Approximate Modularity

Flavio Chierichetti
Sapienza University

Abhimanyu Das
Google MTV

Anirban Dasgupta
IIT Gandhinagar

Ravi Kumar
Google MTV

Modularity

A set function f is modular if it satisfies:

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \qquad \forall S, T \subseteq [n]$$

Modularity

A set function f is modular if it satisfies:

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \qquad \forall S, T \subseteq [n]$$

- Modular functions can be succinctly represented as linear functions: $f(S) = w_0 + \sum_{i \in S} w_i$
- It is thus very easy to optimize them

Modularity

A set function f is modular if it satisfies:

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \qquad \forall S, T \subseteq [n]$$

- Modular functions can be succinctly represented as linear functions: $f(S) = w_0 + \sum_{i \in S} w_i$
- It is thus very easy to optimize them

$$\arg\max_{S \subseteq [n]} f(S) = \{i \mid w_i > 0\}$$

Modularity

A set function f is modular if it satisfies:

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \qquad \forall S, T \subseteq [n]$$

- Modular functions can be succinctly represented as linear functions: $f(S) = w_0 + \sum_{i \in S} w_i$
- It is thus very easy to optimize them

$$\arg\max_{S \subseteq [n]} f(S) = \{i \mid w_i > 0\}$$

$$\arg\max_{S \in \binom{[n]}{k}} f(S) = \{w_{\pi(1)}, \ldots, w_{\pi(k)}\} \quad \left(w_{\pi(1)} \geq \cdots \geq w_{\pi(n)}\right)$$

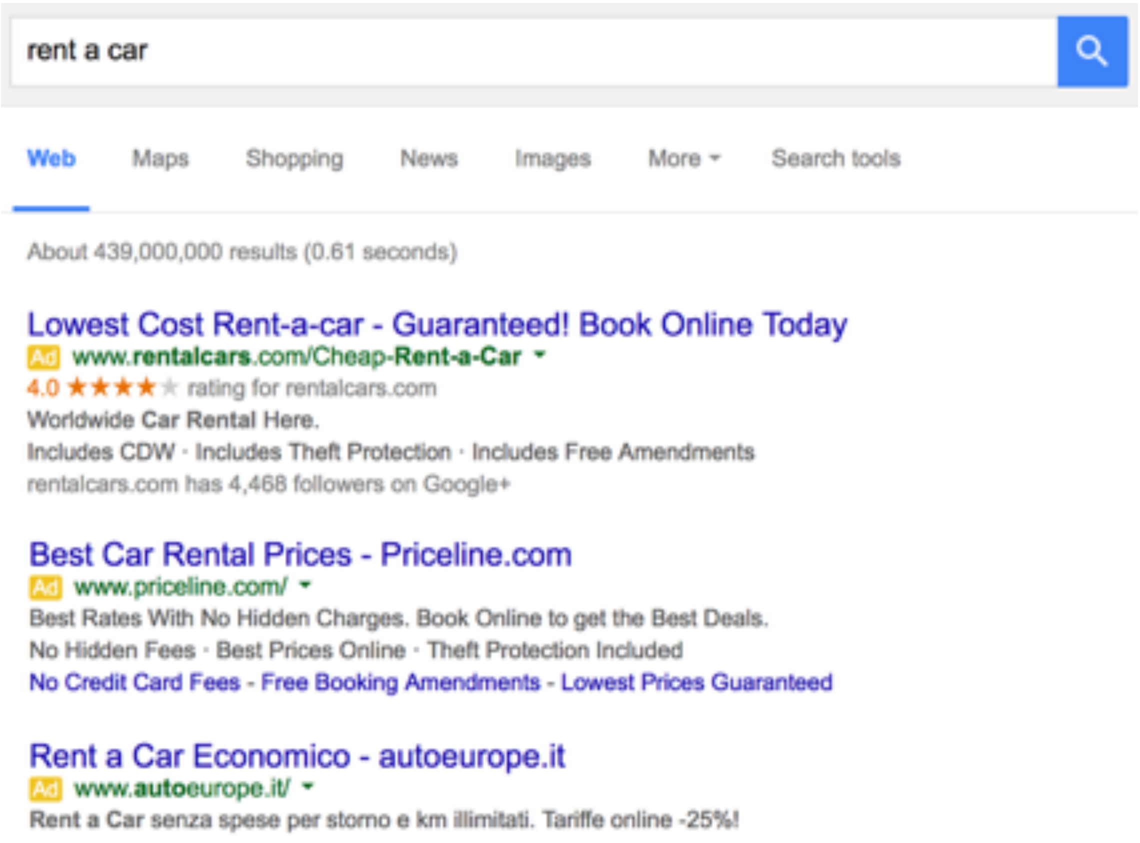
Modularity

A set function f is modular if it satisfies:

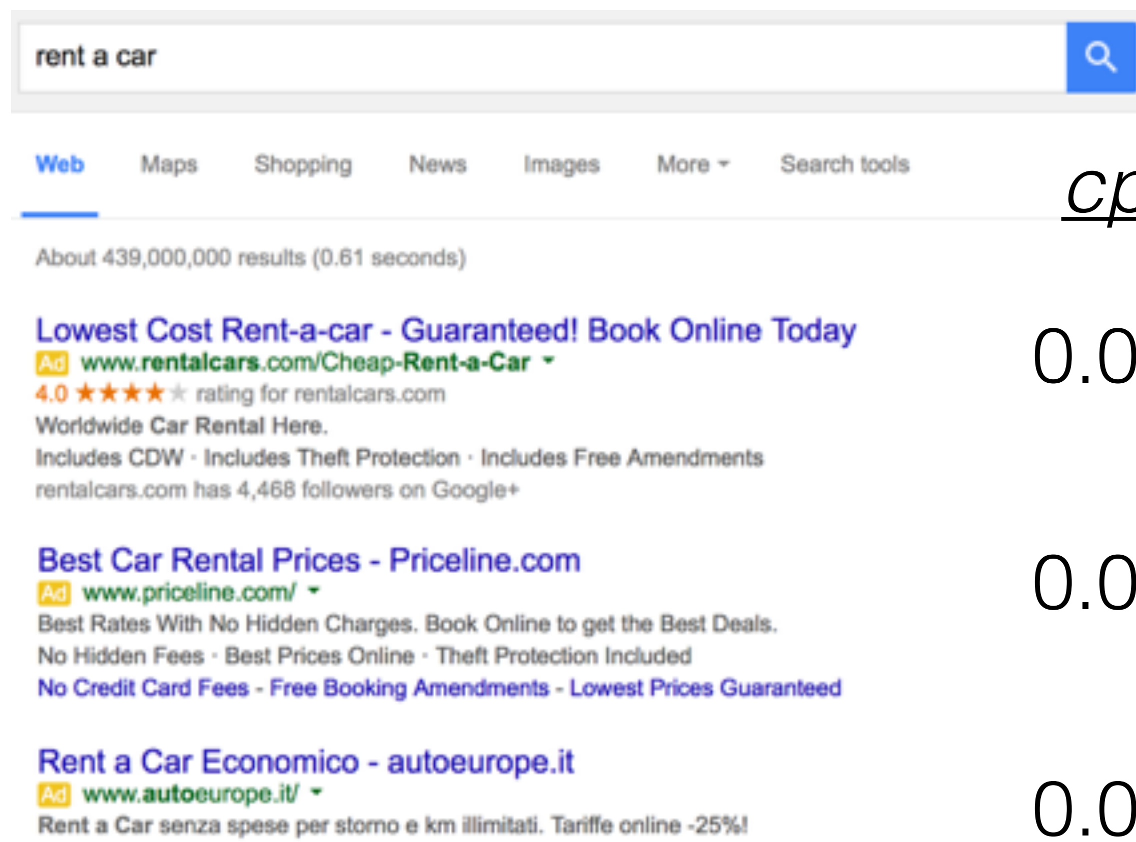
$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \qquad \forall S, T \subseteq [n]$$

- Modular functions can be succinctly represented as linear functions: $f(S) = w_0 + \sum_{i \in S} w_i$
- It is thus very easy to optimize them $(O(n))$
even though the search space can be huge $\left(O\left(2^n\right)\right)$

An example: Ads



An example: Ads



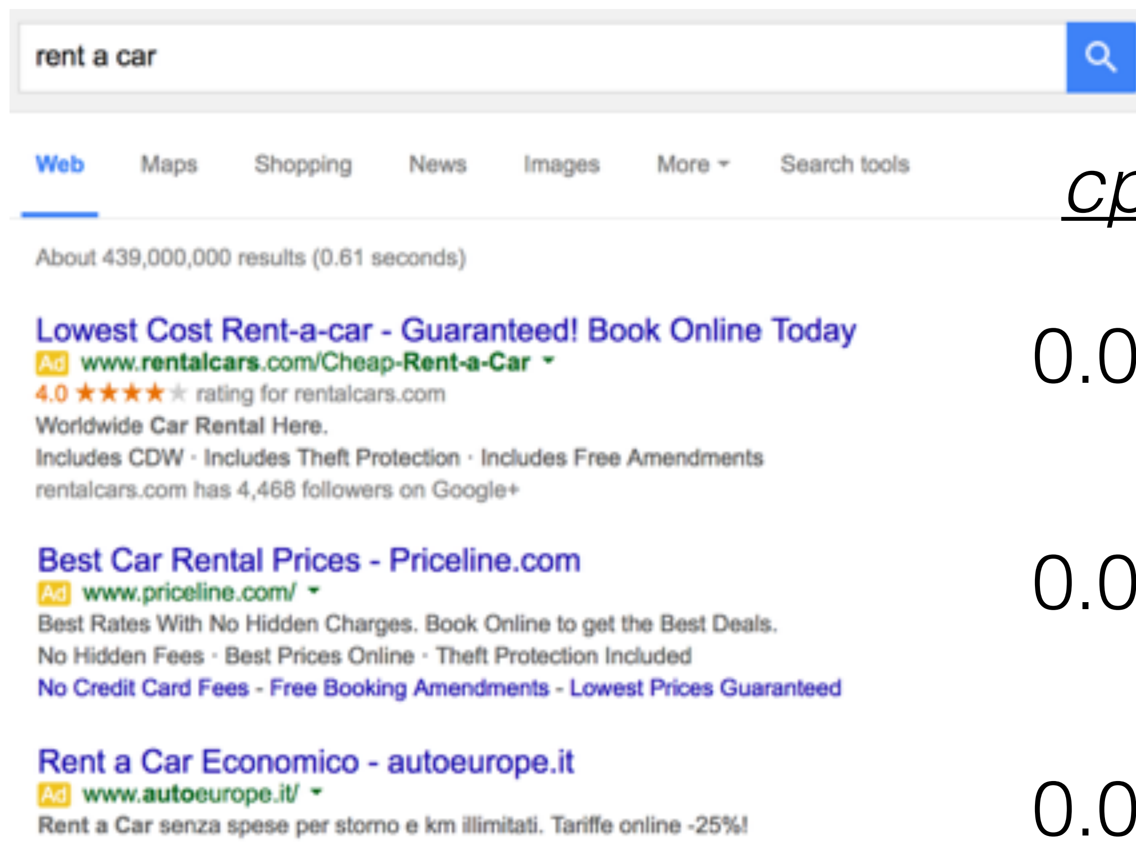
cpc

0.03€

0.05€

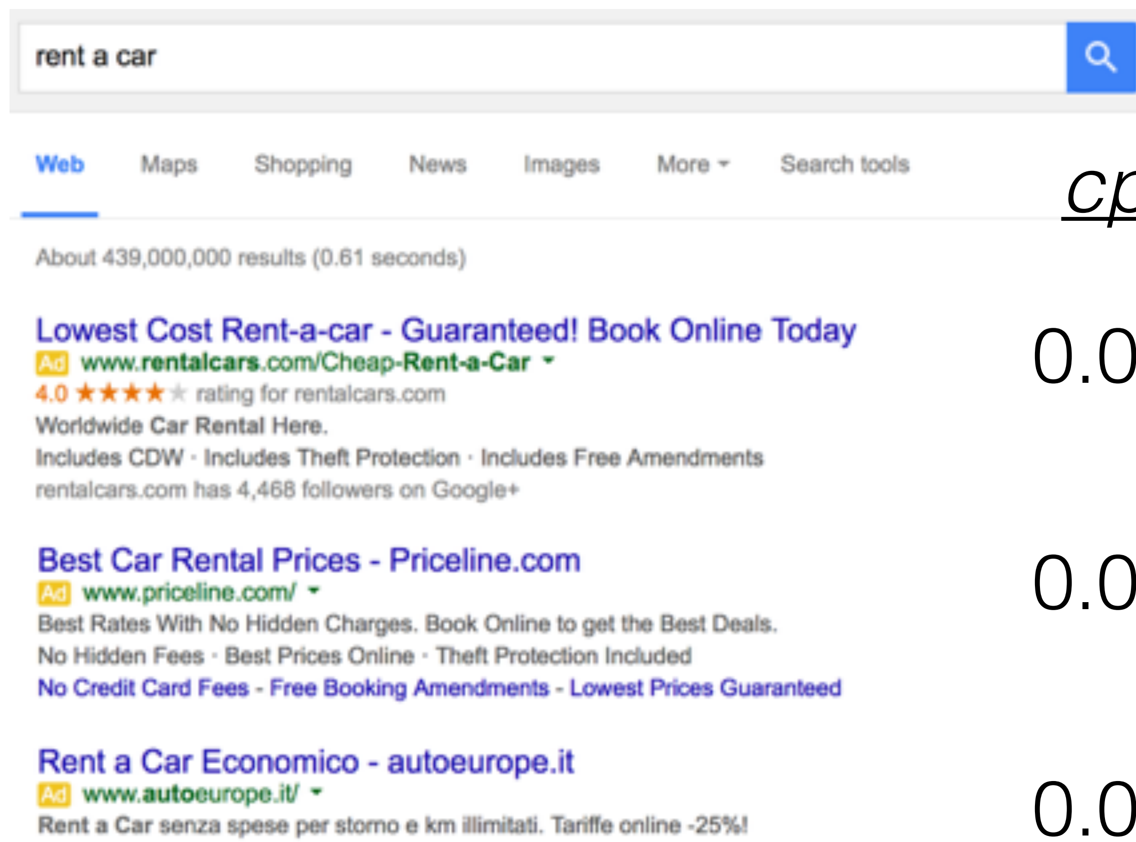
0.02€

Independent Clicks Model



cpc	*ctr*
0.03€	**5%**
0.05€	**2%**
0.02€	**3%**

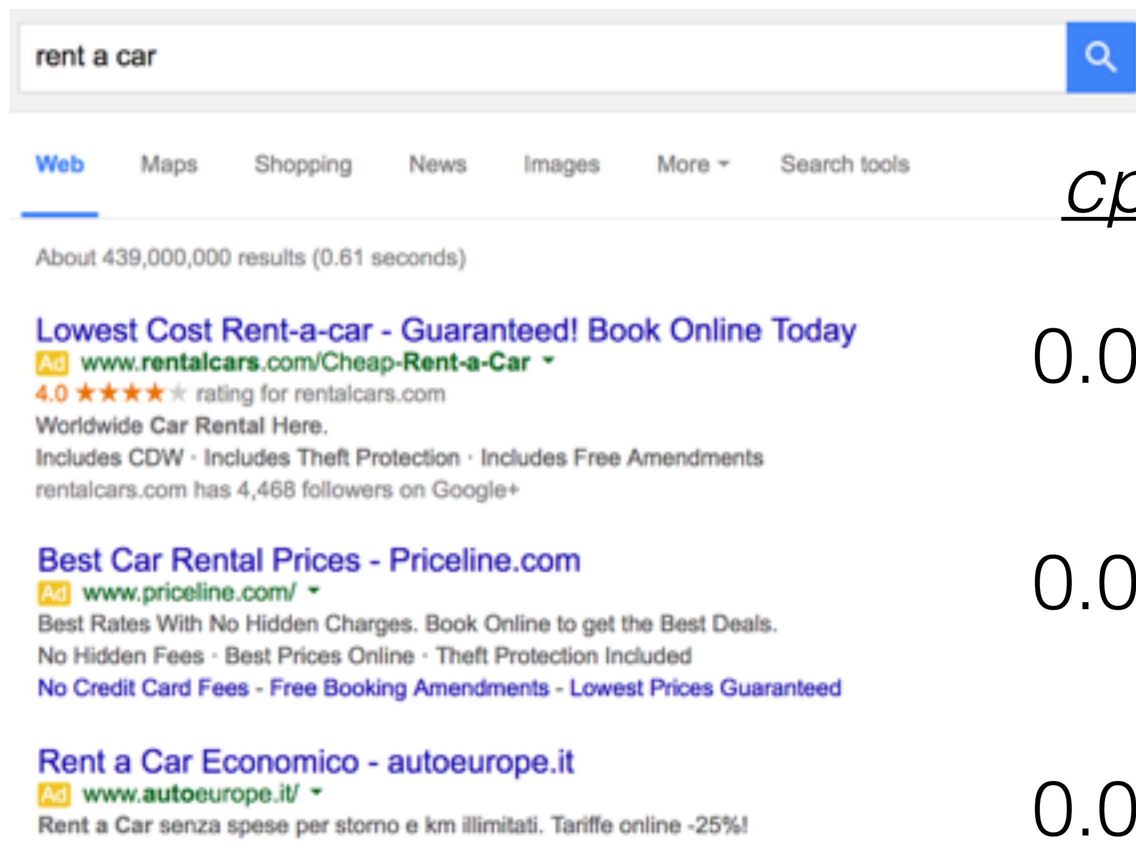
Independent Clicks Model



cpc	ctr
0.03€	**5%**
0.05€	**2%**
0.02€	**3%**

Select a set S of k ads that maximizes $f(S) = \sum_{a \in S} (\text{cpc}(a) \cdot \text{ctr}(a))$

Independent Clicks Model



cpc	*ctr*
0.03€	**5%**
0.05€	**2%**
0.02€	**3%**

Select a set S of k ads that maximizes $f(S) = \sum_{a \in S} (\text{cpc}(a) \cdot \text{ctr}(a))$

In practice, clicks are not independent, but…

Approximate Modularity

…*f* satisfies the ϵ-approximate modularity condition

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

for some small constant $\epsilon > 0$.

Approximate Modularity

…f satisfies the ϵ-approximate modularity condition

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

for some small constant $\epsilon > 0$.

$$\Big|\big(f(S) + f(T)\big) - \big(f(S \cup T) + f(S \cap T)\big)\Big| \leq \epsilon$$

Approximate Modularity

…*f* satisfies the ϵ-approximate modularity condition

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

for some small constant $\epsilon > 0$.

In our paper we analyze the *nature*, and the *approximability*, of approximately modular functions

Approximate Properties

The set function f is ϵ-approximately modular if

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

Approximate Properties

The set function *f* is ϵ-approximately modular if

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

Our definition resembles the *approximate linearity* definition of the *Borsuk-Ulam* theory of approximate functions on convex domains

If *D* is a convex domain, *f* is ϵ-approximately linear on *D* if

$$f(x) + f(y) = f(w) + f(z) \pm \epsilon$$

$$\forall x, y, w, z \in D \text{ such that } x + y = w + z$$

Approximate Properties

The set function *f* is ϵ-approximately modular if

$$f(S) + f(T) = f(U) + f(V) \pm O(\epsilon)$$
$$\text{whenever } S \cap T = U \cap V \wedge S \cup T = U \cup V$$

Our definition resembles the *approximate linearity* definition of the *Borsuk-Ulam* theory of approximate functions on convex domains

If *D* is a convex domain, *f* is ϵ-approximately linear on *D* if

$$f(x) + f(y) = f(w) + f(z) \pm \epsilon$$
$$\forall x, y, w, z \in D \text{ such that } x + y = w + z$$

Approximate Properties

The set function *f* is ϵ-approximately modular if

$$f(S) + f(T) = f(U) + f(V) \pm O(\epsilon)$$
$$\text{whenever } S \cap T = U \cap V \wedge S \cup T = U \cup V$$

Our definition resembles the *approximate linearity* definition of the *Borsuk-Ulam* theory of approximate functions on convex domains

If *D* is a convex domain, *f* is ϵ-approximately linear on *D* if

$$f(x) + f(y) = f(w) + f(z) \pm \epsilon$$
$$\forall x, y, w, z \in D \text{ such that } x + y = w + z$$

One of the original *B-U* theory's goals is to *approximate* an **approximately linear function** with a **truly linear function**

Approximate Modularity

The set function f is ϵ-approximately modular if

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

- In our case, the domain is composed of the vertices of the hypercube - so it is not closed, nor convex.

Approximate Modularity

The set function f is ϵ-approximately modular if

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon \qquad \forall S, T \subseteq [n]$$

- In our case, the domain is composed of the vertices of the hypercube - so it is not closed, nor convex.
  - How well can we optimize these functions in polynomial time?
  - How “close” are these functions to modularity?

Our Results

Polynomial-time Optimization

- Let *f* be an ϵ-approximately modular function on *[n]*. We show that:
  - there exists an algorithm that performs $O(n^2 \log n)$ queries to *f*, and returns a **modular** function *g* such that $\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \sqrt{n})$

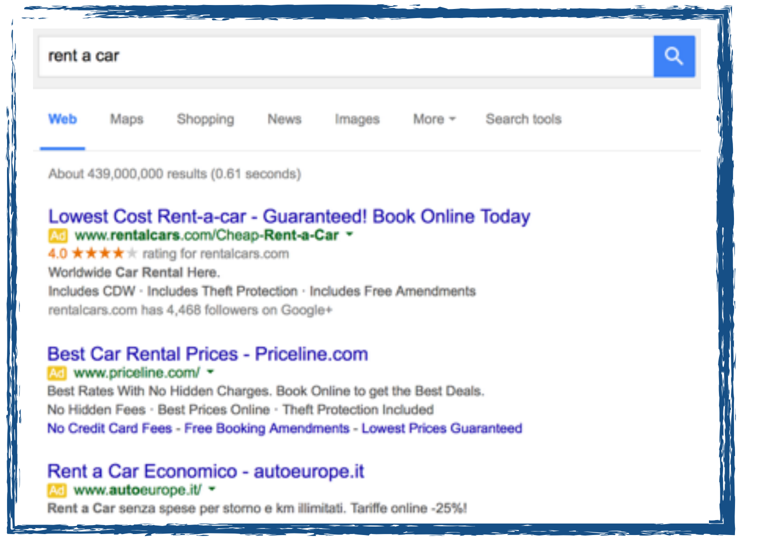
Our Results

Polynomial-time Optimization

- Let *f* be an ϵ-approximately modular function on *[n]*. We show that:

    - there exists an algorithm that performs $O(n^2 \log n)$ queries to *f*, and returns a **modular** function *g* such that $\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \sqrt{n})$

Thus we can, say, approximate our ads problem to an additive $O(\epsilon\sqrt{n})$ error.



Our Results

Polynomial-time Optimization

- Let *f* be an ϵ-approximately modular function on *[n]*. We show that:

  - there exists an algorithm that performs $O(n^2 \log n)$ queries to *f*, and returns a **modular** function *g* such that $\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \sqrt{n})$

  - an algorithm performing $n^{O(1)}$ queries to *f*, **cannot additively approximate** *f* to better than $\Omega(\epsilon \cdot \sqrt{n/\log n})$

Our Results

Polynomial-time Optimization

- Let *f* be an ϵ-approximately modular function on *[n]*. We show that:

  - there exists an algorithm that performs $O(n^2 \log n)$ queries to *f*, and returns a **modular** function *g* such that $\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \sqrt{n})$

  - an algorithm performing $n^{O(1)}$ queries to *f*, **cannot additively approximate** *f* (nor the maximum value of *f)* to better than $\Omega(\epsilon \cdot \sqrt{n/\log n})$

Our Results

Closeness to Modularity

- Let *f* be an ϵ-approximately modular function on *[n]*. We show that:

  - there exists a modular function *g* such that

$$\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \log n)$$

Our Results

Closeness to Modularity

- Let *f* be an ϵ-approximately modular function on *[n]*. We show that:

  - there exists a modular function *g* such that

$$\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \log n)$$

Thus, if we drop the poly-time reconstruction requirement, we can approximate *f* with a modular *g* in an ***exponentially*** better way

Poly-time Approximation

- Our goal is to "quickly" produce a modular function

$$g(S) = w_0 + \sum_{i \in S} w_i$$

that is close to f.

Poly-time Approximation

- Our goal is to “quickly” produce a modular function
$$g(S) = w_0 + \sum_{i \in S} w_i$$
that is close to f .

- That is, our algorithm has to choose $w_0, w_1, \ldots, w_n$, while querying f at most $n^{O(1)}$ times, so to guarantee that $|g(S) - f(S)|$ is small for each $S \subseteq [n]$

Poly-time Approximation

- First, we choose $z = f(\varnothing)$
- Then, we choose:

$$w_i = \text{avg}_{k \in [n]} \text{avg}_{S \in \binom{[n]-\{i\}}{k-1}} \Big(f(S \cup \{i\}) - f(S) \Big)$$

Poly-time Approximation

- First, we choose $z = f(\varnothing)$
- Then, we choose:

$$w_i = \text{avg}_{k\in[n]}\text{avg}_{S\in\binom{[n]-\{i\}}{k-1}}\Big(f(S\cup\{i\}) - f(S)\Big)$$

The average is taken over exponentially many values of f…

Poly-time Approximation

- First, we choose $z = f(\varnothing)$
- Then, we choose:

$$w_i = \mathrm{avg}_{k \in [n]} \mathrm{avg}_{S \in \binom{[n]-\{i\}}{k-1}} \Big(f(S \cup \{i\}) - f(S) \Big)$$

The average is taken over exponentially many values of f…

…but the approximate modularity condition allows us to approximate w_i with only $O(n \log n)$ many random samples

Poly-time Approximation

- First, we choose $z = f(\varnothing)$

- Then, we choose:

$$w_i = \mathrm{avg}_{k \in [n]} \mathrm{avg}_{S \in \binom{[n]-\{i\}}{k-1}} \Big(f(S \cup \{i\}) - f(S) \Big)$$

- With these weights, we can prove that

$$|f(S) - g(S)| \leq 4 \cdot \epsilon \cdot \sqrt{\min(|S|, n - |S|)} \qquad \forall S \subseteq [n]$$

by studying the duals of a number of LPs

Poly-time Approximation

- First, we choose $z = f(\varnothing)$

- Then, we choose:

$$w_i = \text{avg}_{k \in [n]} \text{avg}_{S \in \binom{[n]-\{i\}}{k-1}} \Big(f(S \cup \{i\}) - f(S) \Big)$$

- With these weights, we can prove that

$$|f(S) - g(S)| \leq 4 \cdot \epsilon \cdot \sqrt{\min(|S|, n - |S|)} \qquad \forall S \subseteq [n]$$

by studying the duals of a number of LPs

Observe that the weights, and thus g, are linear combinations of the values of f

Poly-time Approximation

- First, we choose $z = f(\varnothing)$

- Then, we choose:

$$w_i = \text{avg}_{k \in [n]} \text{avg}_{S \in \binom{[n]-\{i\}}{k-1}} \Big(f(S \cup \{i\}) - f(S) \Big)$$

- With these weights, we can prove that

$$|f(S) - g(S)| \leq 4 \cdot \epsilon \cdot \sqrt{\min(|S|, n - |S|)} \qquad \forall S \subseteq [n]$$

by studying the duals of a number of LPs:

$$\begin{cases} \max g(S) - f(S) \text{ subject to} \\ \forall \{X, Y\} \in \binom{2^{[n]}}{2} : \quad -\epsilon \leq f(X \cup Y) + f(X \cap Y) - f(X) - f(Y) \leq \epsilon \end{cases}$$

Poly-time Approximation

- Our algorithm produces a modular function which approximates *f* to within an additive $O\left(\epsilon \cdot \sqrt{n}\right)$ value with a total of $O\left(n^2 \log n\right)$ queries to *f*.

Inapproximability

- We also show that no algorithm querying an ϵ-approximately modular *f* at most *q* times can distinguish WHP whether:
  - *f* is the constant 0 function; or
  - the maximum value of *f* is at least $\Omega\left(\epsilon \cdot \sqrt{n / \log q}\right)$
- Hence, the approximation of our poly-time algorithm is almost optimal.

Our two Questions

- How well can we optimize ϵ-a.m. functions in polynomial time?
- How "close" are ϵ-a.m. functions to modularity?

Our two Questions

- How well can we optimize ϵ-a.m. functions in polynomial time? $\approx \epsilon \cdot \sqrt{n}$ (by a modular approximation)
- How "close" are ϵ-a.m. functions to modularity?

Closeness to Modularity

Let *f* be an ϵ-approximately modular function on *[n]*.
Then, there exists a modular function *g* such that

$$\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \log n)$$

The Rough Idea

sets of f-*maximum* value

S

S'

S''

S'''

$\ldots$

sets of f-*minimum* value

T

T'

T''

T'''

$\ldots$

The Rough Idea

sets of f-*maximum* value

S

S'

S''

S'''

$\ldots$

$$\left(f(S) = f(S') = \cdots = \max_X f(X) \right)$$

sets of f-*minimum* value

T

T'

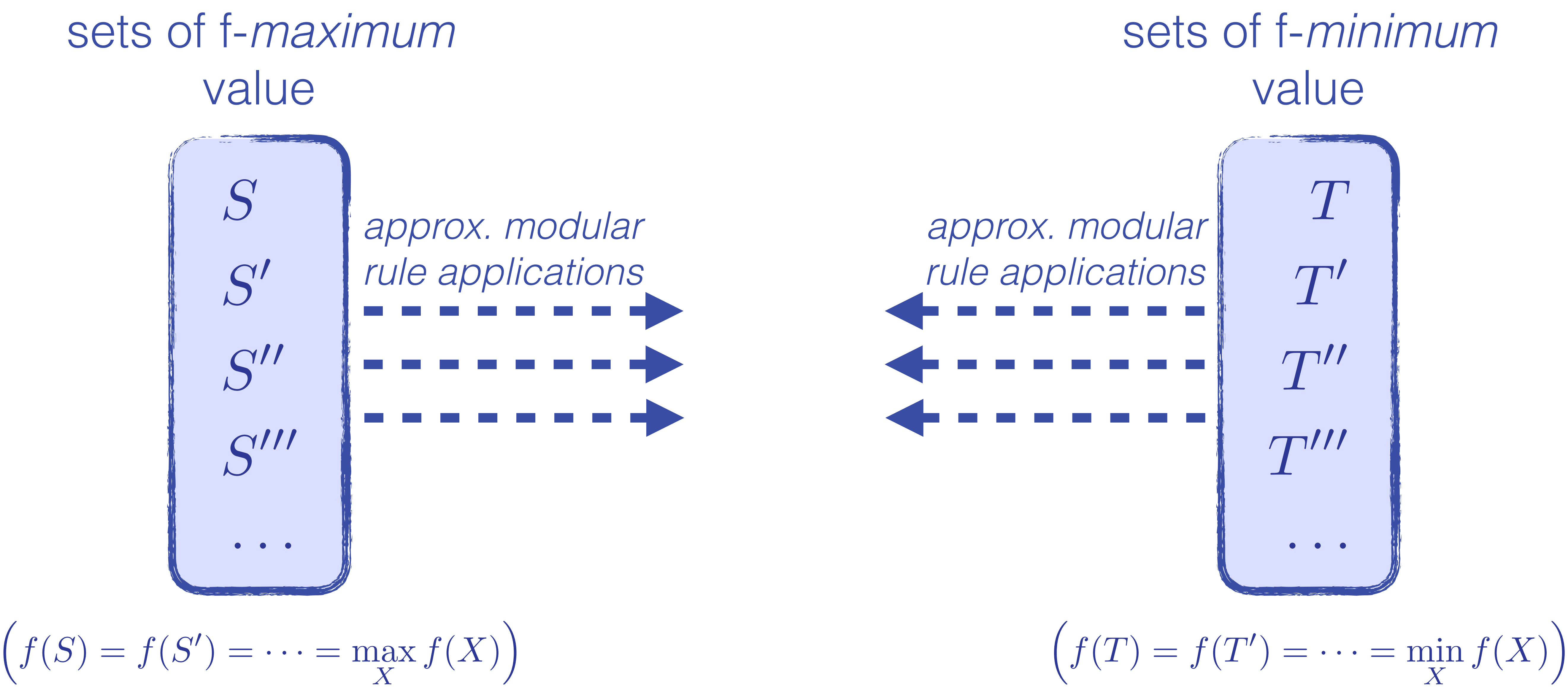
T''

T'''

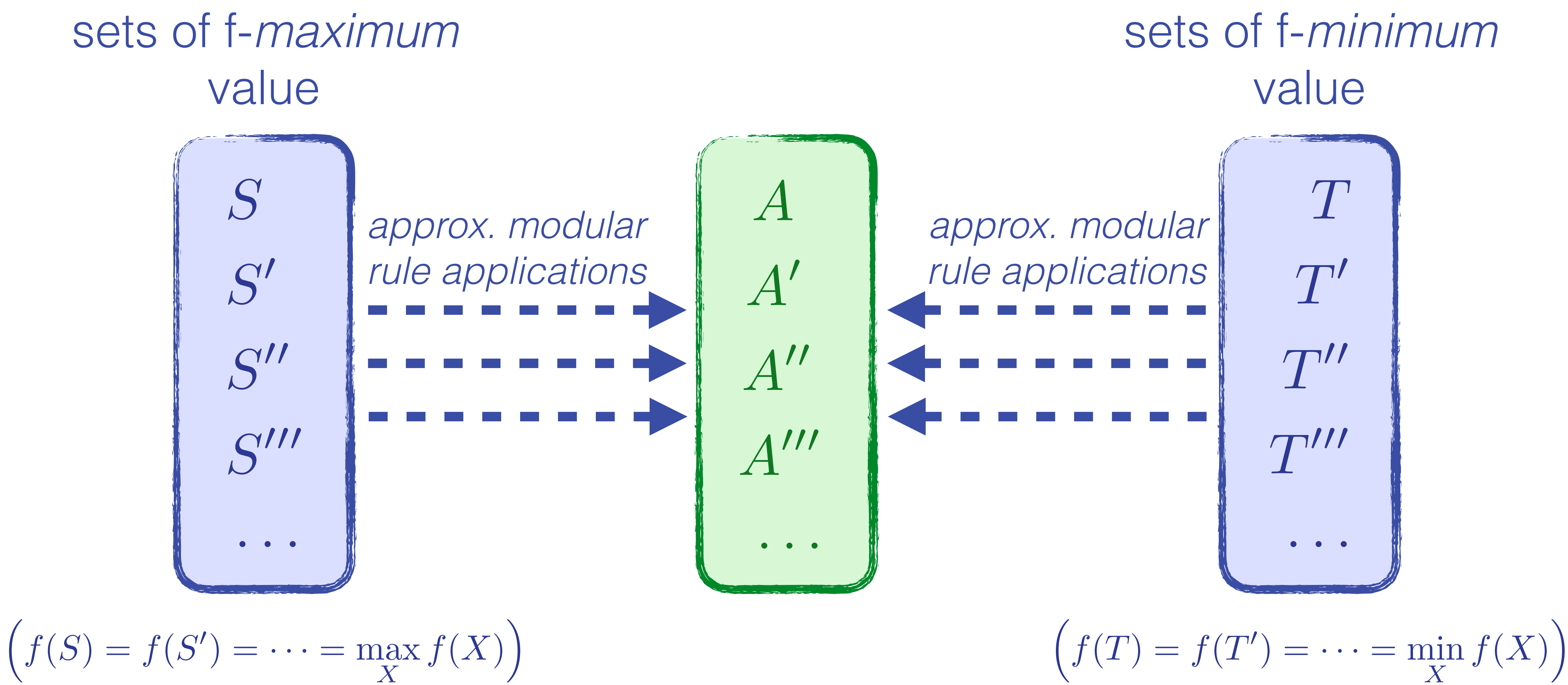
$\ldots$

$$\left(f(T) = f(T') = \cdots = \min_X f(X) \right)$$

The Rough Idea



The Rough Idea



The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*

	$\varnothing$	$\{1\}$	$\{2\}$	$\{1,2\}$
f	0	1	2	2

$\epsilon = 1$

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f*.

	$\varnothing$	$\{1\}$	$\{2\}$	$\{1,2\}$
f	0	1	2	2

$\epsilon = 1$

$$g(S) = w_0 + \sum_{i \in S} w_i$$

$$(w_0 = 1/4, w_1 = 1/2, w_2 = 3/2)$$

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*

	$\varnothing$	$\{1\}$	$\{2\}$	$\{1,2\}$
f	0	1	2	2
g	$1/4$	$3/4$	$7/4$	$9/4$

$\epsilon = 1$

$$g(S) = w_0 + \sum_{i \in S} w_i$$

$$(w_0 = 1/4, w_1 = 1/2, w_2 = 3/2)$$

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f*.

	$\varnothing$	$\{1\}$	$\{2\}$	$\{1,2\}$
f	0	1	2	2
g	$1/4$	$3/4$	$7/4$	$9/4$

$\epsilon = 1$

$$F = f - g$$

$$g(S) = w_0 + \sum_{i \in S} w_i$$

$$(w_0 = 1/4, w_1 = 1/2, w_2 = 3/2)$$

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*

	$\varnothing$	$\{1\}$	$\{2\}$	$\{1,2\}$	
f	0	1	2	2	$\epsilon = 1$
g	$1/4$	$3/4$	$7/4$	$9/4$	
F	$-1/4$	$1/4$	$1/4$	$-1/4$	$\epsilon = 1$

$$F = f - g$$

$$g(S) = w_0 + \sum_{i \in S} w_i$$

$$(w_0 = 1/4, w_1 = 1/2, w_2 = 3/2)$$

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*

- Then,

    - the distance from modularity of *f* equals the distance from modularity of $F = f\text{-}g$

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*
- Then,
  - the distance from modularity of *f* equals the distance from modularity of *F = f-g*
  - the best modular approximation of *F* is the 0-function *z*

The Approach

- Let f be a ϵ-a.m. function, and suppose that g is the best modular approximation of f.
- Then,
  - the distance from modularity of f equals the distance from modularity of $F = f\text{-}g$
  - the best modular approximation of F is the 0-function z

Thus, bounding the distance from modularity of functions F that are best approximated by z is sufficient

The Approach

- Let *f* be a ϵ-a.m. function, and suppose that *g* is the best modular approximation of *f.*
- Then,
  - the distance from modularity of *f* equals the distance from modularity of $F = f\text{-}g$
  - the best modular approximation of *F* is the 0-function *z*
  - the distance from modularity of *F* is equal to the maximum value *M* of *F.*

Marginals of *F*

- We would like to prove that the maximum value of *F* is small ($O(\epsilon \log n)$).

Marginals of *F*

- We would like to prove that the maximum value of *F* is small ($O(\epsilon \log n)$).

Lemma

There exist probability distributions P^+ and P^-, supported respectively on the *maximum* and the *minimum* sets of *F*, such that $\forall i : \Pr\left[i \in P^+\right] = \Pr\left[i \in P^-\right]$.

Marginals of *F*

- We would like to prove that the maximum value of *F* is small ($O(\epsilon \log n)$).

Lemma

There exist probability distributions P^+ and P^-, supported respectively on the *maximum* and the *minimum* sets of *F*, such that $\forall i : \Pr\left[i \in P^+\right] = \Pr\left[i \in P^-\right]$.

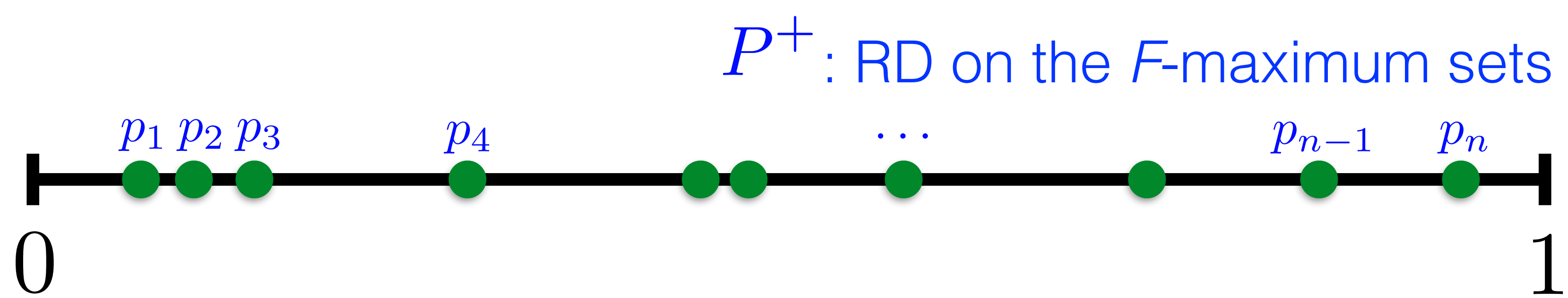
P^+: RD on the *F*-maximum sets

Marginals of *F*

- We would like to prove that the maximum value of *F* is small ($O(\epsilon \log n)$).

Lemma

There exist probability distributions P^+ and P^-, supported respectively on the *maximum* and the *minimum* sets of *F*, such that $\forall i : \Pr\left[i \in P^+\right] = \Pr\left[i \in P^-\right]$.

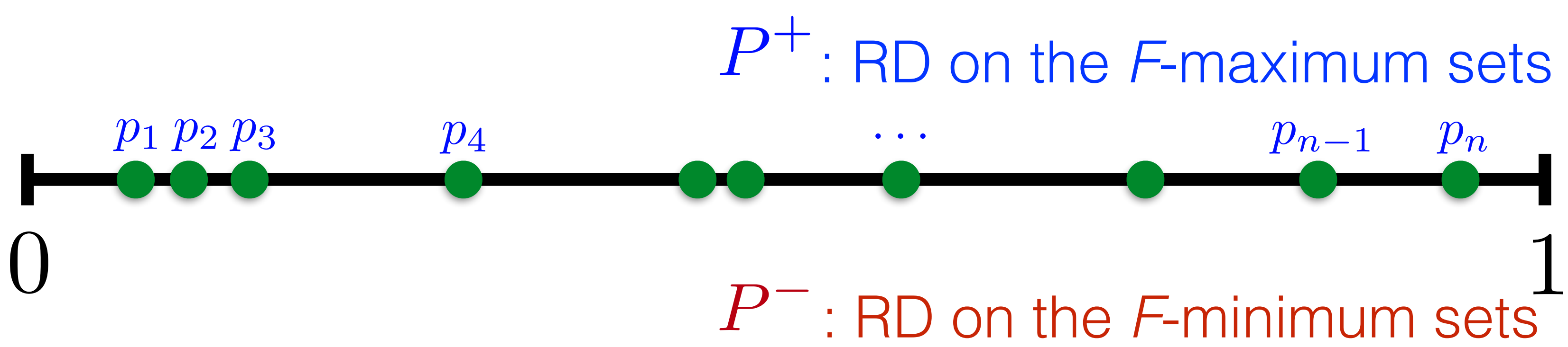


Marginals of *F*

- We would like to prove that the maximum value of *F* is small ($O(\epsilon \log n)$).

Lemma

There exist probability distributions P^+ and P^-, supported respectively on the *maximum* and the *minimum* sets of *F*, such that $\forall i : \Pr\left[i \in P^+\right] = \Pr\left[i \in P^-\right]$.

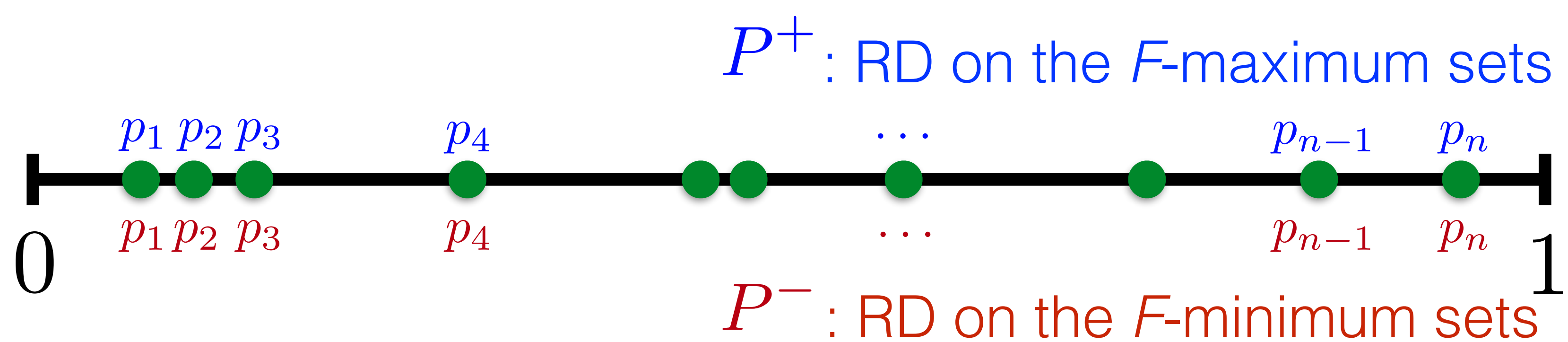


Marginals of F

- We would like to prove that the maximum value of F is small ($O(\epsilon \log n)$).

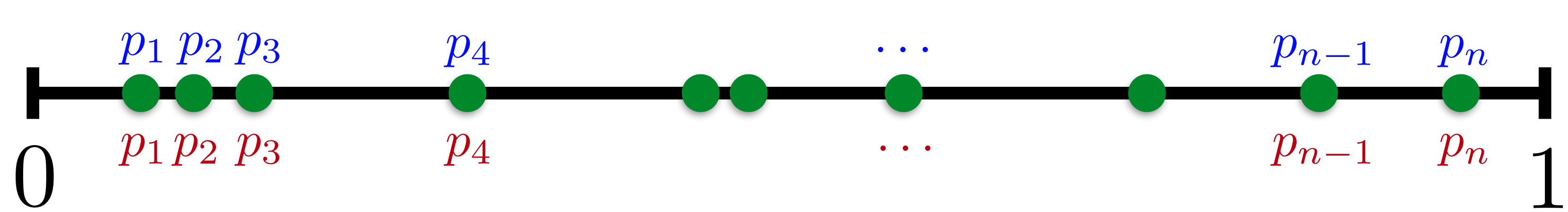
Lemma

There exist probability distributions P^+ and P^-, supported respectively on the *maximum* and the *minimum* sets of F, such that $\forall i : \Pr\left[i \in P^+\right] = \Pr\left[i \in P^-\right]$.



Random Thresholds

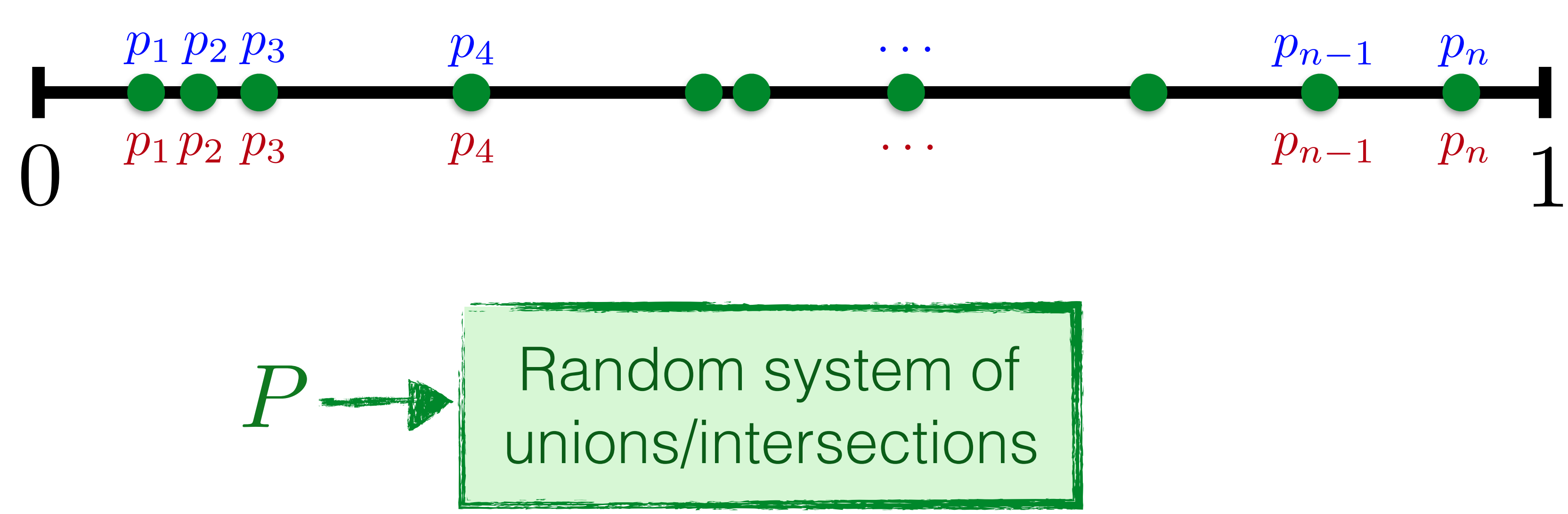
- We use the approximately modular rule to create a random system of unions/intersections that, using any random distribution over sets, produces sets that contain all the elements with a marginal (roughly) smaller than a UAR T



Random system of unions/intersections

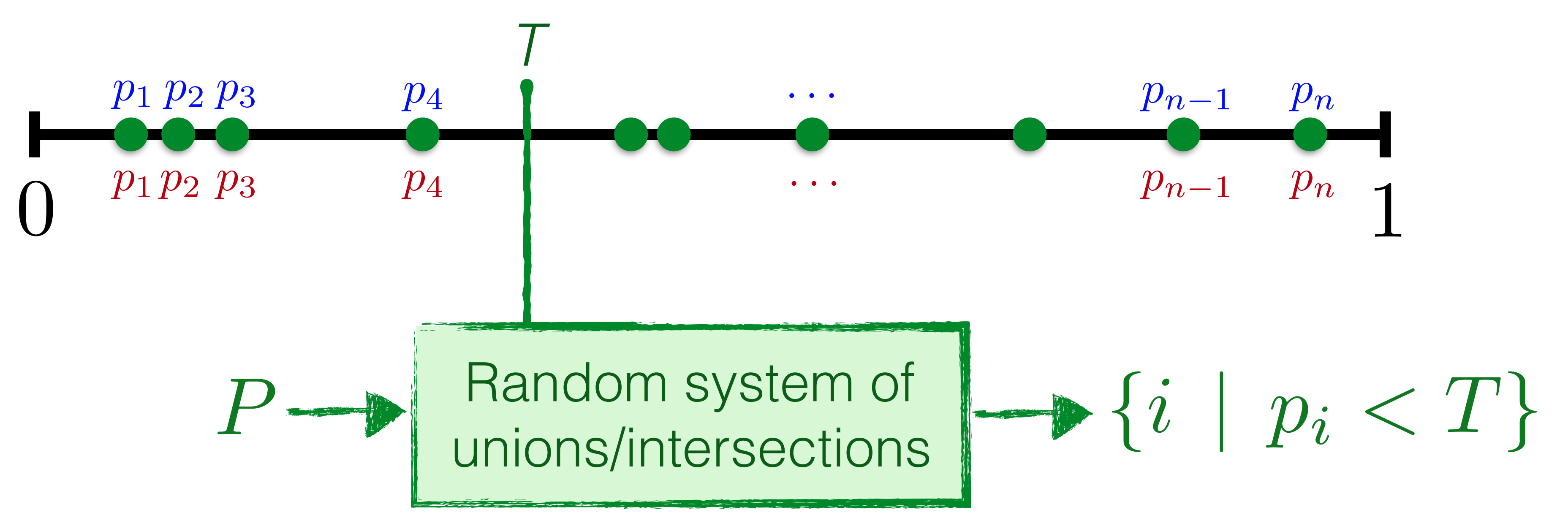
Random Thresholds

- We use the approximately modular rule to create a random system of unions/intersections that, using any random distribution over sets, produces sets that contain all the elements with a marginal (roughly) smaller than a UAR *T*



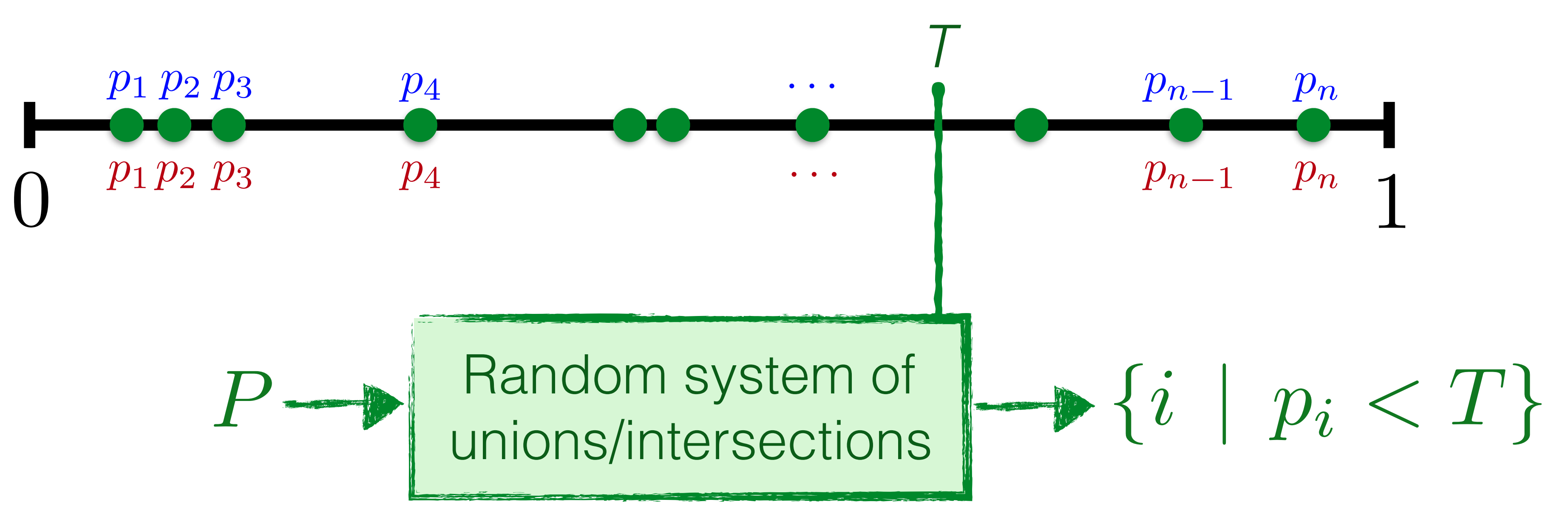
Random Thresholds

- We use the approximately modular rule to create a random system of unions/intersections that, using any random distribution over sets, produces sets that contain all the elements with a marginal (roughly) smaller than a UAR *T*



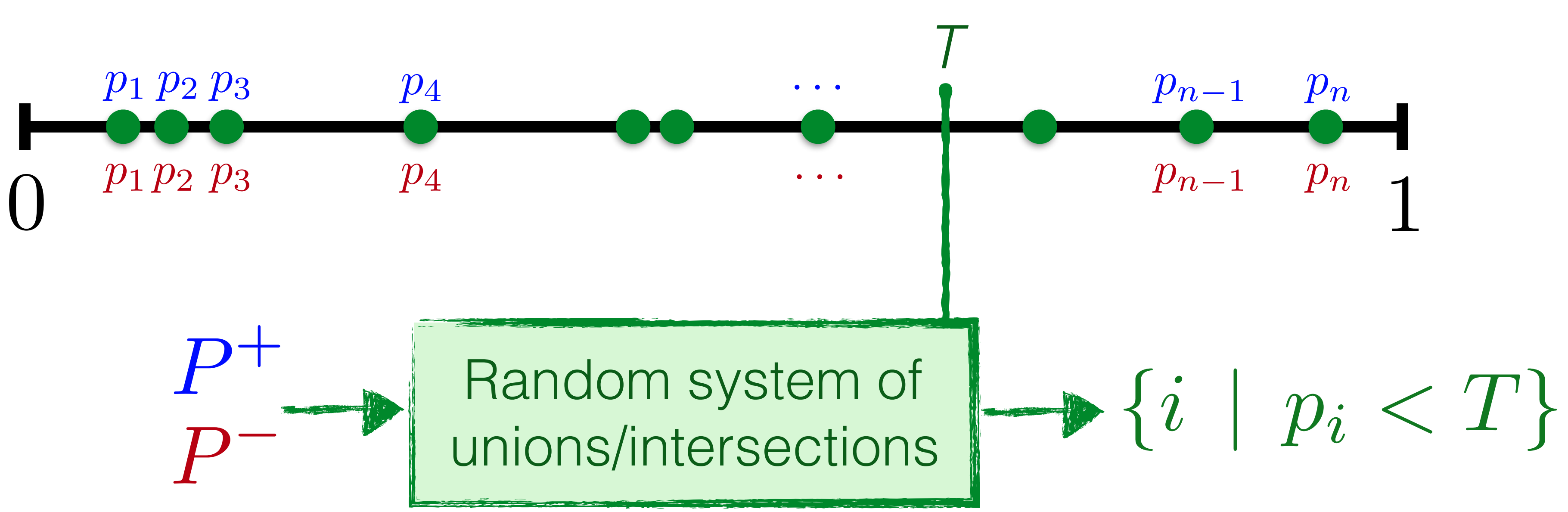
Random Thresholds

- We use the approximately modular rule to create a random system of unions/intersections that, using any random distribution over sets, produces sets that contain all the elements with a marginal (roughly) smaller than a UAR *T*

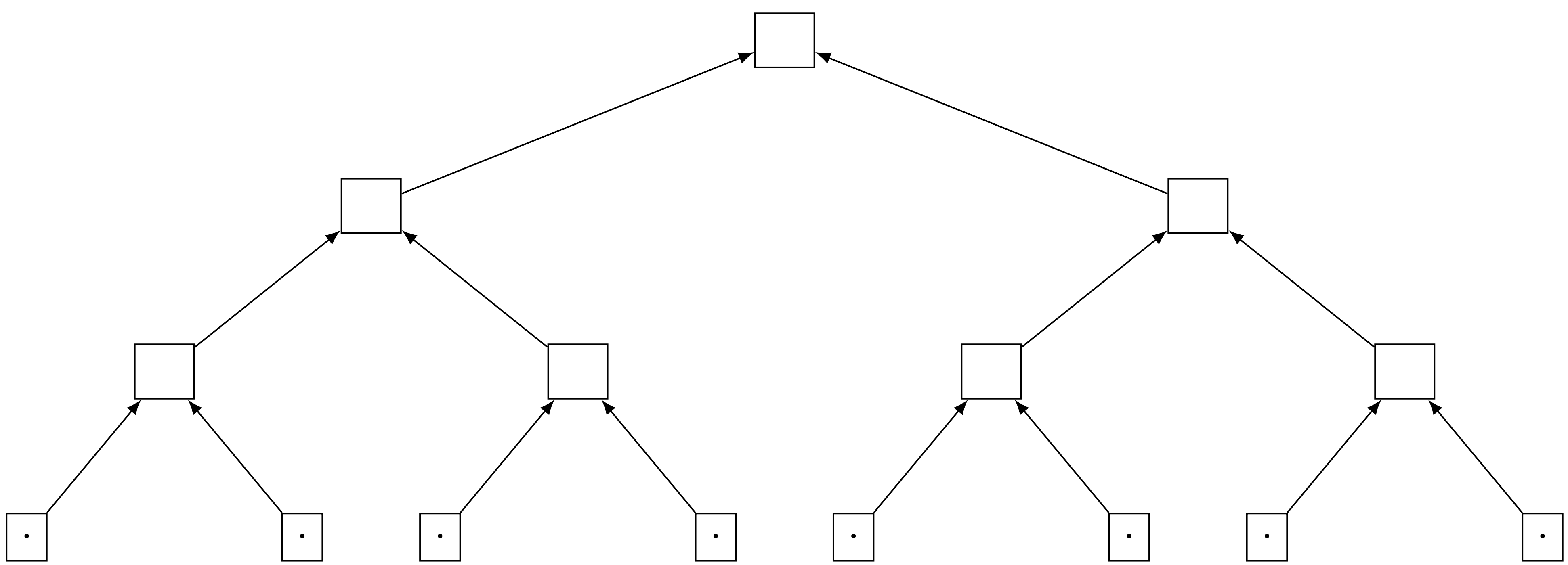


Random Thresholds

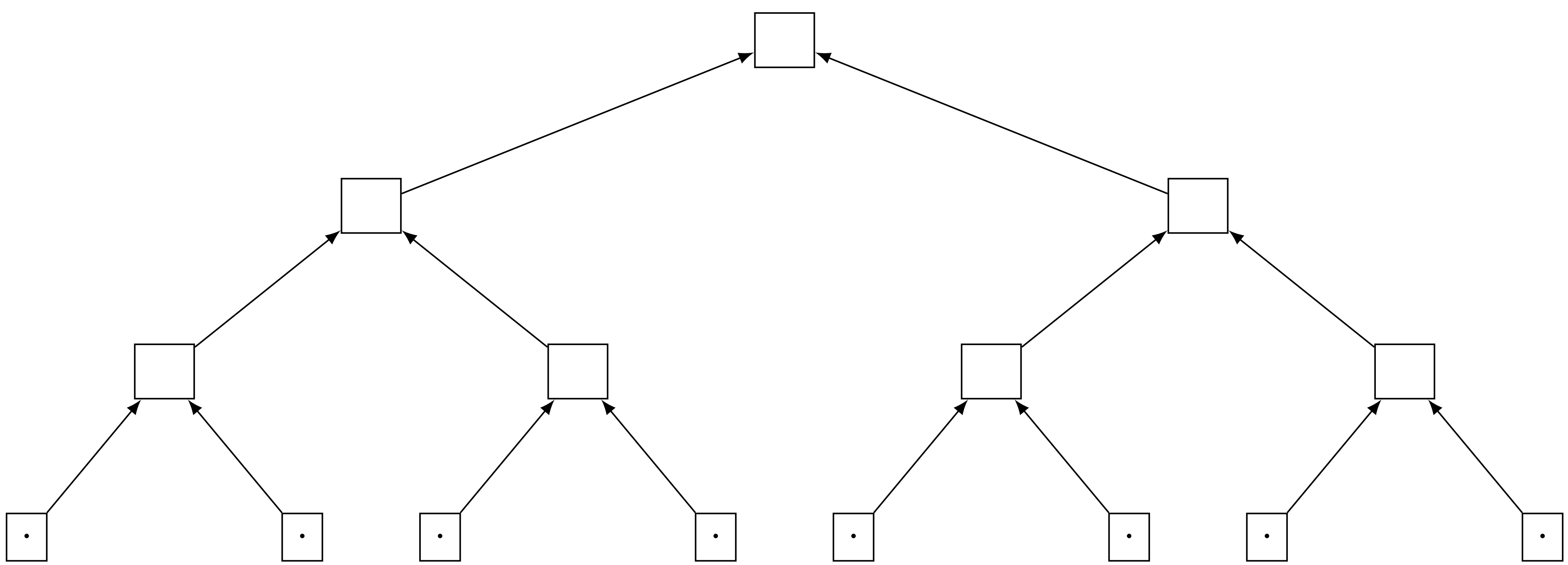
- We use the approximately modular rule to create a random system of unions/intersections that, using any random distribution over sets, produces sets that contain all the elements with a marginal (roughly) smaller than a UAR *T*



Random System of ∪/∩
Striped Tree

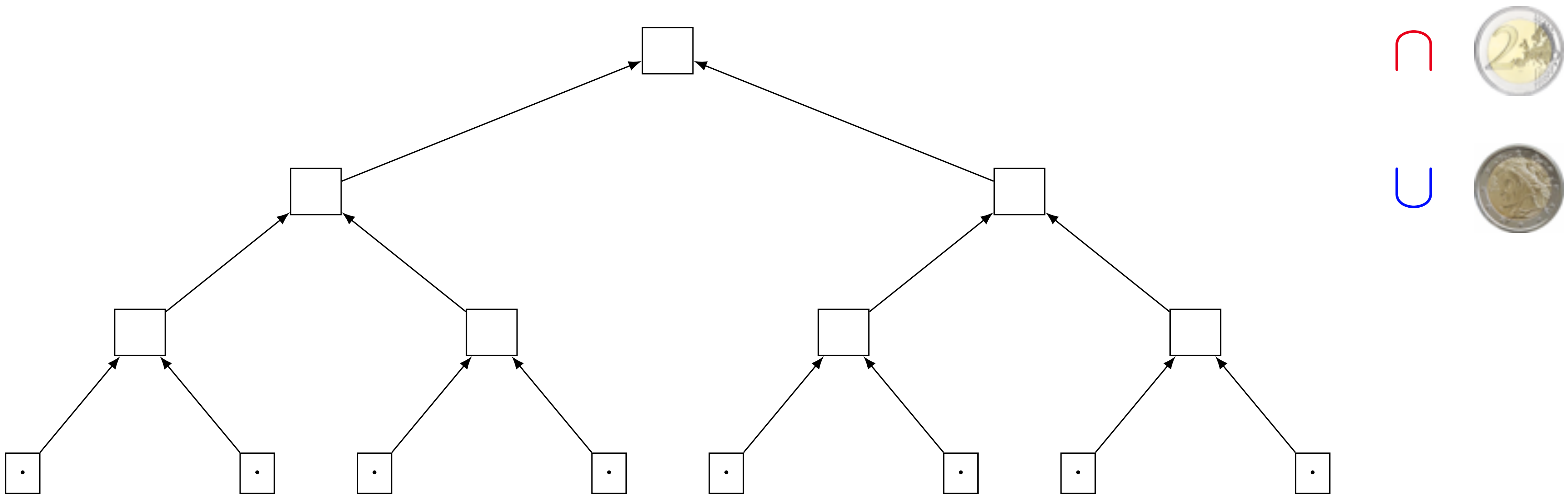


Random System of $\cup/\cap$
Striped Tree

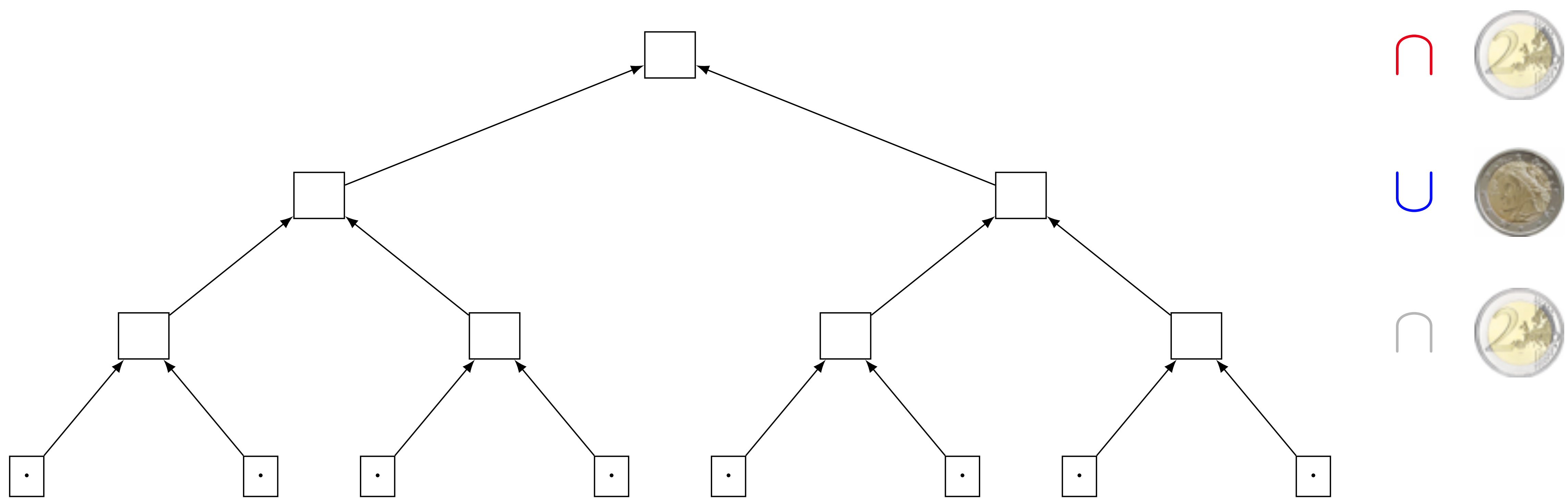


$\cap$

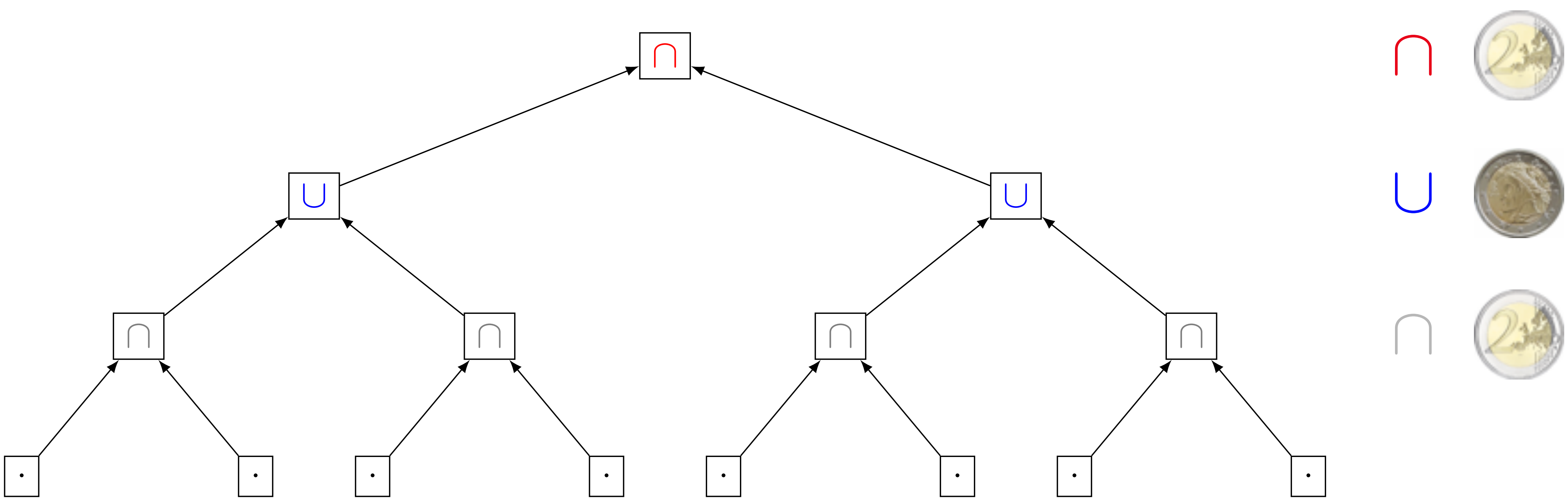
Random System of ∪/∩
Striped Tree



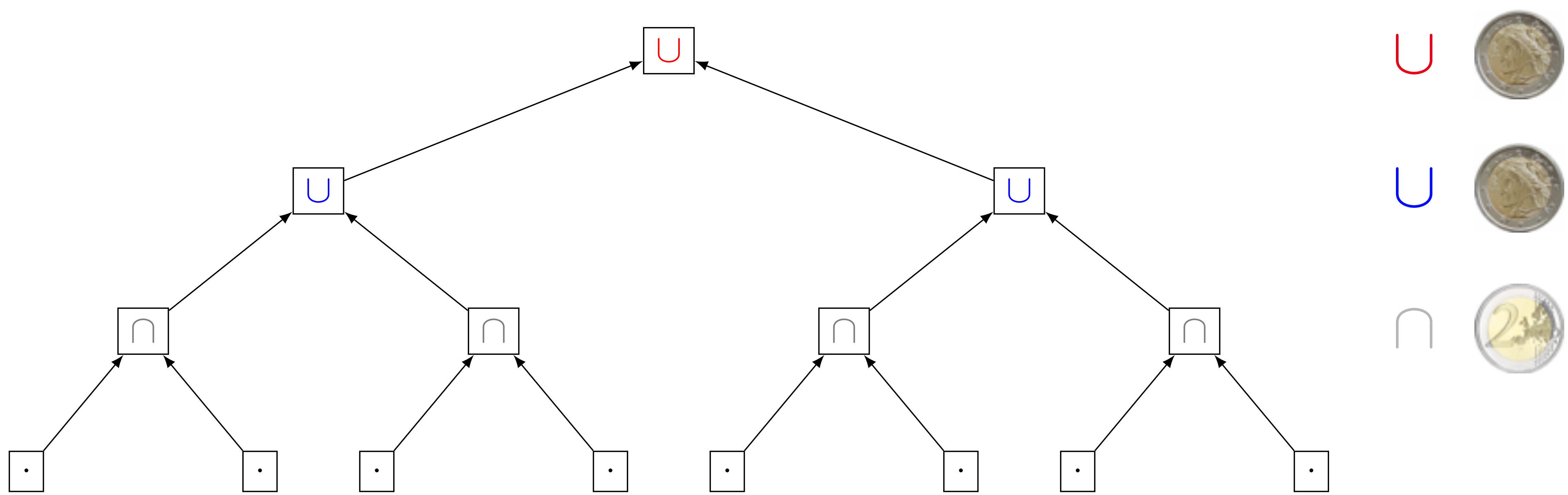
Random System of ∪/∩ Striped Tree



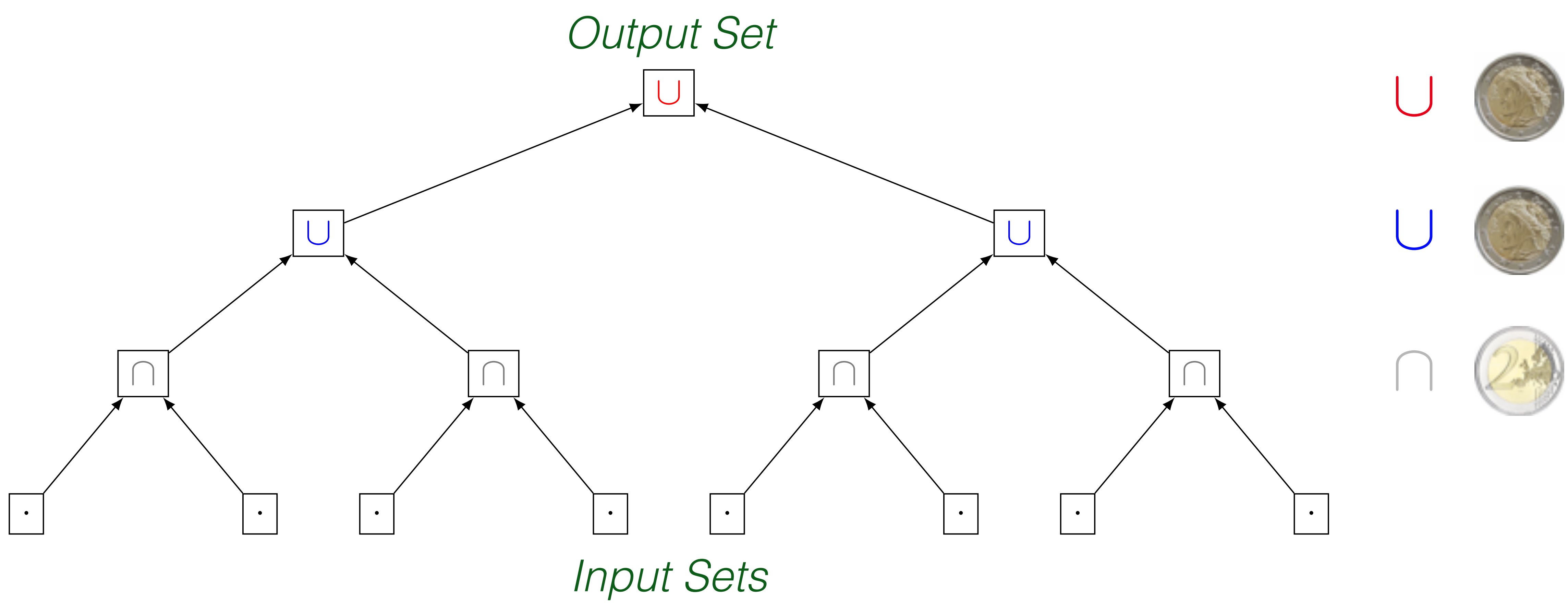
Random System of ∪/∩ Striped Tree



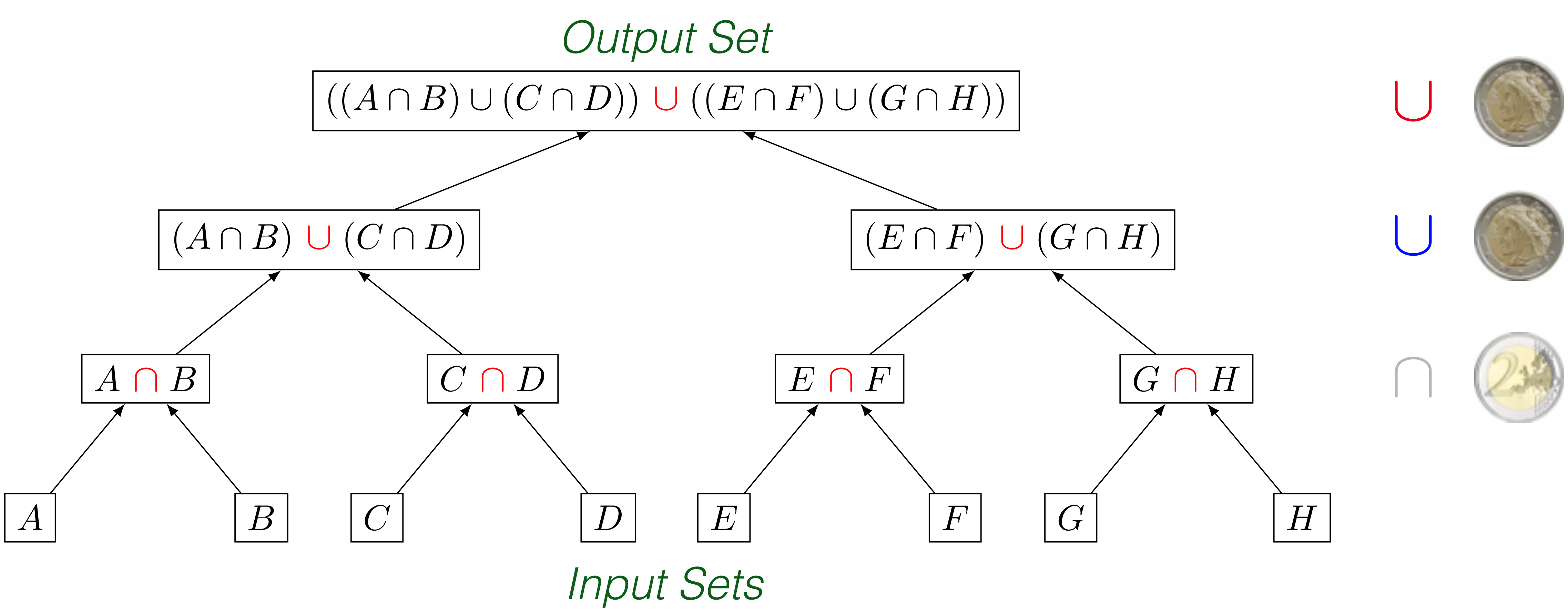
Random System of ∪/∩
Striped Tree



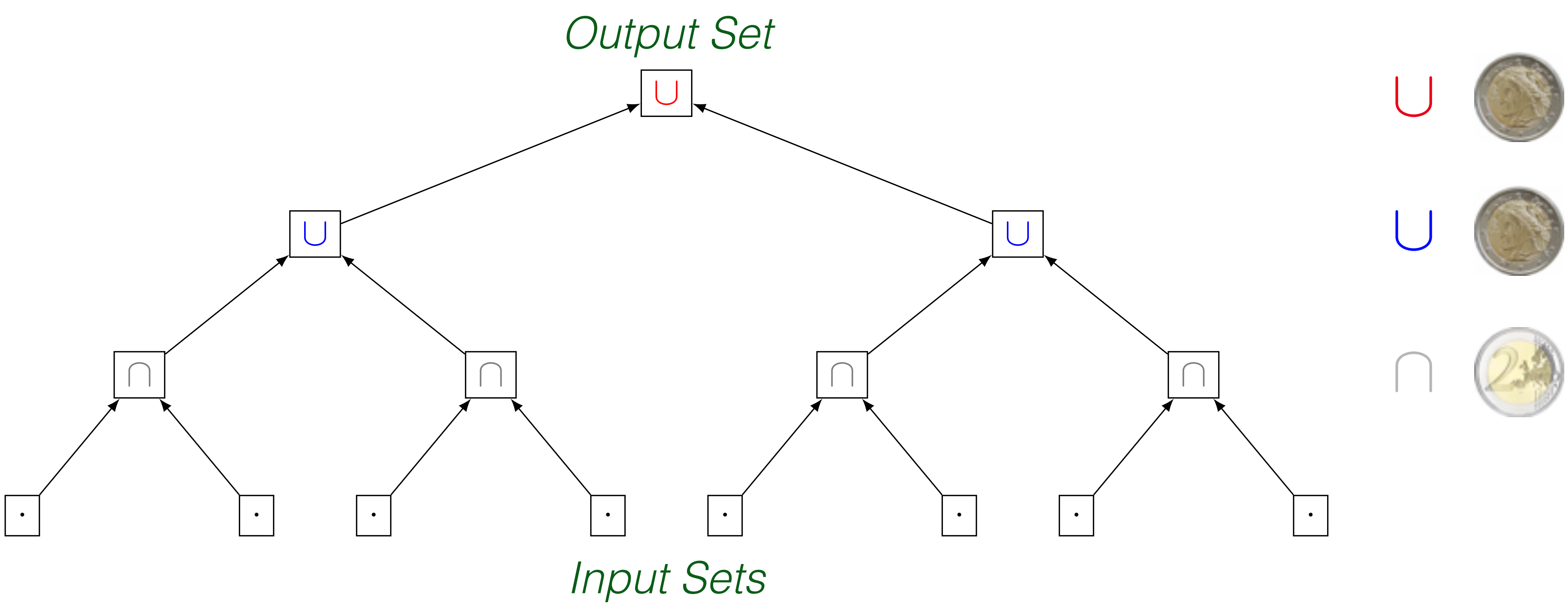
Random System of $\cup/\cap$
Striped Tree



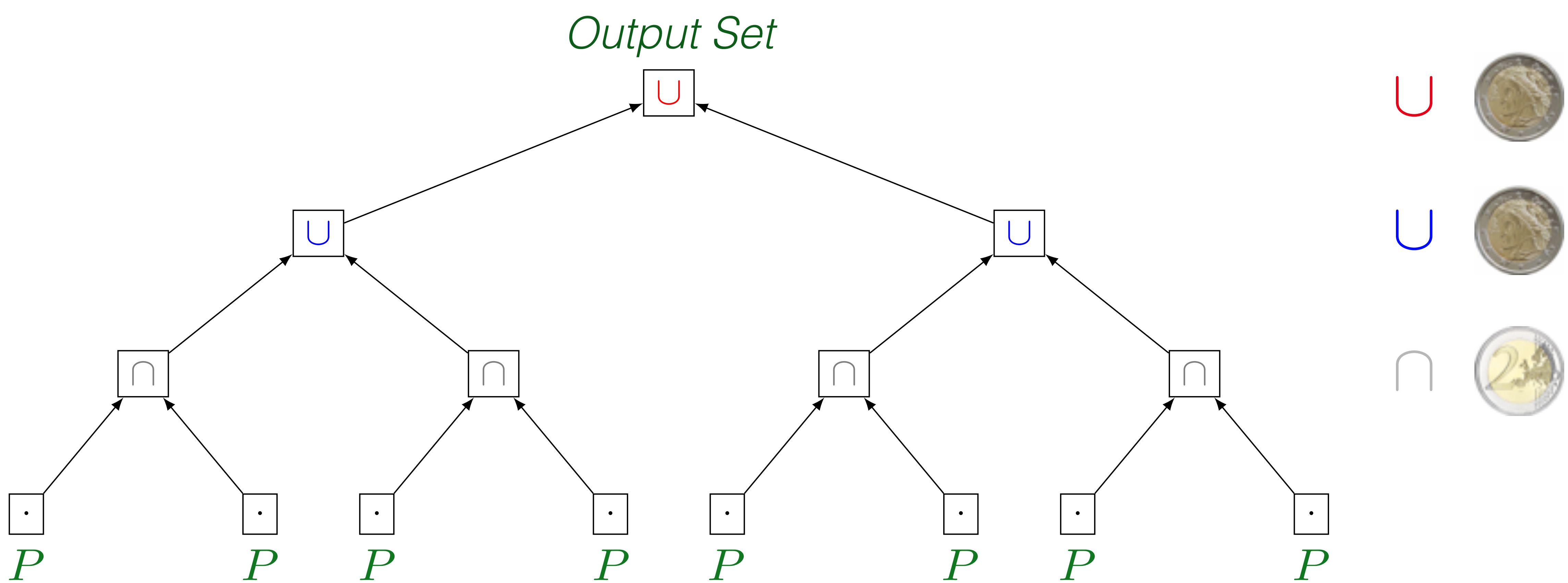
Random System of ∪/∩ Striped Tree



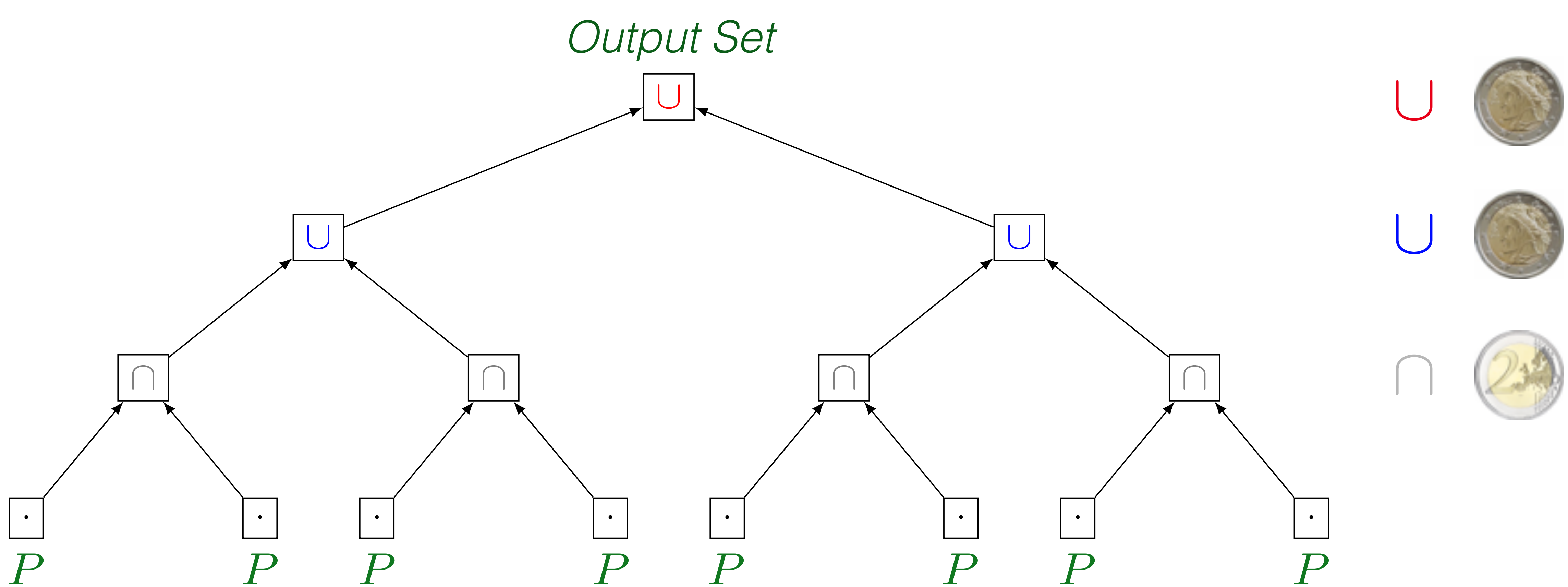
Random System of ∪/∩
Striped Tree



Random System of ∪/∩ Striped Tree

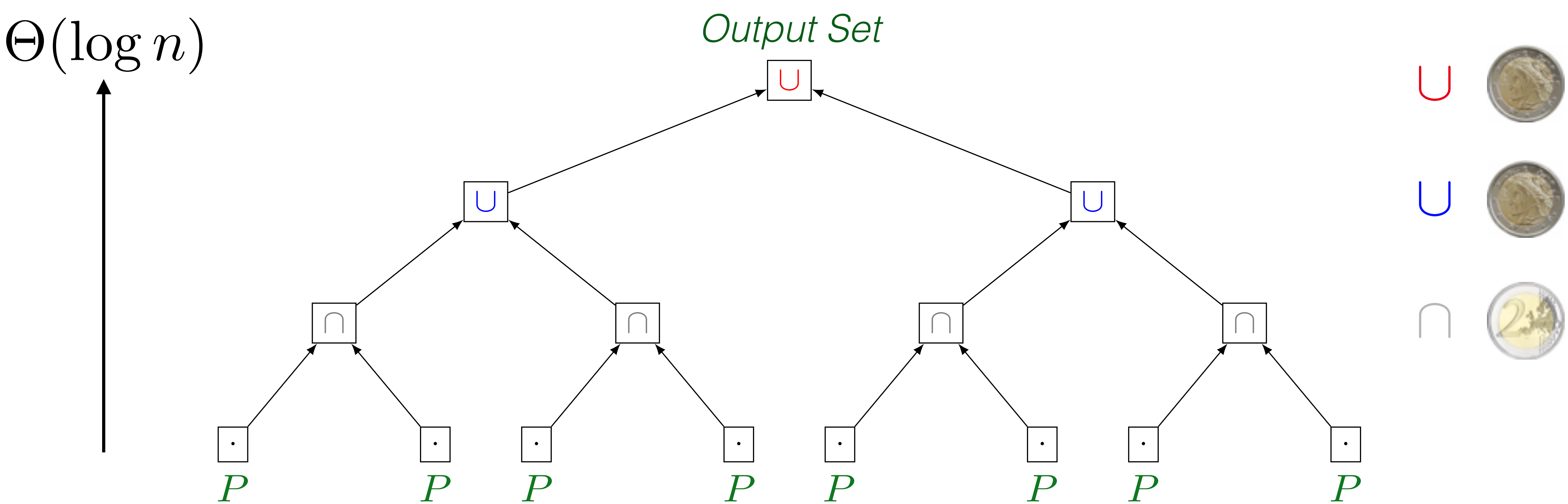


Random System of ∪/∩ Striped Tree



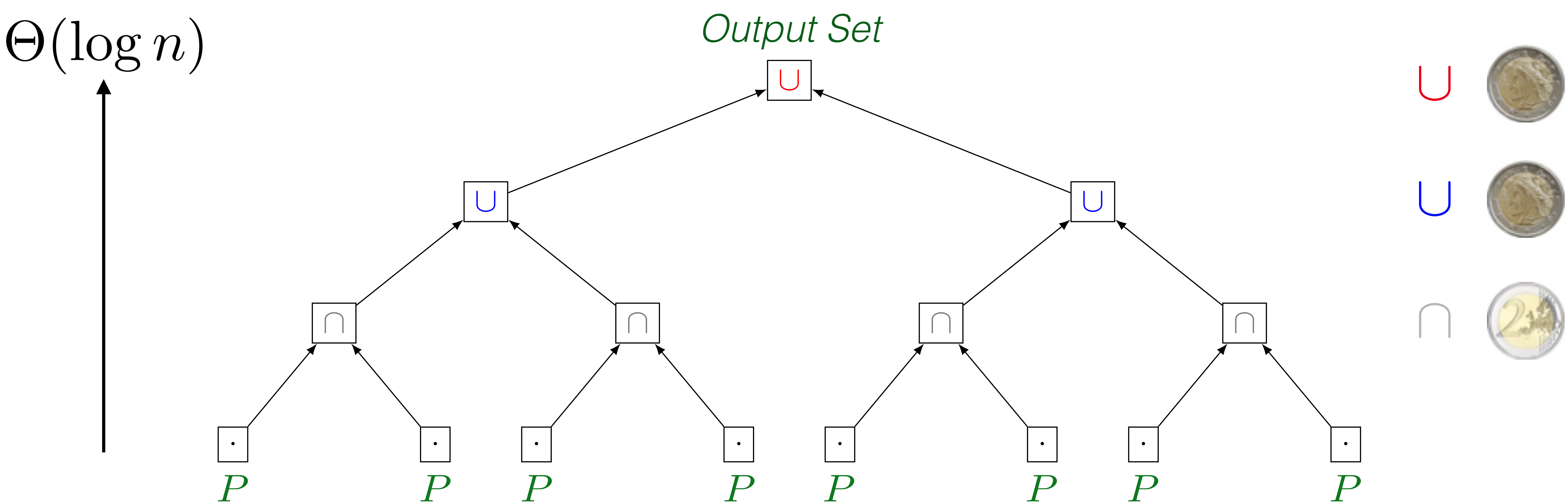
We prove (using the *random iterated systems'* framework) that choosing the levels' actions this way makes the tree act in a way "close to" a UAR threshold in *[0,1]*

Random System of $\cup/\cap$ Striped Tree



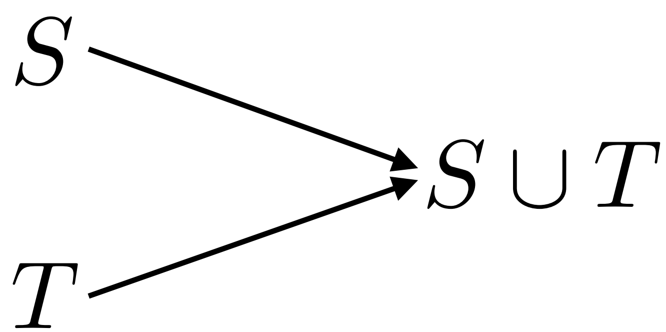
We prove (using the *random iterated systems'* framework) that choosing the levels' actions this way makes the tree act in a way "close to" a UAR threshold in *[0,1]*

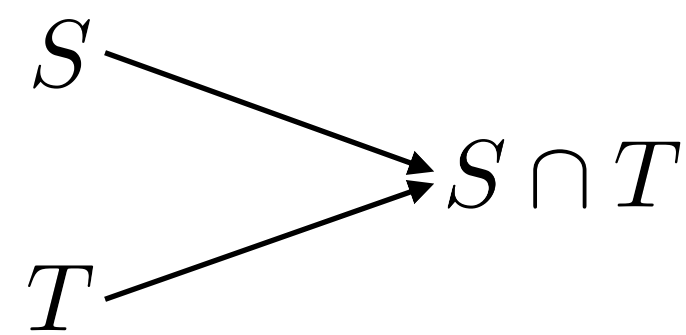
Random System of $\cup/\cap$ Striped Tree



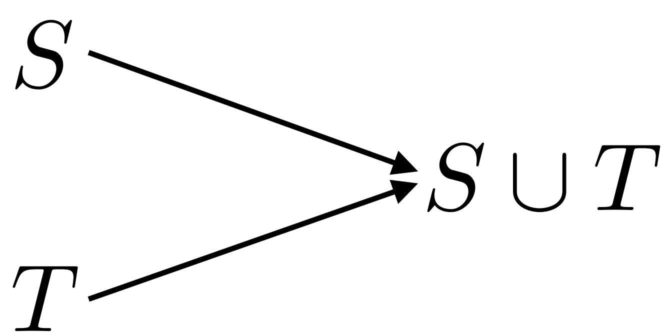
Thus, if we use the *same* random tree, once with input sets selected from P^+ and once from P^-, w.h.p. we will reach the same output set from both the **maximum** and the **minimum** sides

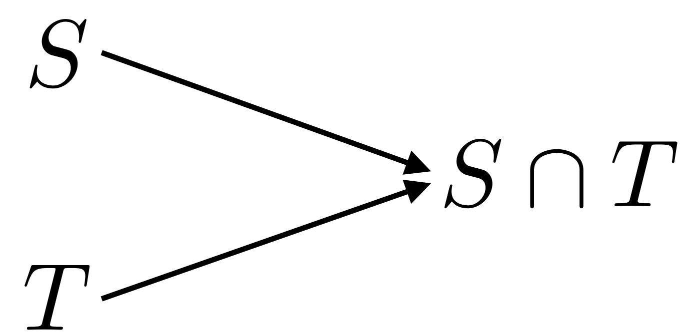
Rule Output(s)





Rule Output(s)

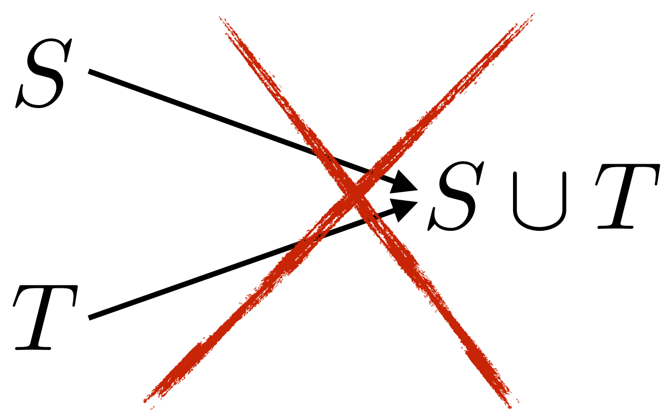


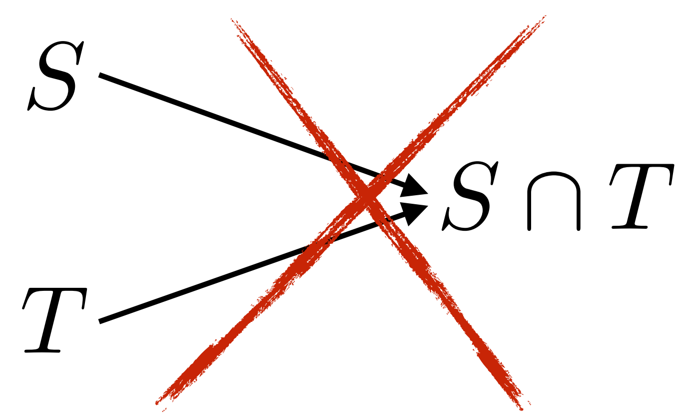


Approximately modular rules have *two* outputs

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon$$

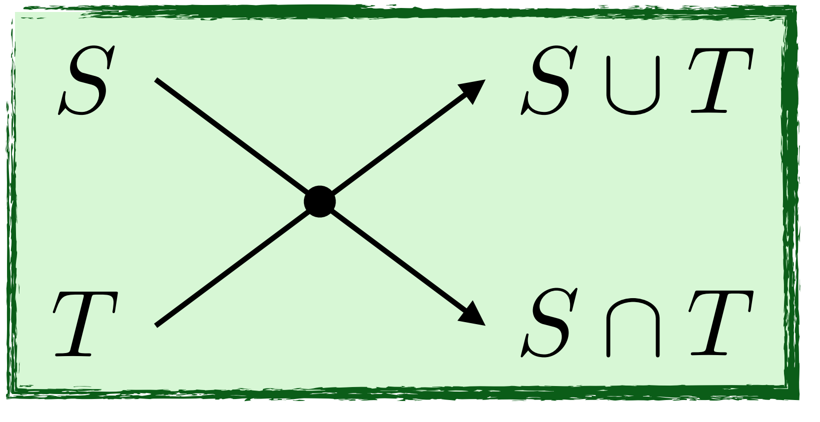
Rule Output(s)





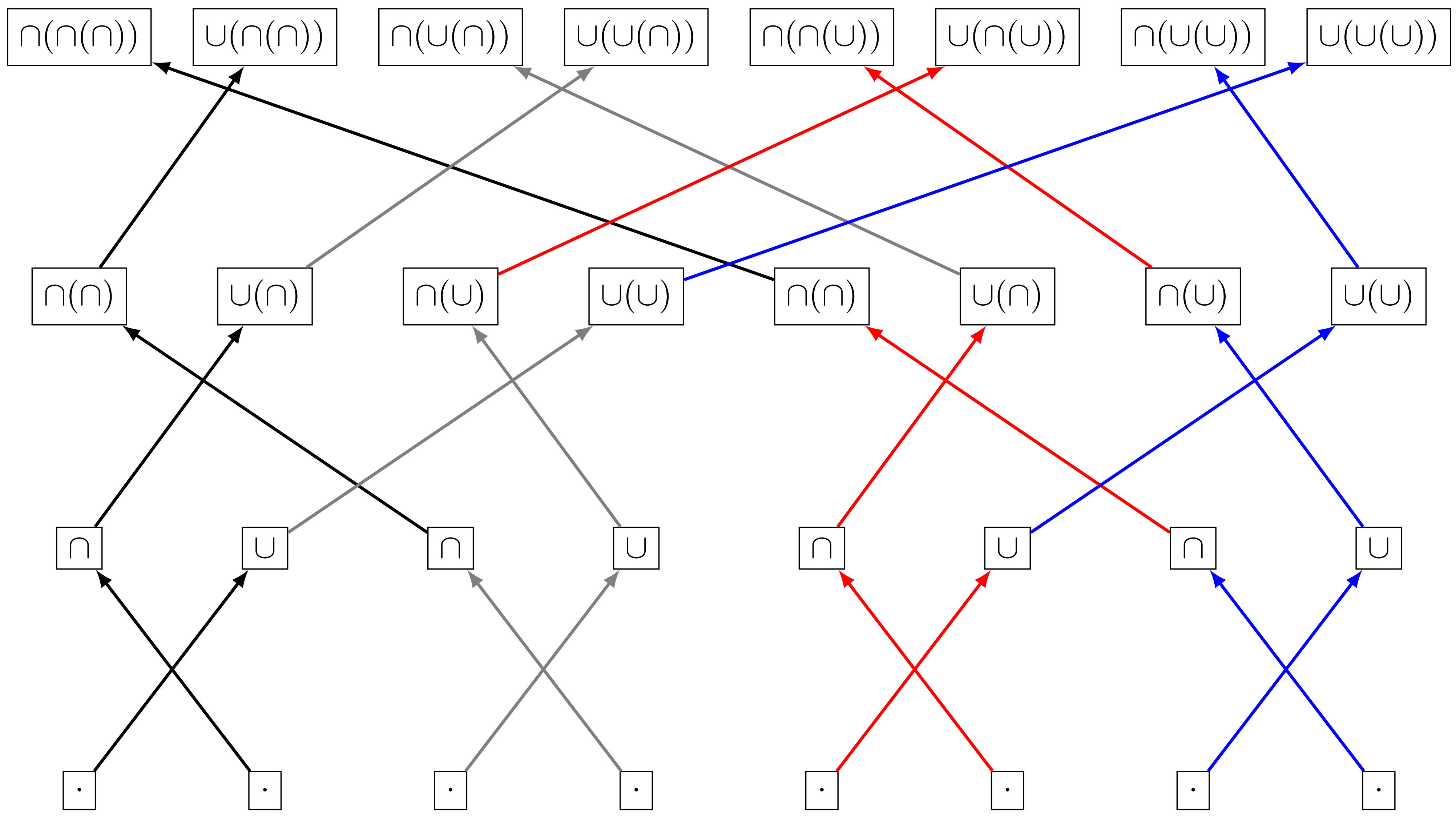
Approximately modular rules have *two* outputs

$$f(S) + f(T) = f(S \cup T) + f(S \cap T) \pm \epsilon$$



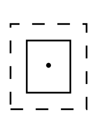
Striped Networks

We use *striped networks* to deal with the 2-outputs issue



Striped Networks

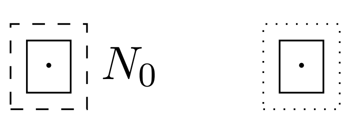
We use *striped networks* to deal with the 2-outputs issue



N_0

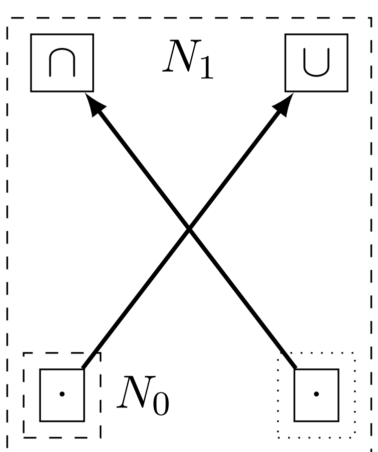
Striped Networks

We use *striped networks* to deal with the 2-outputs issue



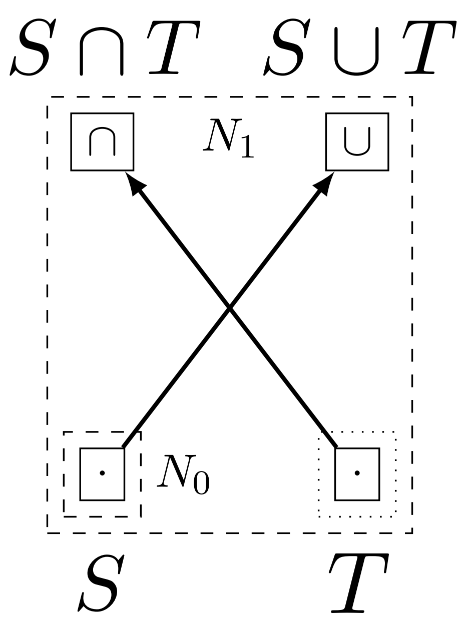
Striped Networks

We use *striped networks* to deal with the 2-outputs issue



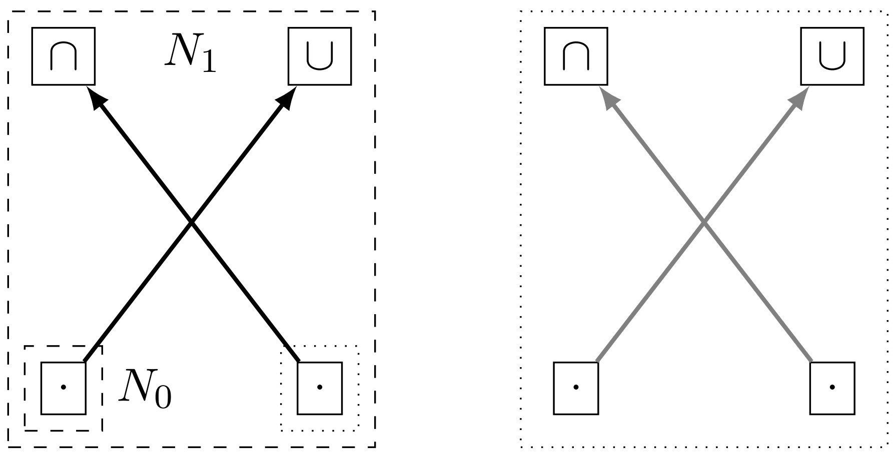
Striped Networks

We use *striped networks* to deal with the 2-outputs issue



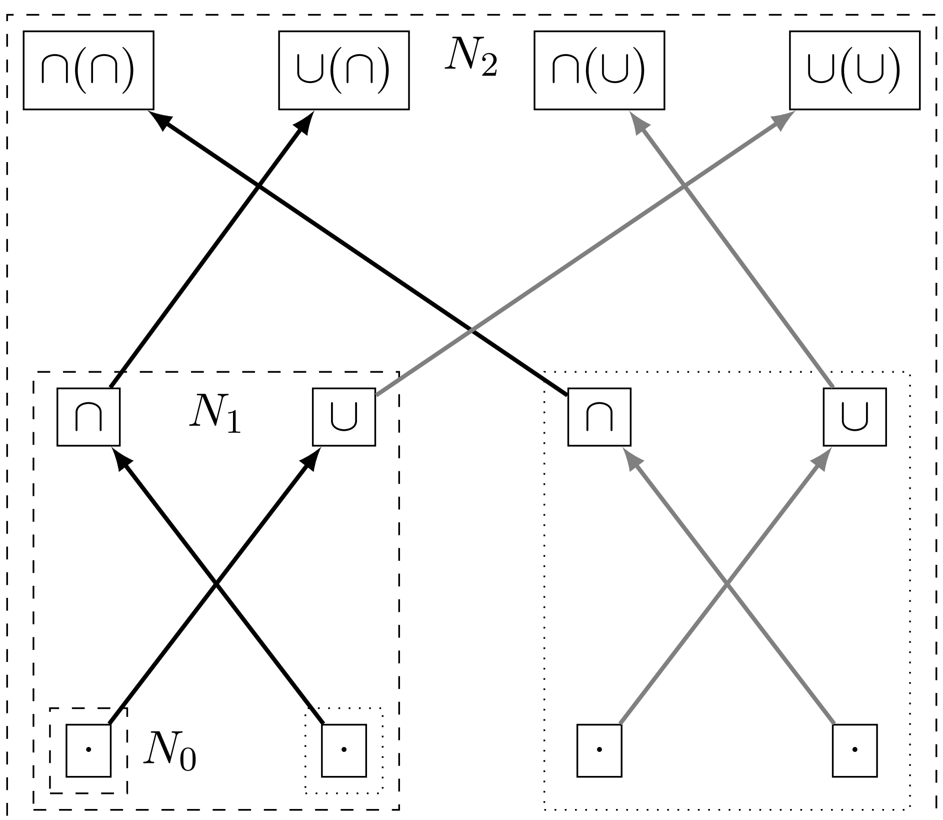
Striped Networks

We use *striped networks* to deal with the 2-outputs issue



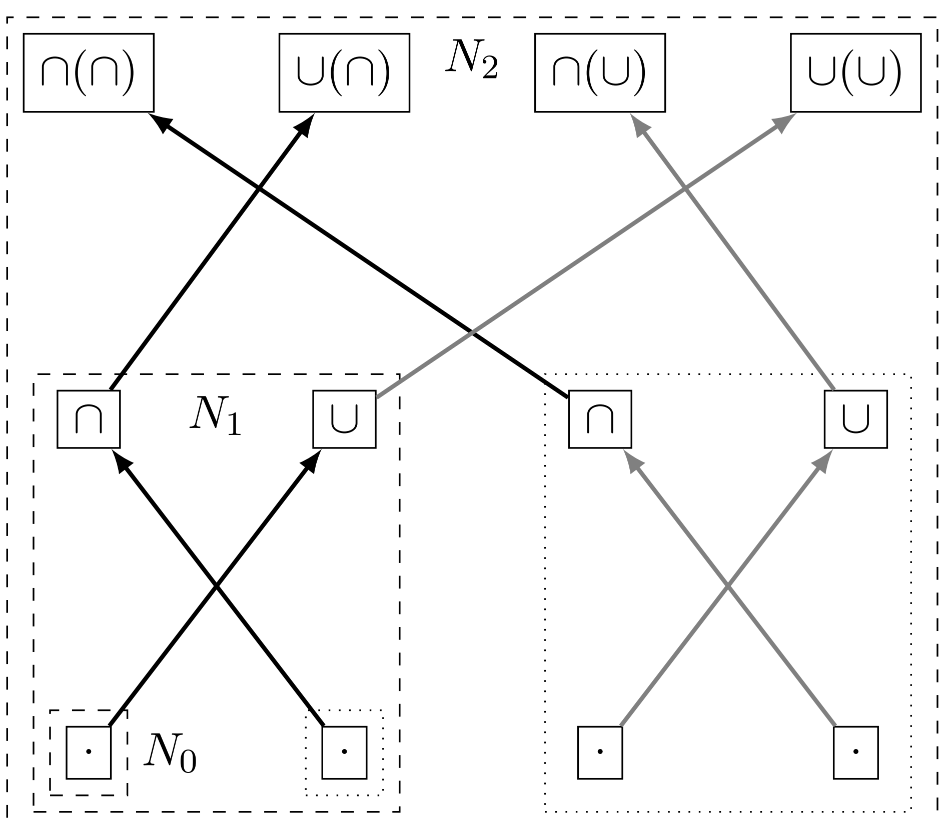
Striped Networks

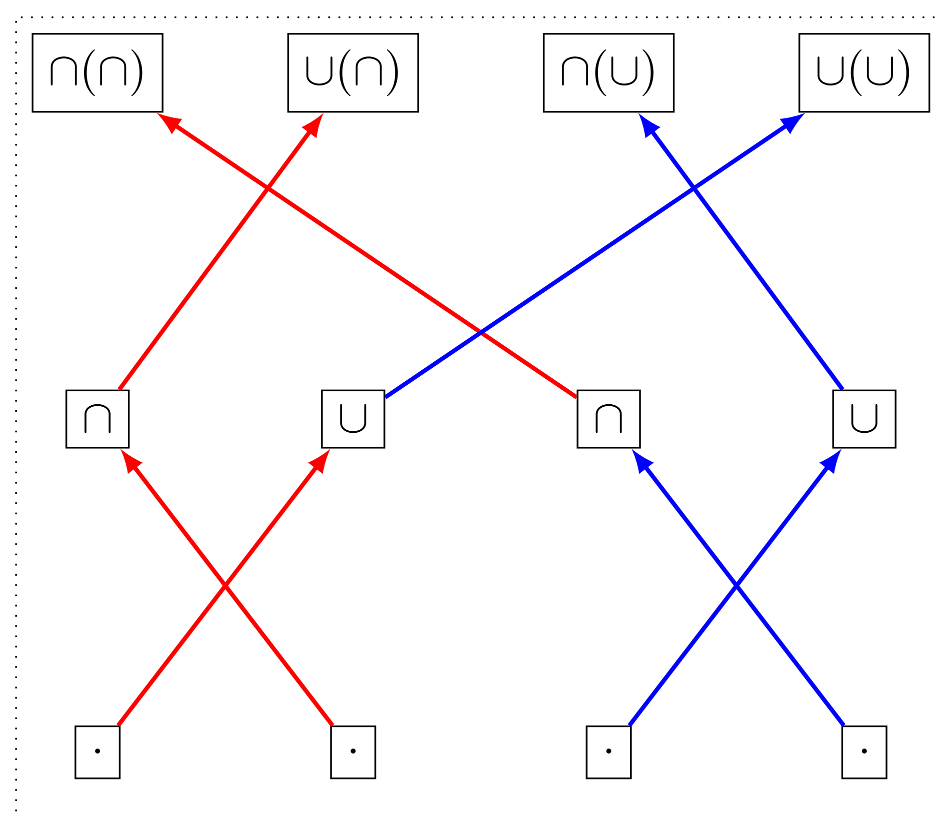
We use *striped networks* to deal with the 2-outputs issue



Striped Networks

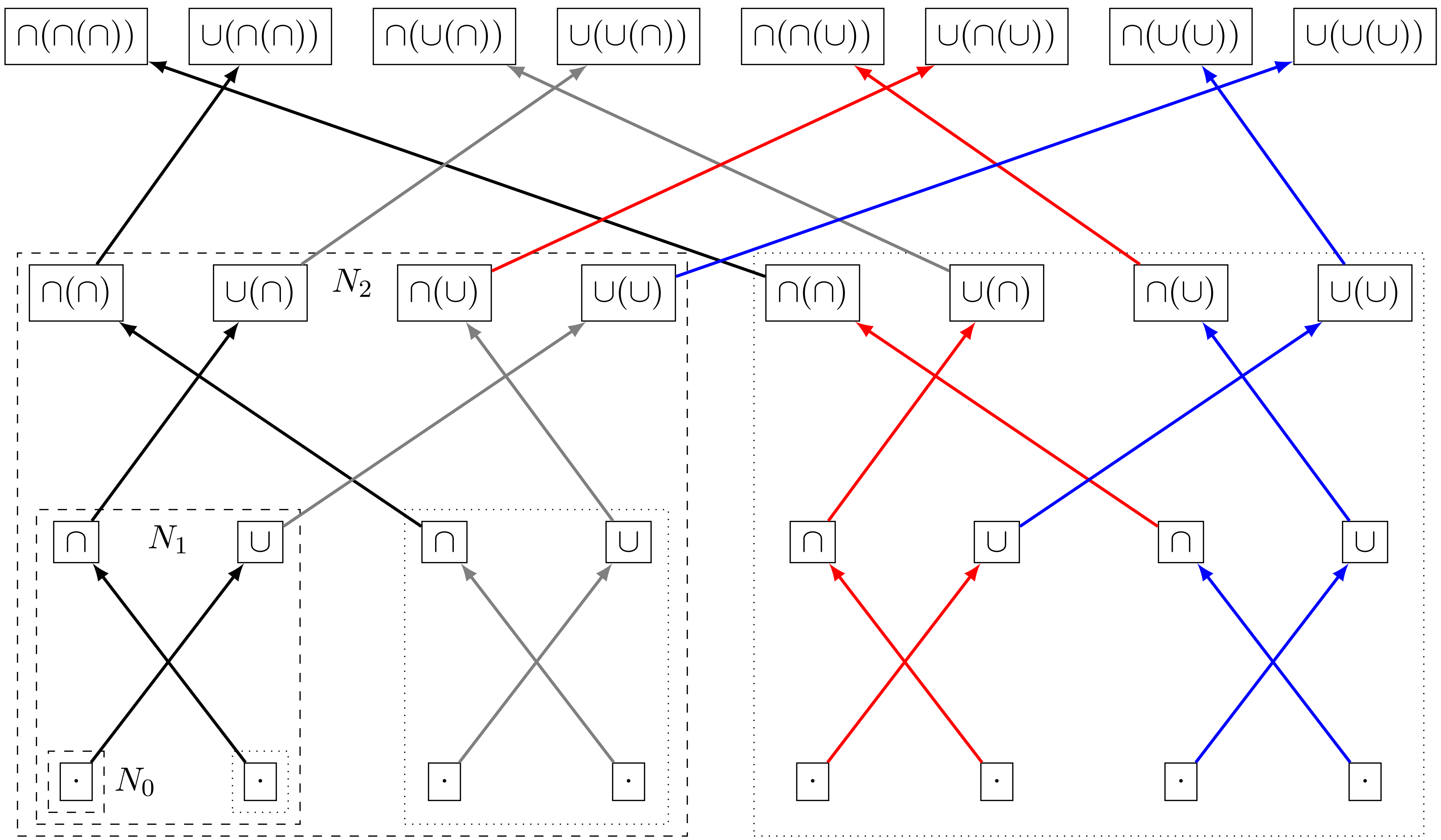
We use *striped networks* to deal with the 2-outputs issue





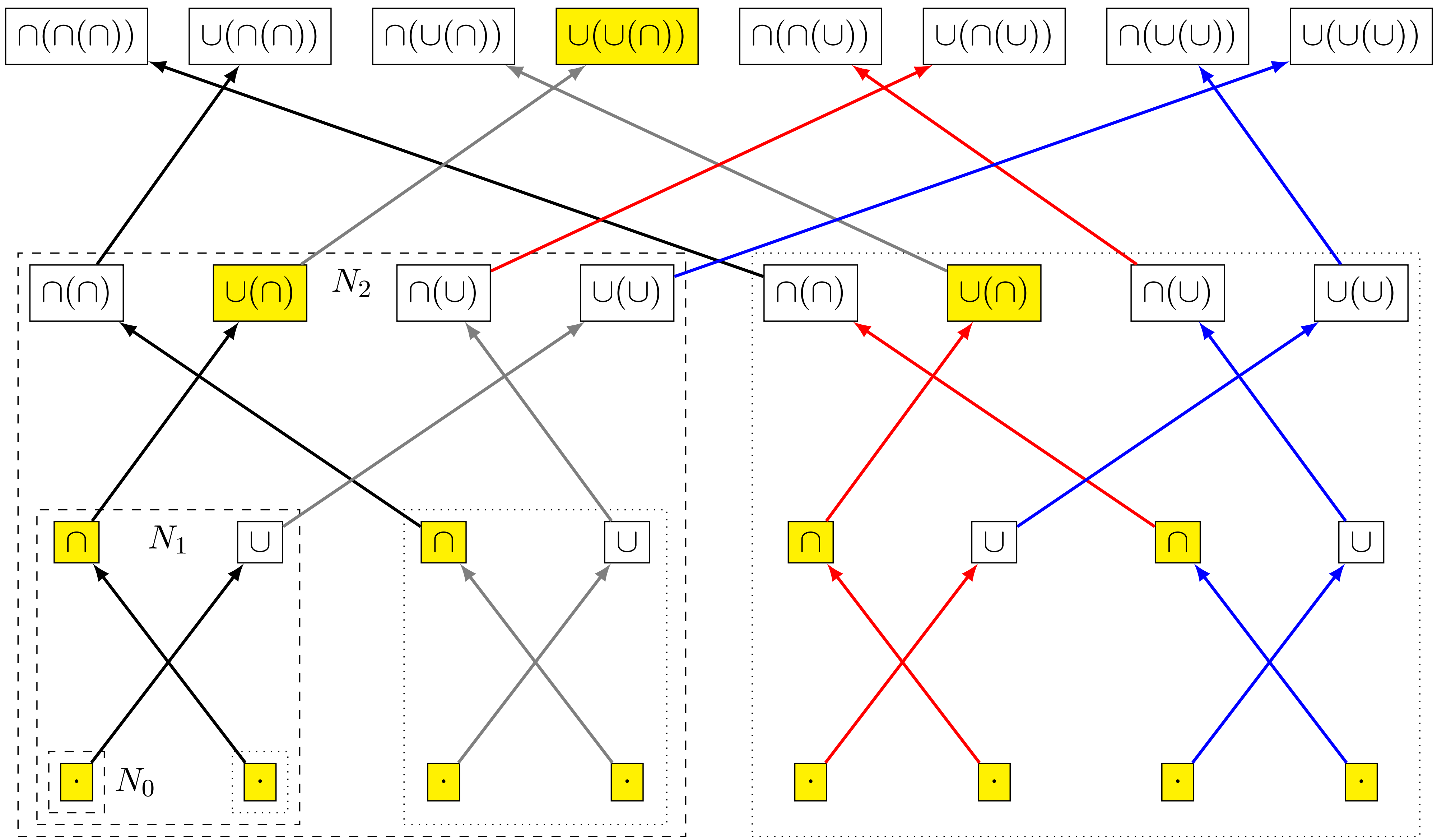
Striped Networks

We use *striped networks* to deal with the 2-outputs issue



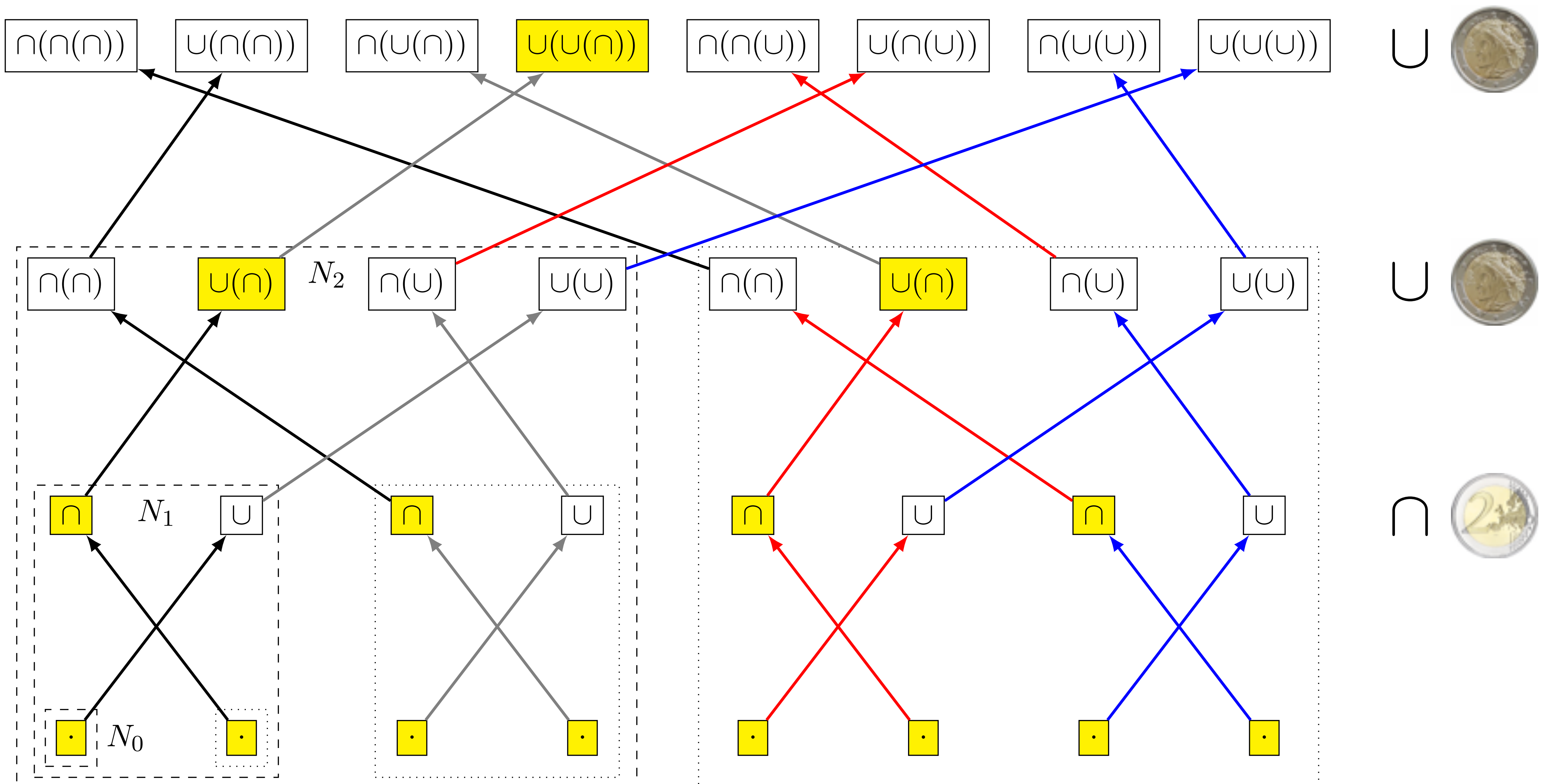
Striped Networks

Striped networks contain all the striped trees of a given height



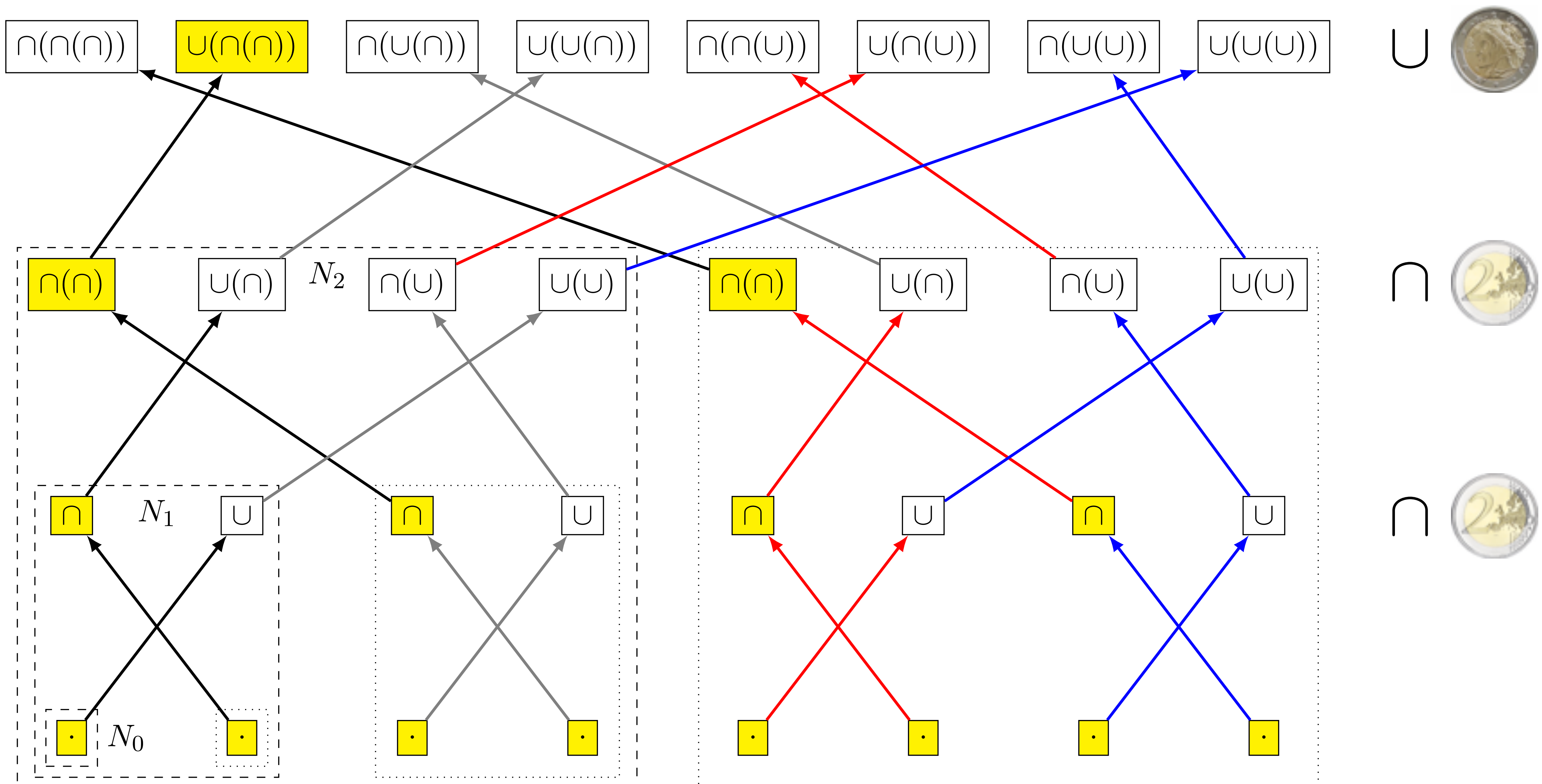
Striped Networks

Striped networks contain all the striped trees of a given height

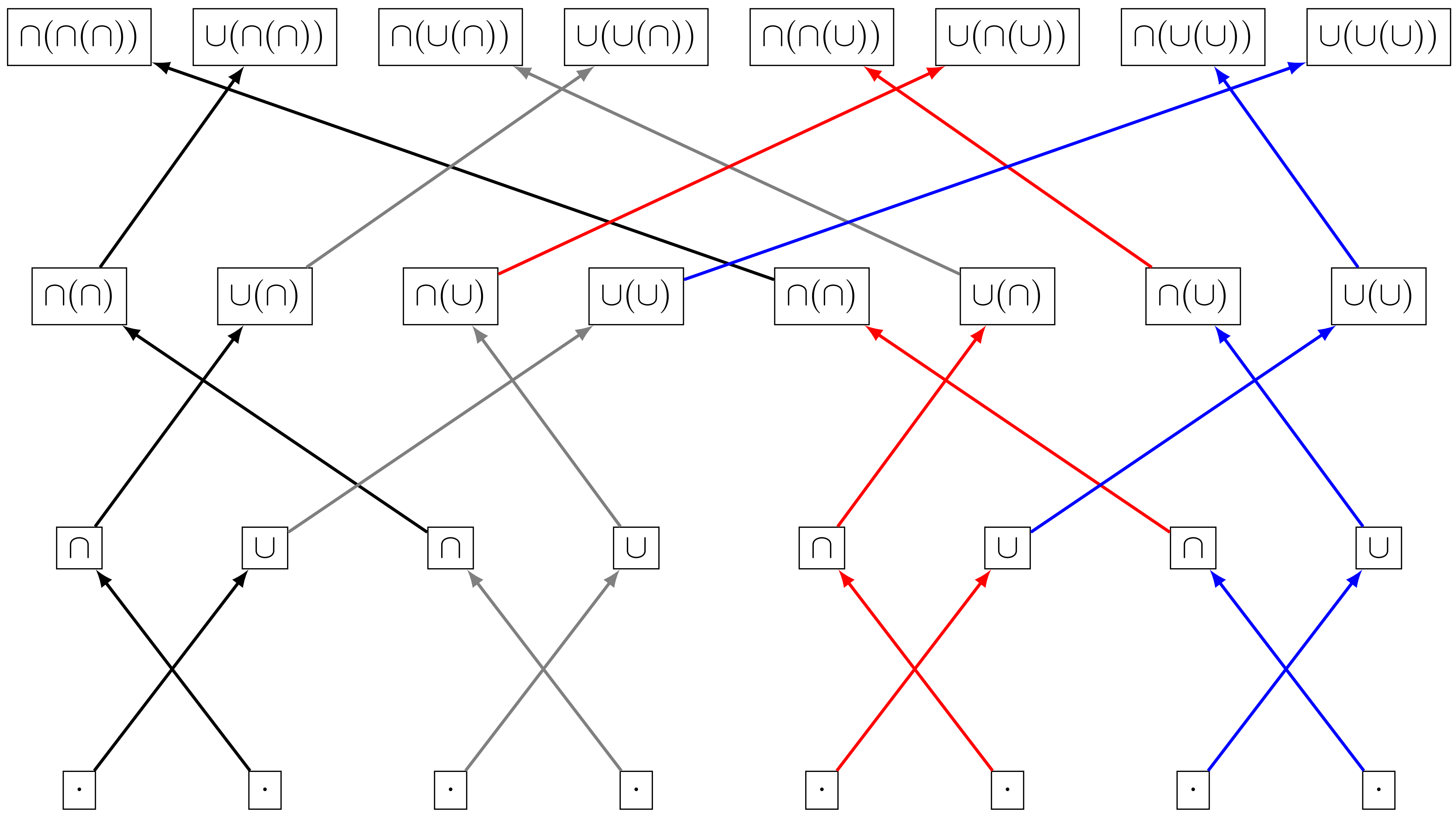


Striped Networks

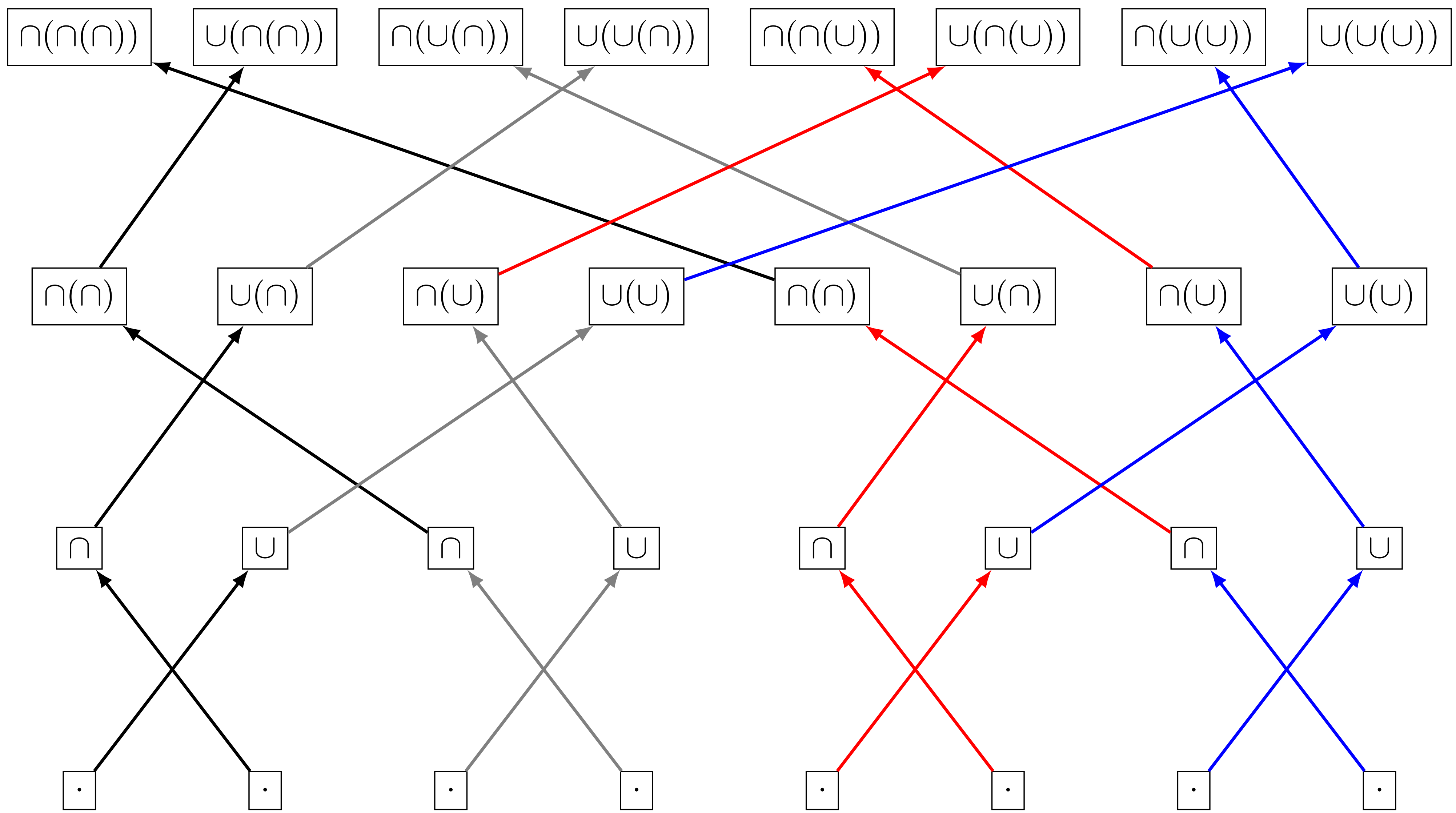
Striped networks contain all the striped trees of a given height



Striped Networks

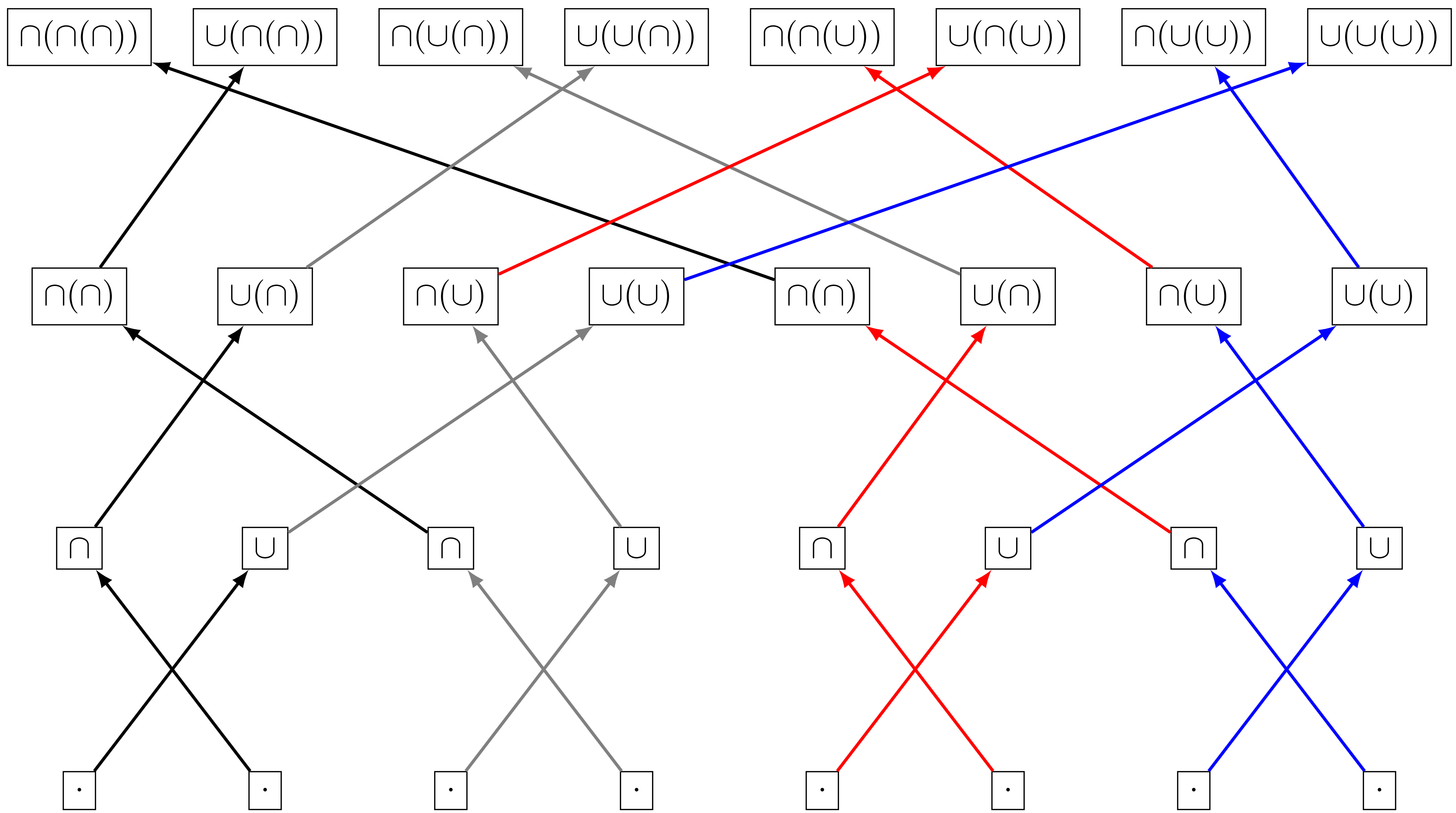


Striped Networks



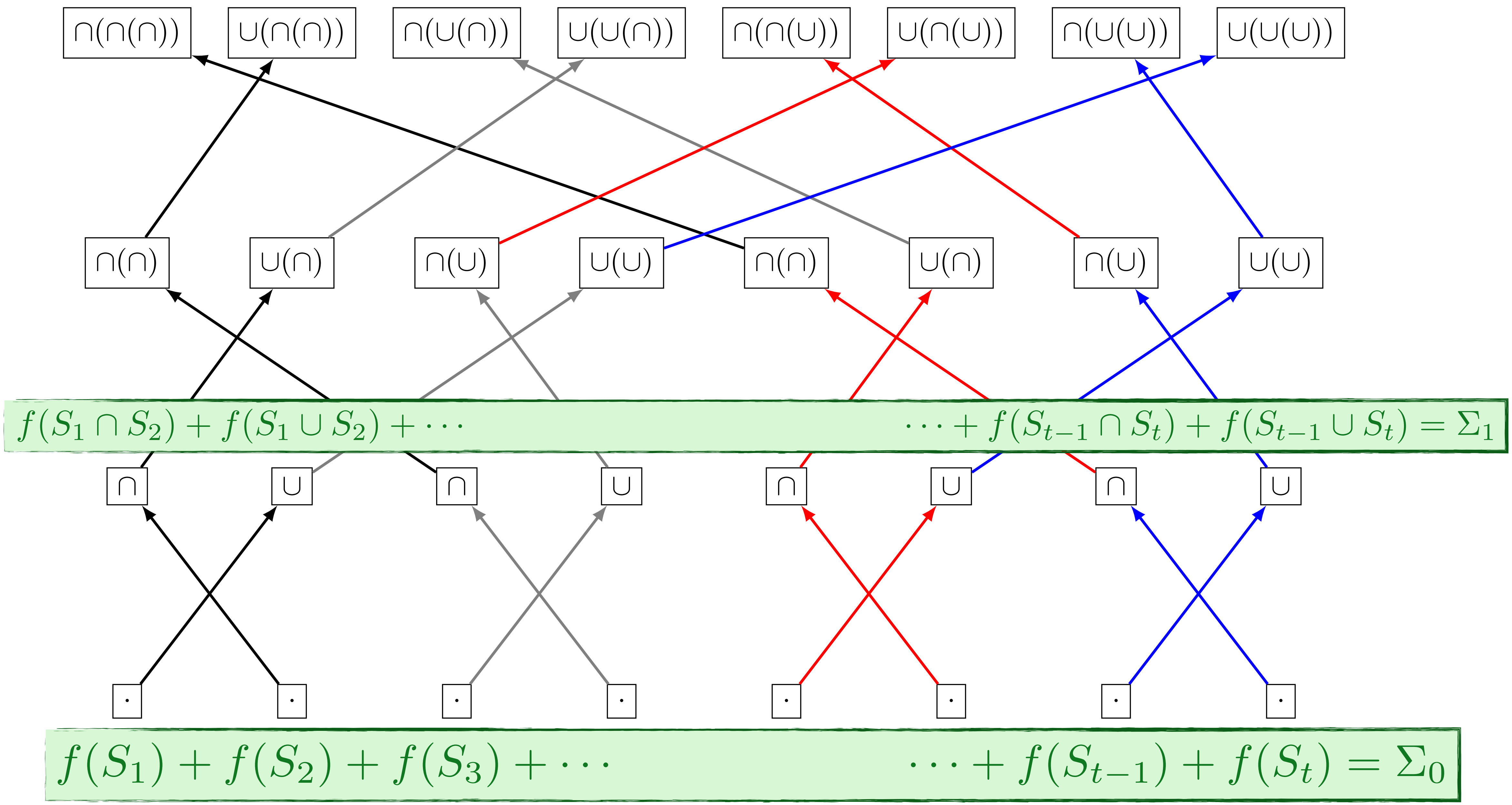
$f(S_1) + f(S_2) + f(S_3) + \cdots$ $\cdots + f(S_{t-1}) + f(S_t)$

Striped Networks

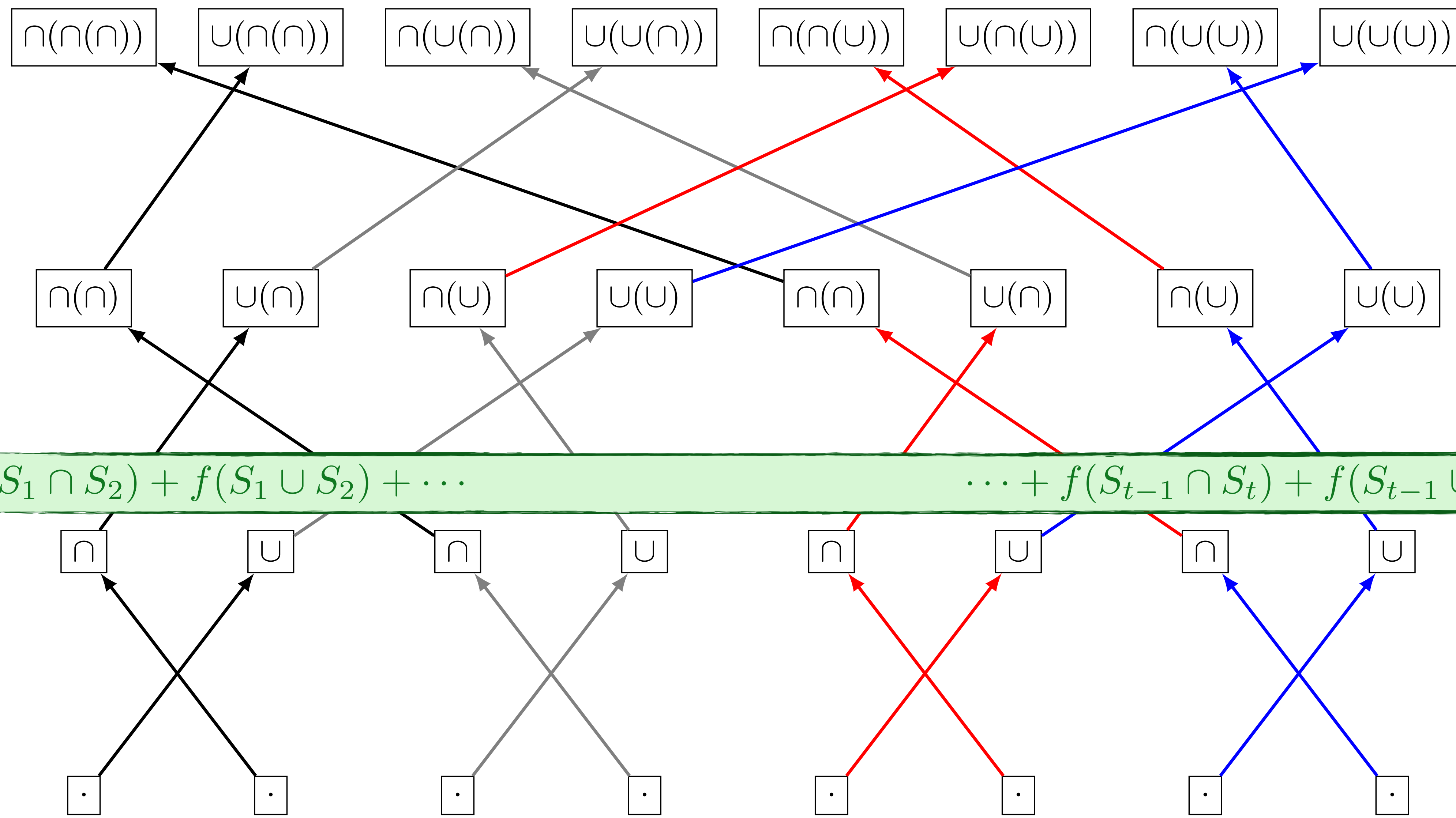


$f(S_1) + f(S_2) + f(S_3) + \cdots \qquad \cdots + f(S_{t-1}) + f(S_t) = \Sigma_0$

Striped Networks



Striped Networks

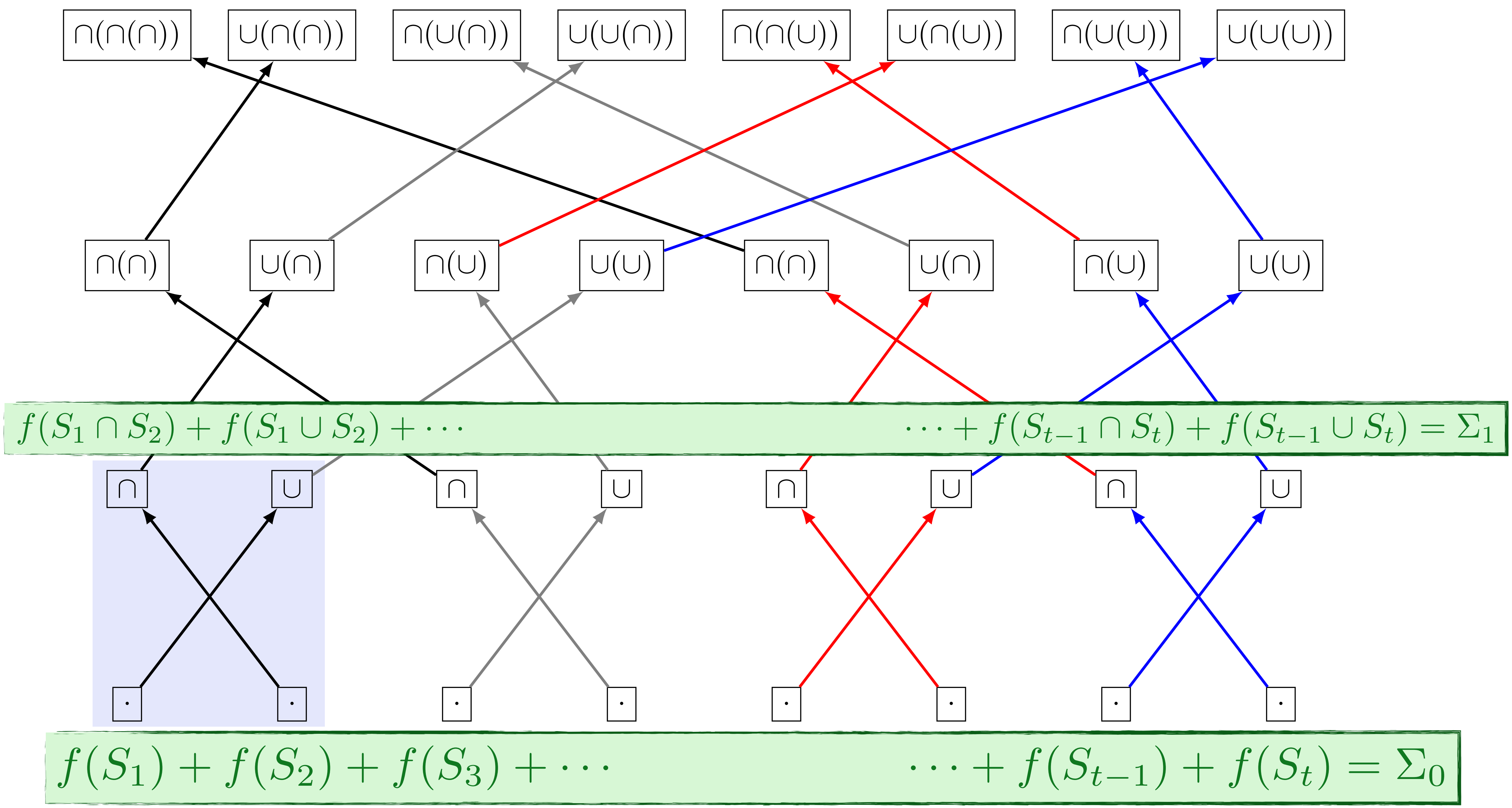


$$f(S_1 \cap S_2) + f(S_1 \cup S_2) + \cdots \quad \cdots + f(S_{t-1} \cap S_t) + f(S_{t-1} \cup S_t) = \Sigma_1$$

$$f(S_1) + f(S_2) + f(S_3) + \cdots \quad \cdots + f(S_{t-1}) + f(S_t) = \Sigma_0$$

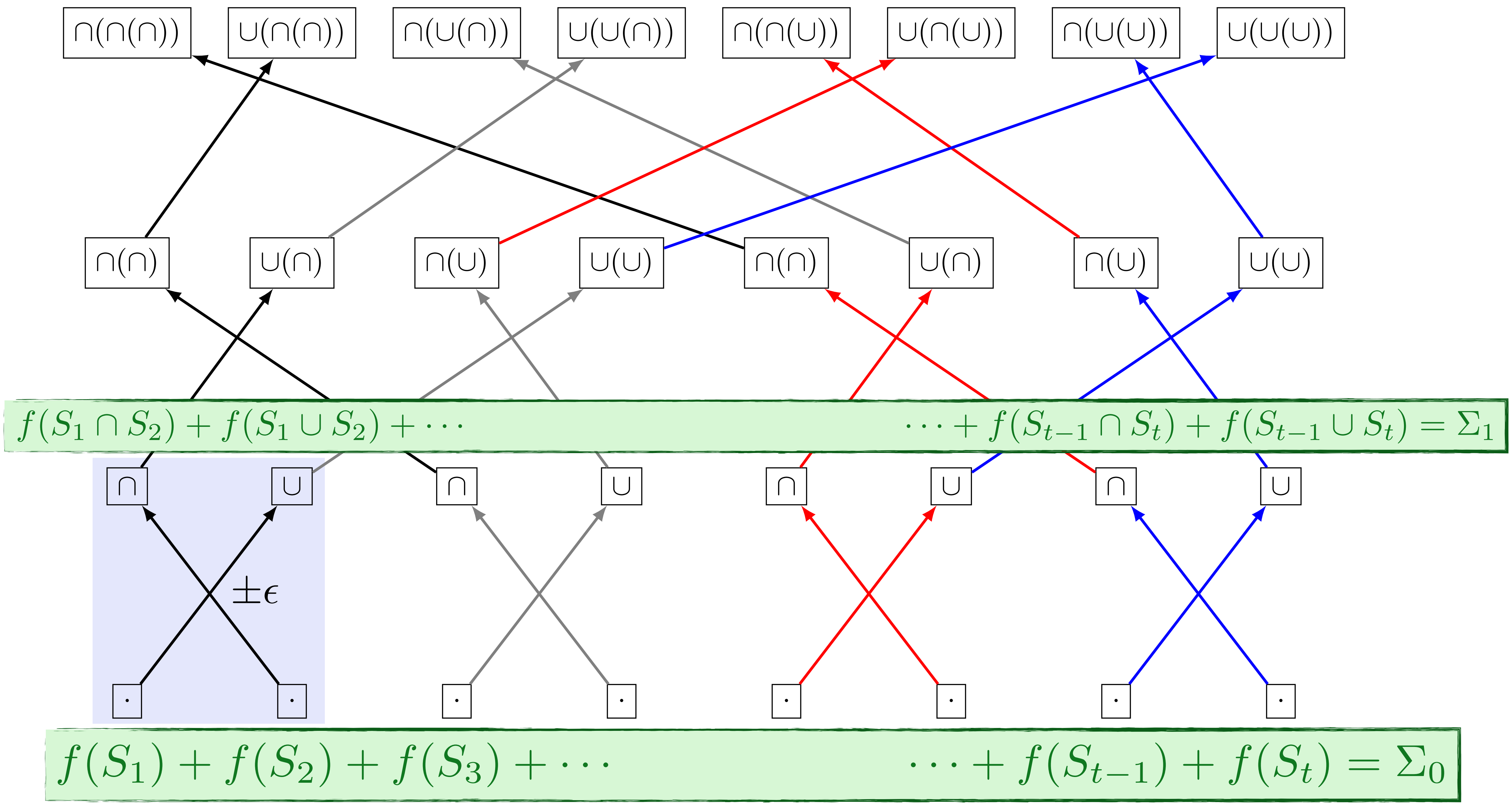
$$f(S \cup T) + f(S \cap T) - \epsilon \leq f(S) + f(T) \leq f(S \cup T) + f(S \cap T) + \epsilon$$

Striped Networks



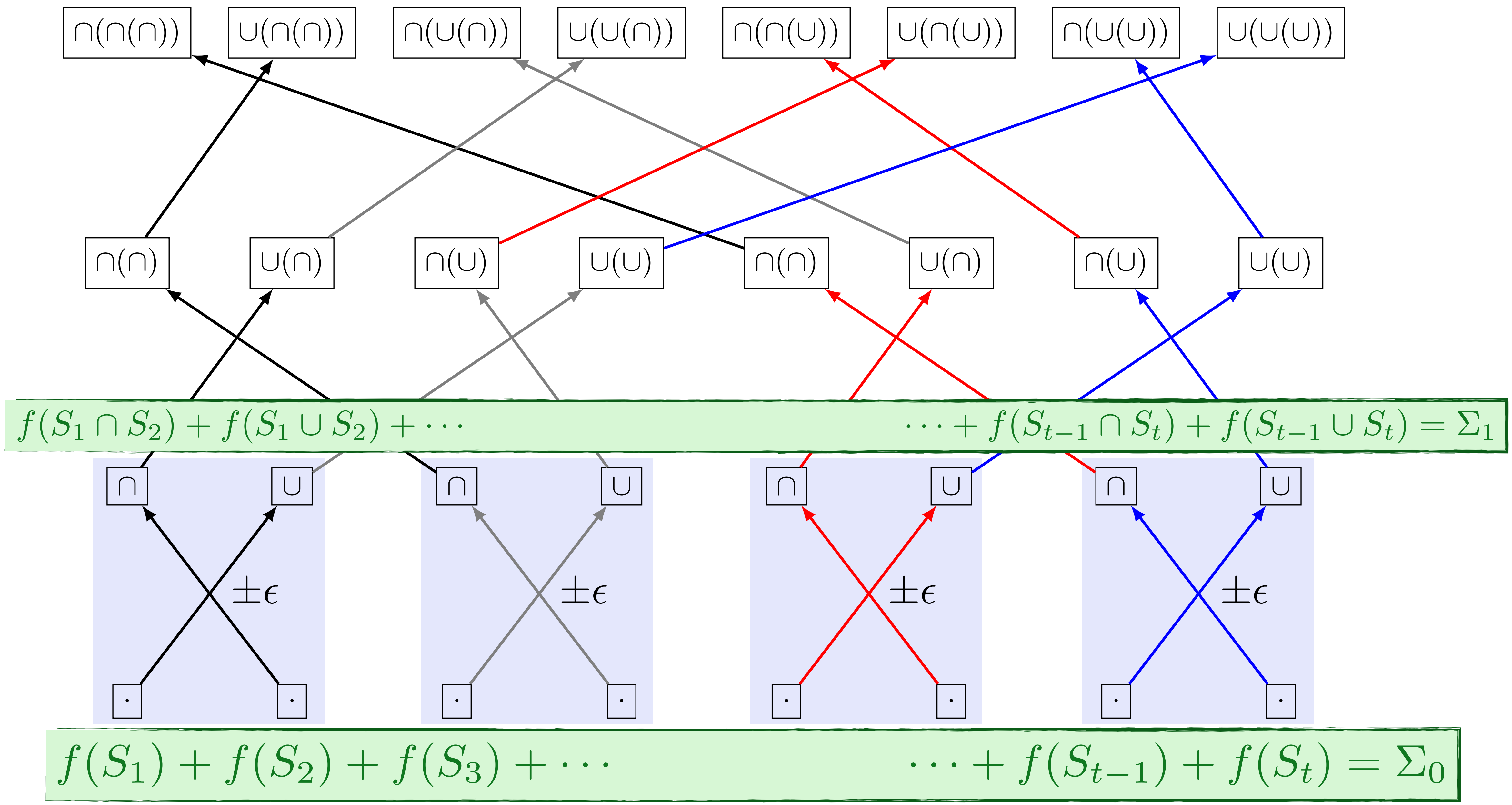
$$f(S \cup T) + f(S \cap T) - \epsilon \leq f(S) + f(T) \leq f(S \cup T) + f(S \cap T) + \epsilon$$

Striped Networks

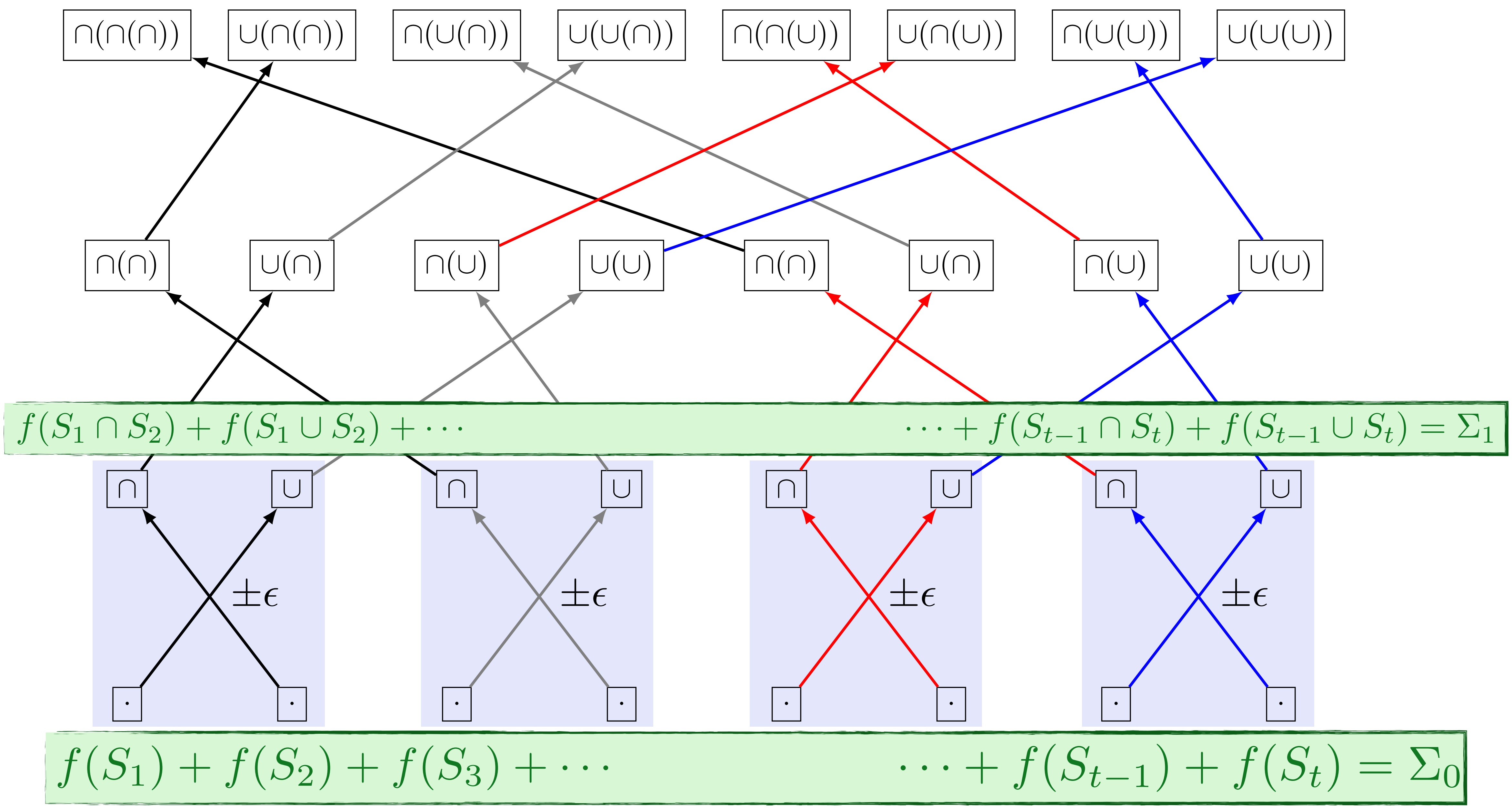


$$f(S \cup T) + f(S \cap T) - \epsilon \leq f(S) + f(T) \leq f(S \cup T) + f(S \cap T) + \epsilon$$

Striped Networks

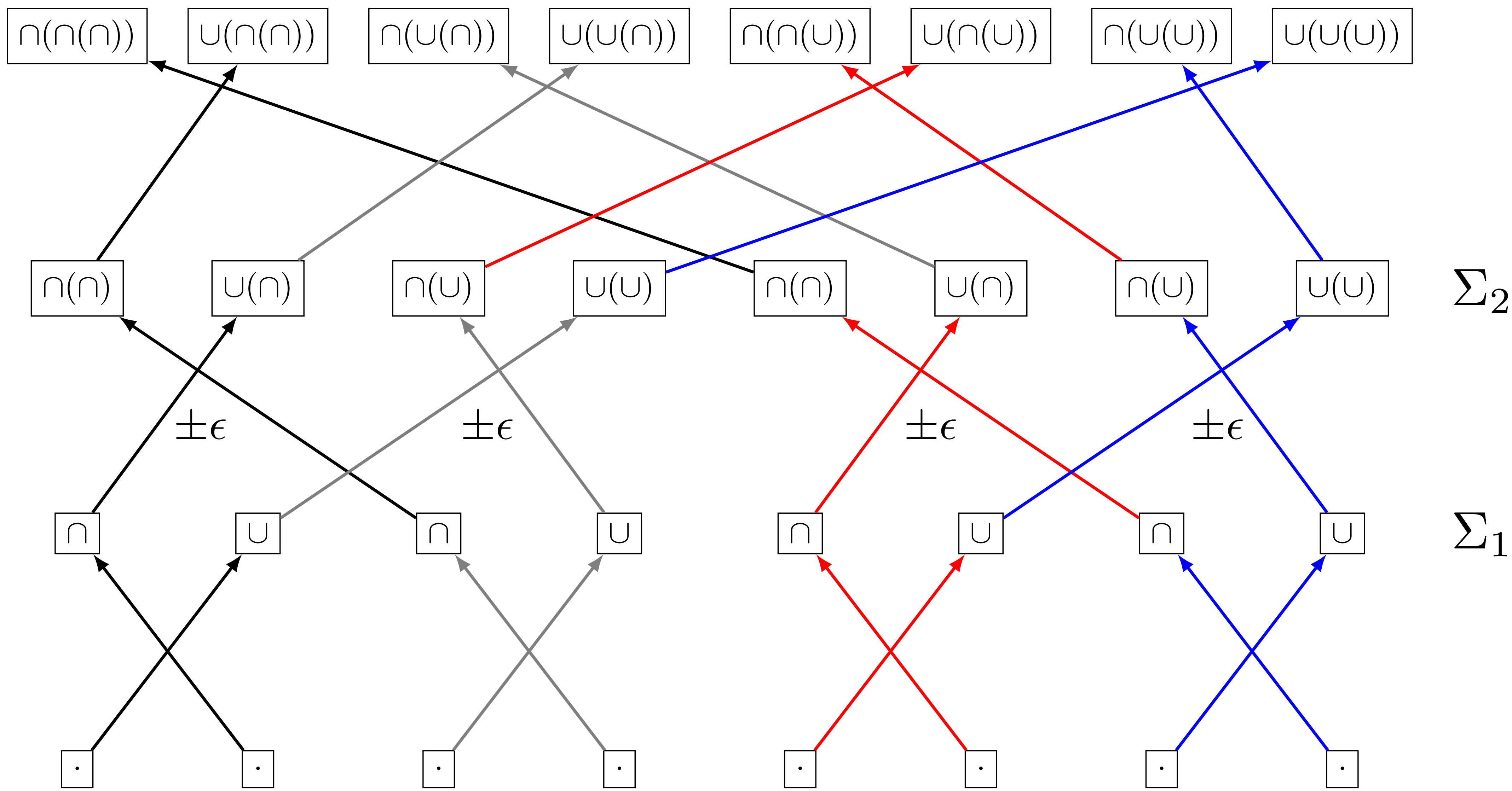


Striped Networks



$$\Sigma_0 = \Sigma_1 \pm \epsilon \cdot {}^{t}\!/_{2}$$

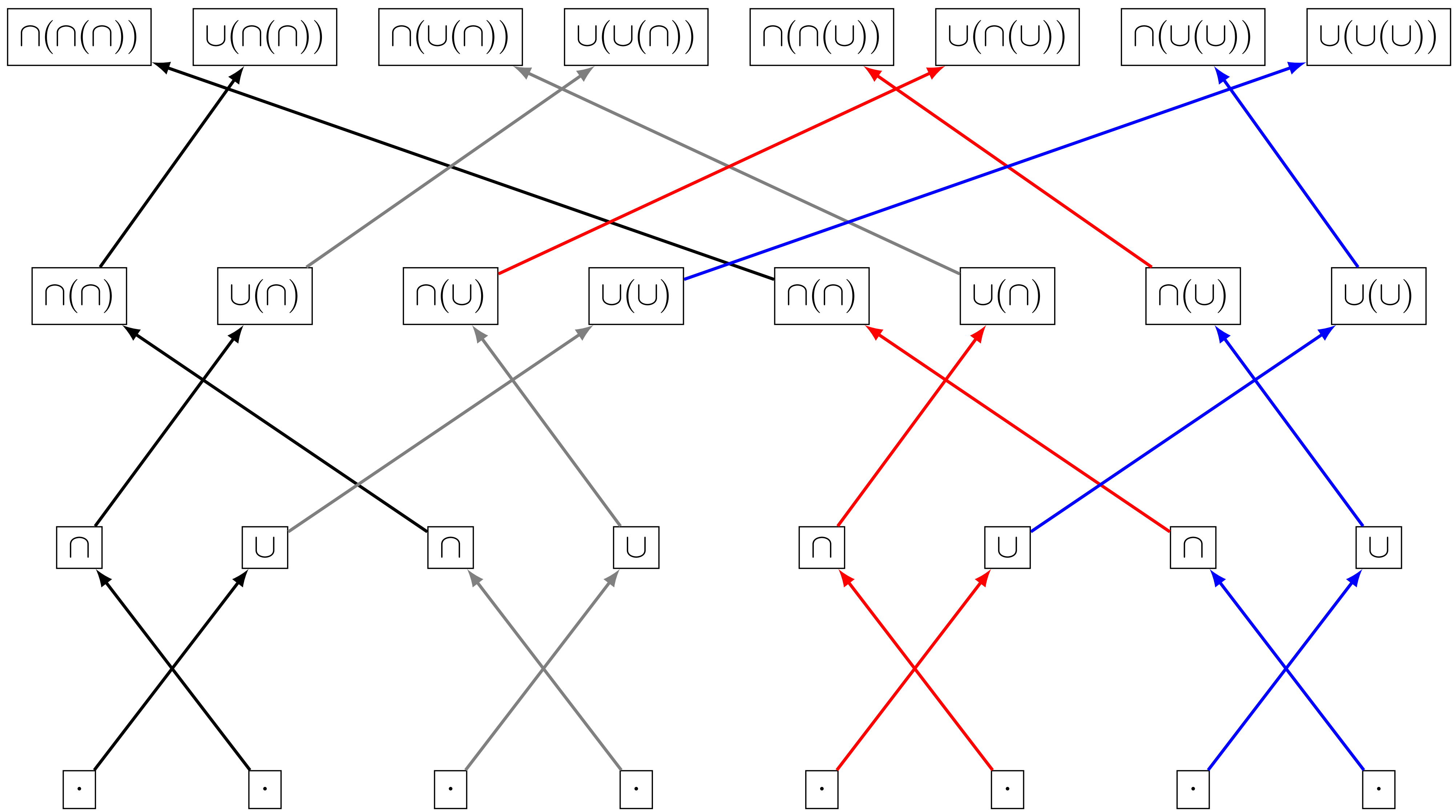
Striped Networks



$\Sigma_0 = \Sigma_1 \pm \epsilon \cdot {}^{t}\!/_{2}$ $\qquad$ $\Sigma_1 = \Sigma_2 \pm \epsilon \cdot {}^{t}\!/_{2}$

Striped Networks

Striped networks "lose" an additive $\epsilon/2$ average term per level

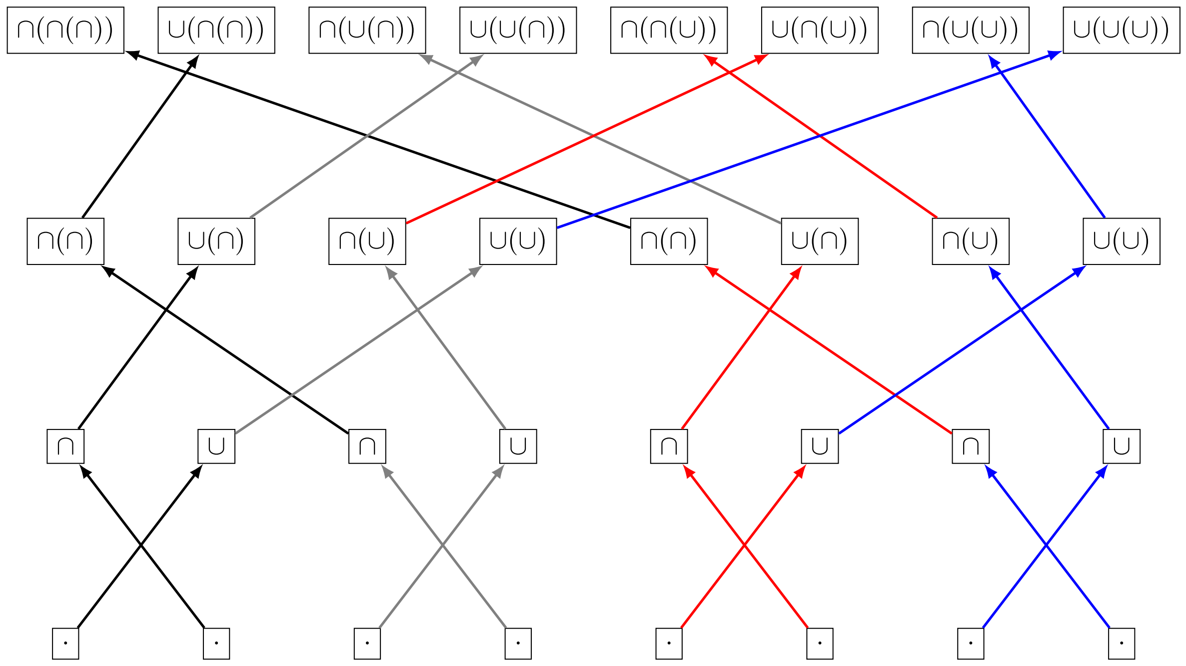


$\Sigma_0 = \Sigma_1 \pm \epsilon \cdot t/2$ $\Sigma_1 = \Sigma_2 \pm \epsilon \cdot t/2$ $\cdots$ $\Sigma_{h-1} = \Sigma_h \pm \epsilon \cdot t/2$

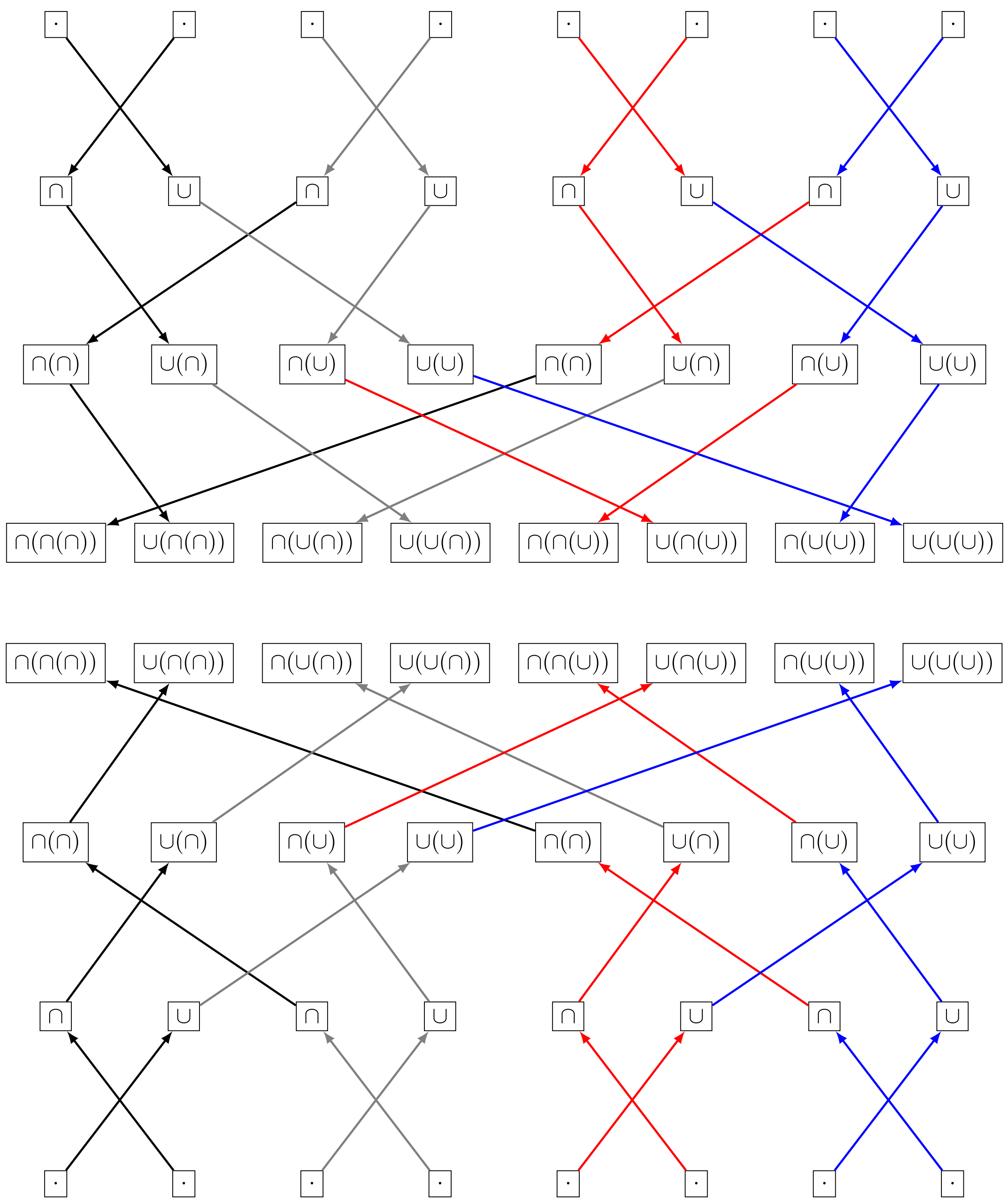
Bounding M

We use striped networks to bound the maximum value M of the ϵ-approximately modular functions F that are best approximated by z

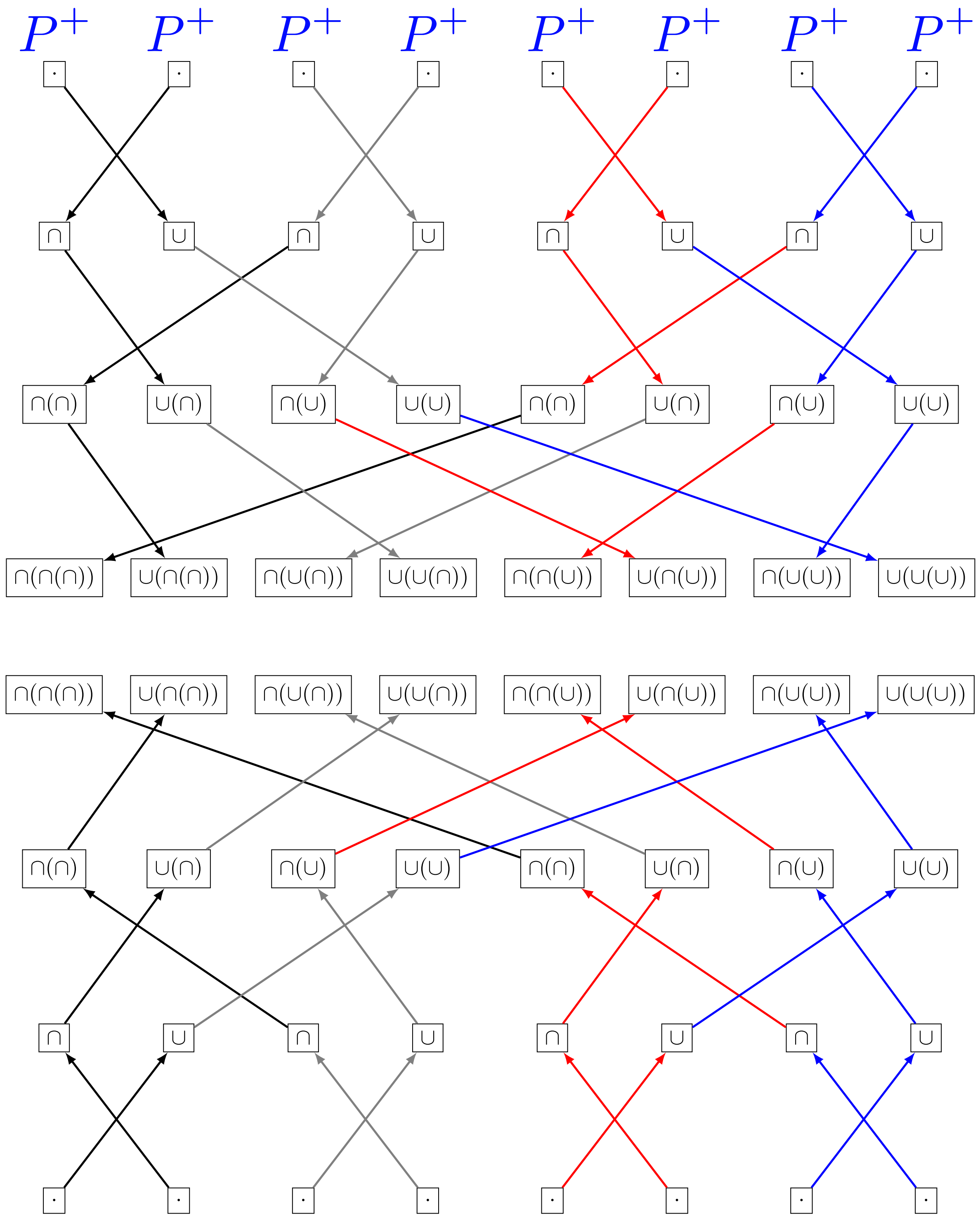
Bounding M



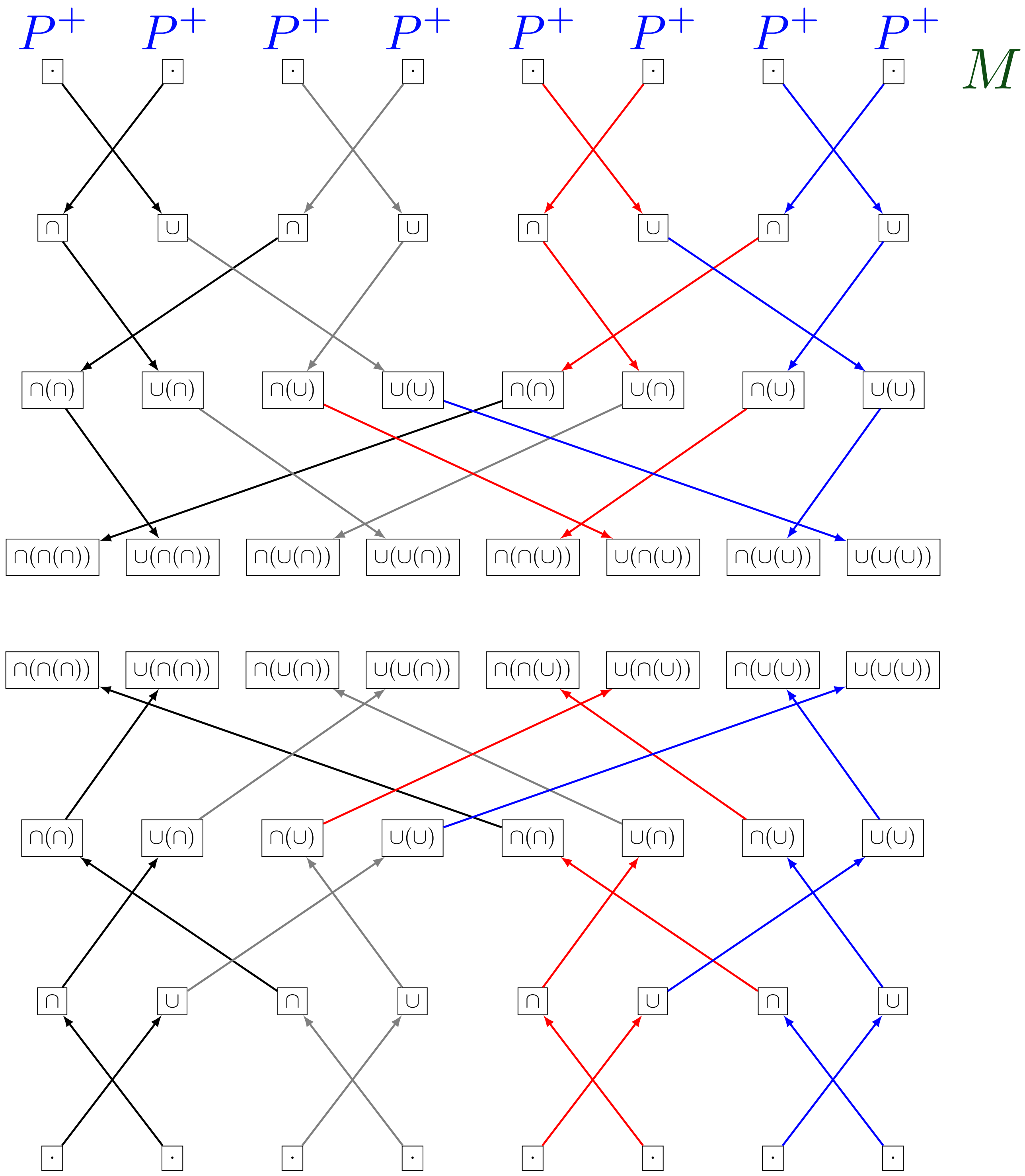
Bounding M



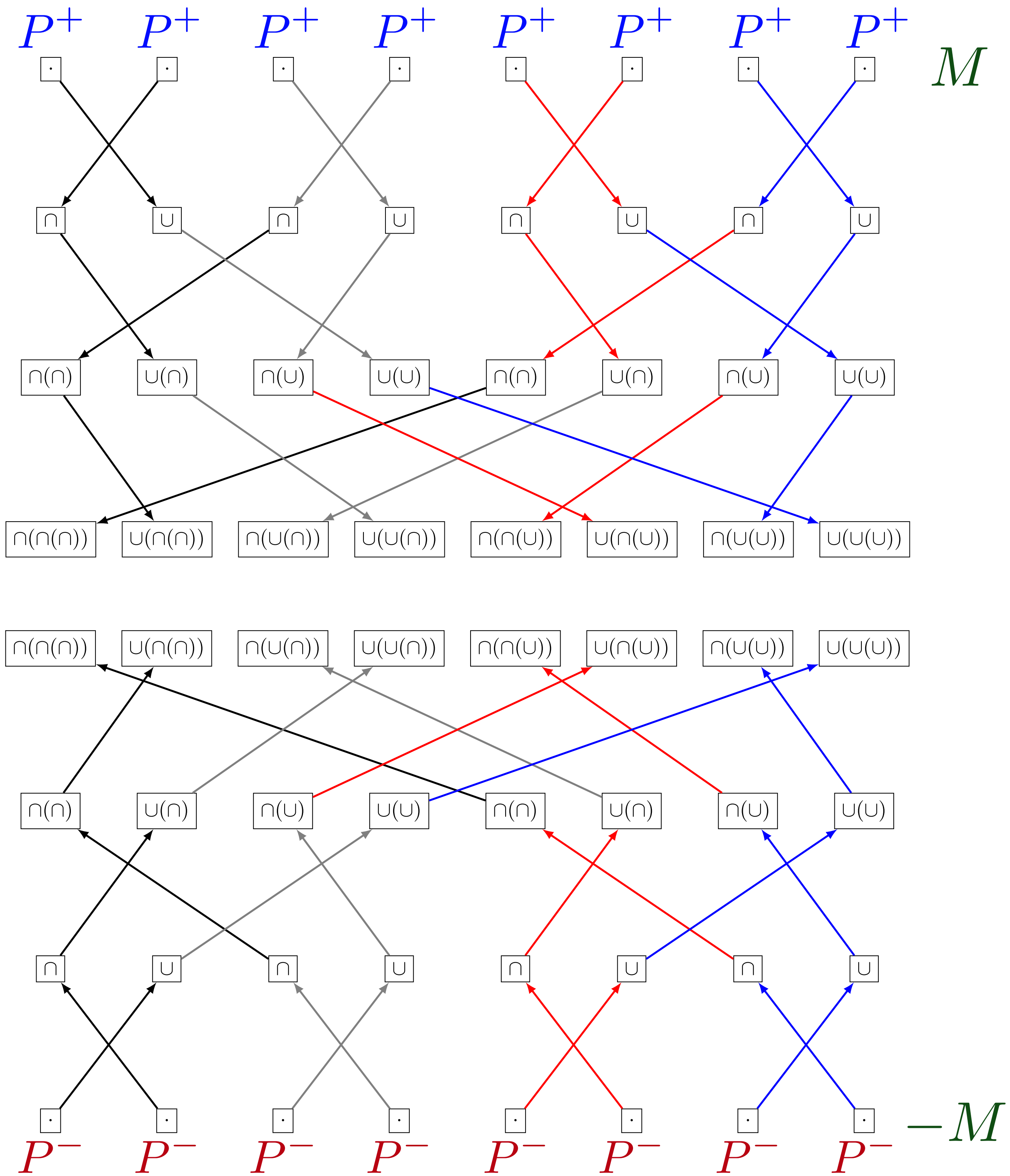
Bounding M



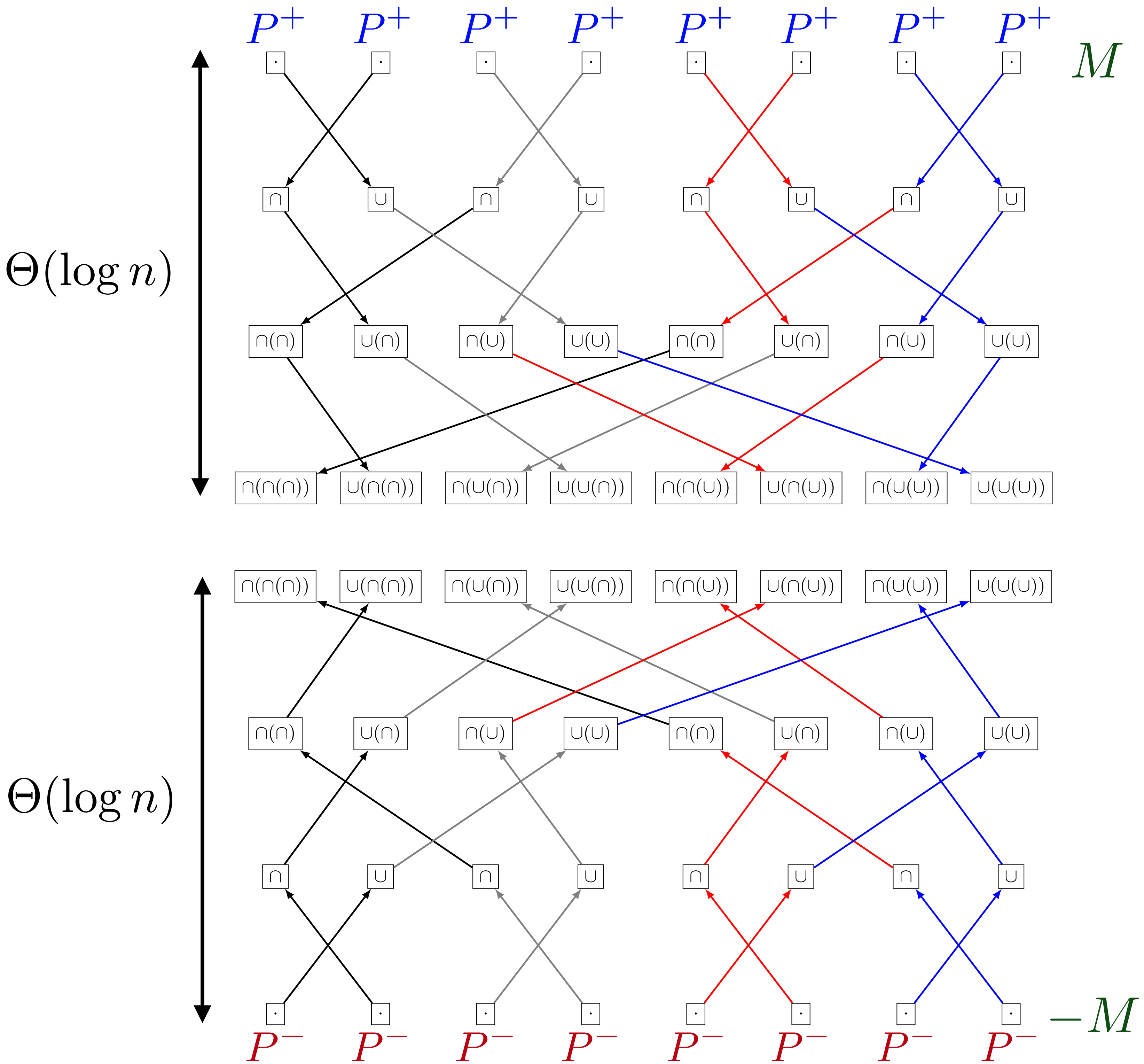
Bounding M



Bounding M

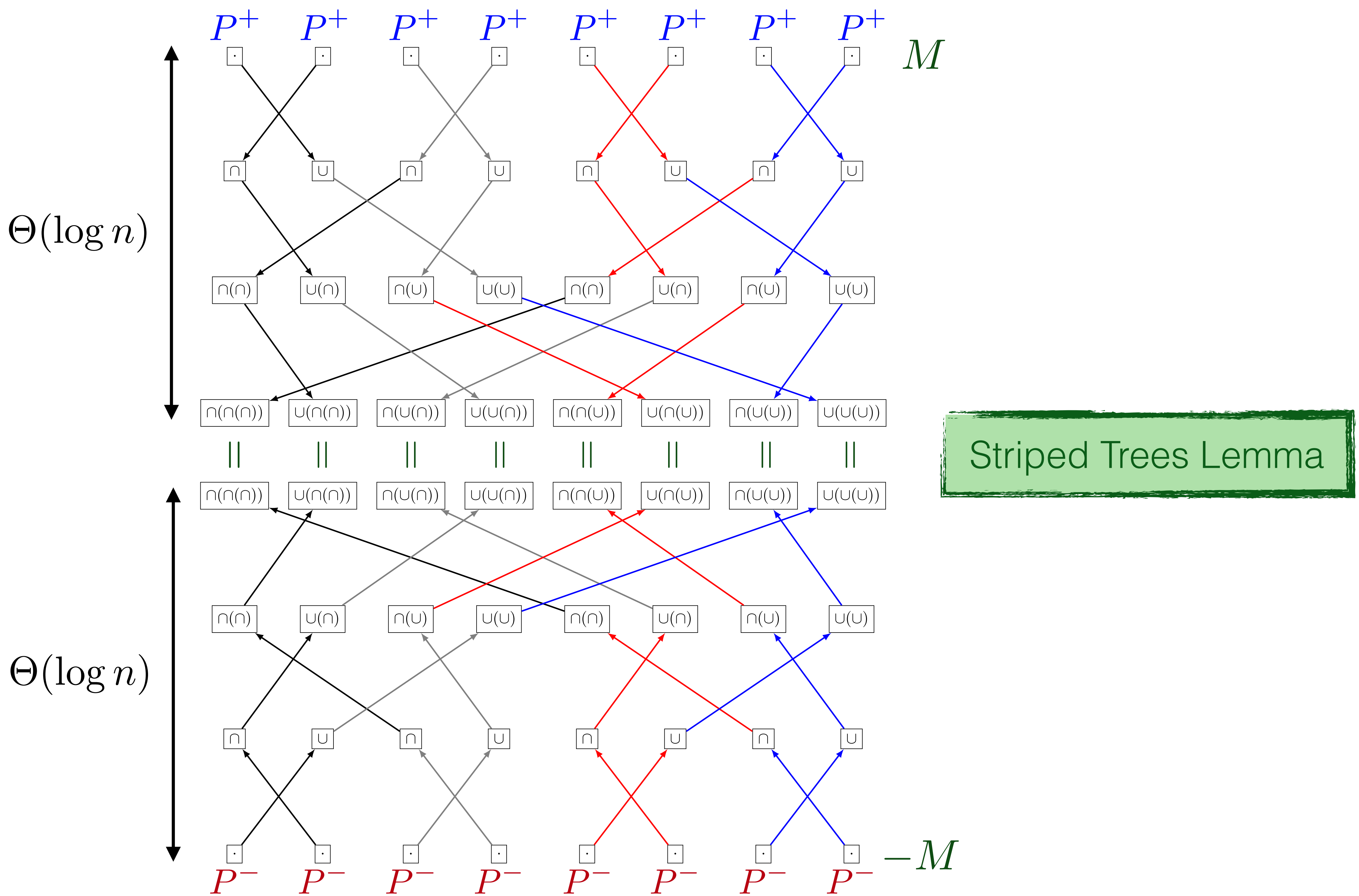


Bounding M



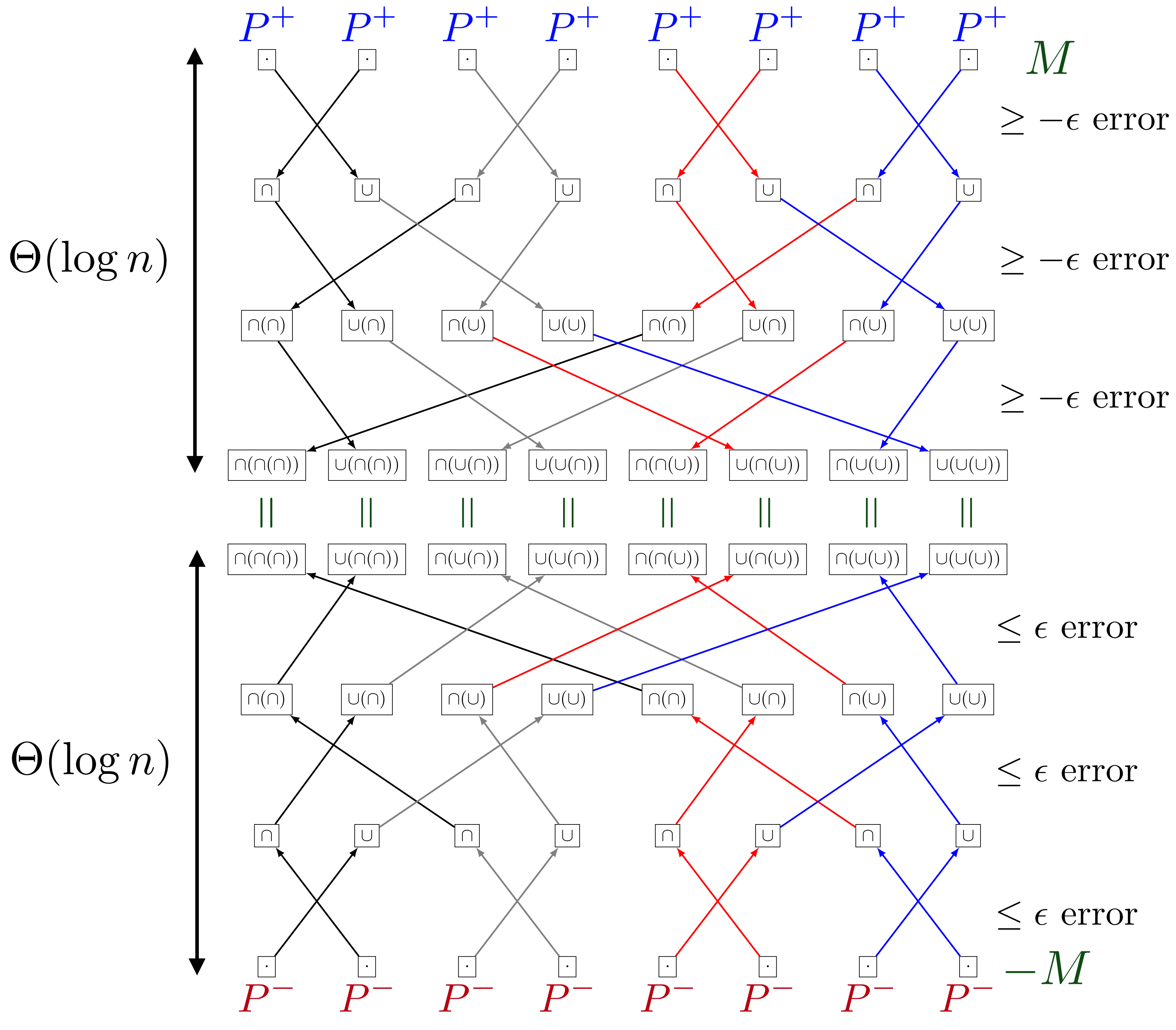
P^+ P^+ P^+ P^+ P^+ P^+ P^+ P^+
M
$\Theta(\log n)$
$\cap$ $\cup$ $\cap$ $\cup$ $\cap$ $\cup$ $\cap$ $\cup$
$\cap(\cap)$ $\cup(\cap)$ $\cap(\cup)$ $\cup(\cup)$ $\cap(\cap)$ $\cup(\cap)$ $\cap(\cup)$ $\cup(\cup)$
$\cap(\cap(\cap))$ $\cup(\cap(\cap))$ $\cap(\cup(\cap))$ $\cup(\cup(\cap))$ $\cap(\cap(\cup))$ $\cup(\cap(\cup))$ $\cap(\cup(\cup))$ $\cup(\cup(\cup))$
$\cap(\cap(\cap))$ $\cup(\cap(\cap))$ $\cap(\cup(\cap))$ $\cup(\cup(\cap))$ $\cap(\cap(\cup))$ $\cup(\cap(\cup))$ $\cap(\cup(\cup))$ $\cup(\cup(\cup))$
$\Theta(\log n)$
$\cap(\cap)$ $\cup(\cap)$ $\cap(\cup)$ $\cup(\cup)$ $\cap(\cap)$ $\cup(\cap)$ $\cap(\cup)$ $\cup(\cup)$
$\cap$ $\cup$ $\cap$ $\cup$ $\cap$ $\cup$ $\cap$ $\cup$
$-M$
P^- P^- P^- P^- P^- P^- P^- P^-

Bounding M



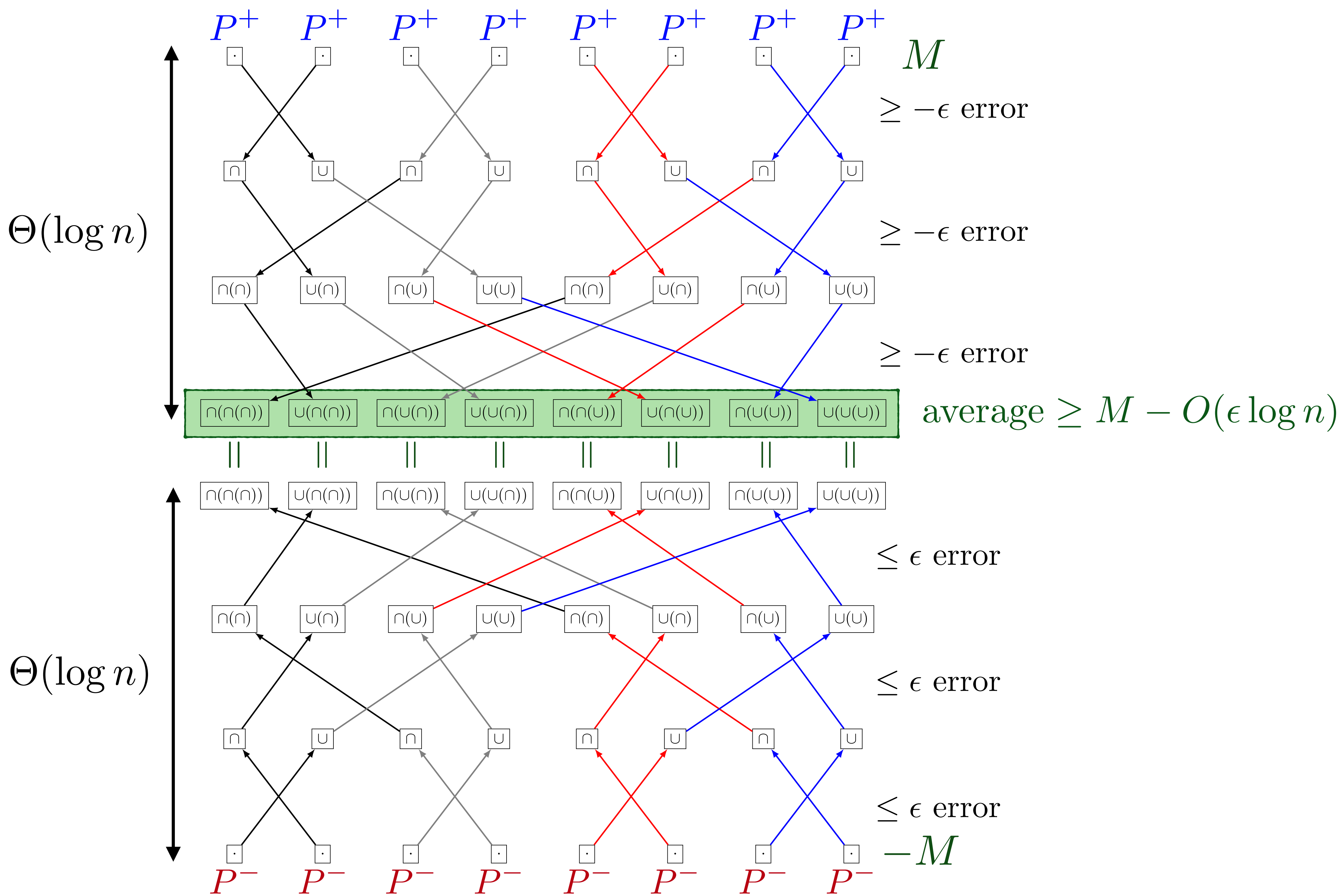
P^+ P^+ P^+ P^+ P^+ P^+ P^+ P^+
M
$\Theta(\log n)$
∩ ∪ ∩ ∪ ∩ ∪ ∩ ∪
∩(∩) ∪(∩) ∩(∪) ∪(∪) ∩(∩) ∪(∩) ∩(∪) ∪(∪)
∩(∩(∩)) ∪(∩(∩)) ∩(∪(∩)) ∪(∪(∩)) ∩(∩(∪)) ∪(∩(∪)) ∩(∪(∪)) ∪(∪(∪))
Striped Trees Lemma
∩(∩(∩)) ∪(∩(∩)) ∩(∪(∩)) ∪(∪(∩)) ∩(∩(∪)) ∪(∩(∪)) ∩(∪(∪)) ∪(∪(∪))
$\Theta(\log n)$
∩(∩) ∪(∩) ∩(∪) ∪(∪) ∩(∩) ∪(∩) ∩(∪) ∪(∪)
∩ ∪ ∩ ∪ ∩ ∪ ∩ ∪
$-M$
P^- P^- P^- P^- P^- P^- P^- P^-

Bounding M



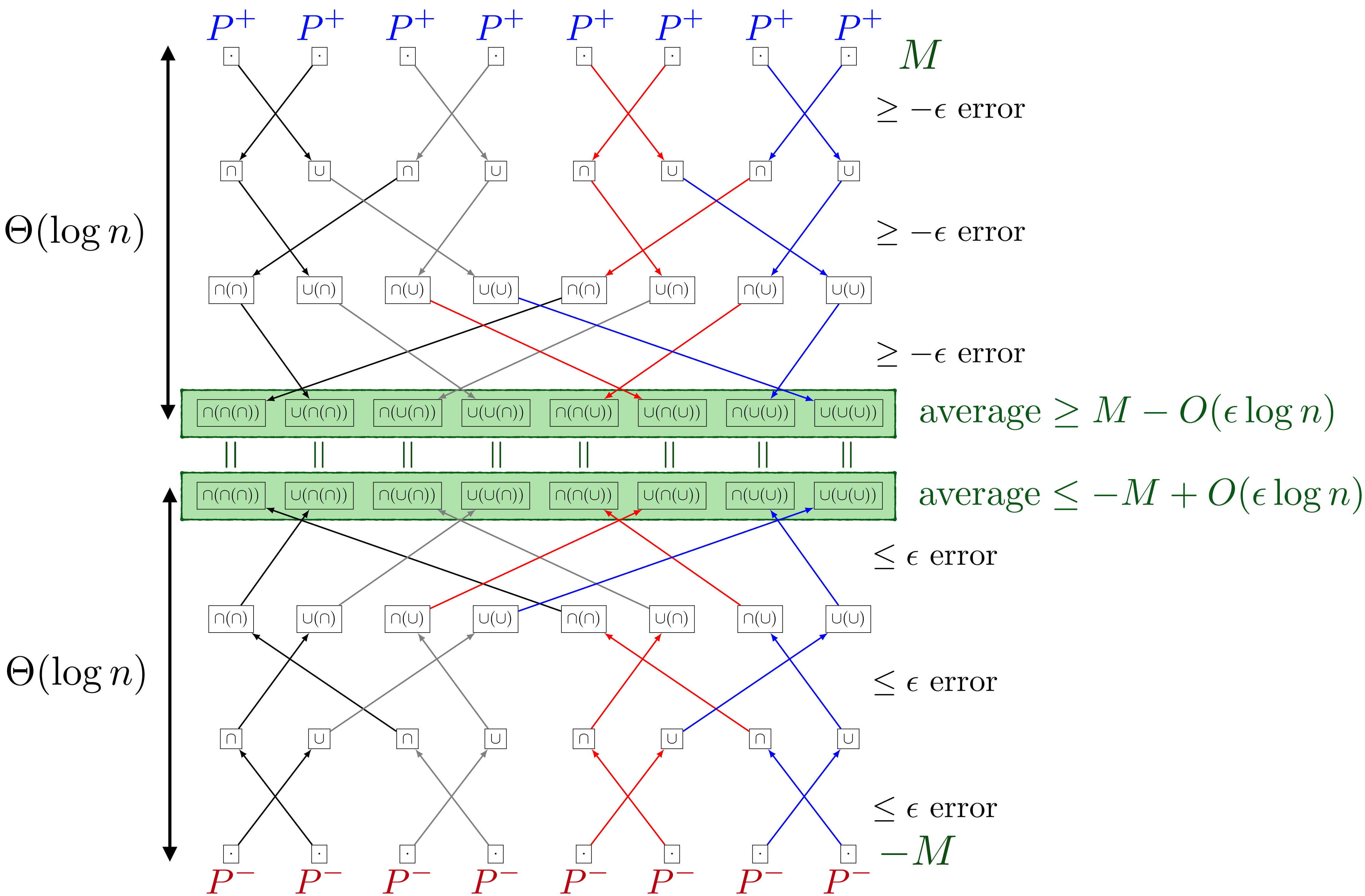
P^+ P^+ P^+ P^+ P^+ P^+ P^+ P^+
M
$\geq -\epsilon$ error
$\geq -\epsilon$ error
$\geq -\epsilon$ error
$\Theta(\log n)$
∩ ∪ ∩ ∪ ∩ ∪ ∩ ∪
∩(∩) ∪(∩) ∩(∪) ∪(∪) ∩(∩) ∪(∩) ∩(∪) ∪(∪)
∩(∩(∩)) ∪(∩(∩)) ∩(∪(∩)) ∪(∪(∩)) ∩(∩(∪)) ∪(∩(∪)) ∩(∪(∪)) ∪(∪(∪))
∩(∩(∩)) ∪(∩(∩)) ∩(∪(∩)) ∪(∪(∩)) ∩(∩(∪)) ∪(∩(∪)) ∩(∪(∪)) ∪(∪(∪))
$\leq \epsilon$ error
$\leq \epsilon$ error
$\leq \epsilon$ error
$\Theta(\log n)$
∩(∩) ∪(∩) ∩(∪) ∪(∪) ∩(∩) ∪(∩) ∩(∪) ∪(∪)
∩ ∪ ∩ ∪ ∩ ∪ ∩ ∪
$-M$
P^- P^- P^- P^- P^- P^- P^- P^-

Bounding M



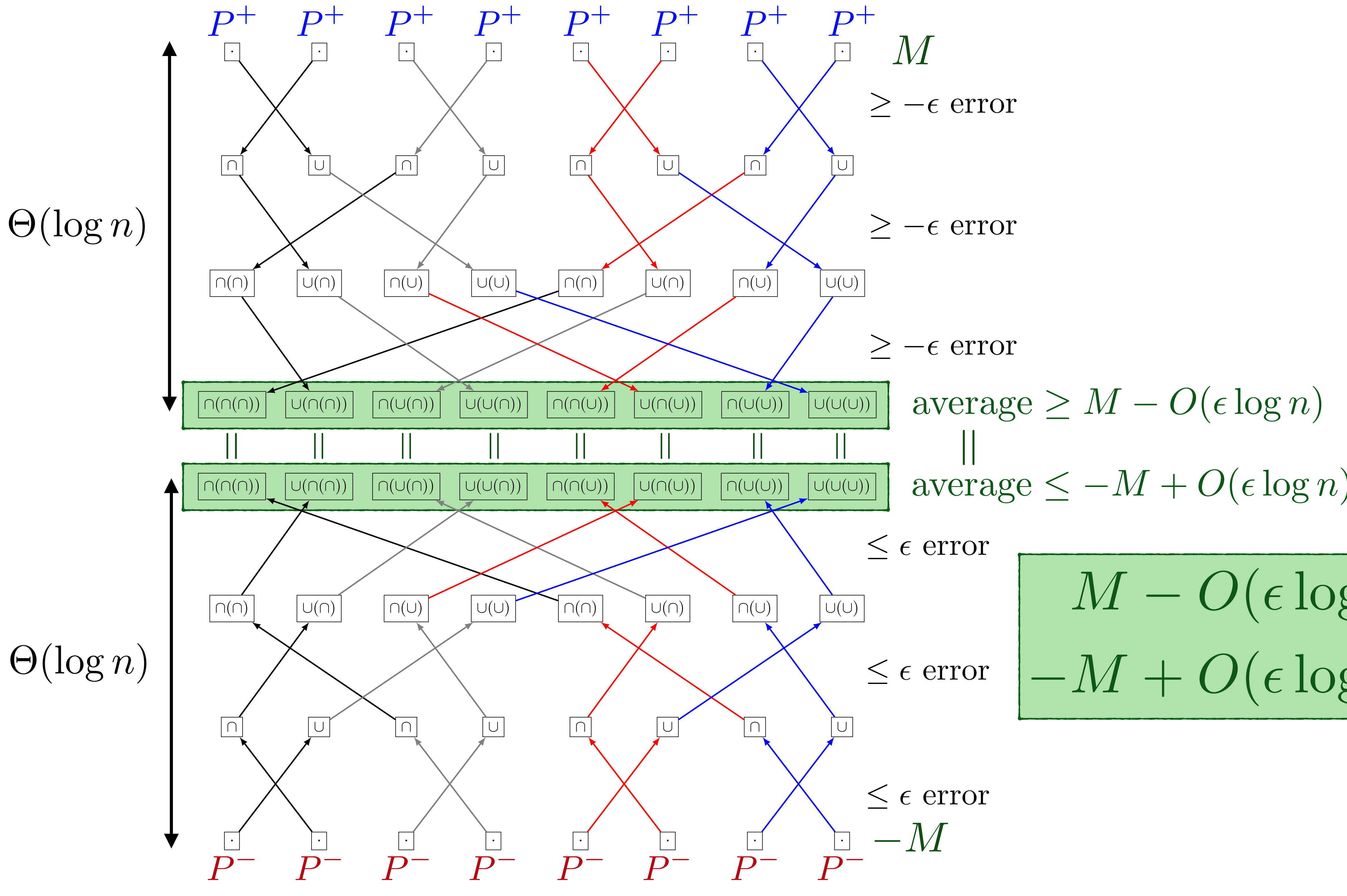
P+ P+ P+ P+ P+ P+ P+ P+
M
≥ −ϵ error
≥ −ϵ error
≥ −ϵ error
Θ(log n)
∩ ∪ ∩ ∪ ∩ ∪ ∩ ∪
∩(∩) ∪(∩) ∩(∪) ∪(∪) ∩(∩) ∪(∩) ∩(∪) ∪(∪)
∩(∩(∩)) ∪(∩(∩)) ∩(∪(∩)) ∪(∪(∩)) ∩(∩(∪)) ∪(∩(∪)) ∩(∪(∪)) ∪(∪(∪))
average ≥ M − O(ϵ log n)
∩(∩(∩)) ∪(∩(∩)) ∩(∪(∩)) ∪(∪(∩)) ∩(∩(∪)) ∪(∩(∪)) ∩(∪(∪)) ∪(∪(∪))
≤ ϵ error
∩(∩) ∪(∩) ∩(∪) ∪(∪) ∩(∩) ∪(∩) ∩(∪) ∪(∪)
≤ ϵ error
Θ(log n)
∩ ∪ ∩ ∪ ∩ ∪ ∩ ∪
≤ ϵ error
−M
P− P− P− P− P− P− P− P−

Bounding M



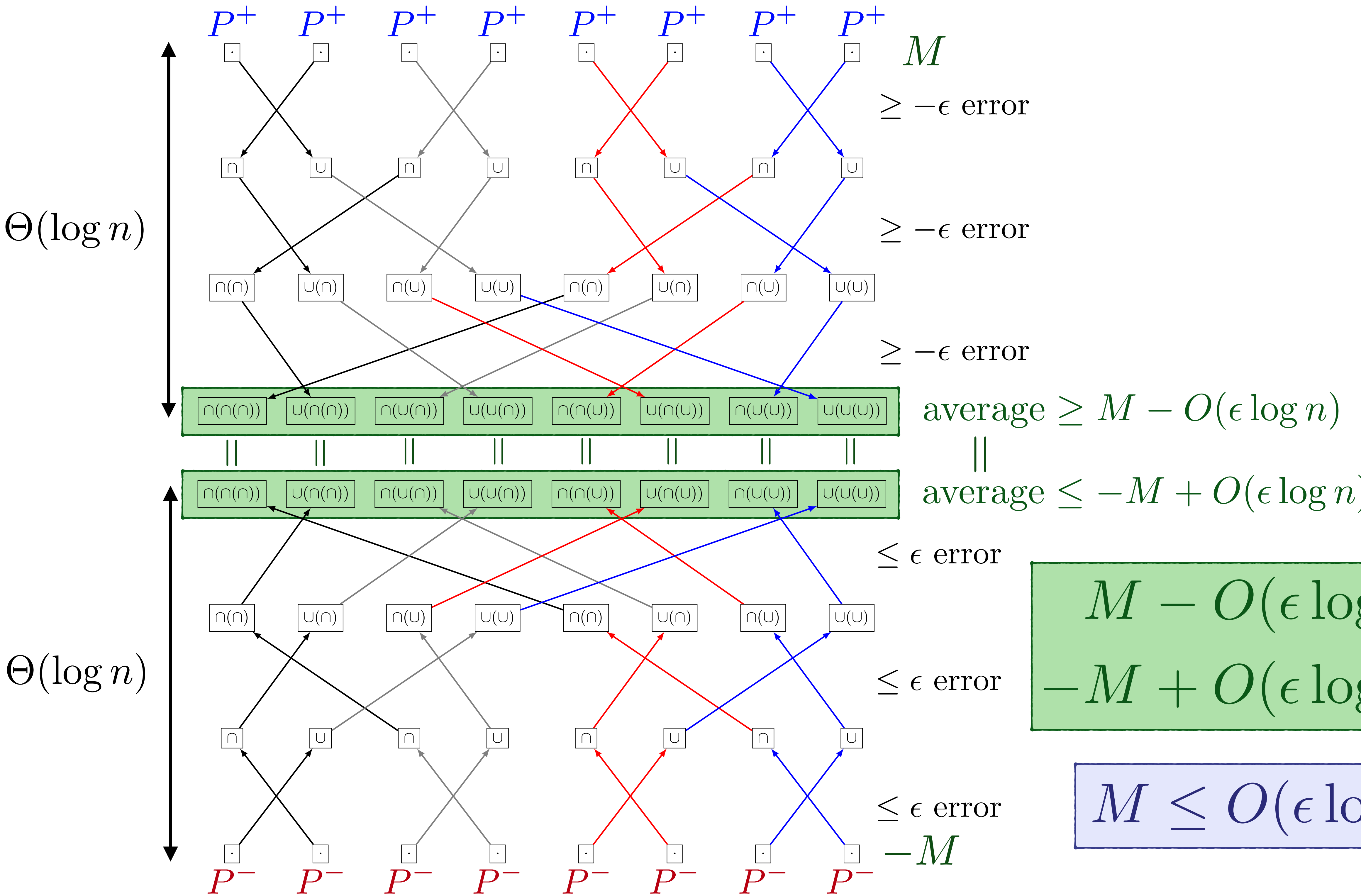
P+ P+ P+ P+ P+ P+ P+ P+
M
≥ −ϵ error
≥ −ϵ error
≥ −ϵ error
Θ(log n)
average ≥ M − O(ϵ log n)
average ≤ −M + O(ϵ log n)
≤ ϵ error
≤ ϵ error
≤ ϵ error
Θ(log n)
−M
P− P− P− P− P− P− P− P−

Bounding M



$$M - O(\epsilon \log n) \leq -M + O(\epsilon \log n)$$

Bounding M



$$M - O(\epsilon \log n) \leq -M + O(\epsilon \log n)$$

$$M \leq O(\epsilon \log n)$$

Closeness to Modularity

- …thus, the maximum value of an ϵ-approximately modular function *F* on *[n]* (that is best approximated by the all-0 function *z*) is $O(\epsilon \cdot \log n)$.

Closeness to Modularity

- …thus, the maximum value of an ϵ-approximately modular function *F* on *[n]* (that is best approximated by the all-0 function *z*) is $O(\epsilon \cdot \log n)$.

- It follows that, if *f* is an ϵ-approximately modular function on *[n]*, there exists a modular function *g* such that

$$\forall S \subseteq [n] \quad |f(S) - g(S)| \leq O(\epsilon \cdot \log n)$$

Conclusion

- We have studied
  - the polynomial-time approximability, and
  - the Borsuk-Ulam approximability,

  of *approximately modular functions.*

Conclusion

- We have studied
  - the polynomial-time approximability, and
  - the Borsuk-Ulam approximability,

  of *approximately modular functions.*
- **Open questions**
  - Is our logarithmic upper bound on the distance to modularity tight?
  - What happens for functions that are (additively) approximately sub-modular?

Thanks!