Supporting Increment and Decrement Operations in Balancing Networks

by
W. Aiello, C. Busch, M. Herlihy, M. Mavronicolas,
N. Shavit and D. Touitou

Costas Busch
Brown University

Balancing Networks

Introduced by Aspnes, Herlihy and Shavit in STOC'91

Distributed data structures used for:

- shared counters
- barriers
- load balancing

Advantages:

- low contention
- non-blocking

Limitation: basic operation is increment (+1)

We would like them to support: decrement (-1)

Decrement is good for:

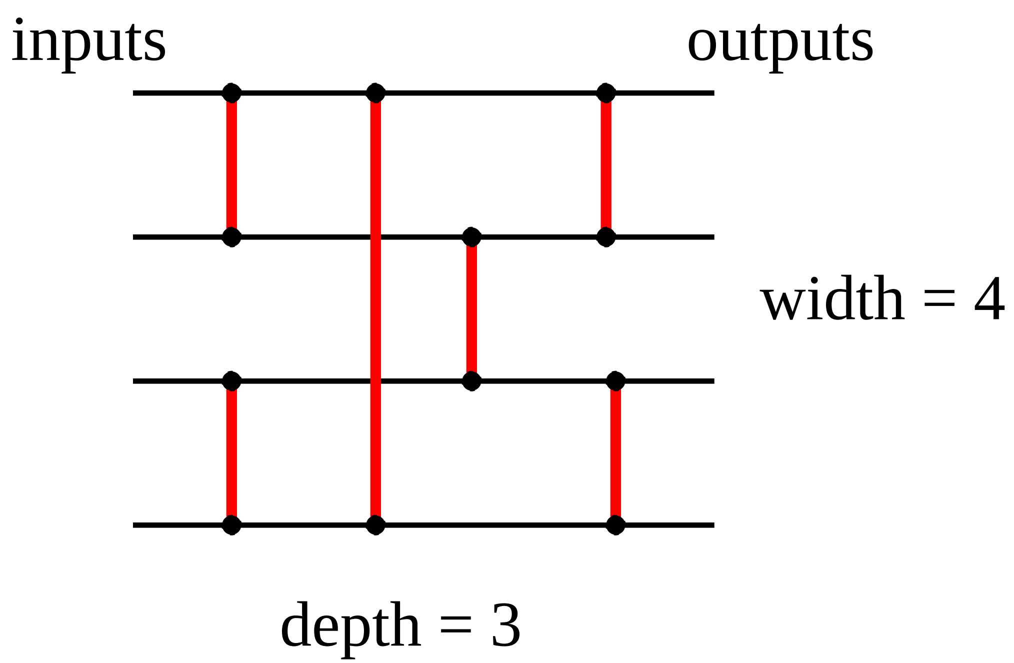
- concurrent “pools” and “stacks”
- semaphores

We show:

certain balancing networks can support both increments and decrements

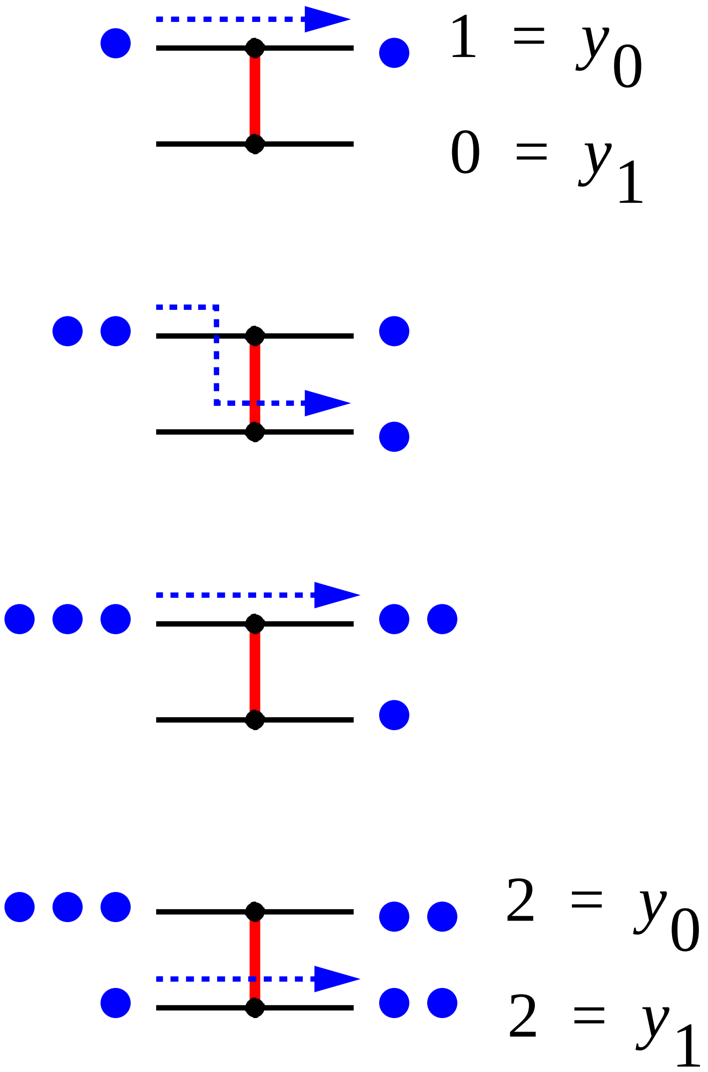
Balancing Networks in Detail

They look like sorting networks:



Constructed from: balancers

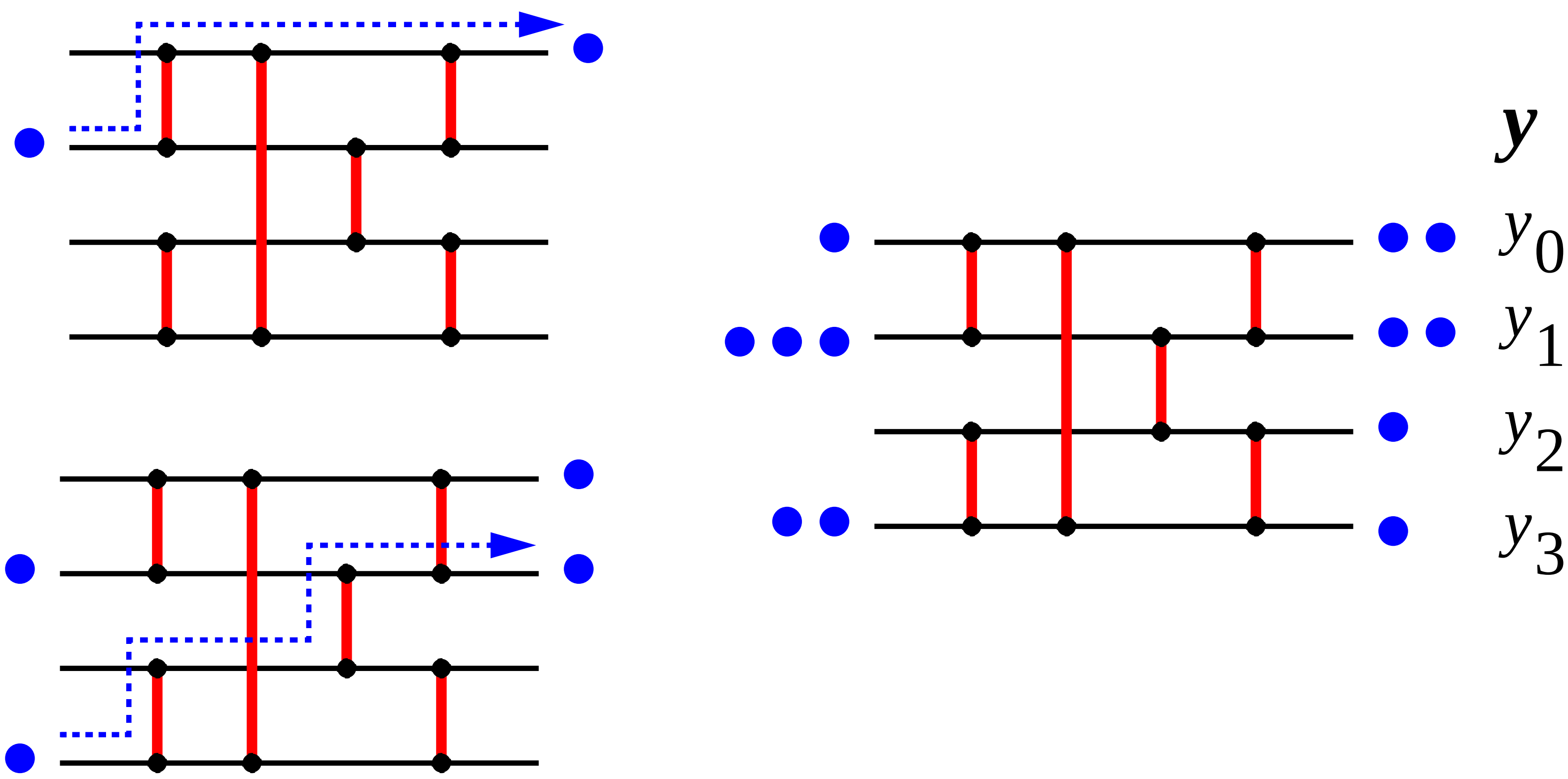
Balancer: routes tokens (increment +1)



Step Property:

$$0 \le y_0 - y_1 \le 1$$

Balancing network



Counting networks satisfy the step property:

on output vector $\boldsymbol{y}$, $0 \leq y_i - y_j \leq 1$ for $i < j$

K-smoothing networks satisfy the *K*-smoothing property:

on output vector $\boldsymbol{y}$, $|y_i - y_j| \leq K$ for all i, j

A boundedness property is:

- a subset of the *K*-smoothing property
- closed under addition with a constant vector

Examples of boundedess properties:

- step property
- *K*-smoothing property (trivially)

Supporting Decrements

For increment (+1) we used: tokens

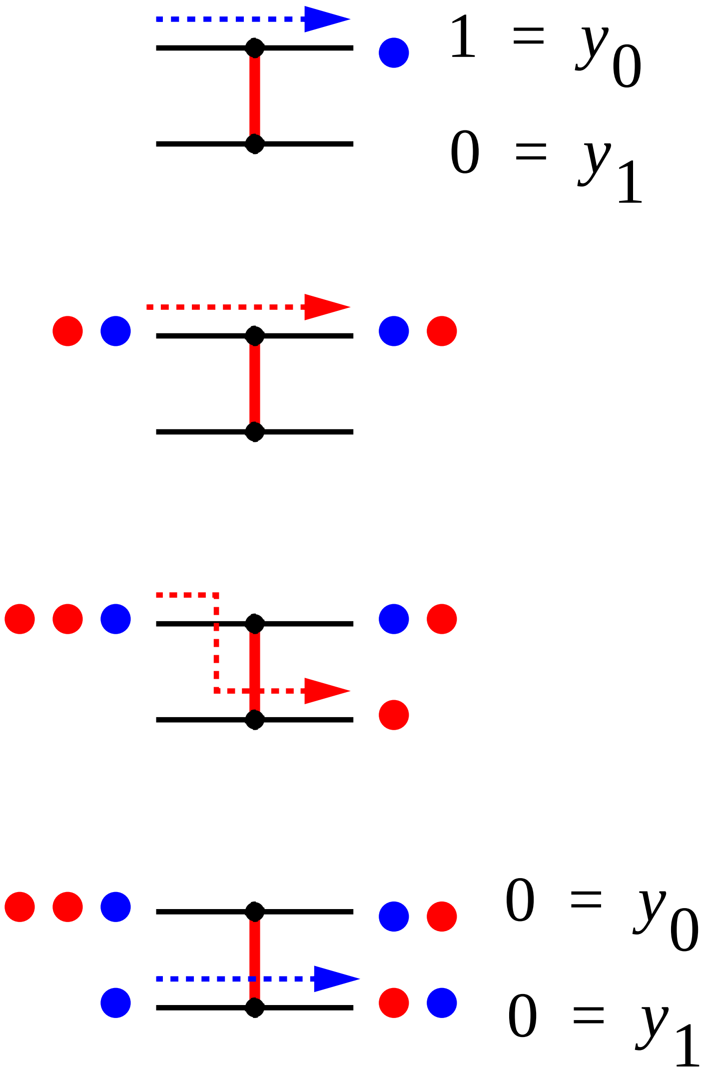
For decrement (-1) we will use: antitokens

Antitokens cancel the effect of the tokens:

Token + Antitoken = 0 (elimination)

Antitokens were introduced by Shavit and Touitou in SPAA'95

Tokens (+1) and antitokens (-1) in a balancer



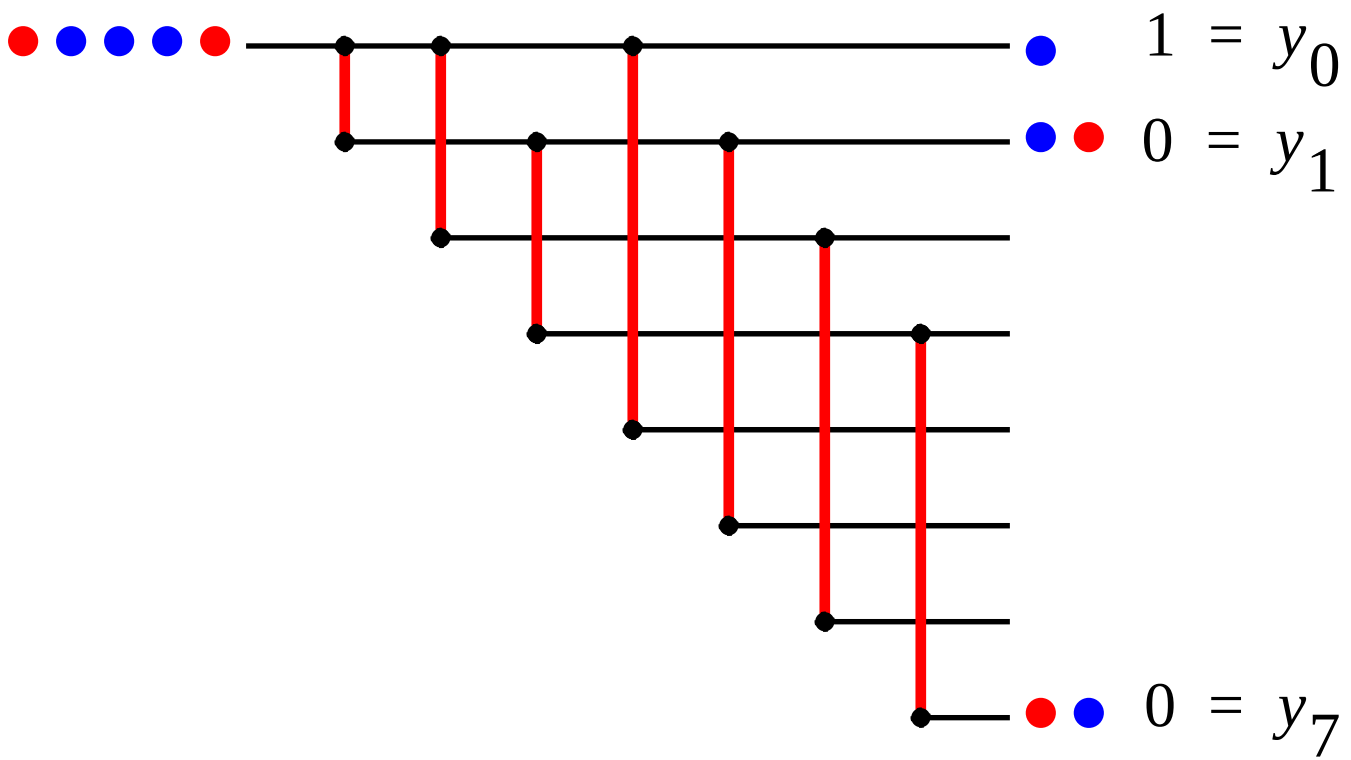
Step Property:

$$0 \le y_0 - y_1 \le 1$$

Shavit and Touitou showed:

counting trees (a kind of counting networks)

support decrements



Step property still holds: $0 \le y_i - y_j \le 1$ for $i < j$

We show a more general result:

if a balancing network satisfies a boundedness property with increments then

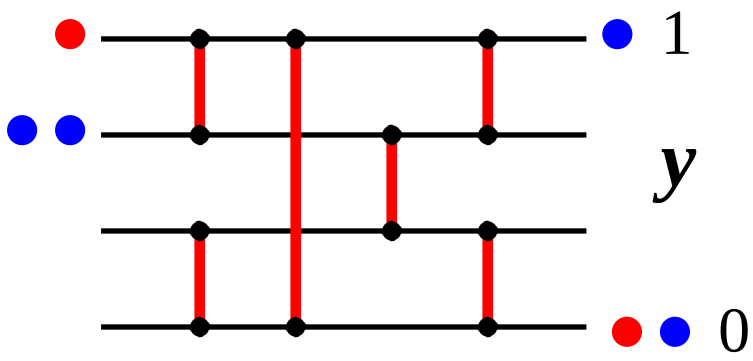
it also supports decrements

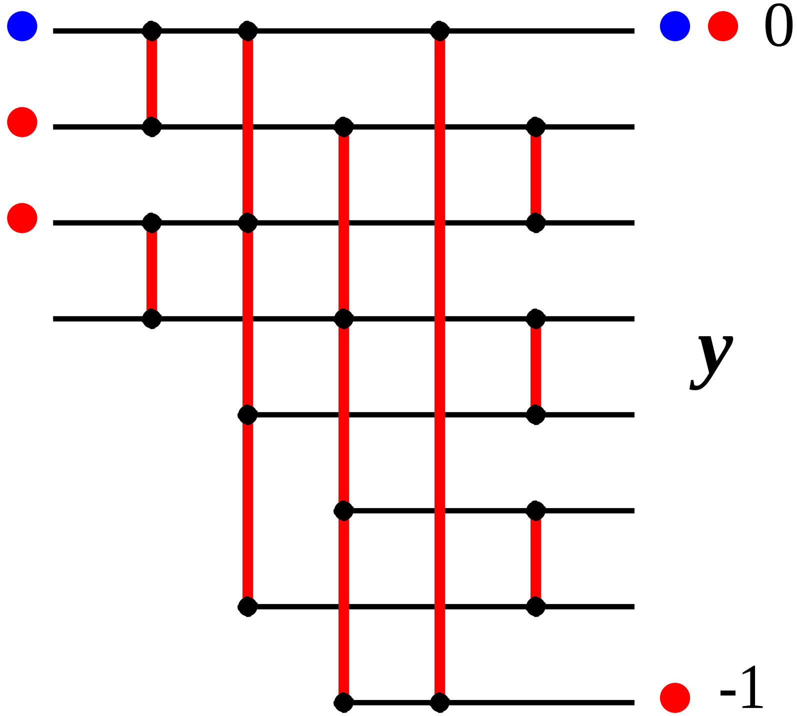
Implication:

All known counting and K-smoothing networks support decrements

Examples

These are known counting networks





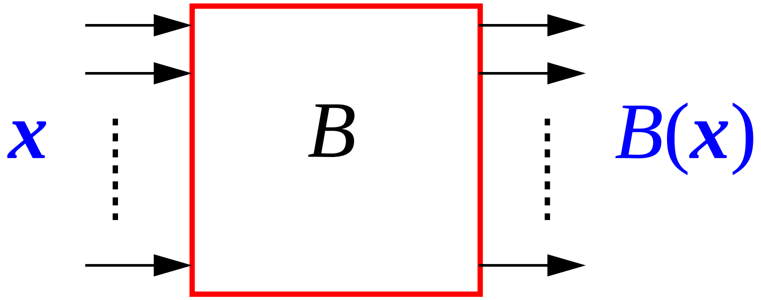
Their step property still holds:

on output vector ***y***, $0 \le y_i - y_j \le 1$ for $i < j$

The Proof

For a balancing network B with input vector $\boldsymbol{x}$:

$B(\boldsymbol{x})$ denotes the output vector



We need to show that:

if $B(\boldsymbol{x})$ has a boundedness property P
whenever $\boldsymbol{x}$ is non-negative then
$B(\boldsymbol{x})$ has the property P for any $\boldsymbol{x}$

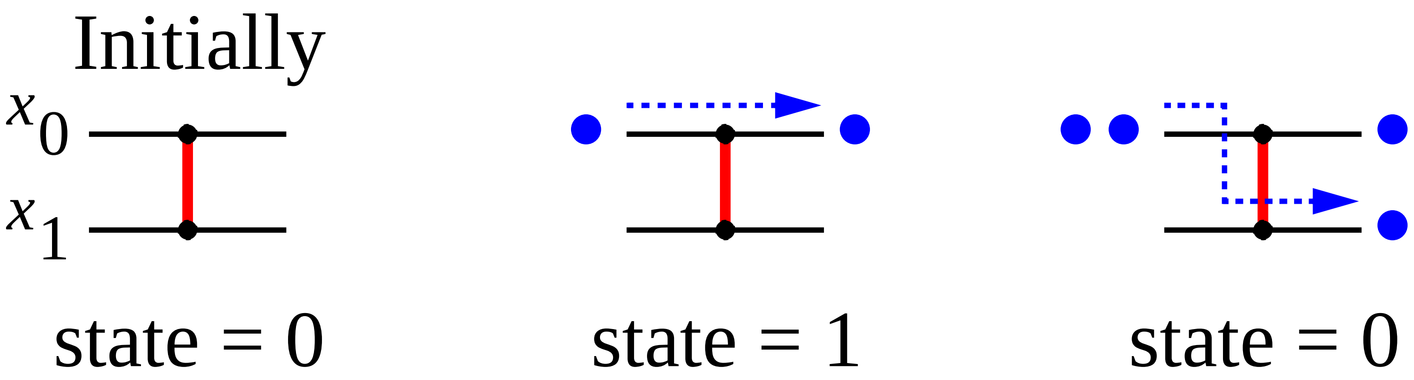
Preliminaries: State of Balancer

The state of balancer *b* with fan-out f_{out}:

- points where the next token will exit

- formally, on input vector $\boldsymbol{x}$:

$$\text{state}_b(\boldsymbol{x}) = \Sigma(\boldsymbol{x}) \bmod f_{\text{out}}$$



Preliminaries: State of Balancing Network

The state of a balancing network *B*:

is the collection of the states of its balancers

A fooling pair to network *B* is:

a pair of input vectors ***x*** and ***x'*** that drive the network *B* to identical states

A null vector to network *B*:

- is any input vector ***x*** that is a fooling pair with the vector **0**
- drives network *B* to its initial state

Preliminaries: Proposition 1

For any balancing network *B*,
any input vector ***z***,
and any fooling pair ***x*** and ***x'****:*

- ***x*** + ***z*** and ***x'*** + ***z*** is a fooling pair

- *B*(***x*** + ***z***) - *B*(***x***) = *B*(***x'*** + ***z***) - *B*(***x'***)

<u>Proof:</u> By induction on the depth of *B*

Preliminaries: Proposition 2

For any balancing network B,
any null vector $\boldsymbol{x}$,
and any integer k:

- $B(k\,\boldsymbol{x}) = k\,B(\boldsymbol{x})$

- $k\,\boldsymbol{x}$ is a null vector

Proof: By induction on k

Preliminaries: Proposition 3

For any balancing network *B*,
and any input vector x
such that *W*(*B*) divides ***x***:

x is a null vector

W(*B*) is the product of the fan-outs of the balancers of network *B*

Proof: By induction on the depth of *B*

Theorem 1

For any balancing network B
that has a boundedness property,
and any input vector $\boldsymbol{x}$
such that $W(B)$ divides $\boldsymbol{x}$:

$B(\boldsymbol{x})$ is a constant vector

Proof: By contradiction

Assume $B(\boldsymbol{x})$ is not constant.

There are elements a and b of $B(\boldsymbol{x})$:

$$|a - b| \geq 1$$

From the boundedness property:

$B(\boldsymbol{x})$ is a K-smoothing vector

By Proposition 3, since $W(B)$ divides $\boldsymbol{x}$:

$\boldsymbol{x}$ is a null vector

By Proposition 2, since x is a null vector:

$B((K + 1)\,\boldsymbol{x}) = (K+1)\,B(\boldsymbol{x})$

Subsequently, for the elements a' and b' of $B((K+1)\,\boldsymbol{x})$:

$$|a' - b'| = (K+1)|a - b| \geq K + 1$$

Therefore, the vector $B((K+1)\,\boldsymbol{x})$ is at least $(K+1)$-smoothing:

a contradiction! (it is K-smoothing)

Main Theorem

For any balancing network *B*
that has a boundedness property *P*
whenever the input vector ***x*** is non-negative:

the network *B* has the property *P* for any input vector ***x***

Proof:

Given any vector x, construct vector ***x'***:

- *W*(*B*) divides ***x'***
- $\boldsymbol{x}' + \boldsymbol{x} \geq \mathbf{0}$

Since $\boldsymbol{x} + \boldsymbol{x}'$ is non-negative

$B(\boldsymbol{x}' + \boldsymbol{x})$ has the property P

By Theorem 1, since $W(B)$ divides $\boldsymbol{x}'$:

$B(\boldsymbol{x}')$ is a constant vector

By proposition 3, since $W(B)$ divides $\boldsymbol{x}'$:

$\boldsymbol{x}'$ is a null vector

By proposition 1, since ***x'*** is a null vectror:

$$B(\boldsymbol{x}) = B(\boldsymbol{x'} + \boldsymbol{x}) - B(\boldsymbol{x'})$$

Subsequently, since

- $B(\boldsymbol{x'} + \boldsymbol{x})$ has the P property
- $B(\boldsymbol{x'})$ is constant,
- property P is closed under addition with a constant vector

$B(\boldsymbol{x})$ has the property P, as needed

Conclusions

We showed that:

balancing networks which satisfy a bounded-ness property can support both increment and decrement operations

Open problem:

Do randomized balancing networks (Aiello et al. PODC '94) support both increment and decrement operations?