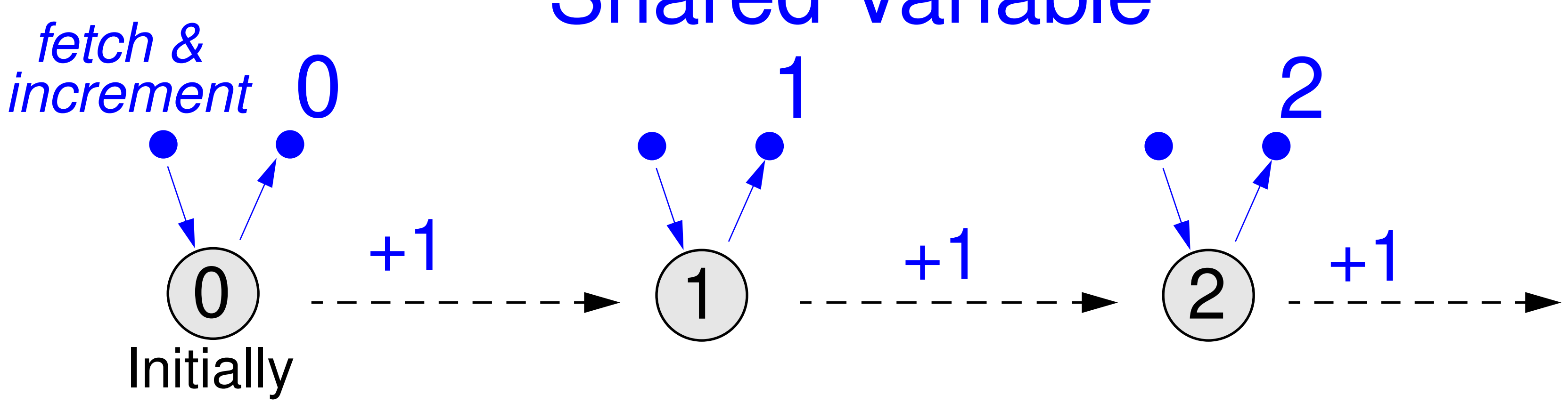
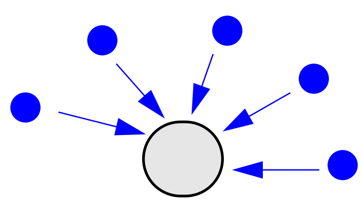
Distributed Counting

- Many distributed processes
- Operation: *fetch & increment*

Shared Variable



Disadvantages:



Centralized

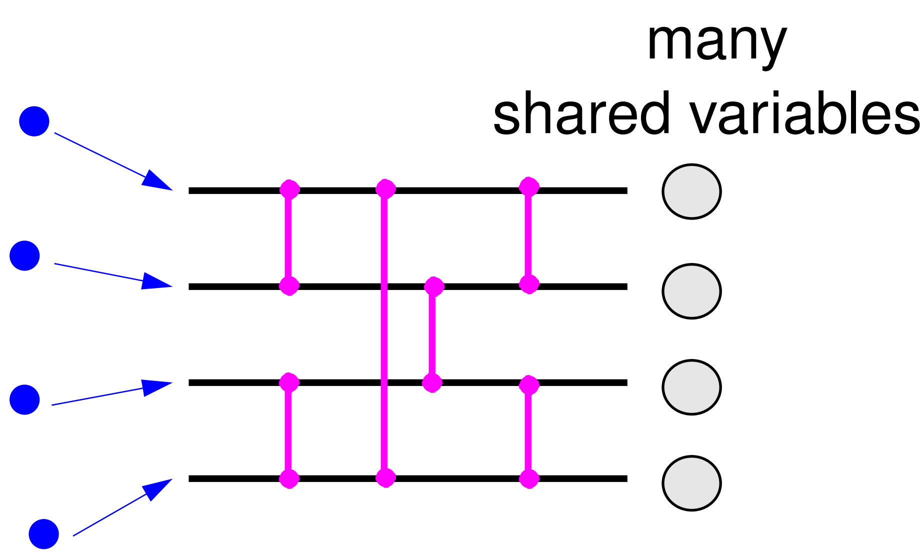
High contention

Blocking

Counting Network

[Aspnes, Herlihy, and Shavit 91]

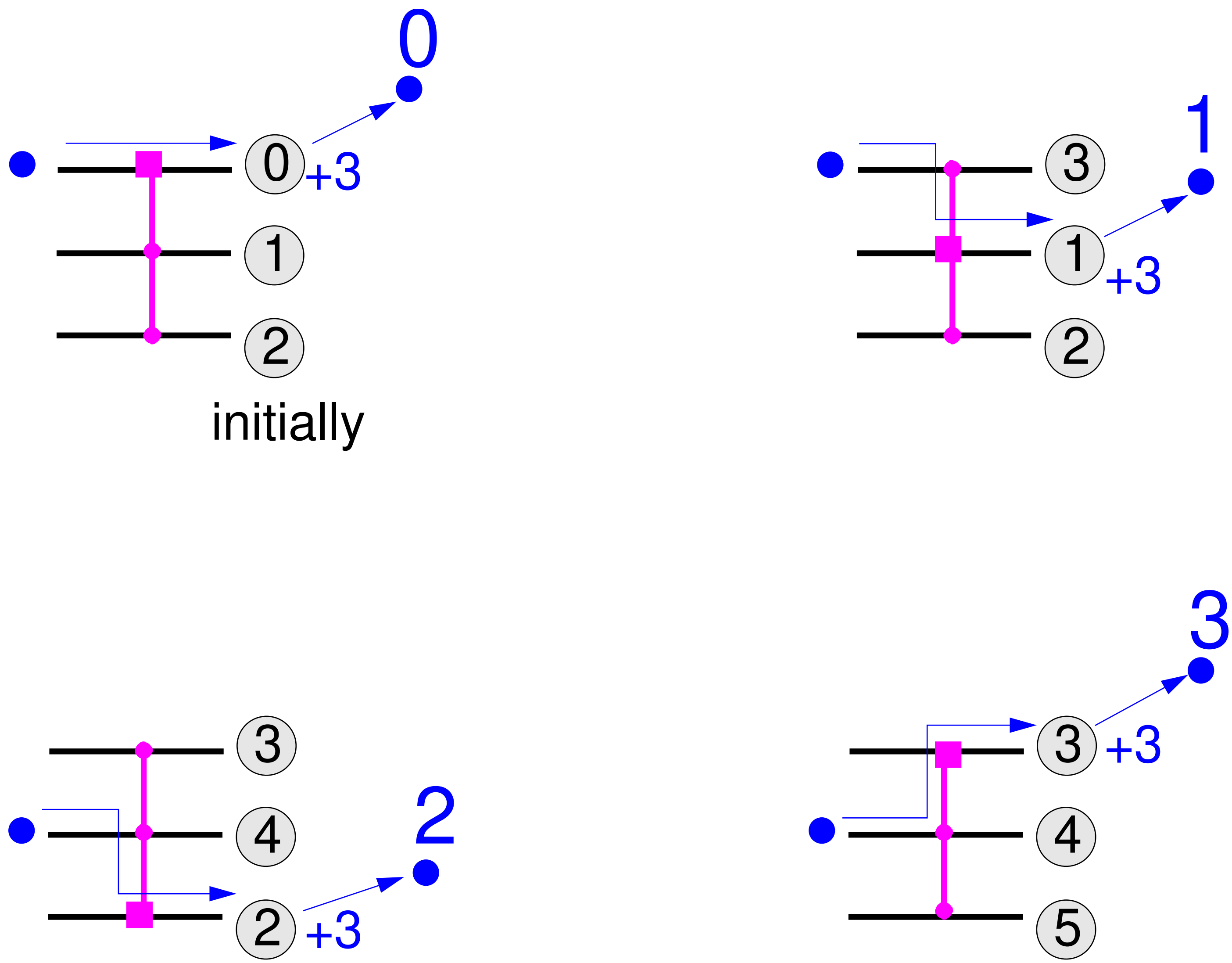
An efficient solution for the distributed counting problem



Advantages:

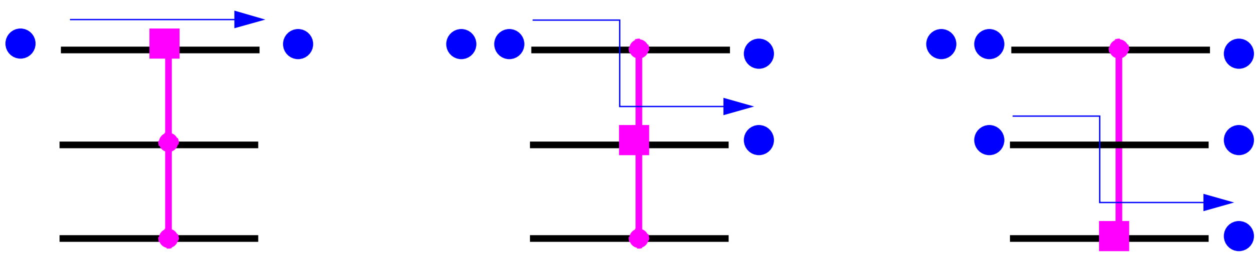
- Highly distributed
- Low contention
- Non-Blocking

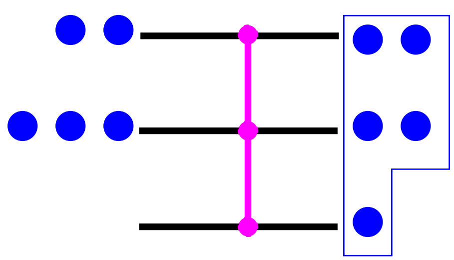
Balancer



Step Property

Guarantees correct counting



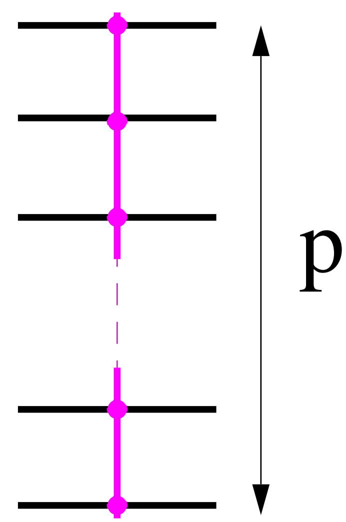


$2 = y_1$

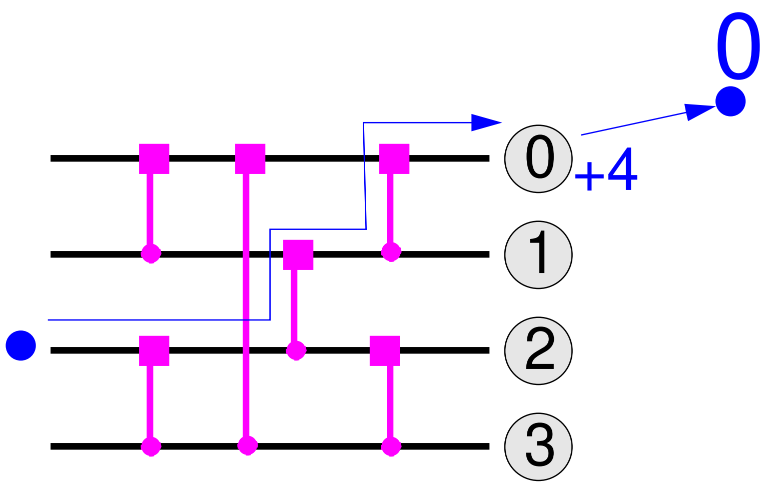
$2 = y_2$

$1 = y_3$

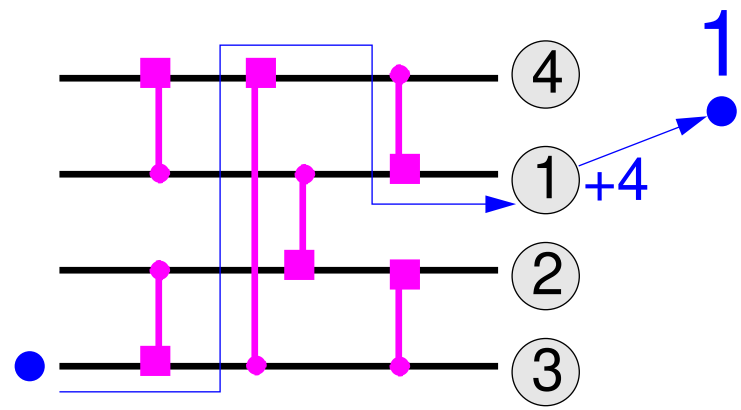
$0 \le y_i - y_j \le 1$, for $i < j$

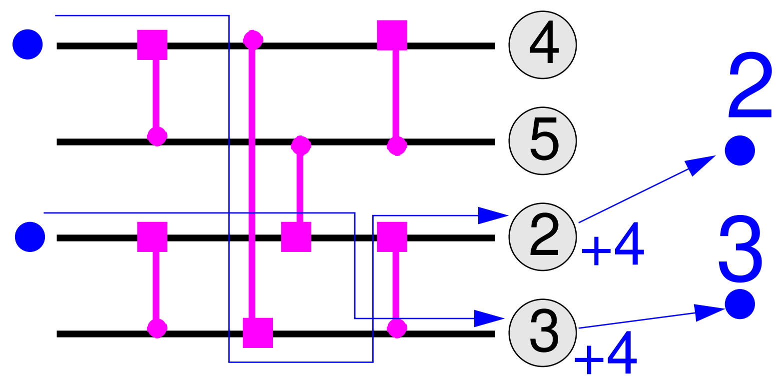


Counting Network (Bitonic [AHS 91])



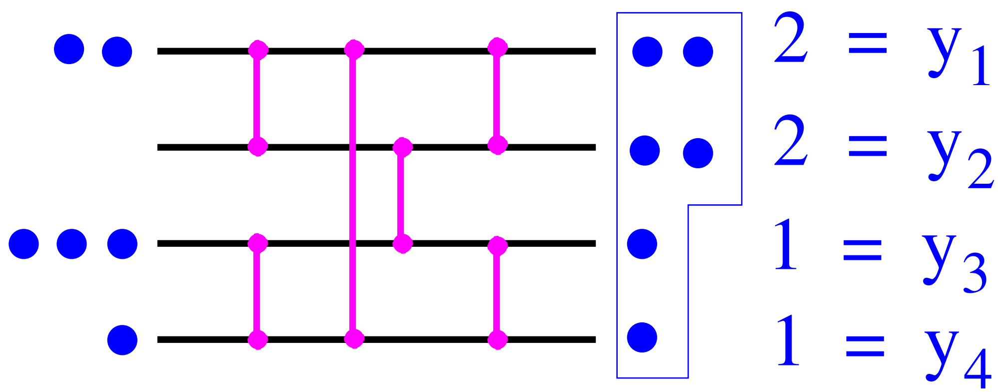
initially





Step Property

Guarantees correct counting

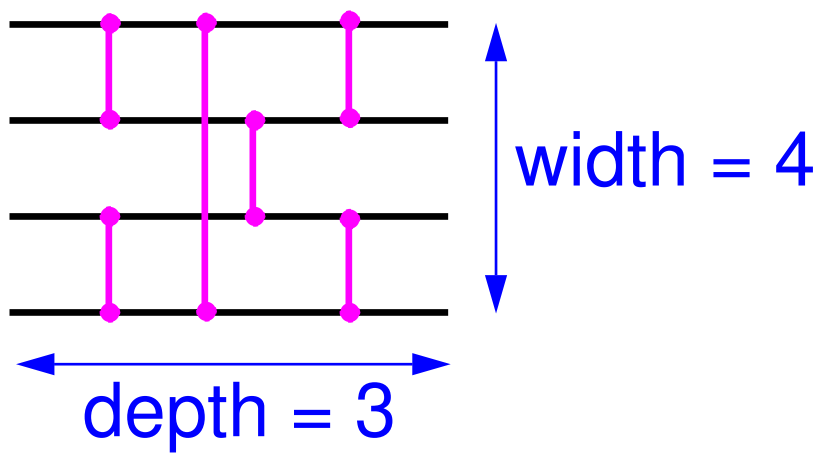


$$0 \le y_i - y_j \le 1, \text{ for } i < j$$

Important Network Features

Width: number of wires

Depth: maximum number of balancers in any path from an input to an output wire



Small depth $\Rightarrow$ Small access latency

An Arbitrary-Width Counting Network

The first **practical** construction:

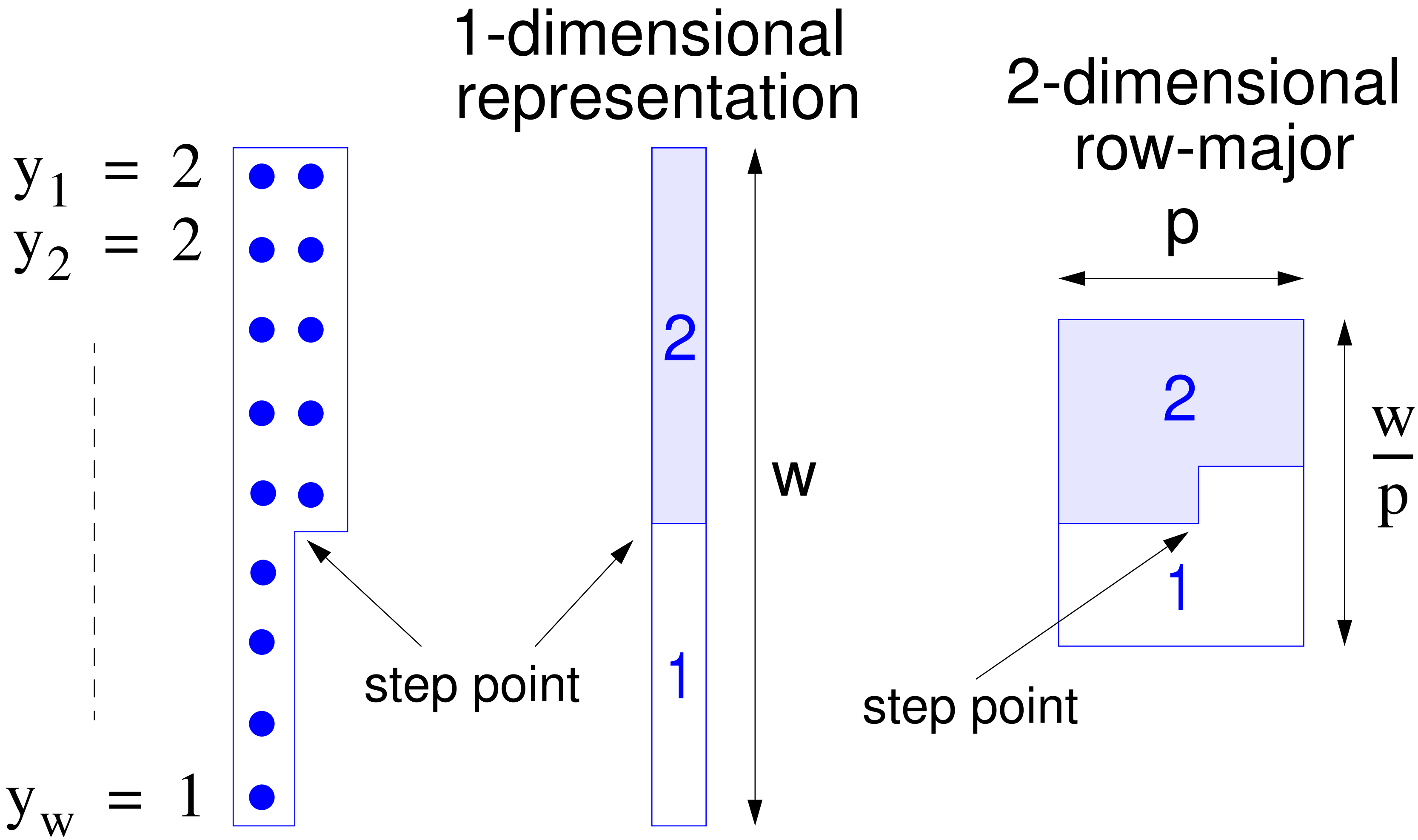
- Arbitrary width $w = p_1 p_2 \ldots p_n$
- Small depth $9.5n^2 - 12.5n + 3$
- Balancers of size at most $\max(p_i)$

Distinct factorizations

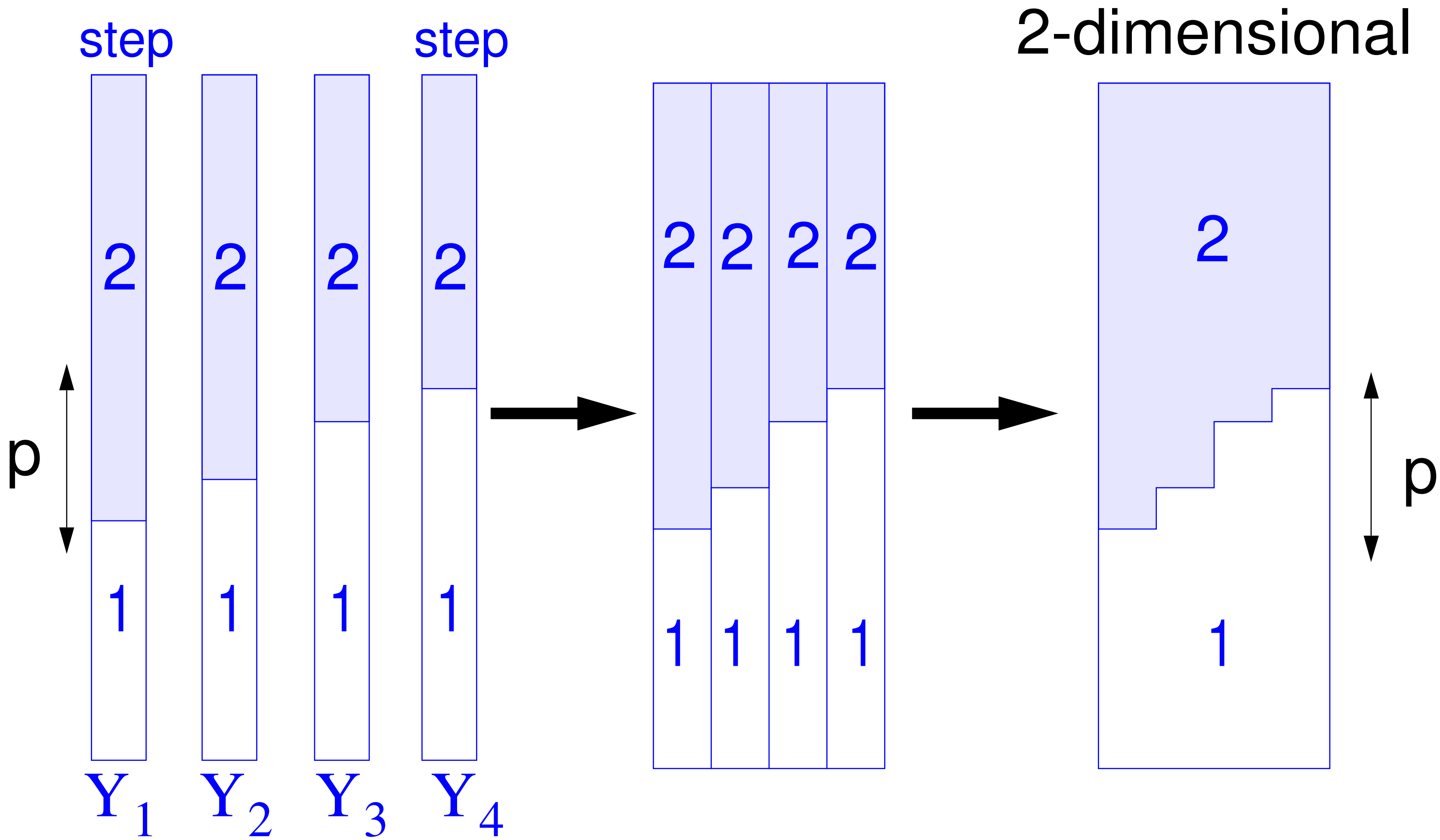
give unique constructions

- Example: width $60 = 2 \cdot 2 \cdot 3 \cdot 5$, depth = 105
  width $60 = 4 \cdot 15$, depth = 28
- Tradeoff: depth ***vs.*** contention

Preliminaries: Step Property

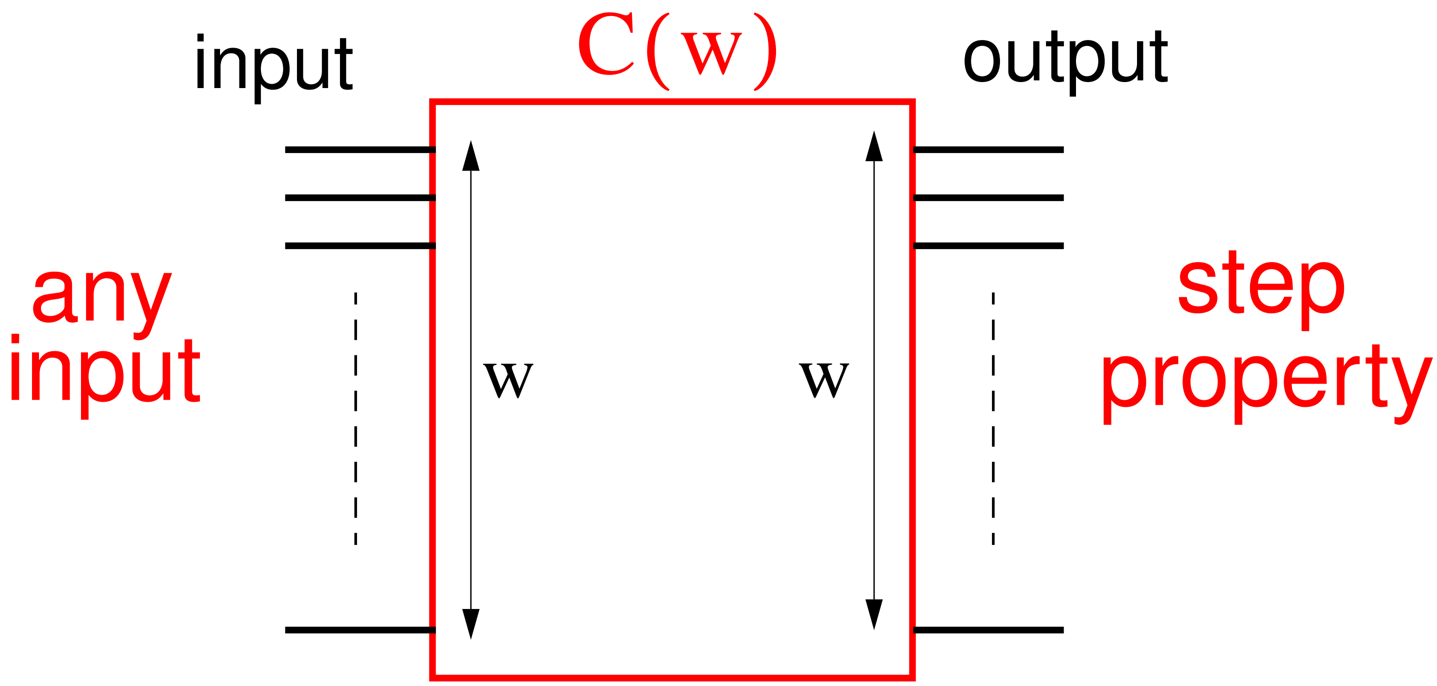


Definition: p-Staircase Property

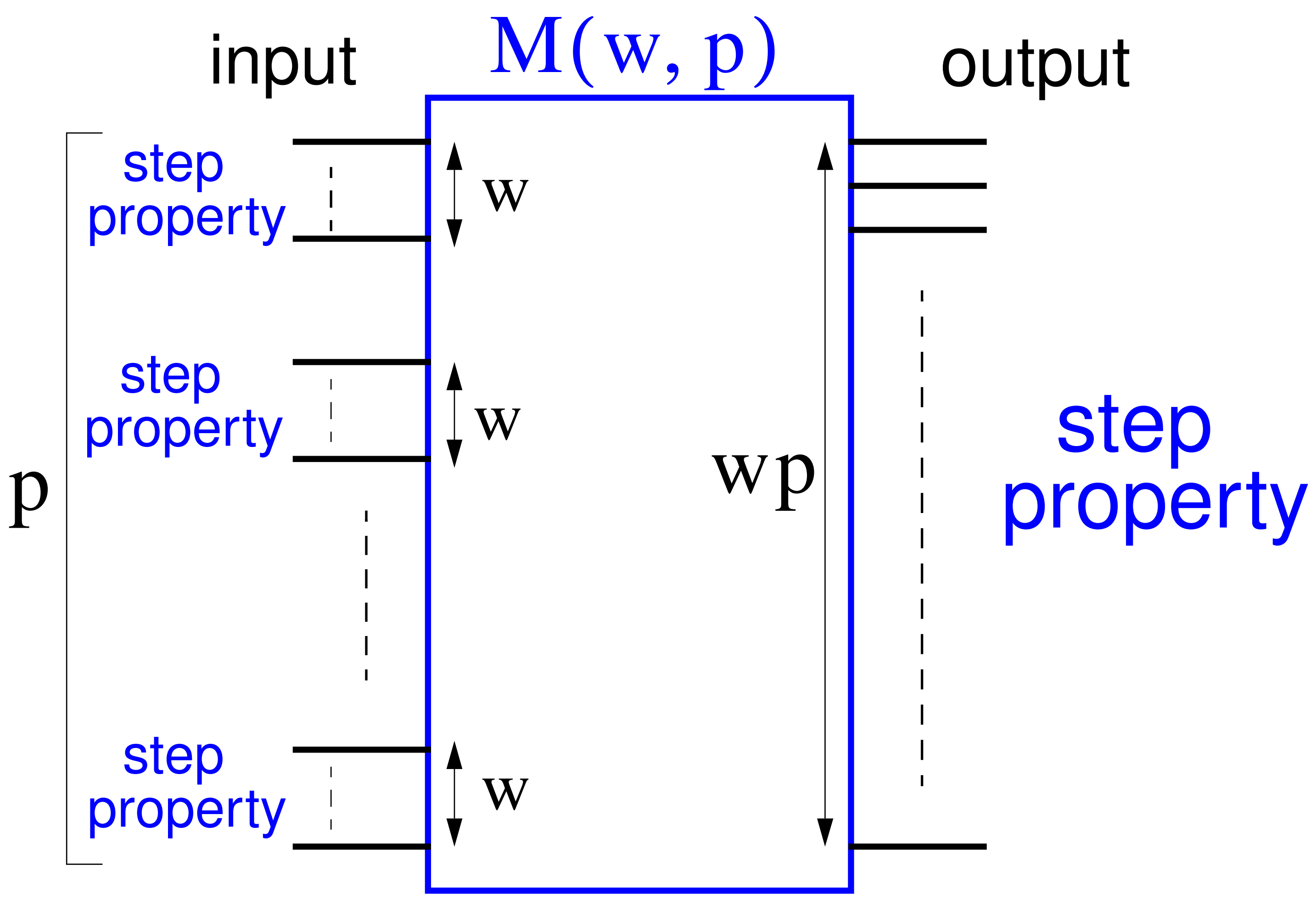


Formally: $0 \leq \Sigma(Y_i) - \Sigma(Y_j) \leq p$, for $i < j$

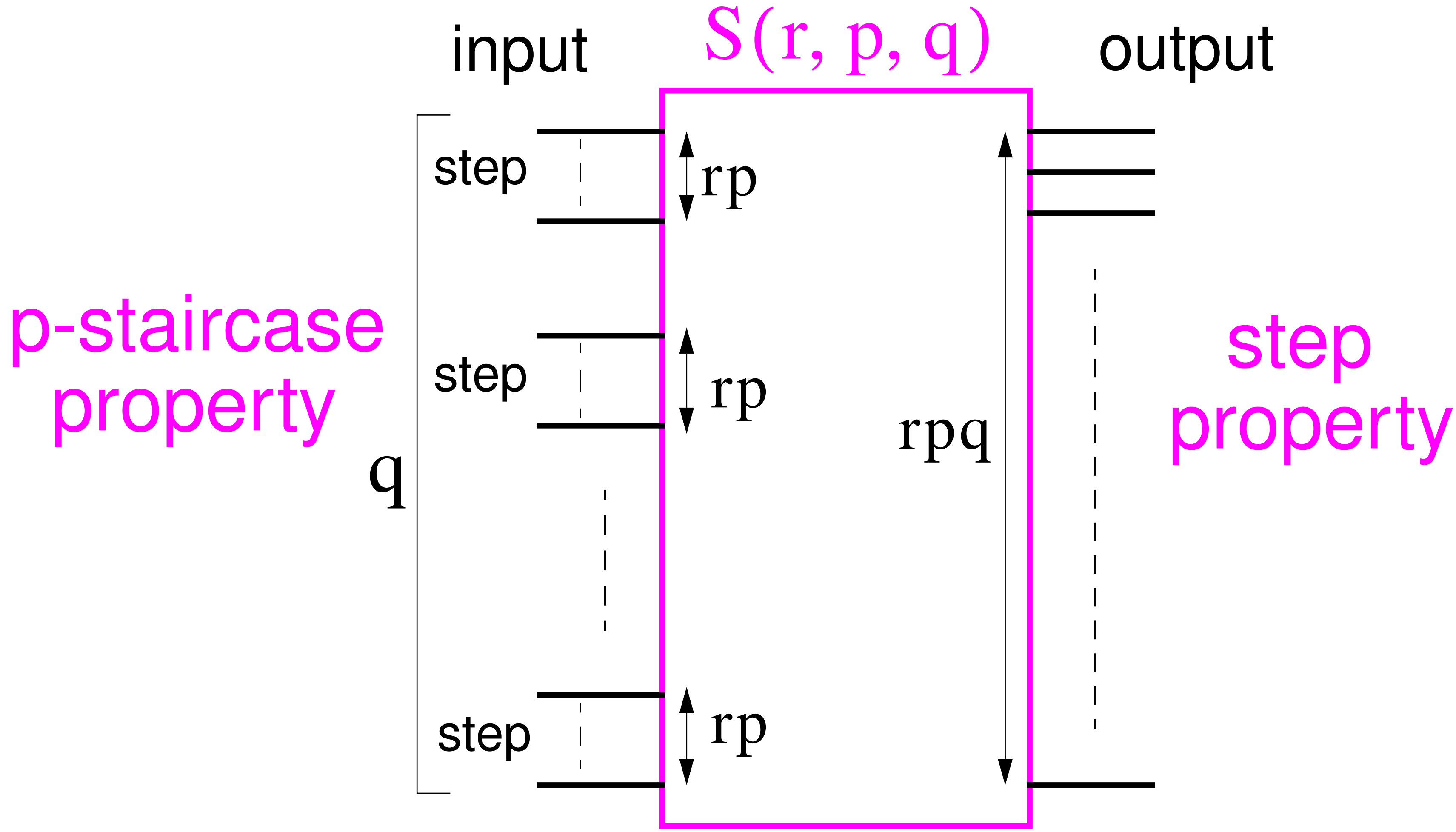
Definition: Counting Network $C(w)$



Definition: Merging Network $\mathrm{M}(w, p)$



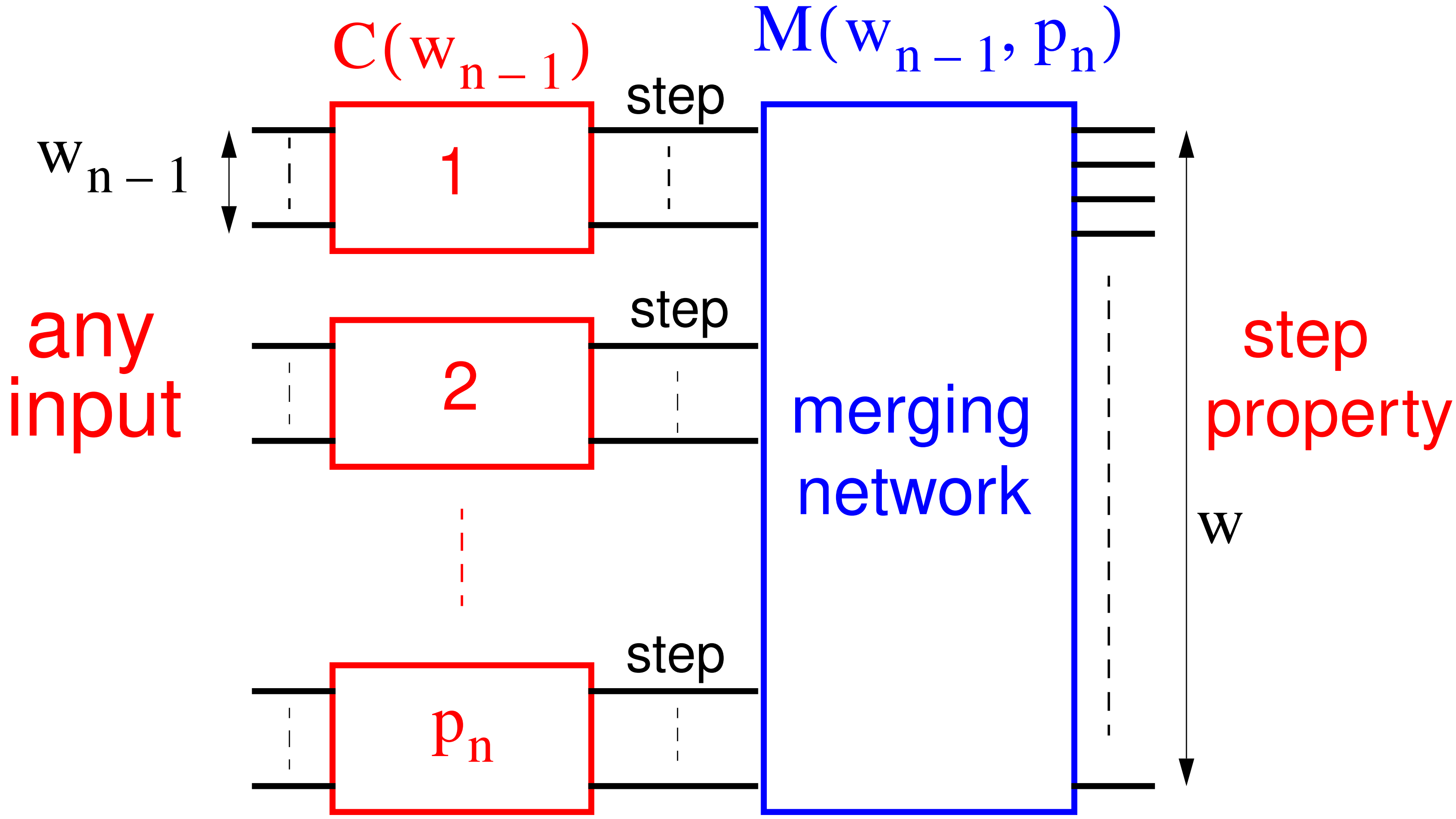
Definition: Staircase Merger $S(r, p, q)$



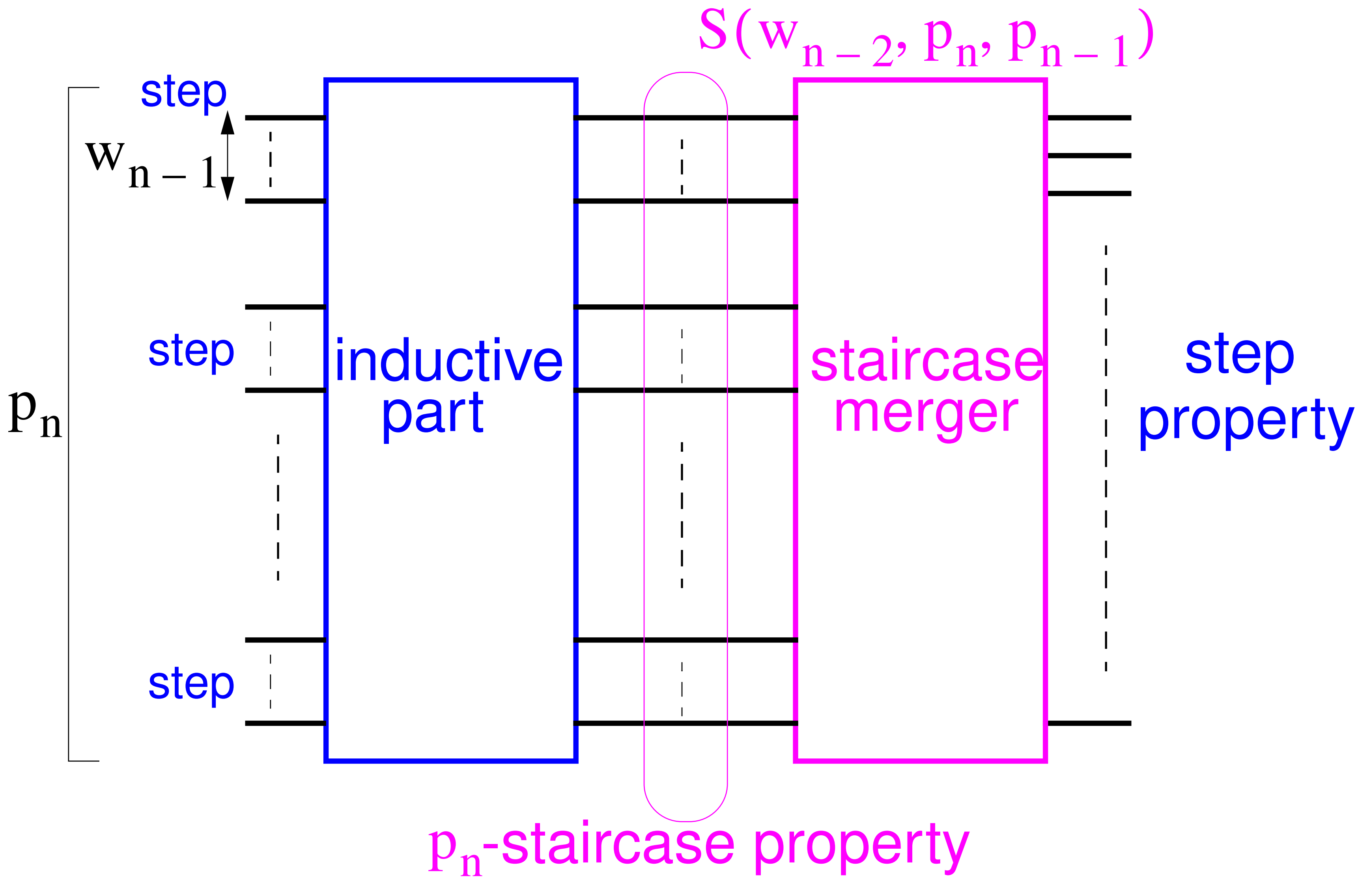
Construction: Counting Network $C(w)$

$$w = p_1 p_2 \ldots p_n$$
$$w_i = p_1 p_2 \ldots p_i$$

Induction on n

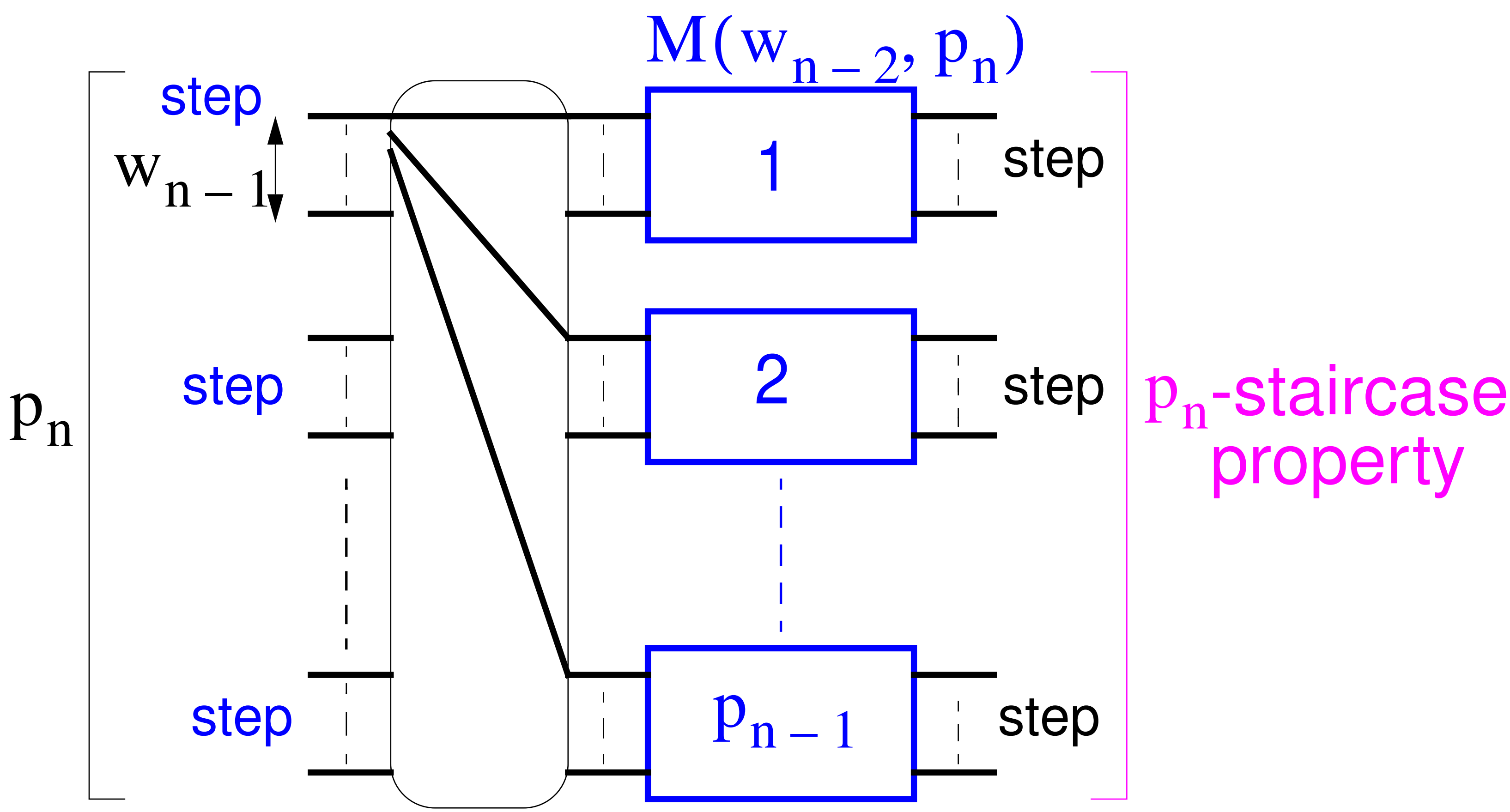


Construction: Merging Network $M(w_{n-1}, p_n)$

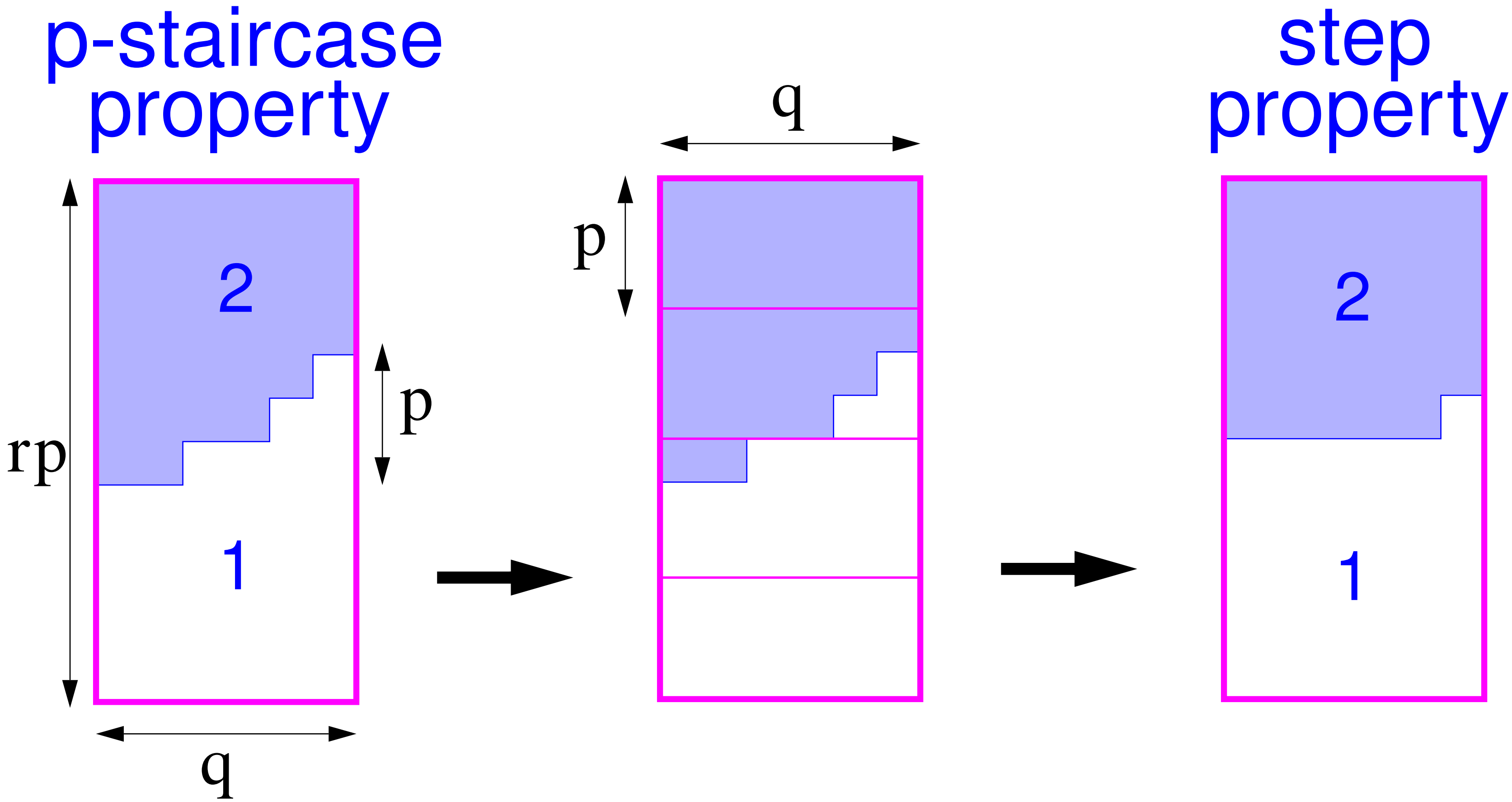


Construction: Merging Network $M(w_{n-1}, p_n)$

Induction on n



Construction: Staircase Merger $S(r, p, q)$



- Balancers sizes: at most $\max(p, q)$
- Depth: 19

Depth for width $w = p_1 p_2 \ldots p_n$

$$d\left(\begin{matrix}\text{Merging}\\\text{network}\end{matrix}\right) = \underbrace{d\left(\begin{matrix}\text{Inductive}\\\text{part}\end{matrix}\right)}^{\text{induction on } n} + \overbrace{d\left(\begin{matrix}\text{Staircase}\\\text{merger}\end{matrix}\right)}^{19} = O(n)$$

$$d\left(\begin{matrix}\text{Counting}\\\text{network}\end{matrix}\right) = \overbrace{d\left(\begin{matrix}\text{Inductive}\\\text{part}\end{matrix}\right)}^{\text{induction on } n} + \overbrace{d\left(\begin{matrix}\text{Merging}\\\text{network}\end{matrix}\right)}^{O(n)} = O(n^2)$$

$$= O(\mathbf{lg}^2 w)$$

constant 9.5

Compare to [K 94]: depth $O(\mathbf{lg}\, w \cdot \mathbf{lg}\,\mathbf{lg}\, w)$

problem: uses [AKS 83], constants are many many thousands