Backward Consistency and Sense of Direction in Advanced Distributed Systems*

(Extended Abstract)

Paola Flocchini
Université du Québec à Hull (flocchini@uqah.uquebec.ca)

Alessandro Roncato
CDL Informatica - Università di Venezia (roncato@dsi.unive.it)

Nicola Santoro
Carleton University, (santoro@scs.carleton.ca)

Abstract: The studies on the relationship between label consistency, computability and complexity assume the existence of *local orientation*; this assumption is in fact at the basis of the *point-to-point* model and is realistic for systems where a communication link can connect only two entities. However, in systems which use more advanced communication and interconnection technology such as *buses*, *optical* networks, *wireless* communication media, etc., and more importantly, heterogeneous systems (such as *internet*) which include any combination of the above, local orientation can *not* be assumed.

In this paper we consider a new type of consistency which we shall call *backward consistency* and which, unlike sense of direction, can exist even without local orientation. Thus, unlike all previous forms of consistency, it can be found (or designed) in advanced distributed systems.

We study backward consistency both in terms of its relationship with the traditional properties of local orientation and (weak) sense of direction, and with respect to symmetries of the edge labelings and of the naming functions. We prove that backward consistency is computationally equivalent to sense of direction; in other words, it is possible to take advantage of the computational power of sense of direction even in absence of local orientation.

1 Introduction

A distributed system is a collection of computational entities communicating by exchanging messages. Depending on how the communication is achieved, different models exist. In the *point-to-point* model, the communication topology of the system is viewed as an edge-labeled undirected graph $(G = (V, E), \lambda)$ where nodes correspond to the system entities, edges represent pairs of neighboring entities (i.e., entities which can communicate directly), each node $x \in V$ has a local label (usually called port number) $\lambda_x(x, y)$ associated to each of its incident edges (x, y), and $\lambda = \{\lambda_x : x \in V\}$.

In the studies of computing in distributed systems, recurring challanging questions are on the relationships between *local views* and *global consistency*, and on their impact on distributed computability and communication complexity. In these studies, *locality* is at the entity level: the local view of an entity x consists of just the labeling λ_x of its communication ports (or, communication links, incident edges) and, sometimes, of the naming β_x it uses to refer to the other entities. A major goal of these investigation is to derive under what properties of λ and/or β it is possible to infer global properties about the entire system. A practical aim is to understand if and how these properties can be usefully employed, e.g., to yield more efficient distributed computations.

In particular, there is a large body of evidence on the positive impact on complexity of the set of global consistency constraints satisfied by labelings with sense of direction (e.g., see [14, 15, 25, 27, 35]). Properties of Sense of Direction have been studied, for example, in [4, 5, 6, 8, 13, 19, 18, 20, 21, 36, 37]; for a recent survey, see [17].

All studies in the relevant literature, with very few exceptions, make the basic (but often unstated) fundamental assumption that the entities are able to distinguish among their incident links, i.e., the labelings λ_x are injective functions. Even if the links might not have a label assigned yet, it is assumed that an entity is capable of such an assignment (e.g., [10, 11, 22, 26]). This assumption, usually called *local orientation*, is in fact at the basis of the *point-to-point* model and is realistic for systems where a communication link can connect only two entities.

If we are to model systems which use more advanced communication and interconnection technology such as *buses*, *optical* networks, *wireless* communication media, etc., and more importantly, heterogeneous systems (such as *internet*) which include any combination of the above, local orientation can *not* be assumed. This is because any direct connection between k entities will correspond, at each of those entities, to $k-1$ edges with the same label; hence, if $k > 2$ (e.g., a bus), λ is not injective. That is, unless the system is point-to-point (i.e., each connection is only between two entities) there are nodes that cannot distinguish between some of their incident edges. This implies that all the established body of results on the relationship between consistency, computability and complexity does not hold and, thus, can not be applied outside the point-to-point model.

*Research supported in part by F.C.A.R and N.S.E.R.C

In other words, very little is known on these subject for systems with advanced communication technology. (A vast body of knowledge does indeed exist on these systems, but not on these topics).

There are few exceptions: the early work on computing in complete graphs with "ambigous" labelings [34, 21], the more recent studies on computability in anonymous "wireless" networks [31, 9], the results on leader election and function evaluation in anonymous networks with different types of "port awareness" [7, 38]. Note that "ambigous", "wireless" and "port aware" all denote the (possible) absence of local orientation.

In this paper we consider a new type of consistency which we shall call *backward consistency.* Backward consistency is strongly related to the classical "forward" consistency implied by sense of direction; however, unlike sense of direction, it can exist even without local orientation. Thus, unlike all previous forms of consistency, backward consistency can be found (or designed) in advanced distributed systems.

In the following, we study backward consistency both in terms of its relationship with the traditional properties of local orientation and (weak) sense of direction, and with respect to symmetries of the edge labelings and of the naming functions. In particular, we focus on the conditions under which a labelling can have simultaneously both forward and backward consistency.

We then consider the power of backward consistency, and prove that *backward consistency is computationally equivalent to sense of direction.* In other wards, it is possible to take advantage of the computational power of sense of direction even in absence of local orientation. This counterintuitive result somehow extends and generalizes the results of [1, 9] on the insensitivity of ring networks to local orientation. In addition to the theoretical result, we also provide a simple effective procedure to efficiently map protocols exploiting sense of direction into ones using "backward sense of direction".

In the following we consider the undirected case (i.e., bidirectional communication capabilities); this is done only for simplicity of exposition, as all results extend to and hold also in the directed case.

2 Basic Definitions and Properties

2.1 Labelings and Sense of Direction

In this section we first recall the basic definitions of consistency, symmetry and sense of direction in labeled graphs; most of the terminology follows [16].

Let $G = (V, E)$ be a simple undirected graph, $E(x)$ denote the set of edges incident to node $x \in V$. Given $G = (V, E)$ and a set Σ of labels, a *local labeling function* of $x \in V$ is any function $\lambda_x : E(x) \to \Sigma$ which associates a label $l \in \Sigma$ to each edge $e \in E(x)$. A set $\lambda = \{\lambda_x : x \in V\}$ of local labeling functions will be called a *labeling* of G, and by (G, λ) we shall denote the corresponding *(edge-)labeled graph.* If λ is injective, we say that it is a *local orientation.* Note that local orientation is assumed in the point-to-point model and in almost all literature on informative labelings, including the one on unlabeled networks (e.g., [10, 22, 26]). A labeling λ is *symmetric* if there exists a bijection $\psi : \Sigma \to \Sigma$ such that for each $\langle x, y\rangle \in E$, $\lambda_y(\langle y, x\rangle) = \psi(\lambda_x(\langle x, y\rangle))$; ψ will be called the edge-symmetry function.

A *walk* π in G is a sequence of edges in which the endpoint of one edge is the starting point of the next edge. Let $P[x]$ denote the set of all the walks starting from $x \in V$, $P[x, y]$ the set of walks starting from $x \in V$ and ending in $y \in V$. Let $\Lambda_x : P[x] \to \Sigma^+$ and $\Lambda = \{\Lambda_x : x \in V\}$ denote the extension of λ_x and λ, respectively, from edges to walks; let $\Lambda[x] = \{\Lambda_x(\pi) : \pi \in P[x]\}$, and $\Lambda[x, y] = \{\Lambda_x(\pi) : \pi \in P[x, y]\}$.

Given an edge symmetry function ψ, we shall denote by $\Psi : \Sigma^+ \to \Sigma^+$ its extension to strings; i.e., for $\alpha = a_1 \cdot a_2 \cdot \ldots \cdot a_p \in \Sigma^+$, $\Psi(\alpha) = \psi(a_p) \cdot \ldots \cdot \psi(a_1)$, where $\cdot$ denotes the concatenation operator.

Informally, a labeled graph (G, λ), has *sense of direction* when it is possible to understand, from the labels associated to the edges, whether different walks from any given node x end in the same node or in different ones.

A *coding* function of (G, λ) is any function $\mathbf{c}$ with domain Σ^+. It is said to be *consistent* if $\forall x, y, z \in V$, $\forall \pi_1 \in P[x, y]$, $\pi_2 \in P[x, z]$ $\mathbf{c}(\Lambda_x(\pi_1)) = \mathbf{c}(\Lambda_x(\pi_2))$ iff $y = z$. In other words, walks *originating* from the same node are mapped to the same value iff they end in the same node. We shall denote by $\mathcal{N}(\mathbf{c})$ the co-domain of $\mathbf{c}$.

Definition 1 $\mathcal{WSD}$ - Weak Sense of Direction
A labeled graph (G, λ) has weak sense of direction $\mathbf{c}$ *iff $\mathbf{c}$ is a consistent coding function of (G, λ). Alternatively, we shall say that $\mathbf{c}$ is a $\mathcal{WSD}$ in (G, λ).*

It is immediate to see that consistency requires local orientation:

Lemma 1 *Let (G, λ) have $\mathcal{WSD}$ $\mathbf{c}$. Then λ has local orientation.*

Given a coding function $\mathbf{c}$, a *decoding function* $\mathbf{d}$ for $\mathbf{c}$ is any function $\mathbf{d} : \Sigma \times \mathcal{N}(\mathbf{c}) \to \mathcal{N}(\mathbf{c})$ such that $\forall \langle x, y\rangle \in E(x)$, $\pi \in P[y, z]$ $\mathbf{d}(\lambda_x(x, y), \mathbf{c}(\Lambda_y(\pi))) = \mathbf{c}(\lambda_x(x, y) \cdot \Lambda_y(\pi))$, where $\cdot$ is the concatenation operator.

Definition 2 $\mathcal{SD}$ - Sense of Direction
A labeled graph (G, λ) has sense of direction $(\mathbf{c}, \mathbf{d})$ *iff $\mathbf{c}$ is a $\mathcal{WSD}$ and $\mathbf{d}$ is a decoding function for $\mathbf{c}$. Alternatively, we shall say that $(\mathbf{c}, \mathbf{d})$ is a $\mathcal{SD}$ in (G, λ).*

Let $\mathcal{L}$, $\mathcal{W}$, and $\mathcal{D}$ denote the set of labeld graphs (G, λ) with local orientation, weak sense of direction, and sense of direction, respectively. The following relationship holds:

Lemma 2 *[5, 16] $\mathcal{D} \subset \mathcal{W} \subset \mathcal{L}$*

The proof follows from the fact that local orientation is necessary but (obviously) not sufficient for consistency [16], and the fact that there exist labeled graphs with $\mathcal{WSD}$ but without $\mathcal{SD}$ [5].

2.2 Backward Consistency and Sense of Direction

In this section we introduce the notion of backward consistency. A coding function $\mathbf{c}$ of (G, λ) is *backward consistent* if $\forall x, y, z \in V$, $\forall \pi_1 \in P[x, z]$, $\pi_2 \in P[y, z]$ $\mathbf{c}(\Lambda_x(\pi_1)) = \mathbf{c}(\Lambda_y(\pi_2)) \Leftrightarrow x = y$. In other words, sequences of labels of walks *terminating* in the same node are mapped to the same value iff they start from the same node.

Note that the difference between the consistency just defined and the traditional one is of "viewpoint". In the latter, the focus is *forwards*, on the sequence of labels on walks leaving from a given node; in the former the focus is *backwards* on the sequences of labels on walks which terminate at a given node.

Definition 3 $\mathcal{WSD}^-$ Weak Backward Sense of Direction
A system (G, λ)*, has* weak backward sense of direction $\mathbf{c}$ *iff* $\mathbf{c}$ *is a backward consistent coding function. Alternatively, we shall say that* $\mathbf{c}$ *is a* $\mathcal{WSD}^-$ *in* (G, λ).

Given a coding function $\mathbf{c}$, a *backward decoding function* $\mathbf{b}$ for $\mathbf{c}$ is any function $\mathbf{b} : \Sigma \times \mathcal{N}(\mathbf{c}) \to \mathcal{N}(\mathbf{c})$ such that $\forall \pi \in P[x, y]$, $(y, z) \in E(y)$, $\mathbf{b}(\mathbf{c}(\Lambda_x(\pi)), \lambda_y(y, z)) = \mathbf{c}(\Lambda_x(\pi) \cdot \lambda_y(y, z))$ where $\cdot$ is the concatenation operator. (Recall that $\mathcal{N}(\mathbf{c})$ denotes the codomain of $\mathbf{c}$). We can now define backward sense of direction

Definition 4 $\mathcal{SD}^-$ Backward Sense of Direction
A system (G, λ) *has* backward sense of direction $(\mathbf{c}, \mathbf{d}_b)$ *iff* $\mathbf{c}$ *is a* $\mathcal{WSD}^-$ *and* $\mathbf{b}$ *is a backward decoding function for* $\mathbf{c}$. *Alternatively, we shall say that* $(\mathbf{c}, \mathbf{d}_b)$ *is a* $\mathcal{SD}^-$ *in* (G, λ).

We shall denote by $\mathcal{W}^-$ and $\mathcal{D}^-$ the set of labeled graphs (G, λ) with backward weak sense of direction and backward sense of direction, respectively. In the following, when no ambiguity arises, we shall omit the reference to (G, λ).

3 Backward Consistency

3.1 Absence of Local Orientation

The first basic difference between forward and backward consistency is striking: Local Orientaton is **not** implied by the definition of $\mathcal{SD}^-$.

Theorem 1 $(\mathcal{SD}^- \not\Rightarrow \exists\mathcal{L})$ $(\mathcal{L} \not\Rightarrow \exists\mathcal{SD}^-)$
Local orientation is neither necessary nor sufficient for $\mathcal{SD}^-$.

Proof Consider the labeled graph (G, λ) of Figure 1. It is easy to verify that there exists both consistent backward coding and decoding functions in (G, λ). However there is no local orientation. The fact that it is not sufficient is obvious (see, for example, the graph of Figure 3 which has local orientation but not sense of direction backward). □

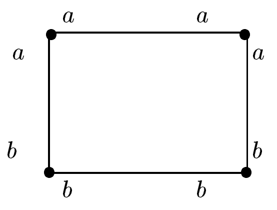

Figure 1: $\mathcal{SD}^- \not\Rightarrow \exists\mathcal{L}$

Thus, in a system with $\mathcal{SD}^-$, nodes might not be able to distinguish among some of their incident edges. As the example of Figure 1 shows, in a system with backward sense of direction, this "blindness" can be even *complete* (i.e., it extends to all incident edges of a node) and, in the extreme case, *total* (i.e., it occurs at every node).

This phenomenon is not restricted to a few special graphs. On the contrary, *every* graph G can be labeled so to have (complete and) total blindness, and still have backward sense of direction.

Theorem 2 *For any graph* G *there exists a labeling* λ *such that* $\forall x \in V$*,* $\forall y, z \in E(x)$*,* $\lambda_x((x, y)) = \lambda_x((x, z))$*, but* (G, λ) *has* $\mathcal{SD}^-$.

Proof Consider the labeling λ defined as follows: $\forall (x, y)$, $(w, z) \in E$, $\lambda_x((x, y)) = \lambda_w((w, z))$ iff $x = w$. With this labeling, all the links starting from the same node x are labeled with the same label (hence, there is total and complete blindness). It is not difficult to show that the coding function $\mathbf{c}$ defined as follows: $\mathbf{c}(a \cdot \alpha) = a$ for all $a \in \Sigma, \alpha \in \Sigma^+$ is backward consistent, and that the function $\mathbf{d}(\mathbf{c}(\alpha), a) = \mathbf{c}(\alpha)$ is a backward decoding for $\mathbf{c}$; hence, (G, λ) has backward sense of direction. □

3.2 Backward Local Orientation

Systems with backward consistency may be without local orientation; however, they do have a property which is the "backward" analog of local orientation. A labeling λ is a *backward local orientation* if: $\forall x, y, z : y, z \in E(x), \lambda_y((y, x)) \neq \lambda_z((z, x))$. Let $\mathcal{L}^-$ denote the set of labeled graphs (G, λ) with backward local orientation.

First observe that backward local orientation is (obviously) *not sufficient* for backward consistency.

Theorem 3 $\mathcal{L}^- \not\Rightarrow \exists\mathcal{WSD}^-$
Backward local orientation does not suffice for backward consistency.

Proof Consider the labeled graph of Figure 2. Such a graph has clearly backward local orientation. By contradiction, let us assume that it has backward weak sense of direction. By definition of backward coding function at node y we would have that $\mathbf{c}(b \cdot a) = \mathbf{c}(c \cdot b)$. However the labels by definition of backward consistency at node x, we must have: $\mathbf{c}(b \cdot a) \neq \mathbf{c}(c \cdot b)$ □

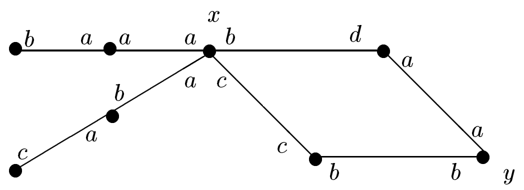

Figure 2: $\mathcal{L}^- \not\Rightarrow \exists\mathcal{WSD}^-$.

Notice that the labeling in the proof of the previous theorem does not have local orientation, thus it actually proves that $(\mathcal{L}^- - \mathcal{W}^-) - \mathcal{L} \neq \emptyset$.
However, in a system with $\mathcal{WSD}^-$, there is always backward local orientation.

Theorem 4 $\mathcal{WSD}^- \Rightarrow \mathcal{L}^-$
If (G, λ) *has* $\mathcal{WSD}^-$*, then* λ *has backward local orientation.*

Proof Let (G, λ) be a labeled graph without backward local orientation and with backward coding function $\mathbf{c}$. Then there exist two edges (y, x) and (z, x) such that $\lambda_y(y, x) = \lambda_z(z, x)$. But this contradicts the existence of backward consistency, in fact by definition of backward consistency $\mathbf{c}(\lambda_y(y, x)) \neq \mathbf{c}(\lambda_z(z, x))$ □

Summarizing, (backward) local orientation is necessary but not sufficient for (backward) consistency. Thus, $\mathcal{W}^- \subset \mathcal{L}^-$.

Two questions naturally arises: is the simultaneous presence of both local orientation and backward local orientation sufficient for consistency ? Is it sufficient for backward consistency ? The answer to these questions is negative as we prove the stronger result that there exist labeled graphs having both forms of local orientation but without either form of consistency.

Theorem 5 $(\mathcal{L} \cap \mathcal{L}^-) - (\mathcal{W} \cup \mathcal{W}^-) \neq \emptyset$
The simultaneous presence of local orientation and backward local orientation does not imply neither weak sense of direction nor backward weak sense of direction.

Proof Consider the labeled graph (G, λ) shown in Figure 3. The labeling has clearly both local orientation and local orientation backward. By contradiction, assume (G, λ) has weak sense of direction $\mathbf{c}$. By definition of consistent coding at node x, it must be: $\mathbf{c}(a \cdot b) = \mathbf{c}(c)$; however, at node y it must be $\mathbf{c}(a \cdot b) \neq \mathbf{c}(c)$ yielding a contradiction. Assume now, again by contradiction, that (G, λ) has backward weak sense of direction $\mathbf{c}$. By definition of backward consistent coding at node z, it must be: $\mathbf{c}(a \cdot b) = \mathbf{c}(c)$; however at node w,$\mathbf{c}(a \cdot b) \neq \mathbf{c}(c)$, which is a contradiction. □

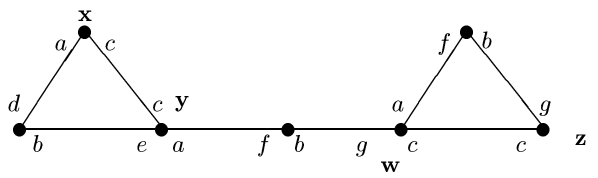

Figure 3: $\mathcal{LO}$, $\mathcal{LO}^- \not\Rightarrow \exists\mathcal{WSD}$

3.3 Orthogonality

In this section we will show that the concept of backward consistency is *orthogonal* to the one of sense of direction; in particular, even the presence of sense of direction is not sufficient to guarantee the existence of a backward consistency.

Theorem 6 *$(\mathcal{D} - \mathcal{L}^- \neq \emptyset)$, $(\mathcal{L}^- - \mathcal{D} \neq \emptyset)$*
Sense of direction is neither necessary nor sufficient for the existence of backward local orientation.

Proof Consider any graph with more than two nodes labeled with a *neighbouring* labeling (e.g., see Figure 4); all such labeled graphs have sense of direction [16]: the coding function $\mathbf{c}$ is $\mathbf{c}(\alpha \cdot a) = a$ for all $a \in \Sigma, \alpha \in \Sigma^+$, and the decoding function is $\mathbf{d}(a, \mathbf{c}(\alpha)) = \mathbf{c}(\alpha)$. On the other hand, (G, λ) does not have local orientation backward. In other words, $\mathcal{SD}$ is not sufficient for the existence of $\mathcal{LO}^-$. The fact that it is also not necessary follows from Theorem 5. □

In other words, sense of direction cannot guarantee backward consistency because it cannot even guarantee backward local orientation. What if there is backward local orientation in addition to sense of direction? The answer is still negative:

Theorem 7 $(\mathcal{D} \cap \mathcal{L}^-) - \mathcal{W}^- \neq \emptyset$
Simultaneous presence of sense of direction and backward local orientation is neither necessary nor sufficient for the existence of a backward consistent coding function.

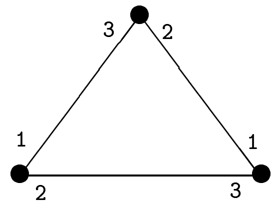

Figure 4: $\mathcal{D} - \mathcal{L}^- \neq \emptyset$

Proof To prove that it is not necessary, consider the labeled graph (G, λ) of Figure 5. It is easy to verify that there exist both a consistent coding and decoding function in (G, λ). By contradiction, assume that there exists a backward consistent coding function $\mathbf{c}$ in (G, λ). Then, by definition of backward consistency at node x, we have that $\mathbf{c}(1 \cdot 1) = \mathbf{c}(2 \cdot 2)$. However, by definition of backward consistency at node y, $\mathbf{c}(1 \cdot 1) \neq \mathbf{c}(2 \cdot 2)$; yielding the contraddiction.
Backward consistency clearly implies local orientation backward. However, from Theorem 6 it follows that backward consistency does not imply local orientation and, thus, does not imply sense of direction. Which means that the simultaneous presence of sense of direction and backward local orientation is not sufficient for backward consistency. □

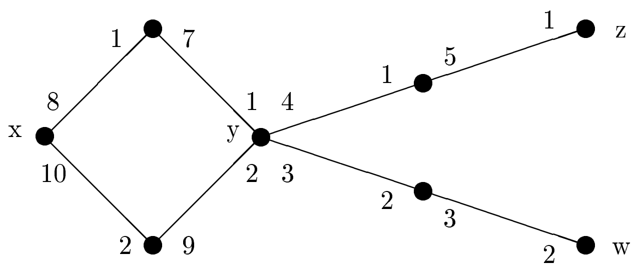

Figure 5: $(\mathcal{D} \cap \mathcal{L}^-) - \mathcal{W}^- \neq \emptyset$

The results of this section indicate that the presence of some additional property is necessary for one type of consistency to imply the other. We will show in the next section that *edge-symmetry* is one such property.

4 Symmetry and Backward Consistency

The results of the previous section indicate that the presence of some additional property is necessary for one type of consistency to imply the other. We will show that *edge-symmetry* is one such property.

Recall that a labeling λ is *symmetric* if there exists a bijection $\psi : \Sigma \to \Sigma$ such that for each $(x, y) \in E$, $\lambda_y((y, x)) = \psi(\lambda_x((x, y)))$; ψ will be called the edge-symmetry function, and $\Psi : \Sigma^+ \to \Sigma^+$ is its extension to strings. Notice that all common labelings (e.g., "dimensional" in hypercubes, "compass" in meshes and tori, "left-right" in rings, "distance" in chordal rings, etc.) are symmetric.

4.1 Edge-Symmetry and Backward Consistency

We will show that, while in general not true (as shown by Theorem 1), in systems with edge symmetry local orientation is necessary for backward consistency. In fact, we shall show that, in systems with edge symmetry, $\mathcal{L} = \mathcal{L}^-$.

Theorem 8 *($\mathcal{ES}$: $\mathcal{L} \Leftrightarrow \mathcal{L}^-$)*
Let (G, λ) be a system with edge symmetry; then, there is local orientation iff there is local orientation backward.

Proof Let (G, λ) be a system with edge symmetry.
$\Rightarrow$) By contradiction, suppose there is no local orientation backward. Then, there must exist two edges (x, y), (x, z) such that $\lambda_y((y, x)) = \lambda_z((z, x))$. However, by local orientation, we have that: $\lambda_x((x, y)) \neq \lambda_x((x, z))$ which contradicts the fact that there is edge symmetry.
$\Leftarrow$) By contradiction, suppose there is no local orientation. Then, there must exist two edges (z, x), (z, y) such that:$\lambda_z((z, x)) = \lambda_z((z, y))$. However, by local orientation backward we must have $\lambda_x((x, z)) \neq \lambda_x((y, z))$ which contradicts the fact that there is edge symmetry. $\Box$

Thus, a system with edge symmetry either has both types of local orientation or none; still, presence of both and edge symmetry are not sufficient for backward consistency.

Theorem 9 $\mathcal{ES} : \mathcal{L}, \mathcal{L}^- \not\Rightarrow \mathcal{WSD}^-$
The simultaneous presence of local orientation and edge symmetry is not sufficient for the existence of backward consistency.

Proof Consider the labeled graph of Figure 6. The labeling is a coloring, i.e. the edge symmetry function is the identity function; moreover, it is a local orientation. By contradiction, let us assume that there exist a backward consistent coding function $\mathbf{c}$. By definition of backward consistency at node x, we have that $\mathbf{c}(a \cdot b) = \mathbf{c}(c \cdot d)$ and at node y, $\mathbf{c}(c \cdot d) = \mathbf{c}(e \cdot f)$, which implies $\mathbf{c}(a \cdot b) = \mathbf{c}(e \cdot f)$. The sequence of labels $b \cdot a$ corresponds to a path in $P[w, z]$, and $f \cdot e$ corresponds to a path in $P[v, z]$; thus we have that $\mathbf{c}(a \cdot b) \neq \mathbf{c}(e \cdot f)$ which is a contradiction. $\Box$

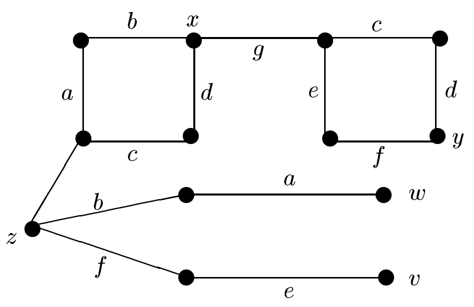

Figure 6: $\mathcal{ES} : \mathcal{L}, \mathcal{L}^- \not\Rightarrow \mathcal{W}^-$

We will now show that edge symmetry in a system with (weak) sense of direction suffices for the system to have (weak) backward sense of direction.

Theorem 10 *($\mathcal{ES}$, $\exists(\mathcal{W})\mathcal{SD} \Rightarrow \exists(\mathcal{W})\mathcal{SD}^-$)*
Let (G, λ) be a system with edge symmetry. If (G, λ) has $\mathcal{WSD}$ $\mathbf{c}$, then, there exists a $\mathcal{WSD}^-$ $\mathbf{c}_b$. Furthermore, if $\mathbf{c}$ has a consistent decoding, there exists a consistent backward decoding of $\mathbf{c}_b$.

Conversely, edge symmetry in a system with (weak) backward sense of direction suffices for the system to have (weak) sense of direction.

Theorem 11 *($\mathcal{ES}$, $\exists(\mathcal{W})\mathcal{SD}^- \Rightarrow \exists(\mathcal{W})\mathcal{SD}$)*
Let (G, λ) be a system with edge symmetry. If (G, λ) has $\mathcal{WSD}^-$ $\mathbf{c}_b$, then, there exists a $\mathcal{WSD}$ $\mathbf{c}$. Furthermore, if $\mathbf{c}_b$ has a consistent backward decoding, there exists a consistent decoding of $\mathbf{c}$.

From Theorems 10 and 11, it follows that systems with edge symmetry either have both types of consistency or none. In other words, in systems with edge symmetry, $\mathcal{W} = \mathcal{W}^-$ and $\mathcal{D} = \mathcal{D}^-$.

An immediate question is whether systems with edge-symmetry are the only ones with such a property.

Theorem 12 *Edge symmetry is not necessary for a system to have both forward and backward consistency.*

4.2 Edge Symmetry and Biconsistency

In the previous section we have seen that all systems with edge symmetry either have both types of consistency or neither. For those wo do have them, the corresponding coding functions are in general different.

The question we ask now is under what circumstances a single coding function suffices; that is, when a coding function is both forward and backward consistent. We shall call any such function *biconsistent*. The first result is that

Theorem 13 *($ES, \mathcal{WSD} \not\Rightarrow \mathcal{WSD} = \mathcal{WSD}^-$)*
Edge symmetry is not sufficient for a (backward) consistent coding function to be biconsistent.

However, if there is name symmetry, then any consistent coding function is biconsistent. A weak sense of direction $\mathbf{c}$ has *name symmetry* iff there exists a function $\mu : \mathcal{N}(\mathbf{c}) \to \mathcal{N}(\mathbf{c})$ such that $\forall \pi \in P[x, y]$: $\mu(\mathbf{c}(\Lambda_x(\pi))) = \mathbf{c}(\Lambda_y(\overline{\pi}))$. A useful property is the following [18]:

Lemma 3 *[18] Let λ be a symmetric labeling and ψ be the corresponding function. A consistent coding function $\mathbf{c}$ has name symmetry iff $\forall \pi_1 \in P[s, t], \pi_2 \in P[w, z]$*
$\mathbf{c}(\Lambda_s(\pi_1)) = \mathbf{c}(\Lambda_w(\pi_2)) \Rightarrow c(\Psi(\Lambda_s(\pi_1))) = \mathbf{c}(\Psi(\Lambda_w(\pi_2)))$.

Theorem 14 *($\mathcal{ES}$, $\mathcal{NS} \Rightarrow \mathcal{WSD} = \mathcal{WSD}^-$)*
In a system (G, λ) with edge symmetry, any $\mathcal{WSD}$ with name symmetry is also $\mathcal{WSD}^-$.

Proof Let $\mathbf{c}$ be a $\mathcal{WSD}$ in (G, λ) with edge and name symmetry and let ψ be the edge symmetry function. By contradiction, suppose that c is not a consistent backward coding function; that is, suppose $\exists x, y, z$, $\pi_1 \in P[x, z]$, $\pi_2 \in P[y, z]$ with $x \neq y$, such that $\mathbf{c}(\Lambda_x(\pi_1)) = \mathbf{c}(\Lambda_y(\pi_2))$. Since there is name symmetry, by Lemma 3, we have that $\mathbf{c}(\Psi(\Lambda_x(\pi_1))) = \mathbf{c}(\Psi(\Lambda_y(\pi_2)))$. But $\Lambda_z(\overline{\pi}_1) = \Psi(\Lambda_x(\pi_1))$, $\Lambda_z(\pi_2) = \Psi(\Lambda_x(\pi_1))$ and $\overline{\pi}_1 \in P[z, x]$, $\overline{\pi}_2 \in P[z, y]$ which contradicts the consistency of c in z. $\Box$

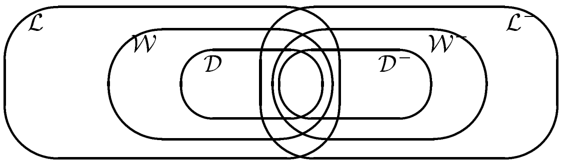

Figure 7: The consistency landscape

In other words, in systems with edge symmetry, any consistent coding function with name symmetry is also backward consistent. We will now show that if any such coding function is decodable, then it is also backward decodable; that is, if there is sense of direction, there is also backward sense of direction with exactly the same coding function.

Theorem 15 *Let* $(\mathbf{c}, \mathbf{d})$ *be a* $\mathcal{SD}$ *in a system* (G, λ) *with edge and name symmetry. Then, there exists a backward decoding* $\mathbf{d}_b$ *of* $\mathbf{c}$*; that is,* $(\mathbf{c}, \mathbf{d}_b)$ *is a* $\mathcal{SD}^-$ *in* (G, λ)*.*

5 Consistency Landscape

The results of the previous sections can be seen, and have sometimes been stated, in terms of the "consistency landscape" (see figure 7), i.e., of the relationship among the sets $\mathcal{L}, \mathcal{W}, \mathcal{D}, \mathcal{L}^-, \mathcal{W}^-, \mathcal{D}^-$.

In this section, we continue the analysis of the "consistency landscape'.

5.1 Transformations and Constructions

In this section, we introduce two useful operations on labeled graphs. In the end, we shall derive general results on the structure of the consistency landscape.

The first operation, called "doubling", allows to transform a labeling into a *symmetric* one. Hence, it allows to transform a system with only one type of consistency into one which has both.

Given (G, λ), the *doubling* of λ is the mapping $\lambda^2_x((x,y))$ defined as follows: $\forall (x,y) \in E$, $\lambda^2_x((x,y)) = (\lambda_x((x,y)), \lambda_y((y,x)))$. Its extension from edges to walks will be denoted by Λ^2. The most important property of double labeling is that if (G, λ) has either form of consistency, then (G, λ^2) has both. Given two strings of equal lenght $\alpha = (a_0, a_1, \ldots, a_k) \in \Sigma^+$ and $\beta = (b_0, b_1, \ldots, b_k) \in \Sigma^+$, let $\alpha \oplus \beta = ((a_0, b_0), (a_1, b_1), \ldots, (a_k, b_k)) \in (\Sigma^2)^+$ denote their product.

Theorem 16 *Double Labeling*
If (G, λ) *has either* $(\mathcal{W})\mathcal{SD}$ *or* $(\mathcal{W})\mathcal{SD}^-$ *then* (G, λ^2) *has both* $(\mathcal{W})\mathcal{SD}$ *and* $(\mathcal{W})\mathcal{SD}^-$*.*

Proof Let c be a coding function in (G, λ). Consider the coding function c_2 defined as follows: for all $\alpha \oplus \beta \in (\Sigma^2)^+$ $c_2(\alpha \oplus \beta) = c(\alpha)$. Clearly, c_2 is (resp. backward) consistent in (G, λ^2) if and only if c is (resp. backward) consistent in (G, λ).
Let c, and hence c_2, be (resp. backward) consistent. Given a function $d : \Sigma \times \mathcal{N}(\mathbf{c}) \rightarrow \mathcal{N}(\mathbf{c})$ define the function d_2 as follows: for all $(a,b) \in \Sigma^2$ and $\alpha \oplus \beta \in (\Sigma^2)^+$, $d_2((a,b), c_{\lambda^2}(\alpha \oplus \beta)) = d(a, c(\alpha))$. Clearly, d_2 is a decoding of c_2 if and only if d is a decoding of c. Similiarly, given $d^- : \mathcal{N}(\mathbf{c}) \times \Sigma \rightarrow \mathcal{N}(\mathbf{c})$, the function $d^-_2(c_2(\alpha \oplus \beta), (a,b)) = d(a, c(\alpha))$ is a backward decoding of c_2 if and only if d^- is a backward decoding of c. In other words, $(\mathcal{W})\mathcal{SD}$ in (G, λ) implies $(\mathcal{W})\mathcal{SD}$ in (G, λ^2), and $(\mathcal{W})\mathcal{SD}^-$ in (G, λ) implies $(\mathcal{W})\mathcal{SD}^-$ in (G, λ^2)
Finally observe that λ^2 is symmetric; in fact, for every $(x,y) \in E$, $\lambda^2_x(x,y) = (\lambda^2_y(y,x))^R$. Thus, by Theorems 10 and 11, (G, λ^2) if it has one type of consistency, it has both.
□

The operation of doubling is important in that it allows to transform a labeling into a *symmetric* one, and thus to extend the existing consistency to include also the other type. Another important aspect of the double labeling is that it can be constructed distributively; i.e., starting from the local labeling λ_x and the given form of coding and (when appropriate) decoding, each node x can compute the labeling λ^2_x with one round of communication.

The proof of Theorem 16 show also how to employ the (resp. backward) coding and decoding of the original system to construct the analogous ones of the new system. However, it does not give any indication on the nature of the coding and decoding of the opposite type. As we will see later, such a nature is useful also for the other transformation to be discussed in this section.

The purpose of the next Lemmas is to show how to construct in the new systems the coding and decoding of the opposite type. Given a string $\alpha = (a_0, a_1, \ldots, a_k) \in \Sigma^+$, let $\alpha^R = (a_k, a_{k-1}, \ldots, a_0)$ denote the reverse string.

Lemma 4 *Let* c *be a* $\mathcal{WSD}$ *in* (G, λ)*; then* $c_b(\alpha \oplus \beta) = c(\beta^R)$ *is a* $\mathcal{WSD}^-$ *in* $(G, \lambda)^2$*. Furthermore, if* c *has a decoding* d*, then* $d_b(c_b(\alpha \oplus \beta), (a,b)) = d(b, c(\beta^R))$ *is a backward decoding of* c_b*.*

Proof Let $x, y, z \in V$, $\pi_1 \in P[x,z]$, $\pi_2 \in P[y,z]$. Let $\Lambda^2_x(\pi_1) = ((a_0, a'_0), (a_1, a'_1), \ldots, (a_h, a'_h)$, and $\Lambda^2_y(\pi_2) = ((b_0, b'_0), (b_1, b'_1), \ldots, (b_k, b'_k)$
Let $x = y$. By definition of c_b we have that $c_b(\Lambda^2_x(\pi_1)) = c(a'_h \ldots a'_0) = c(\Lambda_z(\overline{\pi_1})$. Since $x = y$, by definition of consistency of c we have that $c(\Lambda_z({\pi_1}^R)) = c(\Lambda_z({\pi_2}^R) = c(b'_k \ldots b'_0)$. Since $c(b'_k \ldots b'_0) = c_b(\Lambda^2_y(\pi_2))$, it follows that $c_b(\Lambda^2_x(\pi_1)) = c_b(\Lambda^2_y(\pi_2))$.
Let $x \neq y$. By definition of c_b we have that $c_b(\Lambda^2_x(\pi_1)) = c(a'_h \ldots a'_0)$. Since $x \neq y$, by definition of consistency of c we have that $c(\Lambda_z({\pi_1}^R) \neq c(\Lambda_z({\pi_2}^R)$. But $\Lambda_z({\pi_1}^R) = (a'_h \ldots a'_0)$, and $\Lambda_z({\pi_2}^R) = c(b'_k \ldots b'_0)$. Since $c(b'_k \ldots b'_0) = c_b(\Lambda^2_y(\pi_2))$, it follows that $c_b(\Lambda^2_x(\pi_1)) \neq c_b(\Lambda_y(\pi_2))$.
We now show that the corresponding backward consistent decoding function is the following: $\forall \pi \in P[x,y]$, $\forall (y,z) \in E(y)$, let $\Lambda^2_x(\pi) = (\alpha \oplus \beta))$ and $\lambda^2_y((y,z)) = (a,b)$, then: $d_b(c_b(\alpha \oplus \beta), (a,b)) = d(b, c(\beta^R)$

By definition of consistent decoding function $d(b, c(\beta^R) = c(b \cdot \beta^R)$. By definition of c_b we have that $c(b \cdot \beta^R) = c_b((\alpha \oplus \beta) \cdot (a,b))$ Thus, it follows that d_b is consistent, i.e,: $d_b(c_b(\alpha \oplus \beta), (a,b)) = c_b((\alpha \oplus \beta) \cdot (a,b))$.
□

Similarily,

Lemma 5 *Let* c *be a* $\mathcal{WSD}^-$ *in* (G, λ)*. Then* $c_f(\alpha \oplus \beta) = c(\beta^R)$ *is a* $\mathcal{WSD}$ *in* $(G, \lambda)^2$*. Furthermore, if* c *has a backward decoding* d*, then* $d_f((a,b), c_f(\alpha \oplus \beta)) = d(c(\beta^R), b)$ *is a decoding of* c_f*.*

The second transformation is "reversal". Given a labeling λ) of a graph G, the *reverse labeling* $\tilde{\lambda}$ of G is obtained in the following way: $\forall (x,y) \in E$, $\tilde{\lambda}_x((x,y)) = \lambda_y((y,x))$.

The reversal operation is related to doubling as follows, where c_b, c_f, d_b and d_f are as defined in Lemmas 4 and 5:

Lemma 6 *Let c be $\mathcal{WSD}$ in (G,λ) then c_b is $\mathcal{WSD}^-$ in $(G,\tilde{\lambda})$. Furthermore, if c has a decoding d, then (c_b, d_b) is $\mathcal{SD}^-$ in $(G,\tilde{\lambda})$.*

Lemma 7 *Let c be $\mathcal{WSD}^-$ in (G,λ) then c_f is $\mathcal{WSD}$ in $(G,\tilde{\lambda})$. Furthermore, if c has a backward decoding d, then (c_f, d_f) is $\mathcal{SD}$ in $(G,\tilde{\lambda})$.*

Proof It directly follows from Theorem 16 that the following coding function is consistent: $\forall \pi \in P[x_0]$, $\pi = [(x_0,x_1), \ldots, (x_{m-1},x_m)]$: $c_b(\tilde{\Lambda}_x(\pi)) = c(\tilde{\lambda}_{x_{m-1}}((x_{m-1},x_m)), \ldots, \tilde{\lambda}_{x_0}((x_0,x_1)))$. The corresponding backward decoding function is the following: $\forall \pi \in P[x_0]$, $\pi = ((x_0,x_1), \ldots, (x_{m-1}, x_m))$, $\forall (x_m,y) \in E(x_m)$: $d_b(c_b(\tilde{\Lambda}_{x_0}(\pi)), \tilde{\lambda}_{x_m}((x_m,y))) = d(\tilde{\lambda}_{x_m}((x_m,y)), c(\tilde{\lambda}_{x_{m-1}}((x_{m-1},x_m)), \ldots, \tilde{\lambda}_{x_0}((x_0,x_1)))$. $\square$

The above Lemmas have a very important consequence:

Theorem 17 *(G,λ) has $(\mathcal{W})\mathcal{SD}^-$ if and only if $(G,\tilde{\lambda})$ has $(\mathcal{W})\mathcal{SD}$*

This result implies that the structure of the "forward" consistency landscape and that of the backward consistenct are specular. This fact will be instrumental to simplify the proof of some of the results to be established in the following of this section.

5.2 Core

We now focus on the structure of the "core" of the consistency landscape; i.e the set $\mathcal{W} \cup \mathcal{W}^-$.

The first result we establish is that backward sense of direction is a stronger form of consistency than weak sense of direction.

From Lemmas 2 and 7, we have:

Theorem 18 $\mathcal{D}^- \subset \mathcal{W}^-$
There are systems with backward consistency where no coding function is backward decodable.

Note that this is the specular of Lemma 2.

The next question we consider is whether the simultaneous presence of both consistencies suffices for the existence of either form of decoding. The answer is negative, as there exist systems with both $\mathcal{WSD}$ and $\mathcal{W}^-$ where no (backward) coding function is (backward) decodable. Let $\mathbf{G}_w$ be the labeled graph shown in Figure 8.

Lemma 8 [5]
$\mathbf{G}_w \in \mathcal{W} - \mathcal{D}$

Theorem 19 $(\mathcal{W} \cap \mathcal{W}^-) - (\mathcal{D} \cup \mathcal{D}^-) \neq \emptyset$
Existence of both types of consistency is not sufficient to decodability of either type.

Proof The graph G_w has weak sense of direction and any coding function is not decodable. Such a labeled graph has edge symmetry since the labeling is a coloring. Thus, by Theorem 10 we have that it has also backward sense of direction. However, it is easy to verify that no backward consistent coding function can be backward decodable. $\square$

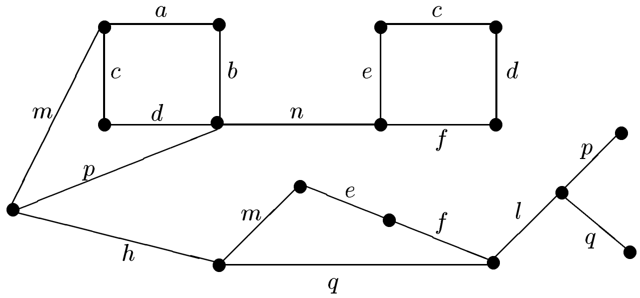

Figure 8: $\mathbf{G}_w$

Theorem 20 $(\mathcal{D} \cap \mathcal{W}^-) - \mathcal{D}^- \neq \emptyset$
Simultaneous presence of sense of direction and backward weak sense of direction is neither necessary nor sufficient for the existence of a backward decoding function.

From Theorems 17 and 20, we have.

Theorem 21 $(\mathcal{D}^- \cap \mathcal{W}) - \mathcal{D} \neq \emptyset$

5.3 Outer Structure

In this section we focus on the outer structure of the consistency landscape: $(\mathcal{L} \cup \mathcal{L}^-) - (\mathcal{W} \cap \mathcal{W}^-)$.

We first introduce two useful properties.

Given two labeled graphs (G_1,λ_1) and (G_2,λ_2), the *melding* of (G_1,λ_1) and (G_2,λ_2) by $x_1 \in V_1, x_2 \in V_2$, denoted by $\mathbf{G_1}[x_1,x_2]\mathbf{G_2}$, is the union of the two graphs restricted by imposing $x_1 = x_2$.

Lemma 9 *Let $\mathbf{G_1}$ and $\mathbf{G_2}$ be vertex- and label-disjoint labeled graphs with $\mathcal{WSD}$. Then $\forall x_1 \in V_1, x_2 \in V_2$, also $\mathbf{G} = \mathbf{G_1}[x_1,x_2]\mathbf{G_2}$ has $\mathcal{WSD}$. Furthermore, if $\mathbf{G_1}$ and $\mathbf{G_2}$ have $\mathcal{SD}$ also (G,λ) has $\mathcal{SD}$.*

Theorem 22 $(\mathcal{W} - \mathcal{D}) - \mathcal{L}^- \neq \emptyset$

Proof Consider the labeled graph of Figure 9. Notice that such a graph is the melding in x of the graph of Figure 8 and a line with two edges (x,y) and (y,z).
Since the labeled line has trivially $\mathcal{WSD}$ and since, by Lemma 8, the the graph of Figure 8 has $\mathcal{WSD}$, it follows from lemma 9 that the graph of Figure 9 has weak sense of direction as well. On the other hand, by Lemma 8, the graph of Figure 8 does not have $\mathcal{SD}$, as a consequence, also the graph of Figure 9 does not have $\mathcal{SD}$. Moreover this labeled graph does not have local orientation backward, in fact $\lambda_x(x,y) = \lambda_z(z,y)$.

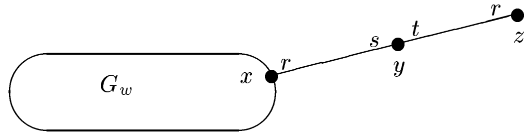

Figure 9: $(\mathcal{W} - \mathcal{D}) - \mathcal{L}^- \neq \emptyset$

$\square$

Specularly, by Theorems 17 and 22, we have that:

Theorem 23 $(\mathcal{W}^- - \mathcal{D}^-) - \mathcal{L} \neq \emptyset$

Theorem 24 $((\mathcal{W} - \mathcal{D}) \cap \mathcal{L}^-) - \mathcal{W}^- \neq \emptyset$

Proof Consider the labeled graph (G, λ) of Figure 10. Notice that it the melding in x of G_w (Figure 8 and a labeled graph (G', λ'). It is easy to verify that (G', λ') has $\mathcal{WSD}$ which implies that (G, λ) has $\mathcal{WSD}$. However, since (G_1, λ_1) does not have $\mathcal{SD}$, it follows that also (G, λ) does not have $\mathcal{SD}$. Clearly there is no local orientation backward. By contradiction, let us assume that there is $\mathcal{WSD}^-$ in (G, λ). By definition of backward consistency at node x we would have that $\mathbf{c}(1{\cdot}1) = \mathbf{c}(2{\cdot}2)$; however by definition of backward consistency at node y it must be $\mathbf{c}(1 \cdot 1) \neq \mathbf{c}(2 \cdot 2)$.

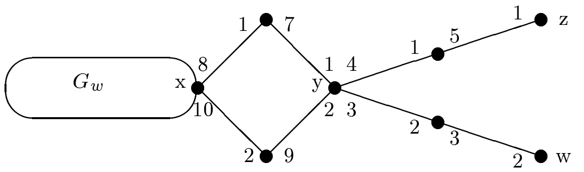

Figure 10: $((\mathcal{W} - \mathcal{D}) \cap \mathcal{L}^-) - \mathcal{W}^- \neq \emptyset$

□

Specularly, we have

Theorem 25 $((\mathcal{W}^- - \mathcal{D}^-) \cap \mathcal{L}) - \mathcal{W} \neq \emptyset$

6 Computational Equivalence

From a computational viewpoint, *anonymous* systems are the least powerful distributed systems, and hence the ideal setting to study the capabilities of specific properties (e.g., [2, 12, 18, 23, 28, 29, 30, 32, 33, 39, 40]).

The computational power of sense of direction in anonymous systems has been investigated for specific labeled graphs (e.g., meshes and tori [3], hypercubes [23, 24], Cayley graphs [22], etc.). The general characterization has been given in [18]; as an indication of its capabilities: many unsolvable problems in anonymous networks (e.g., computing the XOR in a regular network without knowledge of the network size) can be solved if the system has sense of direction (and without breaking anonymity). To exist, and thus to be exploited, sense of direction requires the existence of constraints much stronger than local orientation.

What happens, from a computational point of view, if the system does not even guarantee local orientation? What can be computed if nodes cannot even distingush among their links? There are several studies on computability in absence of local orientation, investigating these questions for specific topologies [9], for specific problems [38], or classes of functions [7, 31, 38]. In according to expectation, these results indicate that absence of local orientation drammatically reduce the computational power in almost all graphs; (a noticeable exception is the ring [9]).

In this section we study the computational capability of the newly introduced backward consistency, which can exist even in presence of complete and total blindness. We prove the unexpected result that *backward consistency has the same computational capabilities of sense of direction.* In other words, not only it overcomes the handicap of not having local orientation, but it actually empowers even a totally blind system with the additional capabilities of sense of direction.

6.1 Computability

A basic concept when computing on anonymous networks is the one of *view*, introduced in [40]. The *view* $T_{(G,\lambda)}(v)$ of a node v in a labeled graph (G, λ) is an infinite, labeled, rooted tree " downward locally isomorphic" to G; i.e., such that there exists a map from the vertices of the tree to the vertices of G which maps the root of the tree to v, the children of the root to the neighbors of v and, recursively, the children of a node to the neighbors of that node. The labeling of the arcs are also downward preserved. When no ambiguity arises, we shall denote a view $T_{(G,\lambda)}(v)$ simply by $T(v)$.

Let $\mathcal{G}$ denote the set of labeled graphs. Given $\mathcal{C} \subseteq \mathcal{G}$ and a graph G, let $\mathcal{C}/G$ denote the restriction of $\mathcal{C}$ to $\mathcal{G}$; e.g., $\mathcal{L}^-/G$ is the set of labeled graphs obtained by labeling G with backward local orientations.

A problem $\mathcal{P}$ is *solvable* in $\mathcal{C}/G$ if it is solvable in all labeled graphs $(G, \lambda) \in \mathcal{C}/G$. Given a topological knowledge K, a problem $\mathcal{P}$ is K-solvable in $\mathcal{C}/G$ if it is solvable in all $(G, \lambda) \in \mathcal{C}/G$ where all the nodes are empowered with a-priori knowledge K.

We first report two results on the computational capabilities of sense of direction. The first result is that, from a computational point of view, there is no difference between weak sense of direction and sense of direction.

Theorem 26 *[18]* $\forall G, \forall \mathcal{P}$,
$\mathcal{P}$ is solvable in $\mathcal{W}/G$ iff $\mathcal{P}$ is solvable in $\mathcal{D}/G$

The second result shows that, with sense of direction, no other knowledge is necessary.

Theorem 27 *[18]* $\forall G, \forall \mathcal{P}, \forall K$,
$\mathcal{P}$ is K-solvable in $\mathcal{L}/G \Rightarrow \mathcal{P}$ is solvable in $\mathcal{W}/G$

We will now focus on backward consistency.

Theorem 28 $\forall G, \forall \mathcal{P}$,
$\mathcal{P}$ is solvable in $\mathcal{D}/G$ iff $\mathcal{P}$ is solvable in $\mathcal{D}^-/G$

Proof To prove this theorem we will need several steps. Given two labeled graphs $(G = (V, E), \lambda)$ and $(G' = (V', E'), \lambda')$, a *labeled graph isomorphism* is a bijection $\chi : V \to V'$ which preserves edges and edge labels; that is, $\langle u, v\rangle \in E \Leftrightarrow \langle \chi(u), \chi(v)\rangle \in E'$ and $\lambda(\langle u, v\rangle) = \lambda'(\chi(u), \chi(v)\rangle)$.
The *complete topological knowledge* (denoted by TK) is knowledge of an isomorphic image $\phi(G, \lambda)$ of (G, λ) and of the isomorphism ϕ. The importance of TK follows from the following fact:

Lemma 10 *[18]* $\forall G, \forall \mathcal{P}$,
$\mathcal{P}$ is TK-solvable in $\mathcal{L}/G$ iff $\mathcal{P}$ is solvable in $\mathcal{D}/G$

In other words, TK represents the maximum information obtainable with $\mathcal{SD}$. Hence, to prove the main theorem, we need only to show that every node can construct TK with $\mathcal{WSD}^-$. To do this, first observe that:

Lemma 11 *Let ϕ be an isomorphism of (G, λ). If $(G, \lambda) \in \mathcal{L}$, then given $\phi(G, \lambda)$ and $\phi(x)$, $x \in V$ can reconstruct ϕ.*

Hence, if λ is a local orientation, to contruct the isomorphism it is sufficient for a node to know the image of the labeled graph and its own image.

Lemma 12 *[18]*
Let **c** *be a consistent coding function of* (G, λ). *Then* $\forall u \in V$, $T_{(G,\lambda}/c = (G, \lambda)$

That is, consistency would allow each node to construct an isomorphic image of (G, λ) from each view. Since in its view a node knows its location (it is the root), it knows its location in the image. Hence, by Lemma 11 (since consistency implies $\mathcal{LO}$) consistency would allow a node to construct the entire isomorphism as well. Therefore, to complete the proof, we have to show how to empower each node with local orientation and a cosistent coding.
If there is backward consistency in (G, λ) a local orientation with forward consistency can be easily constructed by employing the result of Lemma 7: if (G, λ) has backward consistency, then $(G, \tilde{\lambda})$ has local orientation and consistency. Since $\tilde{\lambda}$ is distributely constructable (i.e., each node x can construct $\tilde{\lambda}_x$) by a simple round of communication), also $T_{(G,\tilde{\lambda})}(u)$ is constructable at every node.
Thus, we must only show how to construct at each node a consistent coding function **c** for $(G, \tilde{\lambda})$, given **b**, λ_x, and $\tilde{\lambda}_x$. This is done by observing that, by Theorem 16, if (G, λ) has backward consistent coding **b**, then we can construct a consistent coding **c** of (G, λ^2), and by Lemma 7, **c** is a consistent coding for $(G, \tilde{\lambda})$.
Summarizing, if (G, λ) has backward consistency **b**, each node v can construct the view $T_{(G,\tilde{\lambda})}(v)$ and a consistent coding of $(G, \tilde{\lambda})$ which will empower it to construct both an isomorphisc image of (G, λ) and of the isomorphism. This knowledge is equivalent to complete topological knowledge; since the class of problems solvable with TK is exactly the same solvable with $\mathcal{SD}$ alone, it follows that $\mathcal{SD}^-$ is computationally equivalent to $\mathcal{SD}$.

□

6.2 Complexity

The result of Theorem 28 opens the problem of how to effectively use the computational power of backward consistency. The approach suggested by the proof of Theorem 28, is to use backward consistency to "simulate" sense of direction. While theoretically valid, this approach sidesteps the problem since it does not exploit backward consistency directly. Furthermore, the technique employed in the proof is not algorithmically feasible since it requires the construction of the views $T_{(G,\tilde{\lambda})}(v)$, a task with a formidable communication complexity.

While not solving the general problem, we will provide some complexity relief by pointing out how the simulation approach can be carried out simply and efficiently, without construction of the views.

Let A be an algorithm which solves problem $\mathcal{P}$ in any system with $\mathcal{SD}$. The simulation $S(A)$ of A in a system (G, λ) in which there is $\mathcal{SD}^-$ is performed in two stages:
1) Preprocessing: Each processor x computes the set $\Sigma_x(a) = \{b : \lambda_x(x, y) = a \text{ and } \lambda_y(y, x) = b\}$ for each of its labels $a \in \Sigma\lambda_x$. Notice that $\cup\Sigma(a) = \tilde{\lambda}_x(E(x))$.
2) Simulation: whenever, in algorithm A, a node x send a message m on a edge labeled l, in the simulation $S(A)$ it will send a message (m, l) to the set of edges labeled by the (unique) label $p \in \Sigma\lambda_x$ such that $l \in \Sigma_x(p)$; whenever, in algorithm A, a node performs an action O upon reception of a message m from a link labeld l, in the simulation of A, it will perform the same action O upon reception of message (m, l) from any of the edges labeled p, where p is the (unique) label such that $l \in \Sigma_x(p)$.

It is easy to see that algorithm $S(A)$ behaves on (G, λ) in exact the same way as the algorithm A would behave on $(G, \tilde{\lambda})$ (the only difference is that $S(A)$ sends a pair $(m, label)$ every time that A sends a message m. In fact,

Theorem 29 *Algorithm $S(A)$ solves $\mathcal{P}$ in any system with $\mathcal{SD}^-$ iff A solves $\mathcal{P}$ in any system with $\mathcal{SD}$.*

Proof Let A solves P in any system with $\mathcal{SD}$. Since (G, λ) has $\mathcal{SD}^-$, from Theorem 17, we have that $(G, \tilde{\lambda})$ is a system with $\mathcal{SD}$ and, thus, A solves P on $(G, \tilde{\lambda})$. But, by construction, algorithm S behaves on (G, λ) in exact the same way as the algorithm A would behave on $(G, \tilde{\lambda})$ and the proof follows. The vice-versa is analogous. □

Moreover, the number of message transmissions of $S(A)$ it is the same as in the original algorithm A. The number of message *receptions* obviously depend on the cardinality of the sets $\Sigma_x(p)$. Given (G, λ), let $h(G) = Max_{x \in V, a \in \Sigma}\{\Sigma_x(a)\}$; clearly, $h(G) \leq d(G)$.

Theorem 30 $\mathcal{M}_T(S(A), G, \lambda) = \mathcal{M}_T(A, G, \tilde{\lambda})$
$\mathcal{M}_R(S(A), G, \lambda) \leq h(G)\mathcal{M}_R(A, G, \tilde{\lambda})$

In other words, it is possible to use simulation with some level of efficiency. However, the real task is to develop protocols and techniques which exploit backward consistency directly (not just to simulate forward consistency).

References

[1] D. Angluin. Local and global properties in networks of processors. In *Proc. of 12th A.C.M. Symposium on Theory of Computing*, 82–93, 198.

[2] H. Attiya, M. Snir, and M.K. Warmuth. Computing on an anonymous ring. *Journal of the A.C.M.*, 35(4):845–875, 1988.

[3] P.W. Beame and H.L. Bodlaender. Distributed computing on transitive networks: the torus. In *Proc. of 6th Symposium on Theoretical Aspects of Computer Science*, 294–303, 1989.

[4] G. v. Bochmann, P. Flocchini, and D. Ramazani. Distributed objects with sense of direction. In *1st Workshop on Distributed Data and Structures*, 1–15, Orlando, 1998.

[5] P. Boldi and S. Vigna. On the complexity of deciding sense of direction. In *Proc. of 2nd Colloquium on Structural Information and Communication Complexity*, 39–52, Olympia, 1995.

[6] P. Boldi and S. Vigna. On some constructions which preserve sense of direction. In *3rd International Colloquium on Structural Information and Communication Complexity*, number 6, 47–57, Siena, 1996.

[7] P. Boldi and S. Vigna. Computing vector functions on anonymous networks. In *Proc. of 4th Colloquium on Structural Information and Communication Complexity*, 201–214, Ascona, 1997.

[8] P. Boldi and S. Vigna. Minimal sense of direction and decision problems for Cayley graphs. *Information Processing Letters*, 64:299–303, 1997.

[9] K. Diks, E. Kranakis, A. Malinowski, and A. Pelc. Anonymous wireless rings. *Theoretical Computer Science*, 145(1-2):95–109, 1995.

[10] K. Diks, E. Kranakis, and A. Pelc. Broadcasting in unlabeled tori. *Parallel Processing Letters*, 1998. to appear.

[11] S. Dobrev and P. Ružička. Linear broadcasting and $n \log\log n$ election in unoriented hypercubes. In *Proc. of the 4th International Colloquium on Structural Information and Communication Complexity*, 53–68, 1997.

[12] P. Ferragina, A. Monti, and A. Roncato. Trade-off between computational power and common knowledge in anonymous rings. In *Proc. of 1st Colloquium on Structural Information and Communication Complexity*, 35–48, Ottawa, 1994.

[13] P. Flocchini. Minimal sense of direction in regular networks. *Information Processing Letters*, 61:331–338, 1997.

[14] P. Flocchini and B. Mans. Optimal election in labeled hypercubes. *Journal of Parallel and Distributed Computing*, 33(1):76–83, 1996.

[15] P. Flocchini, B. Mans, and N. Santoro. On the impact of sense of direction on message complexity. *Information Processing Letters*, 63(1):23–31, 1997.

[16] P. Flocchini, B. Mans, and N. Santoro. Sense of direction: definition, properties and classes. *Networks*, 32(3):165–180, 1998.

[17] P. Flocchini, B. Mans, and N. Santoro. Sense of direction in distributed computing. In *Proc. of 12th International Symposium on Distributed Computing (DISC 98)*, 1-15, Andros, 1998.

[18] P. Flocchini, A. Roncato, and N. Santoro. Computing on anonymous networks with sense of direction. *Theoretical Computer Science*. To appear.

[19] P. Flocchini, A. Roncato, and N. Santoro. Symmetries and sense of direction in labeled graphs. *Discrete Applied Mathematics*, 87:99–115, 1998.

[20] P. Flocchini and N. Santoro. Topological constraints for sense of direction. *International Journal on Foundations of Computer Science*, 9(2):179–198, 1998.

[21] Z. Harpaz and S. Zaks. Sense of direction in complete distributed networks. in *Proc. of 23rd Allerton Conference on Communication, Control and Computing*, 1985.

[22] E. Kranakis and D. Krizanc. Labeled versus unlabeled distributed Cayley networks. *Discrete Applied Mathematics*, 63(3):223–236, 1995.

[23] E. Kranakis and D. Krizanc. Distributed computing on anonymous hypercubes. *Journal of Algorithms*, 23:32–50, 1997.

[24] E. Kranakis and N. Santoro. Distributed computing on anonymous hypercubes with faulty components. In *Proc. of 6th Int. Workshop of Distributed Algorithms*, 253–263, 1992.

[25] M.C. Loui, T.A. Matsushita, and D.B. West. Election in complete networks with a sense of direction. *Information Processing Letters*, 22:185–187, 1986. see also Information Processing Letters, vol.28, p.327, 1988.

[26] B. Mans. Optimal distributed algorithms in unlabeled tori and chordal rings. *Journal on Parallel and Distributed Computing*, 46(1):80–90, 1997.

[27] B. Mans and N. Santoro. On the impact of sense of direction in arbitrary networks. In *Proc. of 14th International Conference on Distributed Computing Systems*, pages 258–265, Poznan, 1994.

[28] A. Monti and A. Roncato. A gap theorem for anonymous torus. *Information Processing Letters*, 57:279–285, 1996.

[29] S. Moran and M.K. Warmuth. Gap theorems for distributed computation. *SIAM Journal on Computing*, 22(2):379–394, 1993.

[30] N. Norris. Classifying anonymous networks: when can two networks compute the same vector-valued functions ? In *Proc. of 1st Colloquium on Structural Information and Communication Complexity*, pages 83–98, Ottawa, 1994.

[31] N. Norris. Computing functions on partially wireless networks. In *Proc. of 2nd Colloquium on Structural Information and Communication Complexity*, pages 53–64, Olympia, 1995.

[32] N. Norris. Universal covers of graphs: isomorphism to depth $n-1$ implies isomorphism to all depths. *Discrete Applied Mathematics*, 56:61–74, 1995.

[33] A. Roncato. Gap theorems for anonymous rings. In *Proc. of 2nd Colloquium on Structural Information and Communication Complexity*, pages 65–76, Olympia, 1995.

[34] N. Santoro, J. Urrutia, and S. Zaks. Sense of direction and communication complexity in distributed networks. In *Proc. of the 1st International Workshop on Distributed Algorithms*, pages 123–132, Ottawa, 1985. Carleton University Press.

[35] G. Singh. Efficient leader election using sense of direction. *Distributed Computing*, 10:159–165, 1997.

[36] G. Tel. Network orientation. *International Journal of Foundations of Computer Science*, 5(1):1–41, 1994.

[37] G. Tel. Sense of direction in processor networks. In *Proc. of Conference on Theory and Practice of Informatics*, pages 50–82, 1995.

[38] M. Yamashita and T. Kameda. Electing a leader when processor identity numbers are not distinct. In *Proc. of 3rd International Workshop on Distributed Algorithms*, 1989.

[39] M. Yamashita and T. Kameda. Computing functions on asynchronous anonymous networks. *Mathematical Systems Theory*, 29:331–356, 1996.

[40] M. Yamashita and T. Kameda. Computing on anonymous networks, part I: characterizing the solvable cases. *IEEE Transaction on Parallel and Distributed Computing*, 7(1):69–89, 1996.