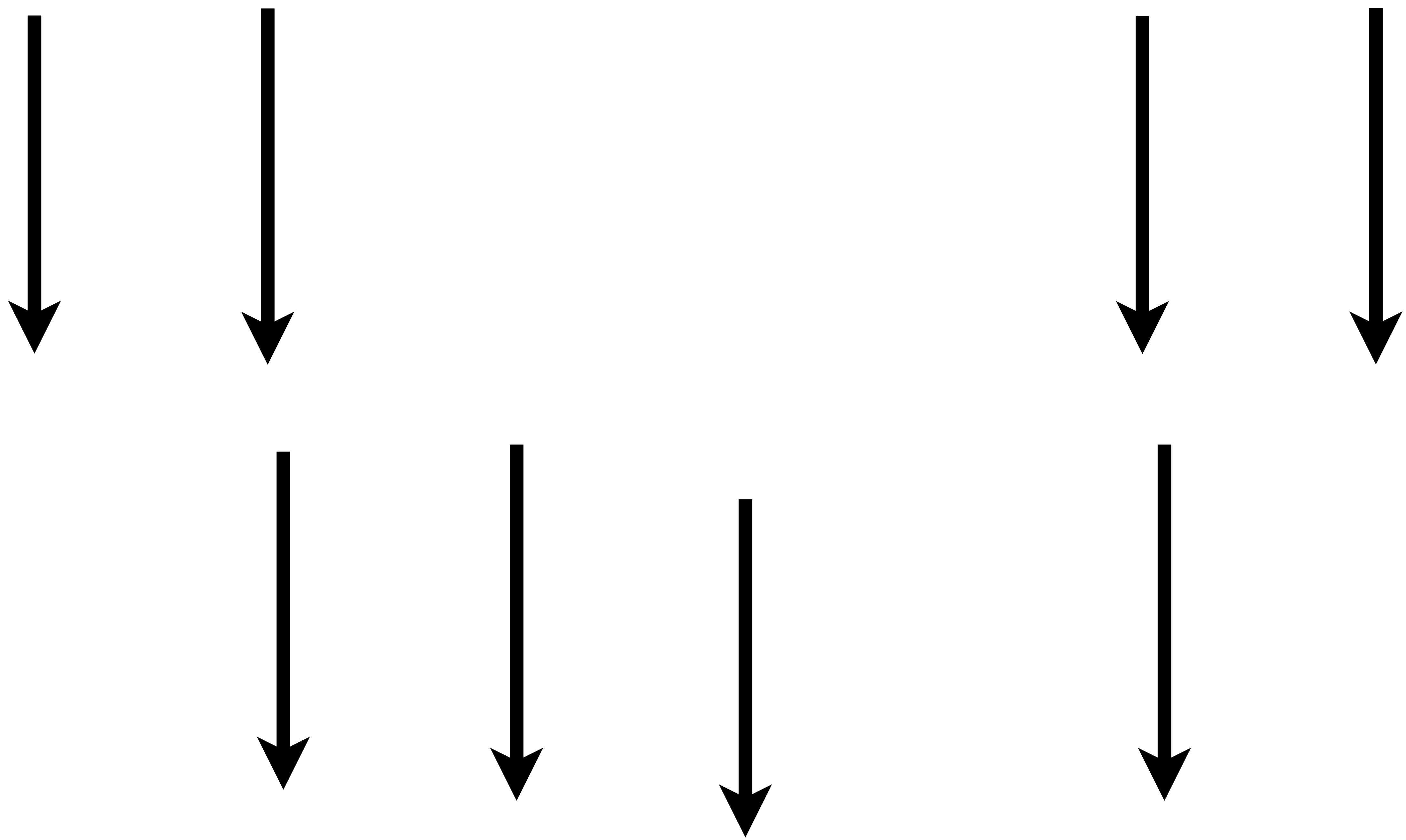


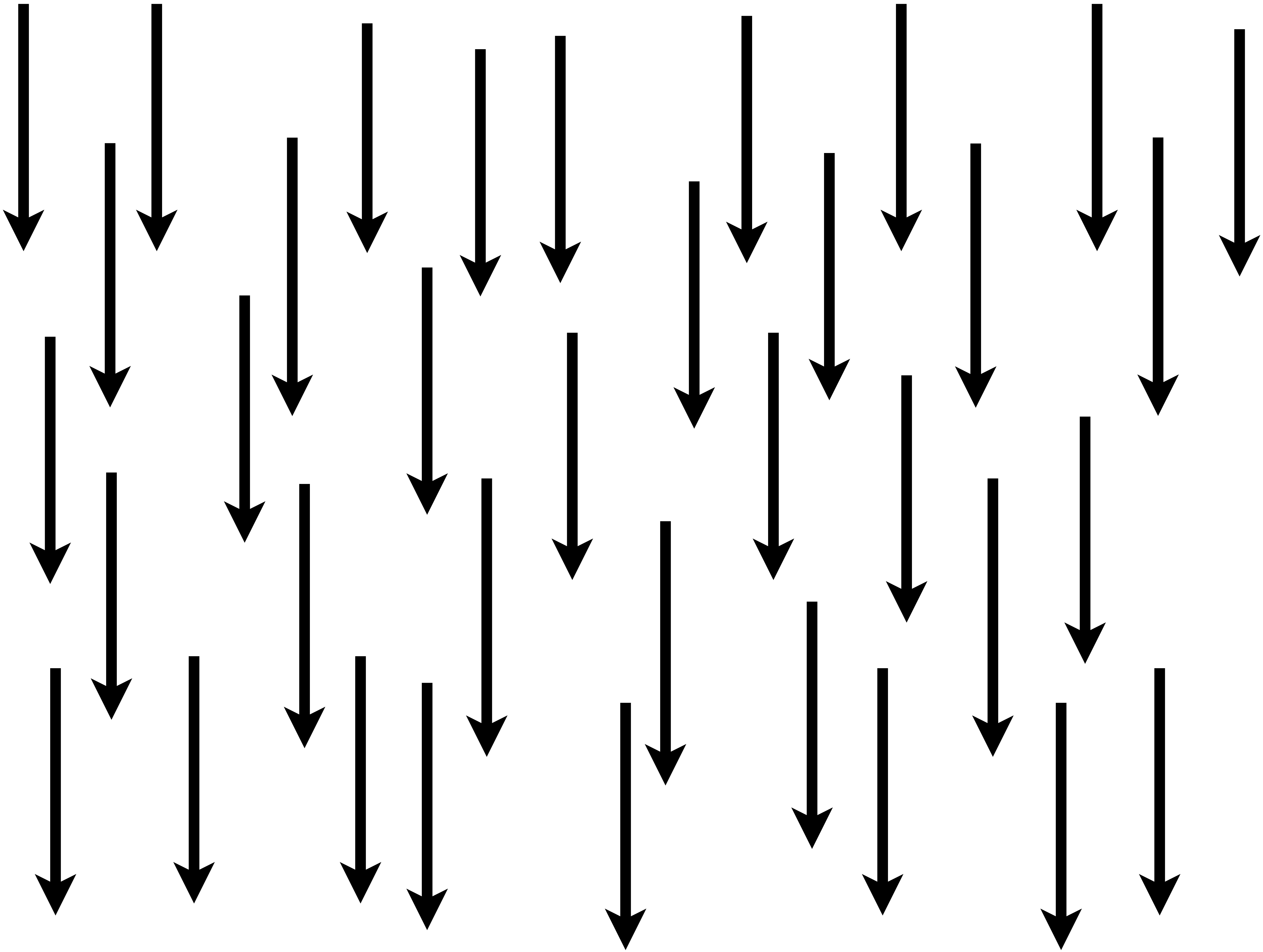


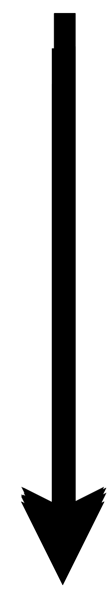






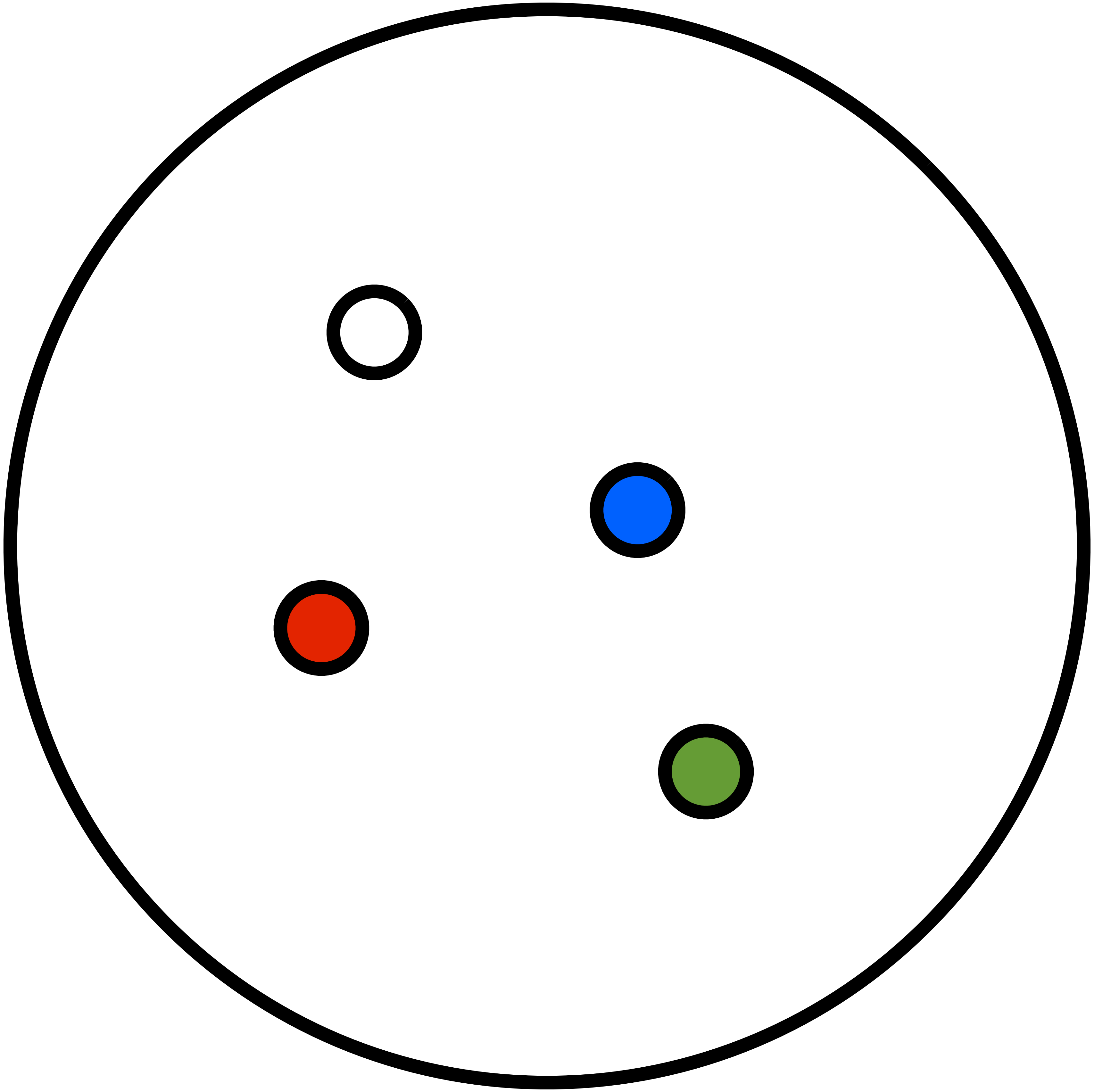


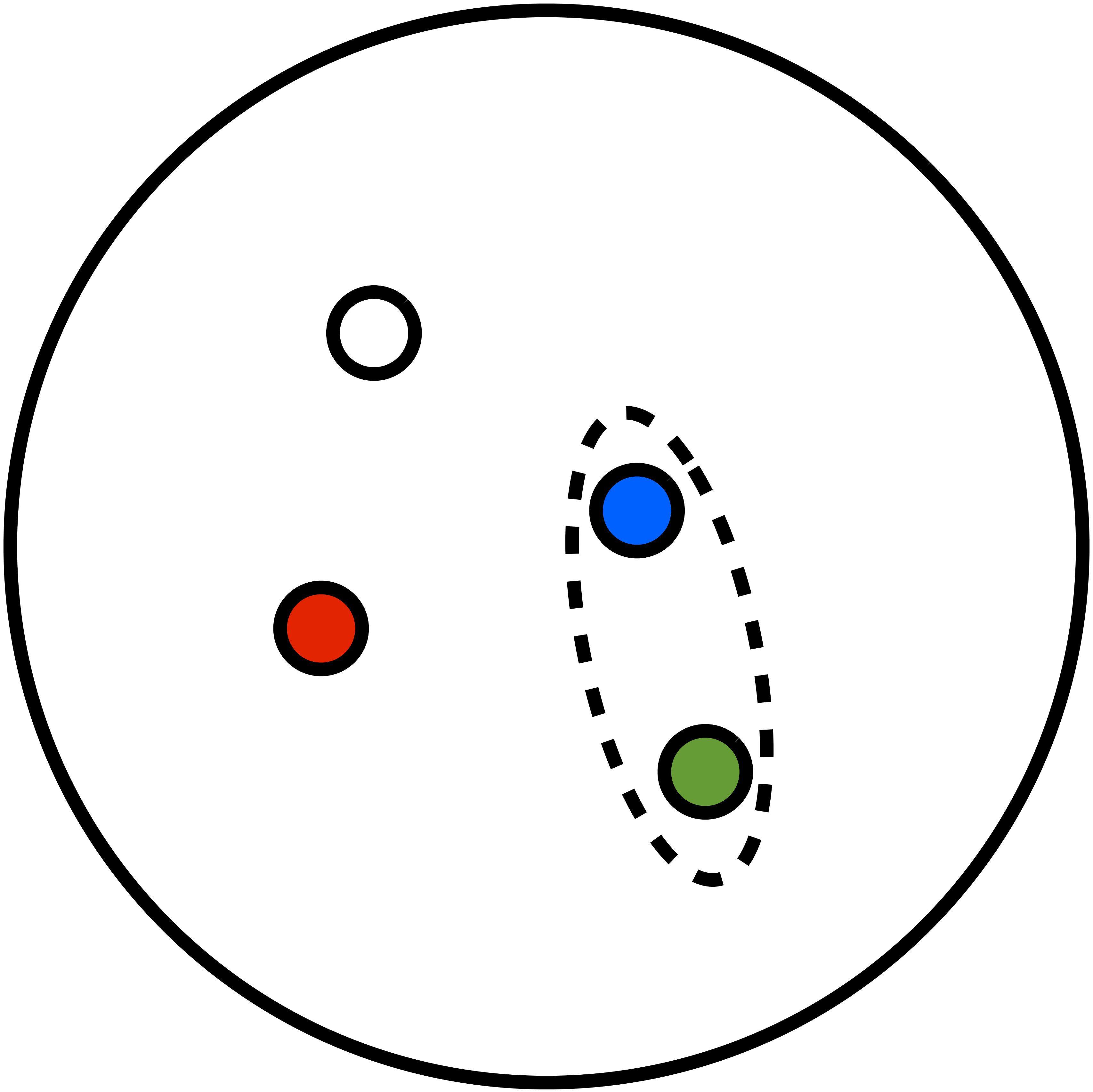


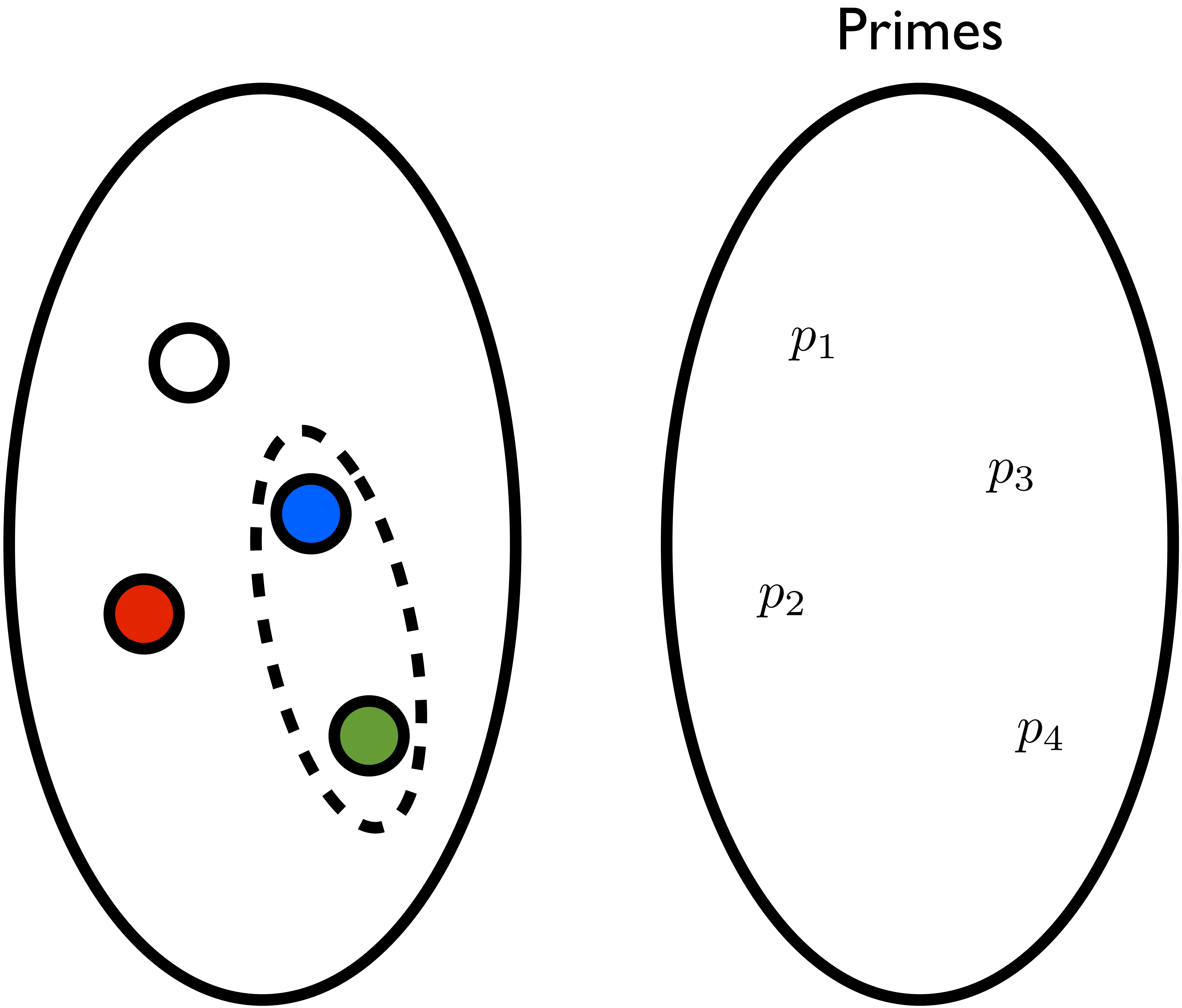


First: Hash sets

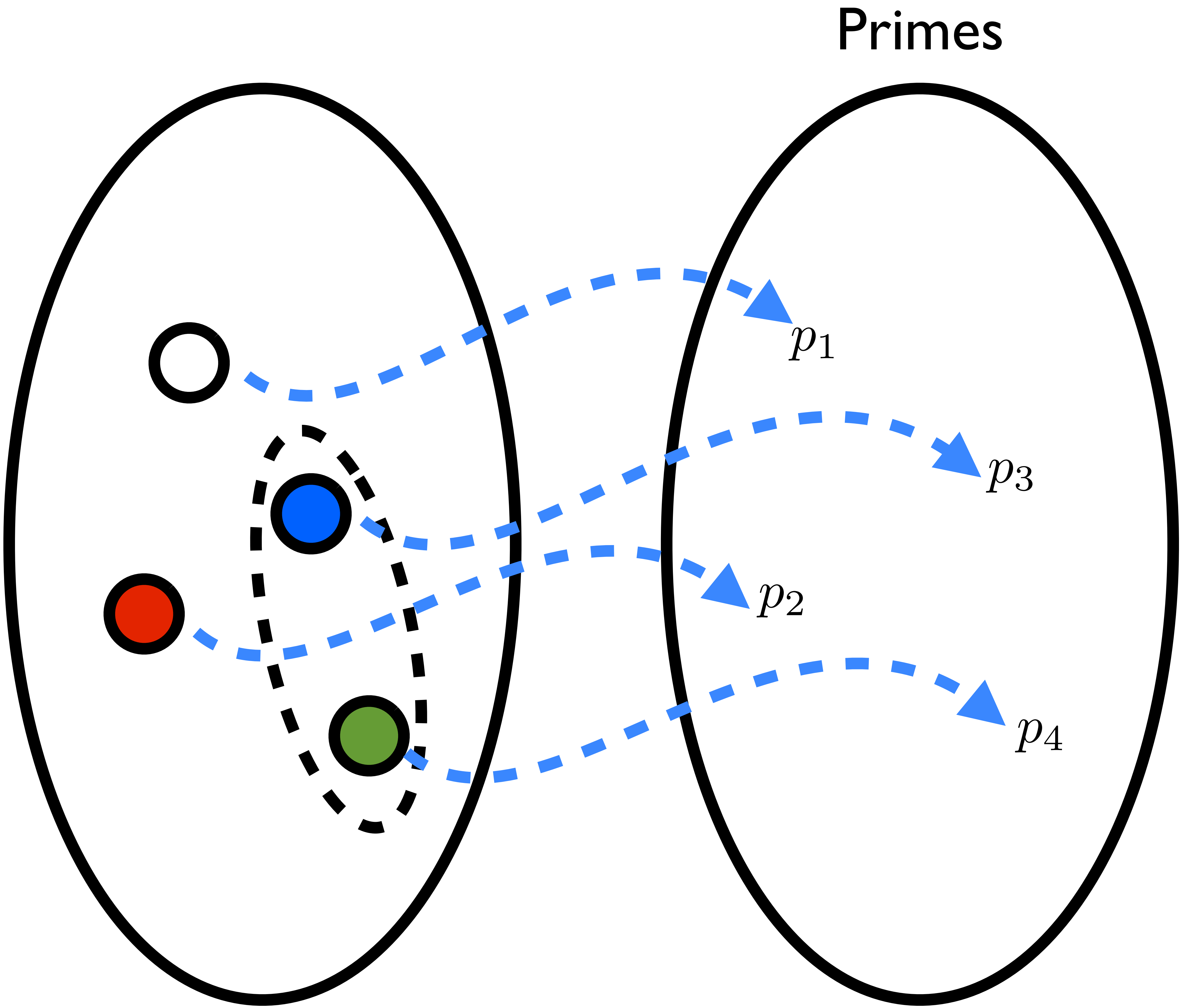
Prime decomposition



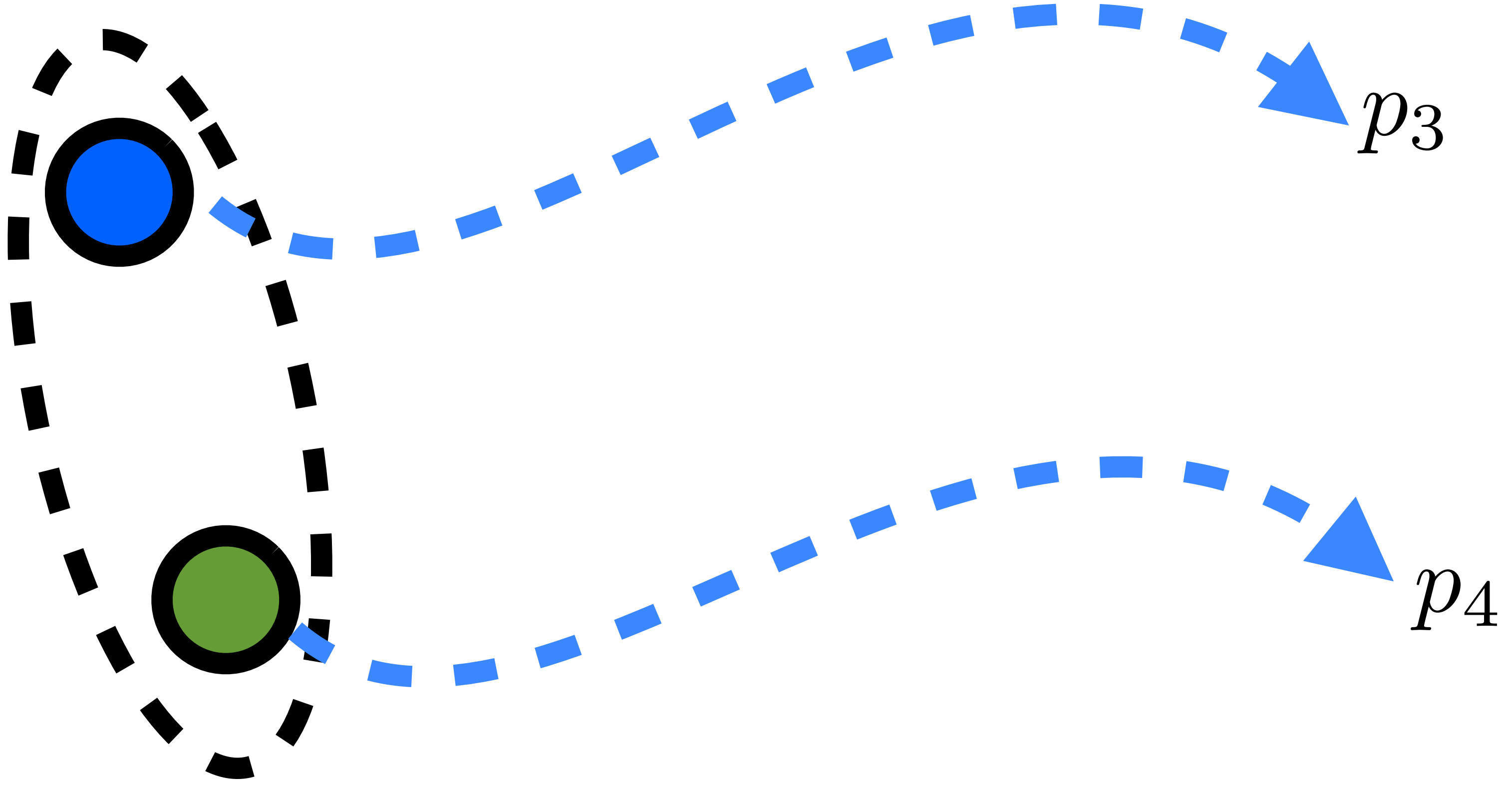




Primes
p_1
p_3
p_2
p_4

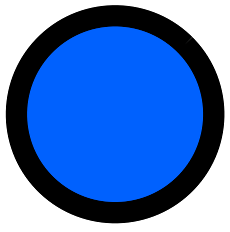
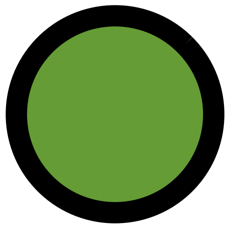


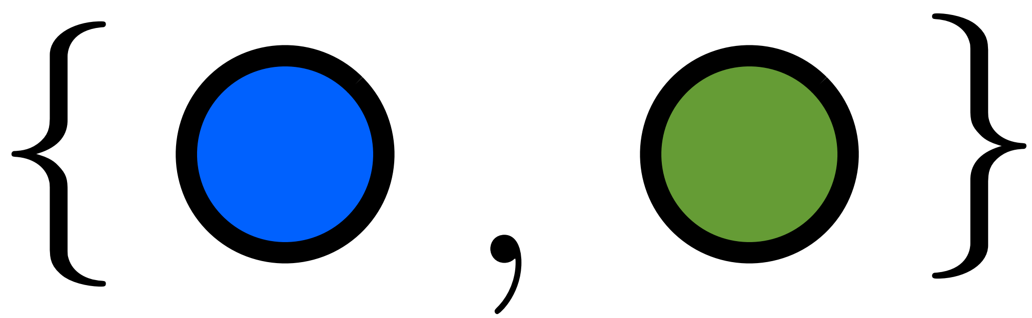
Primes
p_1
p_3
p_2
p_4


p_3
p_4

p_3 p_4

$$p_3 \times p_4$$

{  ,  }



$$A \subseteq B$$

$$[\![B]\!] \bmod [\![A]\!] = 0$$

$$A \cap B$$

$$\gcd(\llbracket A \rrbracket, \llbracket B \rrbracket)$$

$$\operatorname{lcm}(\llbracket A \rrbracket, \llbracket B \rrbracket)$$

$$A \cup B$$

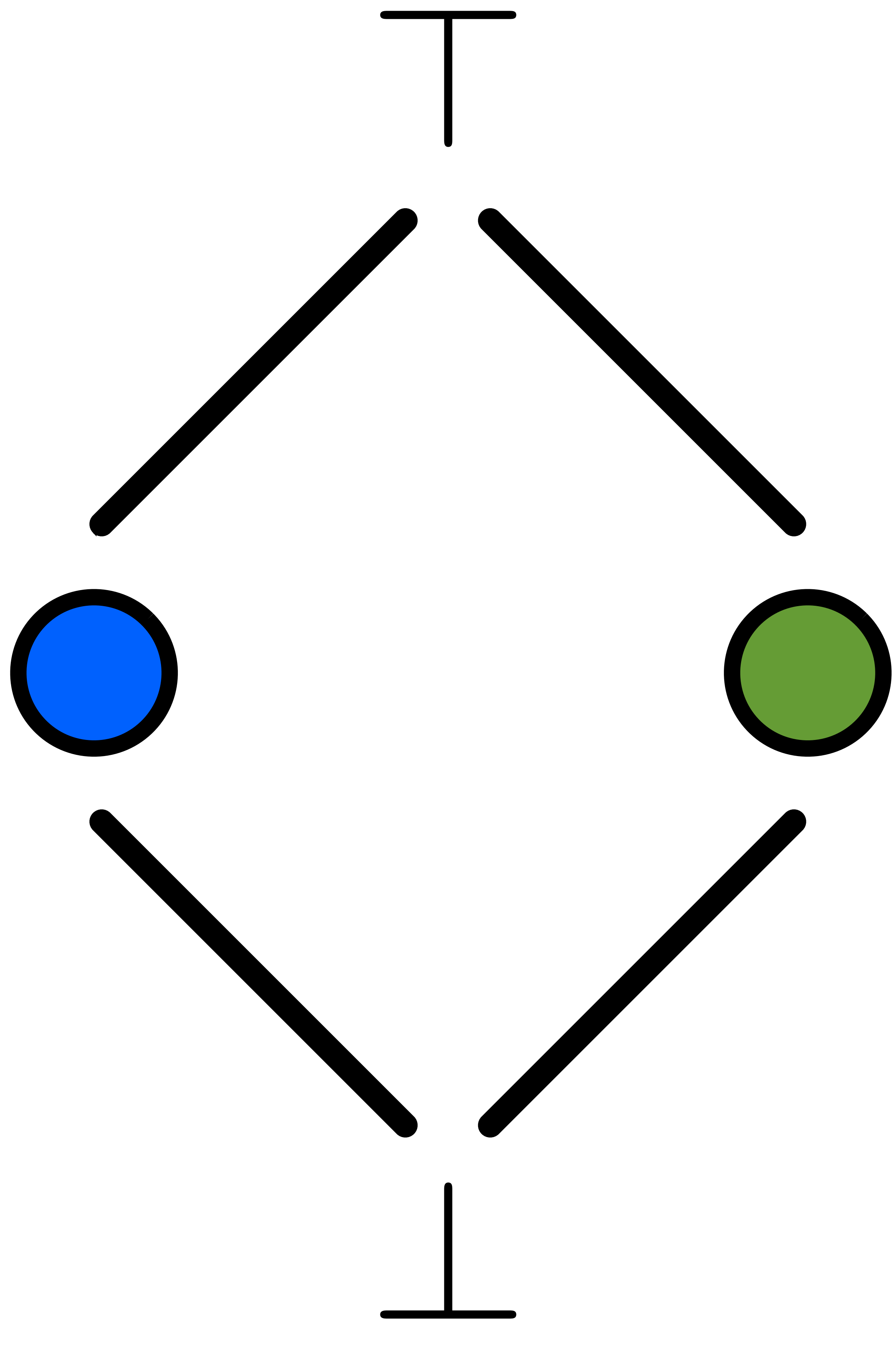
$$A \cup B$$

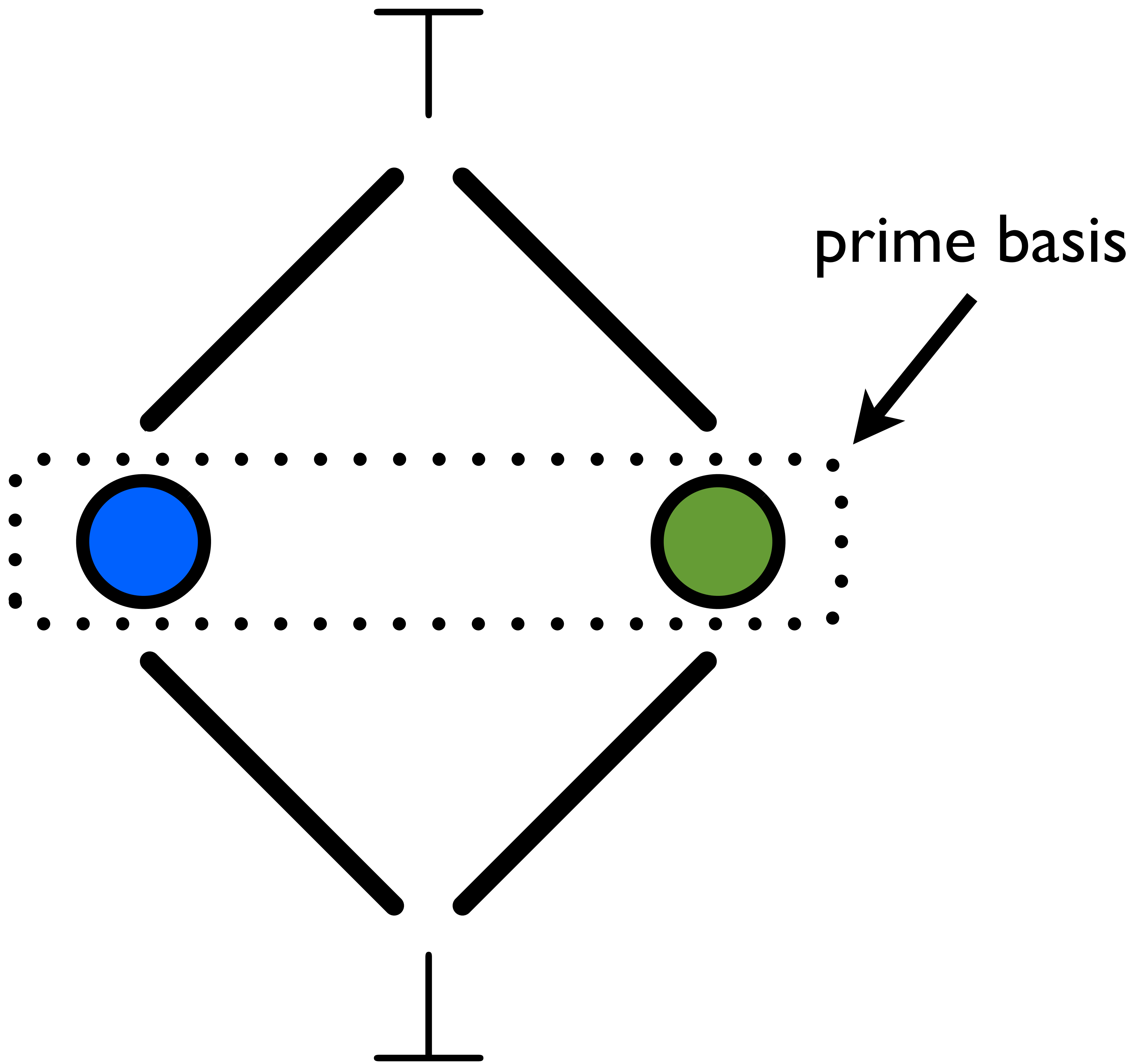
$$A - B$$

$$\llbracket A \rrbracket / \gcd(\llbracket A \rrbracket, \llbracket B \rrbracket)$$

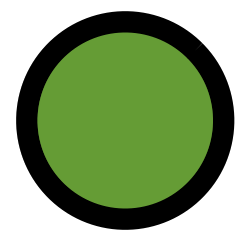
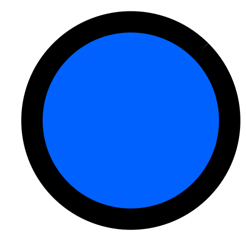
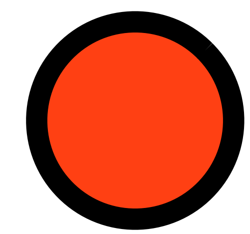
$$\subseteq$$

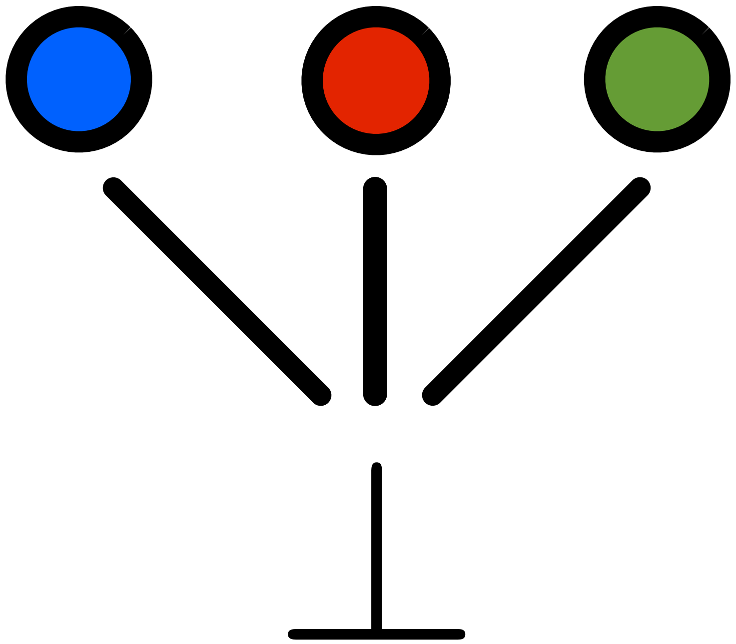
드

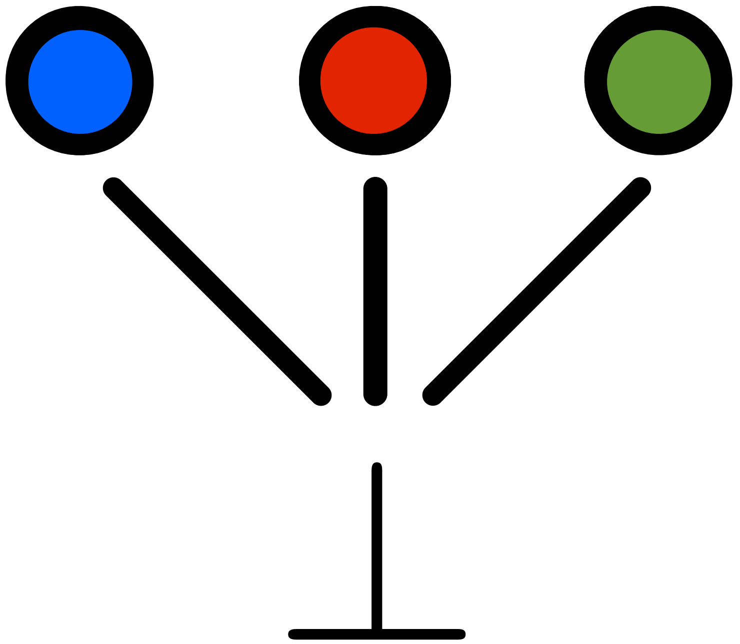


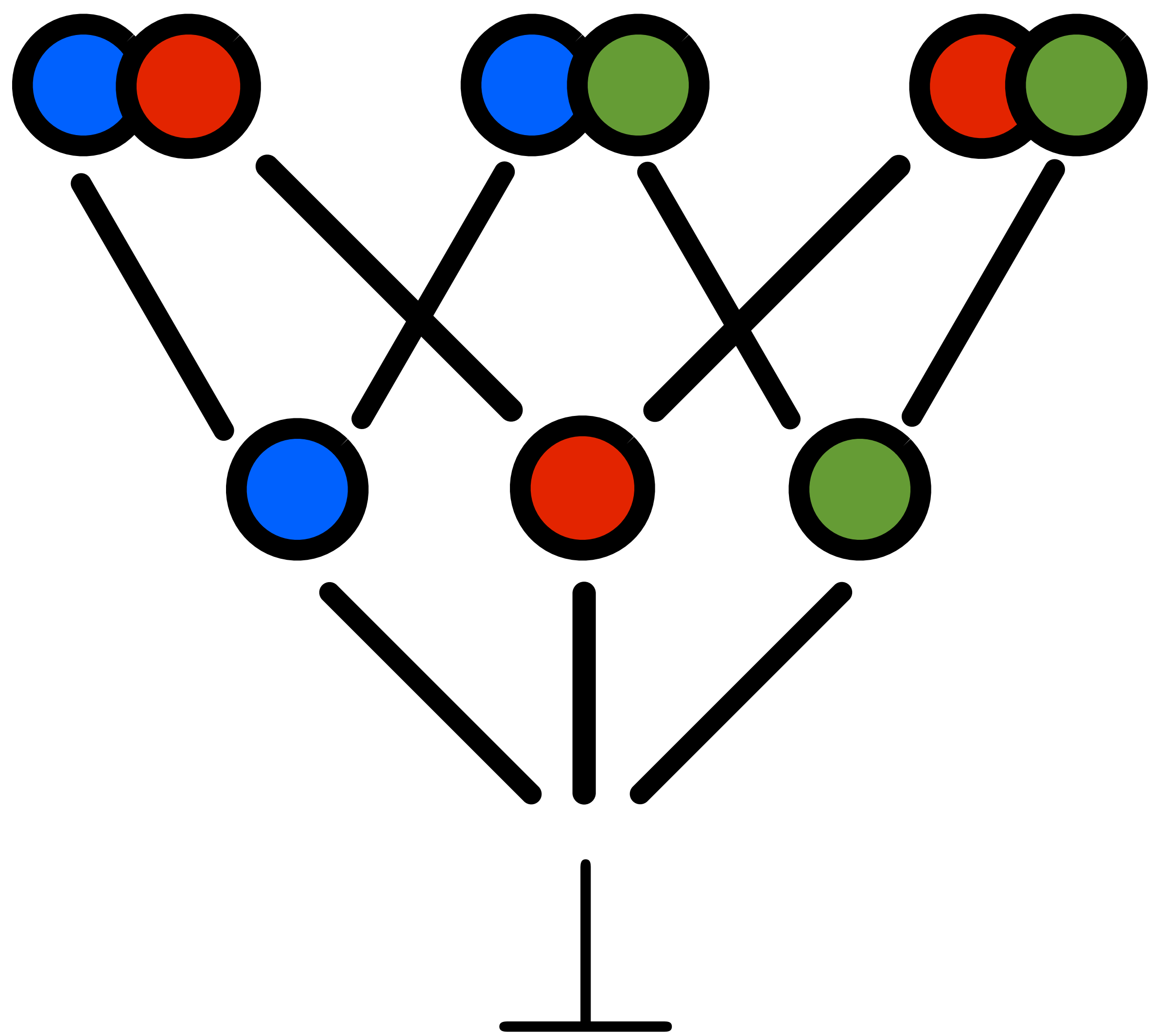

⊤
prime basis
⊥

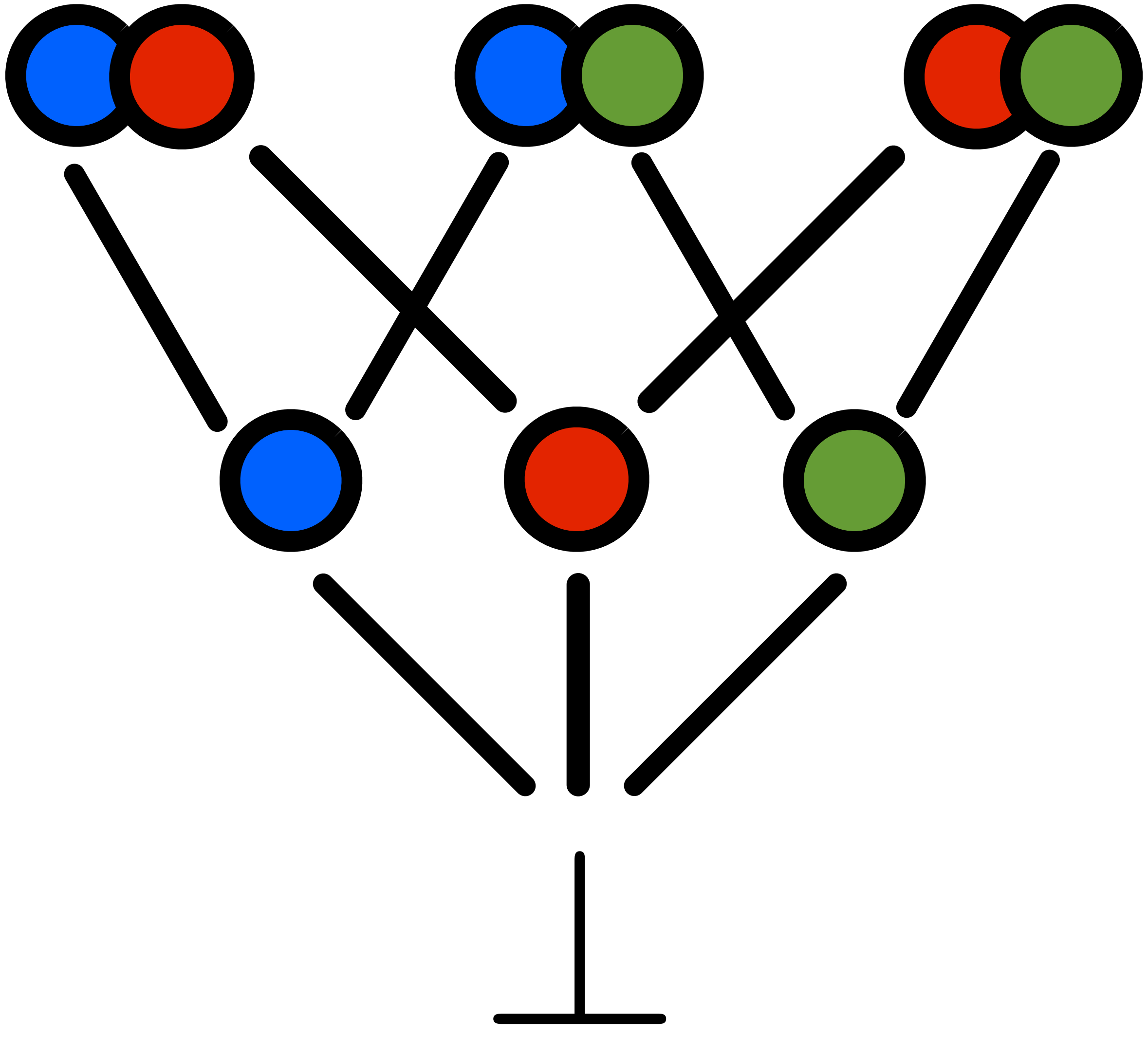
$$n = {p_1}^{m_1} {p_2}^{m_2} {p_3}^{m_3} \cdots$$

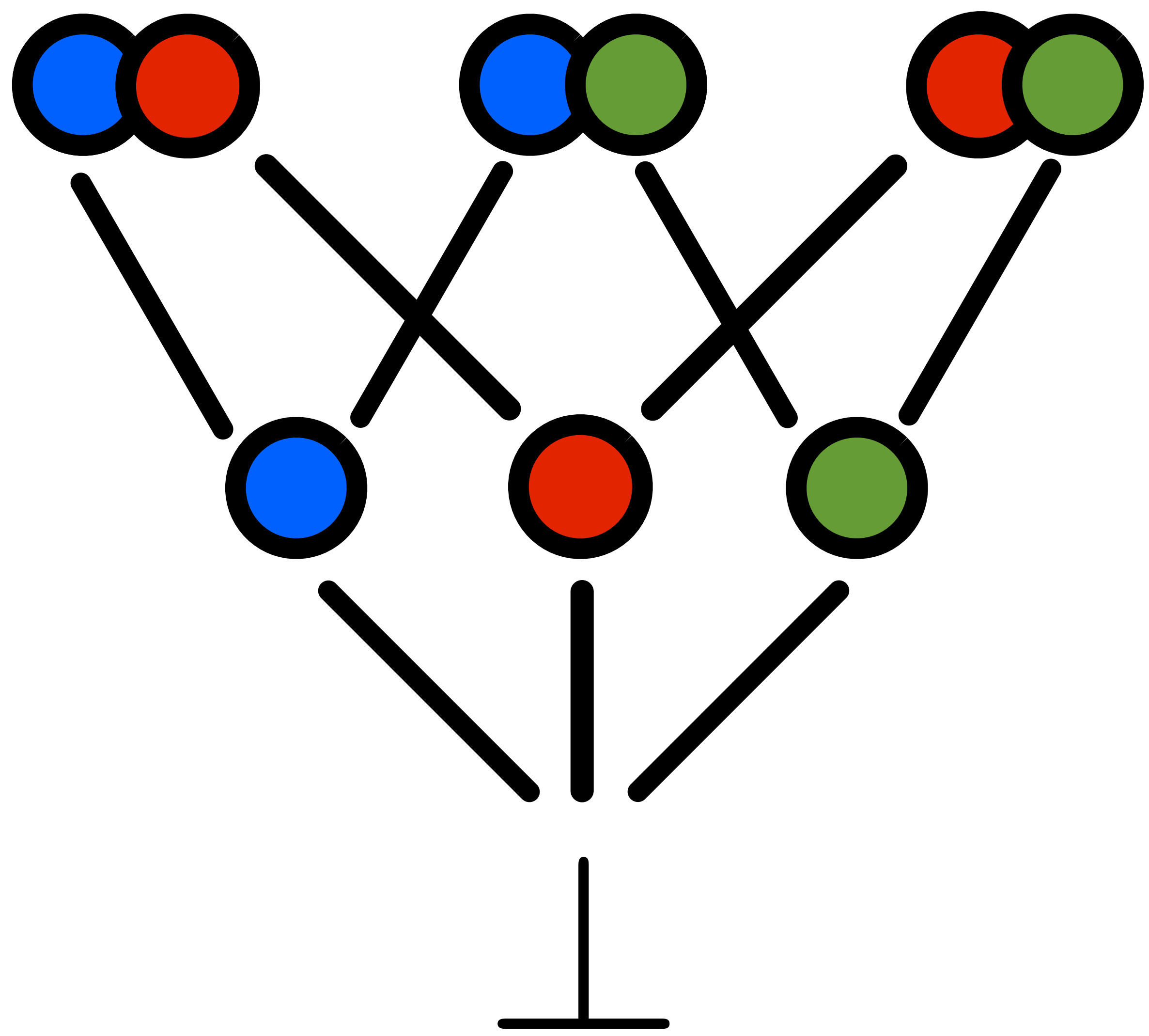
$n = \bigsqcup \{$  ,  ,  $\}$

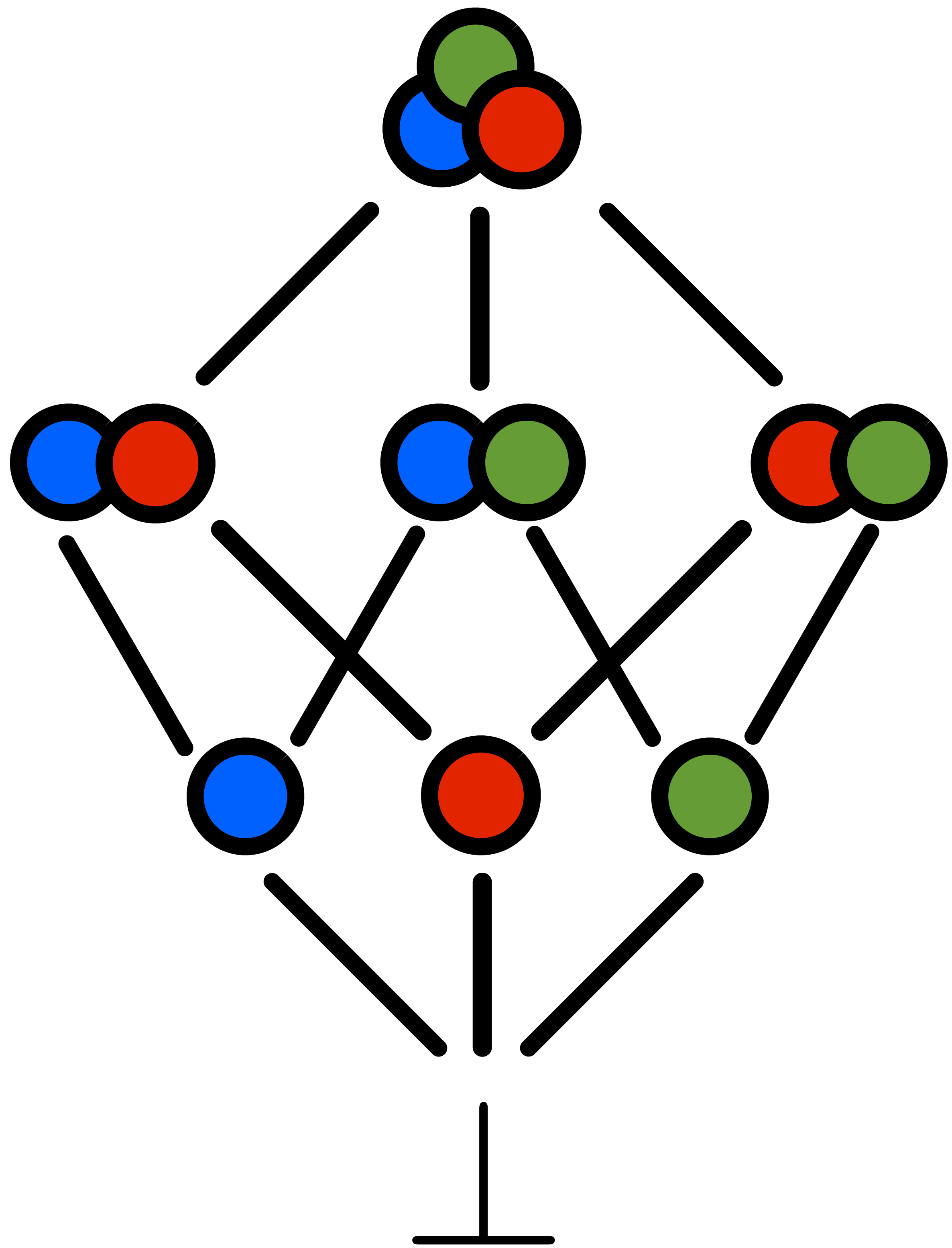


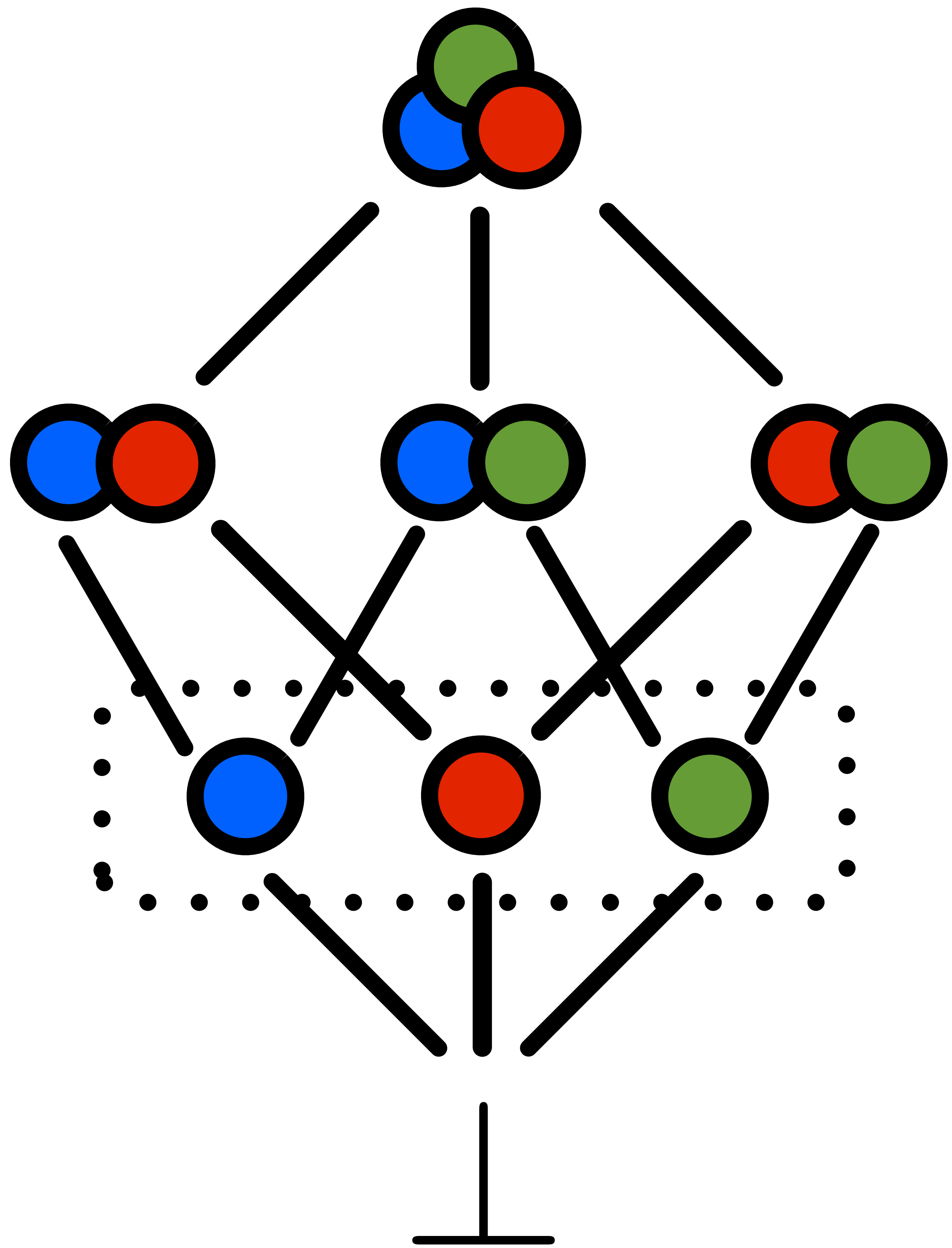


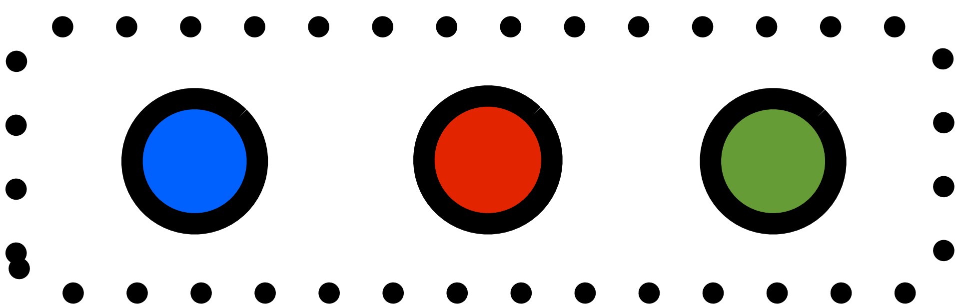


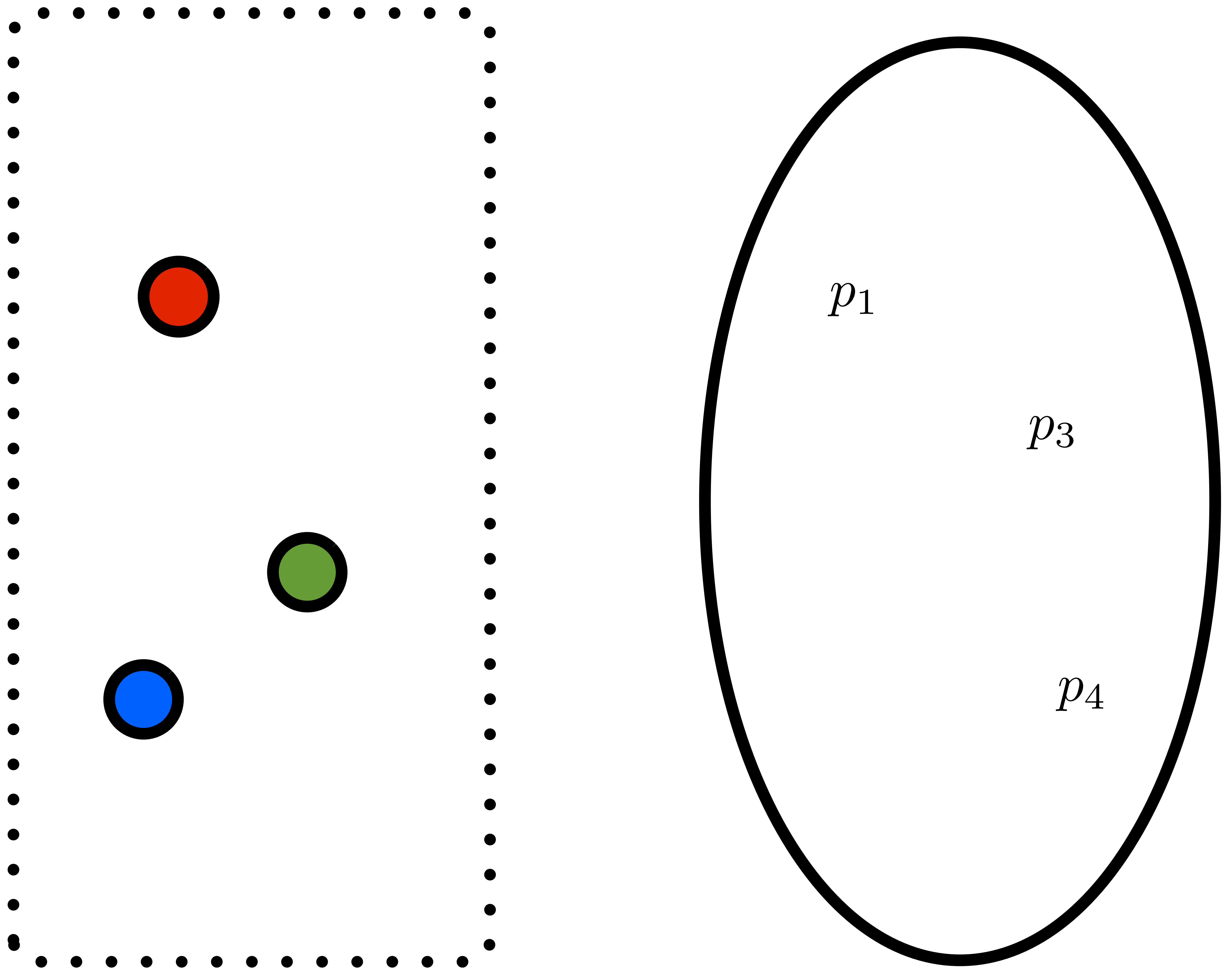


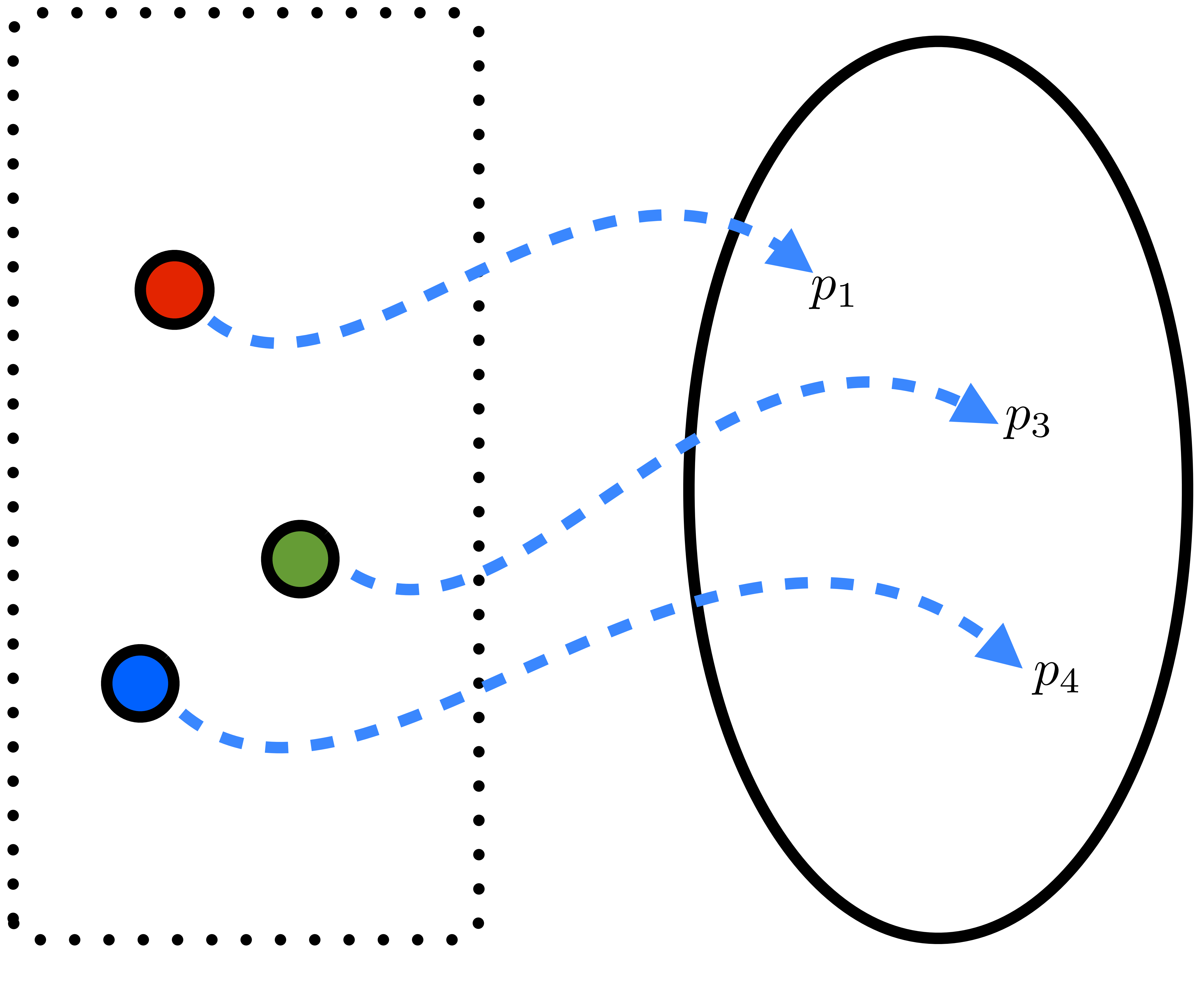









p_1
p_3
p_4


p_1
p_3
p_4

$$x \sqcup y$$

$$\operatorname{lcm}(\llbracket\, x \,\rrbracket, \llbracket\, y \,\rrbracket)$$

$$x \sqsubseteq y$$

$$\llbracket\, y \,\rrbracket \bmod \llbracket\, x \,\rrbracket = 0$$

$$x \sqcap y$$

$$\gcd(\llbracket x \rrbracket, \llbracket y \rrbracket)$$

But, does it work for CFA?

$$\hat{\sigma} : \widehat{Addr} \to \mathcal{P}(\widehat{Value})$$

$$\widehat{Addr}$$

$$\widehat{Value}$$

$$\{\hat{a}_1, \hat{a}_2\}$$

$$\{\hat{v}_1, \hat{v}_2\}$$

$$[\,\hat{a}_1 \mapsto \{\hat{v}_2\}]$$

$$[\,\hat{a}_2 \mapsto \{\hat{v}_1\}]$$

$$[\,\hat{a}_2 \mapsto \{\hat{v}_2\}]$$

$$[\,\hat{a}_1 \mapsto \{\hat{v}_1\}]$$

L_1 has a prime basis.

L_2 has a prime basis.

$L_1 \times L_2$ has a prime basis.

$L_1 + L_2$ has a prime basis.

$X \to L_2$ has a prime basis.

What else?

$$[\![\{a^n, b^m\}]\!]$$

$$[\![a]\!]^n\,[\![b]\!]^m$$

$$A \subseteq B$$

$$[\![B]\!] \bmod [\![A]\!] = 0$$

$$A \cup B$$

$$[\![A]\!] \times [\![B]\!]$$

$$\llbracket \langle a, b, c \rangle \rrbracket$$

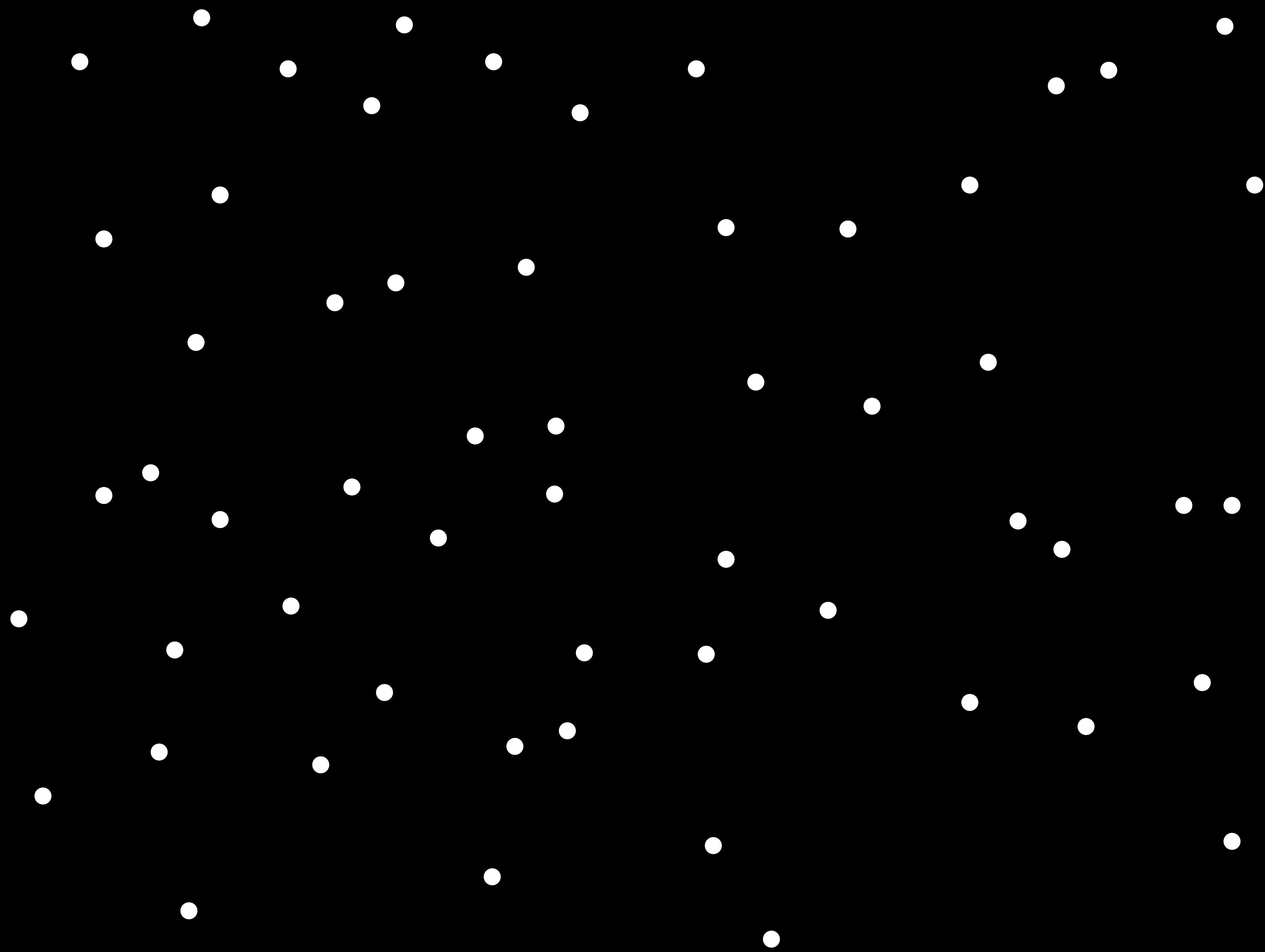
$$\begin{array}{ccc} [\![a]\!] & [\![b]\!] & [\![c]\!] \\ p_1 & p_2 & p_3 \end{array}$$

Wait a minute...

gcd is $O(n^2)$

mod is $O(n^2)$

How is this more efficient?



Flow sets are sparse.

99% of flow sets: < 5 values

Median flow set: 2 values

Primes are dense.

U

$$U \ln U$$

1,000,000 abstract values?

23 bit prime

Most flow sets fit in a word.

Most of the time, $n = 1$.

If not, great locality.

4-6%

2x 5x 8x 100x

Programming

is about

making choices.

3 E's

Elegance

Efficiency

Efficacy

Programmers:

Pick any two

Functional

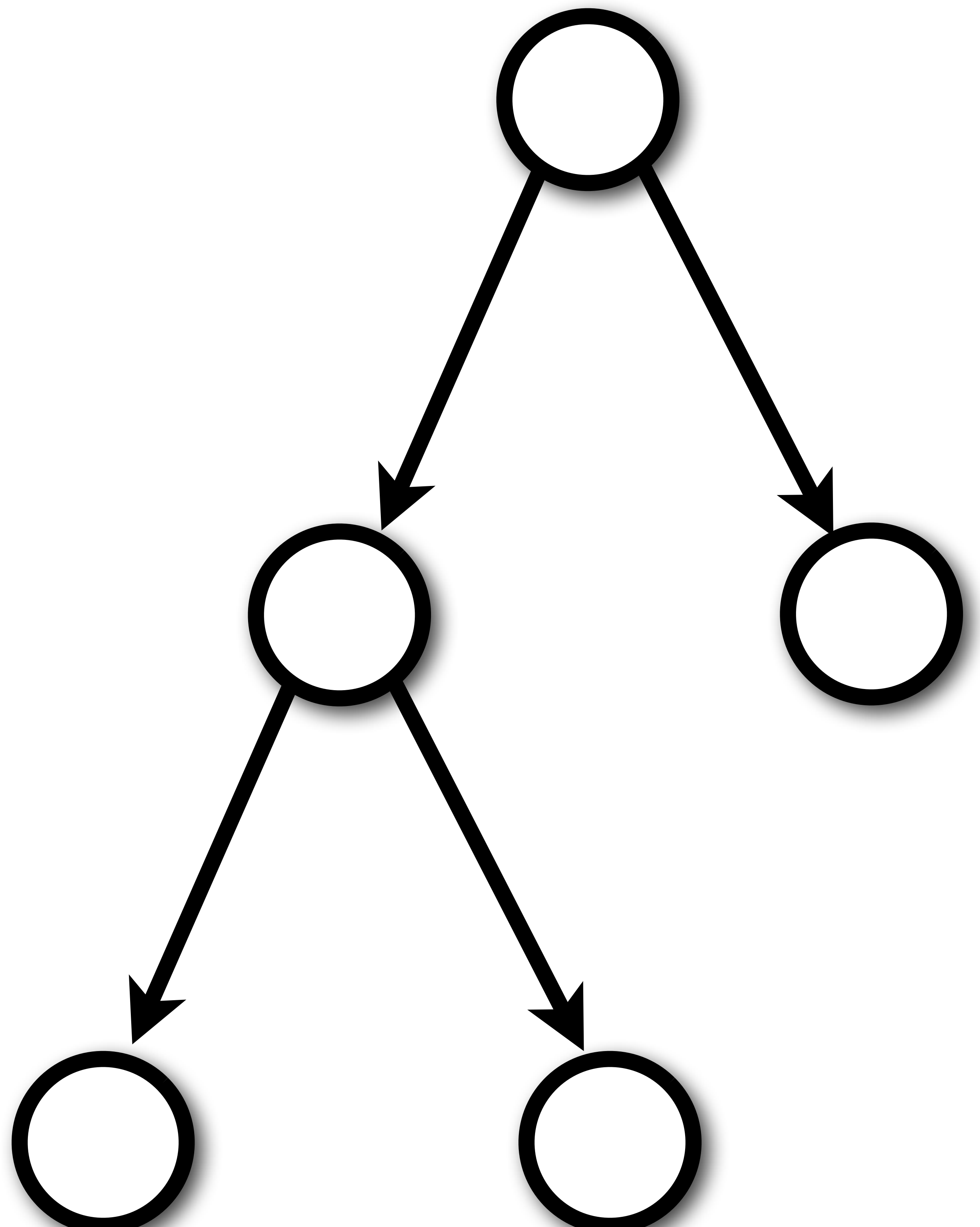
Programmers:

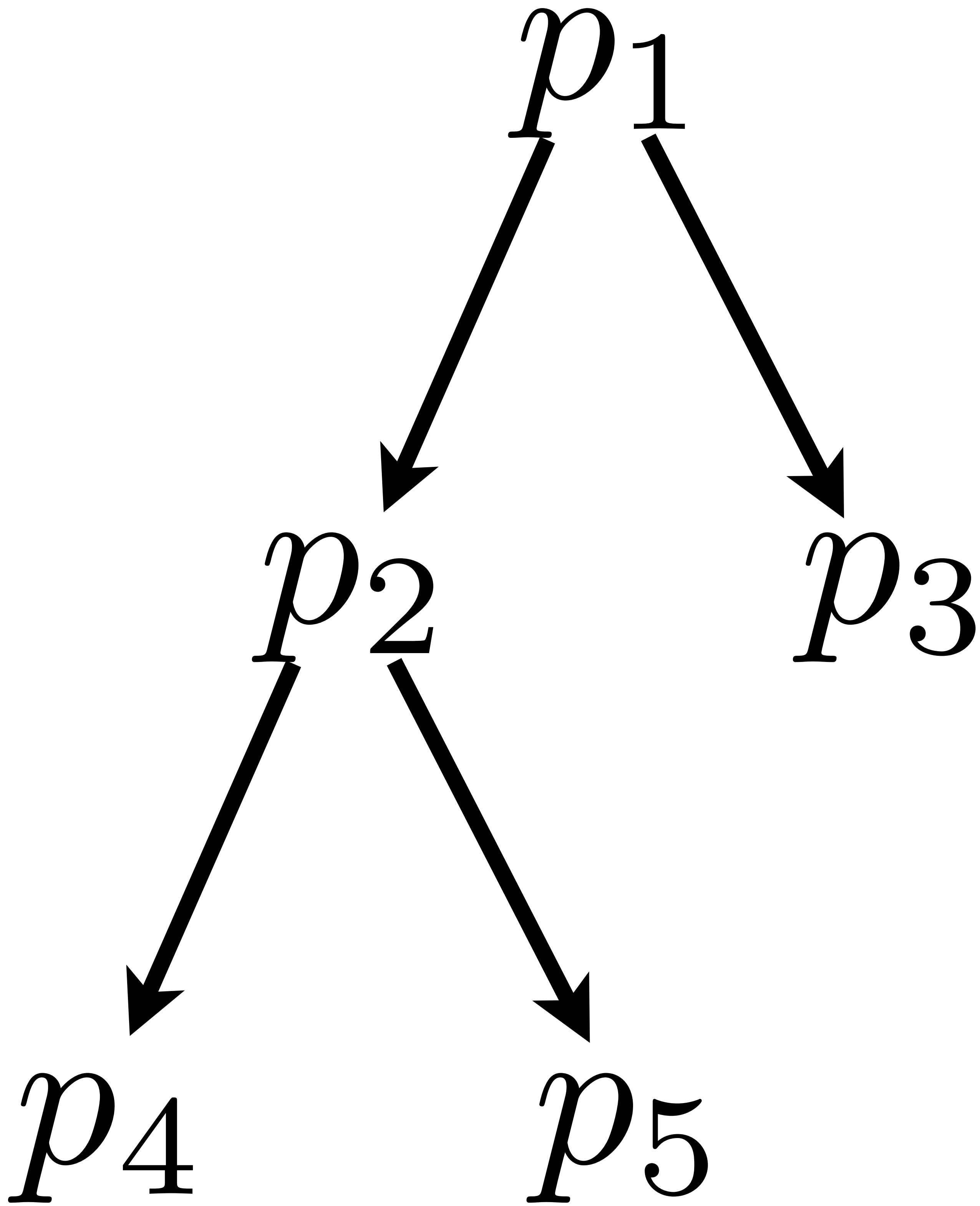
Pick any three

Questions?

Algebraic data types?

```
deriving (Hashable)
```




p_1
p_2
p_3
p_4
p_5

$$p_1 \; p_2 \; p_3 \; p_4 \, p_5$$