

# A Combinatorial Characterization of Properties Preserved by Antitokens

Costas Busch<sup>1</sup>, Neophytos Demetriou<sup>2</sup>, Maurice Herlihy<sup>1</sup>, and  
Marios Mavronicolas<sup>2</sup>

<sup>1</sup> Department of Computer Science,  
Brown University,  
Providence, RI 02912  
{cb,herlihy}@cs.brown.edu

<sup>2</sup> Department of Computer Science,  
University of Cyprus,  
Nicosia CY-1678, Cyprus  
cs97dn1@ucy.ac.cy, mavronic@cs.ucy.ac.cy

**Abstract.** *Balancing networks* are highly distributed data structures used to solve multiprocessor synchronization problems. Typically, balancing networks are accessed by tokens, and the distribution of the tokens on the network's output specify the property of the network. However, tokens represent increment operations only, and tokens alone are not adequate for synchronization problems that require decrement operations. For such kinds of problems, *antitokens* have been used to represent decrement operations. It has been shown that several kinds of balancing networks which satisfy the *step property*, *smoothing property*, and the *threshold property* for tokens alone, preserve their properties even when antitokens are introduced. A fundamental question that was left open was to characterize *all* the properties of balancing networks which are preserved under the introduction of antitokens. In this work, we provide such a simple combinatorial characterization for all the properties which are preserved when antitokens are introduced.

## 1 Introduction

*Balancing networks* were devised by Aspnes *et al.* [4] as a novel class of distributed data structures that provide highly-concurrent, low-contention solutions to a variety of multiprocessor synchronization problems.

A balancing network is constructed from elementary switches with  $p$  *input wires* and  $q$  *output wires*, called  $(p, q)$ -*balancers*. A  $(p, q)$ -balancer accepts a stream of tokens on its  $p$  input wires. The  $i$ -th token to enter the balancer leaves on output wire  $i \bmod q$ , where  $i = 0, 1, \dots$ . One can think of a balancer as having a “toggle” state variable tracking which output wire the next token should exit from. A token traversal amounts to a *Fetch&Increment* operation to the toggle variable. This operation includes reading the current state of the toggle, which is the wire the token will exit from, and then setting the toggle

to point to the next output wire. The distribution of the tokens on the output wires of the balancer satisfies the *step property* (explained below).

A balancing network is an acyclic network of balancers where output wires of some balancers are linked to input wires of other balancers (balancing networks look like sorting networks [15]). The network's *input wires* are those input wires of balancers not linked to any other balancer, and similarly for the network's *output wires*. Tokens enter the network on the input wires, typically several per wire, propagate asynchronously through the balancers and leave from the output wires, typically several per wire.

Balancing networks are classified according to the distribution of the exiting tokens on the output wires. In particular, *Counting networks* [4] are those balancing networks on which the exiting tokens satisfy the *step property*: the exiting tokens are distributed uniformly among the output wires and any excess tokens appear on the upper wires. On *smoothing networks* [1, 4] the output tokens satisfy the *K-smoothing property*: the sum of tokens on any two output wires differ by at most  $K$ . On *threshold networks* [4, 7] the output sequence satisfies the *threshold property*: the number of tokens on the bottom wire is increased by one for every bunch of  $w$  tokens, where  $w$  is the number of output wires of the network.

Based on balancing networks, simple and elegant algorithms have been developed to solve a variety of synchronization problems that appear in distributed computing systems. For example, counting networks are used to implement efficient distributed *Fetch&Increment* counters as well as *linearizable* counters [11]. Furthermore, smoothing networks solve *load sharing* problems [17], and threshold networks provide solutions to *barrier synchronization* problems [9]. For applications of balancing networks see [4, 11, 12, 14, 16].

A limitation of balancing networks is that they are accessed by tokens only. A token can be thought of as an “increment” operation issued by the process which inserts the token in the network. Using tokens only, the capabilities of balancing networks are limited to the use of only increment operations. However, many distributed algorithms require the ability to “decrement” shared objects as well. For example, the classical synchronization constructs of *semaphores* [8], *critical regions* [13], and *monitors* [10] all rely on applying both increment and decrement operations on shared counters.

In order to solve such kinds of problems Shavit and Touitou [16] invented the *antitoken*, an entity that a processor shepherds through the network in order to perform a decrement operation. Unlike a token, which traverses a balancer by fetching the toggle value and then advancing it, an antitoken sets the toggle back and then fetches it. Informally, an antitoken “cancels” the effect of the most recent token on the balancer's toggle state, and vice versa. Furthermore, when an antitoken and a token meet while they traverse a network they can “eliminate” each other without needing to traverse the rest of the network.

In the same paper, Shavit and Touitou provide an operational proof that a specific kind of counting networks which have the form of binary trees count correctly even when they are traversed by both tokens and antitokens. Namely, they

show that in these networks the step property is preserved by the introduction of antitokens.

Subsequently, Aiello *et al.* [2] generalized the results of Shavit and Touitou [16] to far more general classes of balancing networks and properties of balancing networks. More specifically, Aiello *et al.* considered *boundedness properties*, a generalization of the step and  $K$ -smoothing properties. They showed that boundedness properties are preserved by the introduction of antitokens. Busch *et al.* [5] considered the *threshold property* and they showed that this property is also preserved by the introduction of antitokens.

A fundamental question that was left open by the results in [2, 5, 16], is to formally characterize *all* properties of balancing networks that are preserved under the introduction of antitokens. In this work, we provide the first answer to this fundamental question.

We provide a simple, combinatorial characterization for all properties of balancing networks which are preserved when antitokens are introduced. In particular, for any arbitrary balancing network, we define a new, natural class of properties, that we call *closed under the nullity of the balancing network*, which precisely characterizes all the properties preserved by antitokens. This characterization provides necessary and sufficient conditions for all the properties that are preserved. For any property that is satisfied by a balancing network for tokens only, then our characterization implies that this property is preserved when antitokens are introduced if and only if the property is closed under the nullity of the network.

The combinatorial characterization provides a theoretical tool for identifying which properties are preserved by the introduction of antitokens. For example, consider some property of a balancing network for which we know the network satisfies for tokens. In order to prove that this property will be preserved by the antitokens we only need to show that the property is closed under the nullity of the network. Having this theoretical tool, the practitioner can identify if a specific property of balancing can be used to implement algorithms that require decrements.

Moreover, the necessary condition of the characterization enables us to classify all the properties for which we already know are preserved with antitokens. This necessary condition simply says that all these properties must satisfy the characterization and therefore are closed under the nullity of a balancing network. Consequently, from the results of [2], we can infer that the step property, the  $K$ -smoothing property, and in general the boundedness property are all closed under the nullity of a balancing network. Furthermore, from the results of [5], we can infer that the threshold property is closed under the nullity of a balancing network.

The rest of this paper is organized as follows. Section 2 provides some necessary background. In section 3 we describe properties of balancing networks. We present our main combinatorial characterization result in Section 4. We give our conclusions in Section 5.

## 2 Framework

Throughout our discussion we consider integer vectors. For any integer  $g \geq 2$ ,  $\mathbf{x}^{(g)}$  denotes the vector  $\langle x_0, x_1, \dots, x_{g-1} \rangle^T$ . For any vector  $\mathbf{x}^{(g)}$ , denote  $\|\mathbf{x}^{(g)}\|_1 = \sum_{i=0}^{g-1} x_i$ . We use  $\mathbf{0}^{(g)}$  to denote  $\langle 0, 0, \dots, 0 \rangle^T$ , a vector with  $g$  zero entries. In a *constant vector* all entries are equal to some constant  $c$ . We say that a vector is *non-negative*, if all of its entries are non-negative integers. We say that an integer  $d$  *divides* a vector  $\mathbf{x}^{(g)}$  if each entry of  $\mathbf{x}^{(g)}$  is some integer multiple of  $d$ .

For the rest of our discussion, we consider balancers and balancing networks in *quiescent configurations* in which no tokens and antitokens are traversing the network, namely, all the tokens and antitokens that have ever entered the balancer have left it.

We think of a token as a positive unit  $+1$ , and the antitoken as a negative unit  $-1$ . Consider an  $(f_{\text{in}}, f_{\text{out}})$ -balancer. For each input index  $i$ ,  $0 \leq i < f_{\text{in}}$ , we denote by  $x_i$  the algebraic sum of tokens and antitokens that have entered on input wire  $i$ ; that is,  $x_i$  is the number of tokens *minus* the number of antitokens that have entered on input wire  $i$ . We say that the vector  $\mathbf{x}^{(f_{\text{in}})} = \langle x_0, x_1, \dots, x_{f_{\text{in}}-1} \rangle^T$  is an *input vector* of the balancer. Similarly, we define the *output vector* of the balancer. In the same way, we define input and output vectors for balancing networks. Note that when we are considering tokens only the input and output vectors are non-negative. Vectors can take negative values only when we consider antitokens too.

Let  $\mathcal{B}$  be a balancing network with  $w_{\text{in}}$  input wires and  $w_{\text{out}}$  output wires. We call  $w_{\text{in}}$  the fan-in, and  $w_{\text{out}}$  the fan-out of the network. Take any input vector  $\mathbf{x}^{(w_{\text{in}})}$  to  $\mathcal{B}$  and let  $\mathbf{y}^{(w_{\text{out}})}$  be the corresponding output vector. For each input vector  $\mathbf{x}^{(w_{\text{in}})}$ , there is a unique output vector  $\mathbf{y}^{(w_{\text{out}})}$ , and this allows us to treat the network  $\mathcal{B}$  as a function on vectors, and we write  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) = \mathbf{y}^{(w_{\text{out}})}$ . We write also  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$  to denote the network  $\mathcal{B}$ .

Clearly,  $\mathcal{B}(\mathbf{0}^{(w_{\text{in}})}) = \mathbf{0}^{(w_{\text{out}})}$ . In any quiescent configuration it holds that  $\|\mathcal{B}(\mathbf{x}^{(w_{\text{in}})})\|_1 = \|\mathbf{x}^{(w_{\text{in}})}\|_1$ , which means that the algebraic sum of tokens and antitokens that have entered the network is the same with the algebraic sum of tokens and antitokens that have left the network. This also includes the tokens and antitokens that have been “eliminated” in the network, since their algebraic sum is zero.

Consider now an  $(f_{\text{in}}, f_{\text{out}})$ -balancer  $b$ . The *state* of balancer  $b$ , on input sequence  $\mathbf{x}^{(f_{\text{in}})}$ , is defined to be  $\text{state}_b(\mathbf{x}^{(f_{\text{in}})}) = \|\mathbf{x}^{(f_{\text{in}})}\|_1 \bmod f_{\text{out}}$ . We remark that the state of balancer  $b$  is some integer in the set  $\{0, 1, \dots, f_{\text{out}} - 1\}$ , which captures the “position” to which the balancer is set as a toggle mechanism. Consider now a balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$ . The *state* of  $\mathcal{B}$ , denoted  $\text{state}_{\mathcal{B}}(\mathbf{x}^{(w_{\text{in}})})$ , is defined to be the collection of the states of its individual balancers. The *initial state* of network  $\mathcal{B}$  is the state  $\text{state}_{\mathcal{B}}(\mathbf{0}^{(w_{\text{in}})})$ .

In respect to the state of a balancing network, Aiello *et al.* [2] have defined *fooling pairs* and *null vectors* as follows. Say that two input vectors  $\mathbf{x}_1^{(w_{\text{in}})}$  and  $\mathbf{x}_2^{(w_{\text{in}})}$  are a *fooling pair to network*  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$ , if  $\text{state}_{\mathcal{B}}(\mathbf{x}_1^{(w_{\text{in}})}) = \text{state}_{\mathcal{B}}(\mathbf{x}_2^{(w_{\text{in}})})$ . Roughly speaking, a fooling pair “drives” all balancers of the

network to identical states. Say that  $\mathbf{x}^{(w_{\text{in}})}$  is a *null vector to network*  $\mathcal{B}$  if the vectors  $\mathbf{x}^{(w_{\text{in}})}$  and  $\mathbf{0}^{(w_{\text{in}})}$  are a fooling pair to  $\mathcal{B}$ . Intuitively, a null vector “drives” the network back to its initial state. Using results on properties of fooling pairs and null vectors from Aiello *et al.* [2], it is straightforward to obtain the following “linearity” lemma for null vectors.

**Lemma 1.** *Consider a balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$ . Take any input vector  $\mathbf{x}^{(w_{\text{in}})}$  and any null vector  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$  to  $\mathcal{B}$ . Then,*

$$\mathcal{B}(\mathbf{x}^{(w_{\text{in}})} \pm \tilde{\mathbf{x}}^{(w_{\text{in}})}) = \mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \pm \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}).$$

For any balancing network  $\mathcal{B}$ , denote by  $W_{\text{out}}(\mathcal{B})$  the product of the fan-outs of balancers of  $\mathcal{B}$ . Aiello *et al.* [2] show the following.

**Lemma 2 (Aiello et al. [2]).** *Consider a balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$ . Assume that  $W_{\text{out}}(\mathcal{B})$  divides  $\mathbf{x}^{(w_{\text{in}})}$ . Then,  $\mathbf{x}^{(w_{\text{in}})}$  is a null vector to  $\mathcal{B}$ .*

### 3 Properties

A property  $\Pi$  is a (computable) predicate on integer vectors. We identify  $\Pi$  with the set of (integer) vectors satisfying it. Say that a vector  $\mathbf{y}^{(w_{\text{out}})}$  *has the property*  $\Pi$  if  $\mathbf{y}^{(w_{\text{out}})}$  satisfies  $\Pi$ . Say that a balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$  *has a property*  $\Pi$ , if all output vectors  $\mathbf{y}^{(w_{\text{out}})}$  have the property  $\Pi$ , for any input vectors (not only non-negative input vectors). Below we describe in details several interesting properties.

Boundedness properties were introduced by Aiello *et al.* [2]. Fix any integer  $g \geq 2$ . For any integer  $K \geq 1$ , the  *$K$ -smoothing property* [1] is defined to be the set of all vectors  $\mathbf{y}^{(g)}$  such that for any entries  $y_j$  and  $y_k$  of  $\mathbf{y}^{(g)}$ , where  $0 \leq j, k < g$ , it holds  $|y_j - y_k| \leq K$ . A *boundedness property* is any subset of some  $K$ -smoothing property, and this subset is closed under addition with a constant vector, for some integer  $K \geq 1$ . Thus, a boundedness property is a strict generalization of the smoothing property. Clearly, there are infinitely many boundedness properties. The *step property* [4] is defined to be the set of all vectors  $\mathbf{y}^{(g)}$  such that for any entries  $y_j$  and  $y_k$  of  $\mathbf{y}^{(g)}$ , where  $0 \leq j < k < g$ , it holds  $0 \leq y_j - y_k \leq 1$ . Clearly, the step property is a boundedness property, since any vector that has the step property, has also the 1-smoothing property (but not vice versa). The main result of Aiello *et al.* [2] establishes that allowing negative inputs (antitokens) does not spoil the boundedness property of a balancing network.

**Theorem 1 (Aiello et al. [2]).** *Fix any boundedness property  $\Pi$ . Consider any balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$  such that  $\mathbf{y}^{(w_{\text{out}})}$  has the boundedness property  $\Pi$  whenever  $\mathbf{x}^{(w_{\text{in}})}$  is a non-negative input vector. Then,  $\mathcal{B}$  has the boundedness property  $\Pi$ .*

The *threshold property* [4, 7] is the set of all vectors  $\mathbf{y}^{(g)}$ , such that for the entry  $y_{g-1}$  of  $\mathbf{y}^{(g)}$ , it holds  $y_{g-1} = \lfloor \|\mathbf{y}^{(g)}\|_1 / g \rfloor$ . It has been observed in [5] that

the threshold property is not a boundedness property in all non-trivial cases (where  $g > 2$ ). Thus, Theorem 1 does not apply a fortiori to this property. The main result of Busch *et al.* [5] establishes that allowing negative inputs (antitokens) does not spoil the threshold property of a balancing network.

**Theorem 2 (Busch et al. [5]).** *Consider any balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$  such that  $\mathbf{y}^{(w_{\text{out}})}$  has the threshold property whenever  $\mathbf{x}^{(w_{\text{in}})}$  is a non-negative vector. Then,  $\mathcal{B}$  has the threshold property.*

## 4 Combinatorial Characterization

A fundamental question that was left open by the results in Theorems 1 and 2, is to formally characterize all the properties of balancing networks that are preserved under the introduction of antitokens. In this section, we give such a combinatorial characterization as follows.

**Definition 1.** *Consider any balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$ . A property  $\Pi$  is closed under the nullity of  $\mathcal{B}$  if for all non-negative input vectors  $\mathbf{x}^{(w_{\text{in}})}$  and for all non-negative null vectors  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$  to  $\mathcal{B}$ , it holds that  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \in \Pi$  implies  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \pm \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}) \in \Pi$ .*

The use of non-negative vectors in the above definition allows us to determine whether any given property of a balancing network is closed under the nullity of the network by examining how the network behaves for tokens only.

In the next claim we establish our main result. We show that being closed under the nullity of a balancing network is a necessary and sufficient condition for the property to be preserved under the introduction of antitokens.

**Theorem 3.** *Fix a property  $\Pi$ . Consider any balancing network  $\mathcal{B} : \mathbf{x}^{(w_{\text{in}})} \rightarrow \mathbf{y}^{(w_{\text{out}})}$  such that  $\mathbf{y}^{(w_{\text{out}})} \in \Pi$  whenever the input vector  $\mathbf{x}^{(w_{\text{in}})}$  is non-negative. Then,  $\mathcal{B}$  has the property  $\Pi$  if and only if  $\Pi$  is closed under the nullity of  $\mathcal{B}$ .*

*Proof.* First, we prove the “if” direction of the claim. Consider any arbitrary input vector  $\mathbf{x}^{(w_{\text{in}})}$ . We will show that  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \in \Pi$ .

Construct from  $\mathbf{x}^{(w_{\text{in}})}$  an non-negative input vector  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$  such that for each index  $i$ ,  $\tilde{x}_i$  is the least positive multiple of  $W_{\text{out}}(\mathcal{B})$  so that  $0 \leq x_i + \tilde{x}_i$ . Clearly, the vector  $\mathbf{x}^{(w_{\text{in}})} + \tilde{\mathbf{x}}^{(w_{\text{in}})}$  is non-negative. Furthermore,  $W_{\text{out}}(\mathcal{B})$  divides  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$ , and from Lemma 2 it follows that  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$  is a null vector.

By applying Lemma 1 with vectors  $\mathbf{x}^{(w_{\text{in}})}$  and  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$ , we obtain

$$\mathcal{B}(\mathbf{x}^{(w_{\text{in}})} + \tilde{\mathbf{x}}^{(w_{\text{in}})}) = \mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) + \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}),$$

so that

$$\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) = \mathcal{B}(\mathbf{x}^{(w_{\text{in}})} + \tilde{\mathbf{x}}^{(w_{\text{in}})}) - \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}).$$

Since the vector  $\mathbf{x}^{(w_{\text{in}})} + \tilde{\mathbf{x}}^{(w_{\text{in}})}$  is non-negative, we have by assumption that

$$\mathcal{B}(\mathbf{x}^{(w_{\text{in}})} + \tilde{\mathbf{x}}^{(w_{\text{in}})}) \in \Pi.$$

Furthermore, since property  $\Pi$  is closed under the nullity of  $\mathcal{B}$ , and since  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$  is a non-negative null vector, we have by Definition 1 that

$$\mathcal{B}(\mathbf{x}^{(w_{\text{in}})} + \tilde{\mathbf{x}}^{(w_{\text{in}})}) - \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}) \in \Pi.$$

Subsequently,  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \in \Pi$ , as needed.

We continue to show the “only if” part of the claim. Take any non-negative input vector  $\mathbf{x}^{(w_{\text{in}})}$  to  $\mathcal{B}$  and any non-negative null vector  $\tilde{\mathbf{x}}^{(w_{\text{in}})}$  to  $\mathcal{B}$ . Trivially,  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \in \Pi$ , and thus, by Definition 1, we only need to show that  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \pm \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}) \in \Pi$ . By Lemma 1,

$$\mathcal{B}(\mathbf{x}^{(w_{\text{in}})} \pm \tilde{\mathbf{x}}^{(w_{\text{in}})}) = \mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \pm \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}).$$

Obviously,  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})} \pm \tilde{\mathbf{x}}^{(w_{\text{in}})}) \in \Pi$ . Subsequently,  $\mathcal{B}(\mathbf{x}^{(w_{\text{in}})}) \pm \mathcal{B}(\tilde{\mathbf{x}}^{(w_{\text{in}})}) \in \Pi$ , as needed.

Since the boundedness and the threshold properties were shown in [2, 5] to be preserved under the introduction of antitokens (see also Theorems 1 and 2), the necessary condition of Theorem 3 implies that these properties are closed under the nullity of any balancing network.

The sufficient condition of Theorem 3 can be used to determine if any given property is preserved with antitokens. In general, we are given a property  $\Pi$  which we know it is satisfied by a balancing network when the network is accessed by tokens only. We want to find out if this property will still be preserved even when the network is accessed by antitokens too. In order to show this, the sufficient condition of Theorem 3 implies that we only need to prove that the property is closed under the nullity of the network.

We can strengthen Definition 1 and Theorem 3 so that in their statements, the non-negative input vectors and null vectors, are restricted to vectors with entries in the range  $[0, W_{\text{out}}(\mathcal{B})]$ . This way, we obtain a new verification procedure for identifying whether a particular network  $\mathcal{B}$  satisfies a property closed under the nullity of a network. In particular, if property  $\Pi$  is closed under the nullity of a network  $\mathcal{B}$  (for input vectors and null vectors with entries in the range  $[0, W_{\text{out}}(\mathcal{B})]$ ), Theorem 3 implies that in order to verify that  $\mathcal{B}$  satisfies the property  $\Pi$ , it suffices to verify that all vectors with entries in the interval  $[0, W_{\text{out}}(\mathcal{B})]$  satisfy  $\Pi$ . We can simply feed all these vectors to the network and examine if each respective output vector satisfies  $\Pi$ . This is the *first* verification procedure established for properties satisfied by balancing networks that are traversed by both tokens and antitokens. (For more about verification algorithms see [4, 6].)

## 5 Conclusion

We have provided a combinatorial characterization of the properties satisfied by balancing networks traversed by tokens alone that are preserved when antitokens are introduced. Our results close the main problem left open by the results in [2, 5]. An interesting question still left open by our work is to provide a corresponding characterization for *randomized* balancing networks [3], where the balancers distribute the tokens on their output wires following some random permutation.

## References

1. E. Aharonson and H. Attiya, "Counting Networks with Arbitrary Fan-Out," *Distributed Computing*, Vol. 8, pp. 163–169, 1995.
2. W. Aiello, C. Busch, M. Herlihy, M. Mavronicolas, N. Shavit, and D. Touitou, "Supporting Increment and Decrement Operations in Balancing Networks," *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science*, G. Meinel and S. Tison eds., pp. 377–386, Vol. 1563, Lecture Notes in Computer Science, Springer-Verlag, Trier, Germany, March 1999. Also, to appear in the *Chicago Journal of Theoretical Computer Science*.
3. W. Aiello, R. Venkatesan and M. Yung, "Coins, Weights and Contention in Balancing Networks," *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing*, pp. 193–205, Los Angeles, California, August 1994.
4. J. Aspnes, M. Herlihy and N. Shavit, "Counting Networks," *Journal of the ACM*, Vol. 41, No. 5, pp. 1020–1048, September 1994.
5. C. Busch, N. Demetriou, M. Herlihy and M. Mavronicolas, "Threshold Counters with Increments and Decrements," *Proceedings of the 6th International Colloquium on Structural Information and Communication Complexity*, pp. 47–61, Lacanau, France, July 1999.
6. C. Busch and M. Mavronicolas, "A Combinatorial Treatment of Balancing Networks," *Journal of the ACM*, Vol. 43, No. 5, pp. 794–839, September 1996.
7. C. Busch and M. Mavronicolas, "Impossibility Results for Weak Threshold Networks," *Information Processing Letters*, Vol. 63, No. 2, pp. 85–90, July 1997.
8. E. W. Dijkstra, "Cooperating Sequential Processes," *Programming Languages*, pp. 43–112, Academic Press, 1968.
9. D. Grunwald and S. Vajracharya, "Efficient Barriers for Distributed Shared Memory Computers," *Proceedings of the 8th International Parallel Processing Symposium*, IEEE Computer Society Press, April 1994.
10. P. B. Hansen, *Operating System Principles*, Prentice Hall, Englewood Cliffs, NJ, 1973.
11. M. Herlihy, B.-C. Lim and N. Shavit, "Concurrent Counting," *ACM Transactions on Computer Systems*, Vol. 13, No. 4, pp. 343–364, 1995.
12. M. Herlihy, N. Shavit and O. Waarts, "Linearizable Counting Networks," *Distributed Computing*, Vol. 9, pp. 193–203, 1996.
13. C. A. R. Hoare and R. N. Periott, *Operating Systems Techniques*, Academic Press, London, 1972.
14. S. Kapidakis and M. Mavronicolas, "Distributed, Low Contention Task Allocation," *Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing*, pp. 358–365, New Orleans, Louisiana, October 1996.
15. D. E. Knuth, "The Art of Computer Programming III: Sorting and Searching," Vol. 3, Addison-Wesley, 1973.
16. N. Shavit and D. Touitou, "Elimination Trees and the Construction of Pools and Stacks," *Theory of Computing Systems*, Vol. 30, No. 6, pp. 545–570, November/December 1997.
17. S. Zhou, X. Zheng, J. Wang and P. Delisle, "Utopia: A Load Sharing Facility for Large, Heterogeneous Distributed Computer Systems," *Software-Practice and Experience*, Vol. 23, No. 12, pp. 1305–1336, December 1993.