About the local detection of termination of local computations in graphs*

Yves Métivier[1], Anca Muscholl[2] and Pierre-André Wacrenier[1]

[1] LaBRI, Université Bordeaux I, ENSERB,
351 cours de la Libération
33405 Talence, France
[2] Institut für Informatik, Universität Stuttgart,
Breitwiesenstr. 20-22, 70565 Stuttgart, Germany

Abstract. In this paper we give a formalization of the problem of locally detecting the global termination of distributed algorithms on graphs. We introduce the notion of quasi $k-$coverings. Using (quasi) $k-$coverings, we present methods for showing that it is not possible to detect locally the global termination of local computations in certain families of graphs. These methods also allow to show that some knowledge, e.g. the size of the graph, is necessary in order to solve certain problems in a distributed way.

Keywords: local detection of termination, local computations, $k-$covering, quasi $k-$covering.

1 Introduction

Local computations on graphs, as given by graph rewriting systems (with priorities and/or forbidden contexts) are a powerful model for local computations which can be executed in parallel. Rewriting systems provide a general tool for encoding distributed algorithms and for proving their correctness. This paper is concerned with the existence of rewriting systems having the property of local detection of the global termination.

We consider a network of processors with arbitrary topology. It is represented as a connected, undirected graph where vertices denote processors, and edges denote direct communication links. An algorithm is encoded by means of local relabellings. Labels attached to vertices and edges are modified locally, that is, on a subgraph of fixed radius k of the given graph, according to certain rules depending on the subgraph, only (such local computations are called $k-$local). The relabelling is performed until no more transformation is possible (i.e., until a "normal form" is obtained). In the terminology of distributed algorithms, we say that a distributed algorithm terminates whenever it reaches a terminal configuration, i.e. a configuration in which no steps of the algorithm can be applied

* This research was supported by the French-German project PROCOPE. Contact author: metivier@labri.u-bordeaux.fr.

anymore. We say that a local computation system allows the local detection of (global) termination if for any computation there is a vertex for which its neighbourhood of a given radius r determines whether or not a normal form has been reached (we say that global termination is $r-$locally detected).

The problem of local detection of termination is of interest due to connections to the following topics:

- composition of local computations;
- what is the minimal information about the network which is necessary for solving certain problems by distributed algorithms;
- which functions can be computed for a given family of networks.

Our aim in this paper is twofold: first, we present a formalization of the local detection problem and we introduce some new methods for obtaining impossibility results. More precisely, we extend the notion of coverings, which is known from algebraic topology [13] and has been already used in distributed computing for negative results [1,5,7,9], to quasi-coverings. Quasi-coverings capture some topologies which fail to cope with the classical coverings. We show in this paper that one cannot detect locally the global termination for uniformly labelled graphs belonging to certain families of connected graphs $\mathcal{C}$. More specifically, it suffices for our results that $\mathcal{C}$ contains some graph G and a nontrivial $k-$covering of G, see Prop. 13 (or, some graph G and quasi-coverings of G of arbitrary large size, see Thm. 20). Our second aim is to discuss some implications of our results for the borderline between positive and negative results. More precisely, we are interested in the question whether certain additional knowledge about a network, which is used in specific graph rewriting systems, is really necessary. For example, we show that election in T-prime graphs is possible (if and) only if the size of the graph is provided.

Finally, we note that our asynchronous computational model is an extension of models studied by Angluin [1] and more recently by Yamashita et.al. [18,19]. As in those papers, our results are obtained for anonymous networks. (Clearly, if a leader is available, then the termination can be easily detected locally using standard methods). Some of our results were already known for the ring topology, see [17] for a survey.

Our paper is organized as follows. Section 2 contains some basic notions. In Section 3 we define local computations on graphs and the local detection of normal forms. Section 4 contains some examples for graph rewriting systems. In Section 5 we describe the application of coverings to the problem of the local detection of a normal form. In Section 6 we generalize the results to quasi-coverings, and we apply these methods to $T-$prime graphs. We conclude by some remarks about the connections between the election problem, the size problem and the problem of local detection of a normal form.

2 Basic notions and notations

2.1 Graphs

The graph-theoretical notations used here are essentially standard [2]. A graph G is defined as a finite set $V(G)$ of vertices together with a set $E(G) \subseteq \binom{V}{2}$ of edges. We only consider finite, undirected graphs without multiple edges or self-loops. Let $e = \{v, v'\}$ be an edge: we say that e is incident with v, and that v is a neighbour of v'. The set of neighbours of a vertex v, together with v itself, is denoted $N_G(v)$. A vertex of degree one is called a leaf. A path P from v_1 to v_i in G is a sequence $P = (v_1, e_1, v_2, e_2, \dots, e_{i-1}, v_i)$ of vertices and edges such that for all $1 \leq j < i$, e_j is an edge incident with the vertices v_j and v_{j+1}. The length of P is $i - 1$. If $v_1 = v_i$ then P is called a cycle. If each vertex appears only once in a path P, then P is called simple. A *tree* is a connected graph containing no simple cycle. Any two vertices in a tree are connected by precisely one simple path. The distance between two vertices u, v is denoted $d(u, v)$.

G' is a subgraph of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. A subgraph G' of a graph G is called a *spanning subgraph* if $V(G') = V(G)$. An acyclic spanning subgraph of G is called a *spanning forest.* A spanning forest which is connected is called a *spanning tree.*

The subgraph of G induced by a subset V' of $V(G)$ is the subgraph of G having V' as vertex set and containing all edges of G between vertices of V'.

Let v be a vertex, and k a positive integer. We denote by $B_G(v, k)$ the ball of radius k with center v; it is the subgraph of G induced by the vertex set

$$V' = \{v' \in V(G) \mid d(v, v') \leq k\},$$

with a distinguished vertex, namely v, called *center*.

A homomorphism between two graphs G and H is a mapping $\gamma\colon V(G) \to V(H)$ such that if $\{u, v\}$ is an edge of G then $\{\gamma(u), \gamma(v)\}$ is an edge of H. Since we deal only with graphs without self-loops, this implies that $\gamma(u) \neq \gamma(v)$, if $\{u, v\}$ is an edge of G. Note also that $\gamma(N_G(u)) \subseteq N_H(\gamma(u))$. We say that γ is an isomorphism if γ is bijective and the inverse γ^{-1} is also a homomorphism. The notation $G \simeq G'$ means that G and G' are isomorphic. A class of graphs will be any class of graphs in the set-theoretical sense containing all graphs isomorphic to some of its members.

Coverings are known from algebraic topology and are also related to the notion of uniform emulation [3,4]. We say that a graph G is a *covering* of a graph H if there exists a surjective homomorphism γ from G onto H such that for every vertex v of $V(G)$ the restriction of γ to $N_G(v)$ is a bijection onto $N_H(\gamma(v))$. In this paper we use special coverings, namely *$k-$coverings.*

Definition 1. Let G, G' be two labelled graphs and let $\gamma\colon V(G) \to V(G')$ be a graph homomorphism. Let $k > 0$ be a positive integer.

Then G is a *$k-$covering* of G' via γ if for every vertex $v \in V(G)$, the restriction of γ on $B_G(v, k)$ is an isomorphism between $B_G(v, k)$ and $B_{G'}(\gamma(v), k)$.

The k-covering is called strict if G and G' are not isomorphic.

A class of graphs is said to be closed under $k-$coverings (resp. connected $k-$coverings), if it contains all $k-$coverings (resp. connected $k-$coverings) of its elements.

3 Local computations in graphs

3.1 Labelled Graphs

Throughout the paper we will consider only connected graphs where vertices and edges are labelled with labels from a possibly infinite alphabet L. A graph labelled over L will be denoted by (G, λ), where G is a graph and $\lambda \colon V(G) \cup E(G) \to L$ is the function labelling vertices and edges. The graph G is called the underlying graph, and the mapping λ is a labelling of G. The class of labelled graphs over some fixed alphabet L will be denoted by $\mathcal{G}$.

Let (G, λ) and (G', λ') be two labelled graphs. Then (G, λ) is a subgraph of (G', λ'), denoted by $(G, \lambda) \subseteq (G', \lambda')$, if G is a subgraph of G' and λ is the restriction of the labelling λ' to $V(G) \cup E(G)$.

A mapping $\varphi \colon V(G) \cup E(G) \to V(G') \cup E(G')$ is a homomorphism from (G, λ) to (G', λ') if φ is an graph homomorphism from G to G' which preserves the labelling, i.e. such that $\lambda'(\varphi(x)) = \lambda(x)$ holds for every $x \in V(G) \cup E(G)$. The mapping φ is an isomorphism, if it is bijective.

An *occurrence* of (G, λ) in (G', λ') is an isomorphism φ between (G, λ) and a subgraph (H, η) of (G', λ').

3.2 Local computations

Local computations as considered here can be described in the following general framework. Let $\mathcal{G}$ be the class of L-labelled graphs and let $R \subseteq \mathcal{G} \times \mathcal{G}$ be a binary relation on $\mathcal{G}$. Then R will denote a graph rewriting relation. We assume that R is closed by isomorphism, i.e. whenever GRG', if $G_1 \simeq G$ then $G_1 R G'_1$ for some graph $G'_1 \simeq G'$.

Definition 2. Let $R \subseteq \mathcal{G} \times \mathcal{G}$ be a graph rewriting relation and let $k > 0$ be an integer.

1. R is a *relabelling relation* if whenever two labelled graphs are in relation then the underlying graphs are equal (we say equal, not only isomorphic), i.e.:

$$(G, \lambda) R (H, \lambda') \quad \Longrightarrow \quad G = H.$$

2. R is called $k-$*local* if only labels of a ball of radius k may be changed by R, i.e. $(G, \lambda) R (G, \lambda')$ implies that there exists a vertex $v \in V(G)$ such that

$$\lambda(x) = \lambda'(x) \text{ for every } x \notin V(B_G(v, k)) \cup E(B_G(v, k)).$$

The relation R is called *local*, if it is $k-$local for some $k > 0$.

3. An $R-$*normal form* of $G \in \mathcal{G}$ is a graph G' such that GR^*G', but $G'RG''$ holds for no $G'' \in \mathcal{G}$. We say that R is *noetherian* if there is no infinite relabelling chain $G_1RG_2R\cdots$.

The next definition states that a local relabelling relation R is $k-$*locally generated* if its restriction on centered balls of radius k determines its computation on any graph.

Definition 3. Let R be a relabelling relation and $k > 0$ be an integer. Then R is called $k-$*locally generated* if the following holds: For any labelled graphs (G, λ), (G, λ'), (H, η), (H, η') and any vertices $v \in V(G)$, $w \in V(H)$ such that the balls $B_G(v, k)$ and $B_H(w, k)$ are isomorphic via $\varphi\colon V(B_G(v, k)) \longrightarrow V(B_H(w, k))$ and $\varphi(v) = w$, the following three conditions

1. $\lambda(x) = \eta(\varphi(x))$ and $\lambda'(x) = \eta'(\varphi(x))$ for all $x \in V(B_G(v, k)) \cup E(B_G(v, k))$
2. $\lambda(x) = \lambda'(x)$, for all $x \notin V(B_G(v, k)) \cup E(B_G(v, k))$
3. $\eta(x) = \eta'(x)$, for all $x \notin V(B_H(w, k)) \cup E(B_H(w, k))$

imply that $(G, \lambda)R(G, \lambda')$ holds if and only if $(H, \eta)R(H, \eta')$.

R is called *locally generated* if it is $k-$locally generated for some $k > 0$.

Note that if R is $k-$locally generated, then its restriction to graphs of diameter $2k$ determines it uniquely for all graphs. Let us also note that a $k-$locally generated relabelling relation allows parallel rewritings, since non-overlapping k-balls may be relabelled independently.

3.3 The local detection of a normal form

We study local computations such that normal forms are characterized by a set of local configurations.

Definition 4. Let R be a $k-$locally generated relabelling relation. Let $\mathcal{I}$ be a subset of $\mathcal{G}$ called the class of initial graphs and let T be a subset of connected elements of $\mathcal{G}$. We say that T characterizes normal forms obtained from $\mathcal{I}$ if for any $G \in \mathcal{I}$ with GR^*H we have that H is a normal form if and only if H contains a subgraph isomorphic to some $K \in T$. (In this case we also say that H is $K-$characterized).

Let r be a positive integer. If every element of T has radius bounded by r then we say that normal forms are r-*locally* (or *locally*) characterized.

4 Some examples

In this section we give some examples illustrating the notions defined previously. Our examples deal with typical problems as computing a spanning tree in a single-source graph or electing a vertex. We use the notations of Priority Graph Rewriting Systems [10].

4.1 Spanning tree in a single-source graph

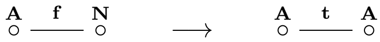
We give in the following two graph rewriting systems which compute a spanning tree of a given graph. We assume that initially the graph has exactly one distinguished vertex labelled **A**. Our graph rewriting systems encode classical distributed algorithms (see e.g. [17]). We have chosen these algorithms for sake of simplicity, although they are not optimal. Proofs and properties of these systems may be found in [8].

Distributed computation of a spanning tree

The idea of the first algorithm is very simple: any vertex belonging to the tree may add a neighbour which is not yet in the tree. Vertices will be labelled by $\{\mathbf{N}, \mathbf{A}\}$, whereas edges will be labelled by $\{\mathbf{f}, \mathbf{t}\}$. Initially exactly one vertex is labelled **A**, all other ones are labelled **N** and every edge is labelled **f**.

The corresponding rewriting system has one rule:

$R:$



This rewriting system is noetherian. For connected graphs initially labelled as described above it yields normal forms where the set of **t**-labelled edges and the set of incident vertices yield a spanning tree. However, it is obvious that the termination of this system cannot be locally detected.

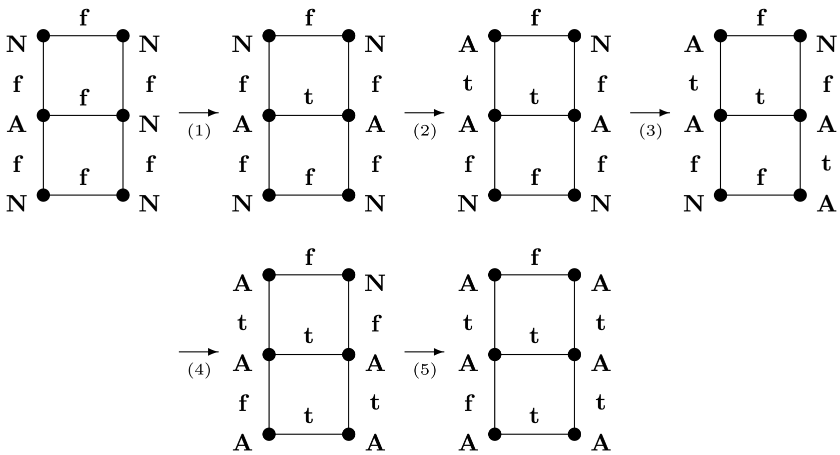


Fig. 1. Computation of a spanning tree using R.

Distributed computation of a spanning tree with acknowledgment

A distributed computation of a spanning tree where termination can be detected locally can be obtained by adding an acknowledgment mechanism. The idea is the following: using the previous system we first remember the direction in which neighbours are added (expressed as parent-child relation). If every neighbour of a vertex v has been visited, then v sends an acknowledgment to its parent node (see rule R_3 below). Moreover, if every child of v has sent its acknowledgment, then v sends itself an acknowledgment to its parent node (see rule R_6 below). The solution given here uses the relation "*parent of*", which is encoded by the labels $\{0, 1, 2\}$ with the meaning "*i is the parent of* $(i + 1)$", where addition is performed modulo 3. This order relation between two vertices labelled **M** is needed in order to ensure that the last rewriting step is performed on the root. Without this relation the rewriting system computes a spanning tree, but we cannot locally detect that the algorithm terminates.

The set of vertex-labels is $\{\mathbf{N}, (\mathbf{A}, 0), (\mathbf{A}, 1), (\mathbf{A}, 2), (\mathbf{M}, 0), (\mathbf{M}, 1), (\mathbf{M}, 2), \mathbf{R}, \mathbf{F}\}$, and edges are labelled as before by $\{\mathbf{f}, \mathbf{t}\}$. Initially the root is labelled $(\mathbf{A}, 0)$, any other vertex is labelled **N**, and every edge is labelled **f**.

We present the rewriting system in two groups of rules. The first group describes what happens when a node is visited: the node is labelled **A** and the edge traversed is labelled **t**. As soon as a visited node has a child, it is labelled **M**.

R_1 : $(\mathbf{A},k) \overset{\mathbf{f}}{\text{---}} \mathbf{N} \longrightarrow (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} (\mathbf{A},k{+}1)$

R_2 : $(\mathbf{M},k) \overset{\mathbf{f}}{\text{---}} \mathbf{N} \longrightarrow (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} (\mathbf{A},k{+}1)$

The second group encodes the acknowledgment; when a vertex is ready to send its acknowledgment it is labelled **R**, and when it has sent its acknowledgment it is labelled **F**.

R_3 : $(\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} (\mathbf{A},k{+}1) \longrightarrow (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} \mathbf{R}$

R_4 : $\mathbf{R} \overset{\mathbf{t}}{\text{---}} (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} \mathbf{R} \longrightarrow \mathbf{F} \overset{\mathbf{t}}{\text{---}} (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} \mathbf{R}$

R_5 : $\mathbf{R} \overset{\mathbf{t}}{\text{---}} (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} (\mathbf{M},k{+}1) \longrightarrow \mathbf{F} \overset{\mathbf{t}}{\text{---}} (\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} (\mathbf{M},k{+}1)$

R_6 : $(\mathbf{M},k) \overset{\mathbf{t}}{\text{---}} \mathbf{R} \longrightarrow \mathbf{R} \overset{\mathbf{t}}{\text{---}} \mathbf{F}$

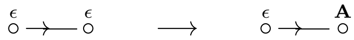
The priority of rules is defined by $R_i \succ R_j \succ R_6$, for all i, j, $1 \leq i \leq 2$, $3 \leq j \leq 5$. The priority mechanism means that if two rule occurrences overlap, then the rule with higher priority has to be applied.

The rewriting system above computes a spanning tree in a distributed way and it has the property that termination can be detected locally: a normal form is obtained as soon as a vertex labelled **R** has only **F**−labelled adjacent vertices.

4.2 Election in a prime ring

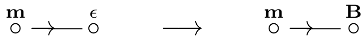
The algorithm below has been given by A. Mazurkiewicz [14] for oriented rings with prime size n. The labels are words over the alphabet $\{\mathbf{A}, \mathbf{B}\}$ of length at most n. Initially all labels are equal to the empty word ϵ. The algorithm can be described by the three rules given in the following. The first rule is

R_1 :



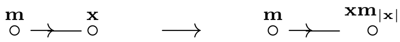
In the next rule we assume that the word $\mathbf{m}$ is not empty.

R_2 :



For the last rule we assume that $0 < |\mathbf{x}| < n$ and $|\mathbf{x}| \leq |\mathbf{m}|$ holds. We denote by $\mathbf{m}_{|\mathbf{x}|}$ the $|\mathbf{x}|$th letter of $\mathbf{m}$.

R_3 :



The relabelling system above is noetherian and in every normal form the labels are conjugated words, i.e. words obtained by cyclic permutations. If n is prime, then this property implies that all labels in a normal form are different. Hence, the vertex elected could be the one with the lexicographical smallest label. Clearly, a vertex labelled by a word of length n knows whether it is the elected vertex. However, no vertex can detect locally that the algorithm terminated.

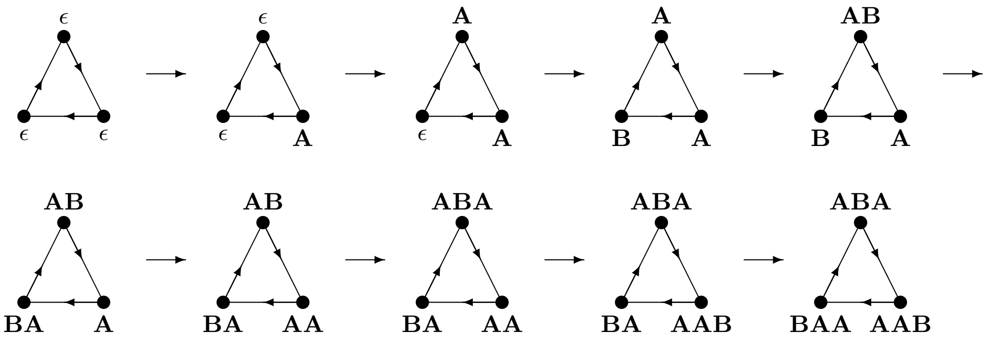


Fig. 2. Election in a prime oriented ring

The above result has been generalized to the family of $T-$prime graphs [15], which is defined as follows. Let $G = (V, E)$ be a connected graph of size n, and let r be an integer dividing n. We say that G is $r-$factorizable if G admits a spanning forest with trees all of size r. The graph G is said to be $T-$prime if it is not $r-$factorizable for any integer $n > r > 1$. The main idea of the algorithm is to construct a partition of the graph into connected subgraphs. Each subgraph is defined by a spanning tree and has a leader (root) with weight equal to the size of the subgraph; all other vertices of the subgraph have zero weight. Initially we consider a partition such that every subgraph consists of a single vertex. We assume that at least one processor starts the computation. Then there is at least one duel between two adjacent vertices from which we obtain a new partition, with at least one element containing two vertices (in this case we say that the algorithm has started). A leader L with weight w looks for an adjacent subgraph having a leader L' with weight w' such that $w > w'$. In this case, their spanning trees (i.e., the two corresponding subgraphs) are combined and L remains the leader with the weight $w+w'$, whereas the weight of L' becomes zero. The algorithm terminates when only one tree is left. Clearly, the elected vertex knows that it has been elected if it knows the size of the graph. From results obtained in the next sections we will deduce that this knowledge is necessary.

5 Local computations, k-coverings and local detection of normal forms

We study in this section the connections between the three notions introduced previously: local detection of normal forms, $k-$coverings and local computations.

5.1 Local computations and $k-$coverings

First we recall some basic properties of local computations and k-coverings [9].

Lemma 5. *Let G be a $k-$covering of G' via γ and let $v_1, v_2 \in V(G)$ be such that $v_1 \neq v_2$. If $\gamma(v_1) = \gamma(v_2)$ then $B_G(v_1, k) \cap B_G(v_2, k) = \emptyset$.*

Proof. Let $v \in B_G(v_1, k) \cap B_G(v_2, k)$. From the hypothesis, the vertices v_1 and v_2 belong to $B_G(v, k)$ and $\gamma(v_1) = \gamma(v_2)$. We get a contradiction to the fact that the restriction of γ to $B_G(v, k)$ is a one-to-one correspondence between $B_G(v, k)$ and $B_{G'}(\gamma(v), k)$.

This lemma yields immediately:

Corollary 6. *Let γ be a graph homomorphism from G to G'.*

1. *Suppose γ is a k-covering and let v_1, v_2 be two different vertices of $V(G)$ verifying $\gamma(v_1) = \gamma(v_2)$. Then $d(v_1, v_2) > 2k$.*
2. *The homomorphism γ is a k-covering if and only if for every $v \in V(G')$ the inverse image $\gamma^{-1}(B_{G'}(v, k))$ is a disjoint union of graphs each isomorphic to $B_{G'}(v, k)$.*

Lemma 7. *Let G' be a connected graph and let G be a $k-$covering of G' via γ. Then there exists an integer q such that*

$$\forall v \in V(G'), \quad card(\gamma^{-1}(v)) = q.$$

Proof. Let v, v' be vertices of G', $v' \in N_{G'}(v)$. Then $v' \in V(B_{G'}(v, k))$ and from Corollary 6 it follows that the inverse image of $B_{G'}(v, k)$ is a union of pairwise disjoint balls in G, each of these balls being isomorphic to $B_{G'}(v, k)$. Thus $\mathrm{card}(\gamma^{-1}(v)) = \mathrm{card}(\gamma^{-1}(v'))$. Since G' is connected we have by transitivity $\mathrm{card}(\gamma^{-1}(u)) = \mathrm{card}(\gamma^{-1}(u'))$ for all $u, u' \in V(G')$, which yields the desired result.

Definition 8. Let G be a $k-$covering of G' via γ, and let q be such that $\mathrm{card}(\gamma^{-1}(v)) = q$ holds for all $v \in V(G')$.

Then the integer q is called the number of *sheets* of the covering G'. In this case we speak of a q-sheeted covering.

Remark 9.
1. If $q = 1$ then G and G' are isomorphic. Moreover, if G is a $k-$covering of G' via γ, then G is a $k'-$covering of G' via γ for every k' with $0 < k' \leq k$.
2. Lemma 7 holds already for classical coverings, hence for k-coverings as well. We have included the proof for the sake of completeness.
3. It is easy to show that if G is a q-sheeted covering of G' via γ, then for every acyclic subgraph H of G' the inverse image $\gamma^{-1}(H)$ is a disjoint union of q graphs isomorphic to H.

The relation between classical coverings and k-coverings is established in the next lemma.

Lemma 10. *Let $k > 0$. Let G, G' be two graphs and $\gamma\colon V(G) \to V(G')$ a graph homomorphism.*

Then G is a k-covering of G' if and only if G is a covering of G' satisfying the following property: for every cycle $C = (v_1, e_1, v_2, e_2, \ldots, e_i, v_{i+1} = v_1)$ of length $i \leq 2k+1$ the inverse image $\gamma^{-1}(C)$ is a disjoint union of graphs isomorphic to C.

Proof. If G is a k-covering of G' then every cycle C of length at most $2k+1$ is contained in some centered ball of radius k, hence the result. Conversely, suppose G is a covering of G' via γ verifying the above property. Assume that $v' \in V(G')$, $v \in \gamma^{-1}(v')$ and H' is a breadth-first spanning tree of $B_{G'}(v', k)$ with root v'. Hence, H' has depth at most k. The inverse image $\gamma^{-1}(H')$ is a disjoint union of graphs isomorphic to H', see Remark 9. Let $H \simeq H'$ be the connected component of $\gamma^{-1}(H')$ containing v, i.e. H is a tree rooted in v. Every non-tree edge $e' = \{x', y'\}$ with $x', y' \in B_{G'}(v', k)$, $e' \notin E(H')$ belongs to a cycle C' contained in $B_{G'}(v', k)$ where all edges up to e' belong to the spanning tree H'. Moreover, $|C'| \leq 2k+1$. Since $\gamma^{-1}(C')$ is a disjoint union of copies of C' it follows that for $x, y \in V(H)$ with $\gamma(x) = x'$, $\gamma(y) = y'$ the edge $\{x, y\} \in \gamma^{-1}(\{x', y'\})$ belongs to $E(G)$. Hence, the subgraph induced by $V(H)$ is isomorphic to $B_{G'}(v')$. By Corollary 6 G is a k-covering of G'.

The question whether a graph has nontrivial finite or infinite connected k-coverings is undecidable [6]. However, we have the following simple special case:

Lemma 11. *Let $k > 0$ be given. Suppose G' and $e' \in E(G')$ are such that $(V(G'), E(G') \setminus \{e'\})$ is connected, but e' belongs to no simple cycle of length at most $2k+1$.*

Then there exists for every $q \geq 1$ a connected, q-sheeted k-covering G of G'.

Proof. Let $G'_e = (V(G'), E(G') \setminus \{e'\})$ and define G_e as the disjoint union of q copies of G'_e. We identify w.l.o.g. $V(G_e)$ with the set $V(G') \times \{1, \ldots, q\}$. Suppose $e' = \{x, y\}$. Then we define G by letting $V(G) = V(G_e)$ and $E(G) = E(G_e) \cup \{\{(x,i),(y,i+1)\} \mid 0 \leq i < q\}$ (with addition modulo q). The natural morphism mapping (x, i) to x is clearly a covering and G is connected. By Lemma 10 G is a k-covering of G'.

The next lemma establishes the connection between k-coverings and k-locally generated relabelling relations.

Lemma 12. *Let R be a k-locally generated relabelling relation and let (G, λ_1) be a k-covering of (G', λ'_1) via γ. Moreover, let $(G', \lambda'_1)R^*(G', \lambda'_2)$. Then a labelling λ_2 of G exists such that $(G, \lambda_1)R^*(G, \lambda_2)$ and (G, λ_2) is a k-covering of (G', λ'_2).*

Proof. It suffices to show the claim for the case $(G', \lambda'_1)R(G', \lambda'_2)$. Suppose that the relabelling step changes labels in $B_{G'}(v, k)$ only, for some vertex $v \in V(G')$. We may apply this relabelling step to each of the (disjoint) labelled balls of $\gamma^{-1}(B_{G'}(v, k))$, since they are isomorphic to $B_{G'}(v, k)$. This yields the labelling λ_2 of G which satisfies the claim.

5.2 Applications to termination detection

Proposition 13. *Let $\mathcal{I} \subseteq \mathcal{G}$ be a class of connected labelled graphs and let R be a $k-$locally generated relabelling relation. Assume that $\mathcal{I}$ contains graphs G, G' and G is a non-isomorphic $k-$covering of G'. Let $r \leq k$.*

Then normal forms obtained from $\mathcal{I}$ cannot be $r-$locally characterized.

Proof. Let G be a non-isomorphic, connected k-covering of G' via $\gamma \colon V(G) \to V(G')$. Moreover, let $G'R^{n-1}H'_1RH'_2$. By Lemma 12 we obtain a new labelling of G, say H_1, such that H_1 is a k-covering of H'_1 (via γ) and GR^*H_1. Suppose that $H'_1RH'_2$ holds via vertex v'. Then we may apply this relabelling step to exactly one of the connected components of $\gamma^{-1}(B_{H'_1}(v, k))$ (being isomorphic to $B_{H'_1}(v, k)$) obtaining the labelled graph H_2. Now, if normal forms are r-locally characterized for some $r \leq k$, then H'_2 is K-characterized for some $K \in T$. This implies that H_2 is K-characterized, too, which contradicts the fact that H_2 is not a normal form.

Proposition 13 may be easily generalized in the following way

Proposition 14. *Let $\mathcal{I} \subseteq \mathcal{G}$ be a class of labelled graphs, and let R be a $k-$locally generated relabelling relation. Assume that $\mathcal{I}$ contains labelled graphs G, G' and G is a connected, q-sheeted $k-$covering of G' via γ with $q \geq 2$. Then normal forms obtained from G are not $K-$characterized for any labelled graph K such that $\gamma^{-1}(K)$ is a disjoint union of graphs isomorphic to K.*

From Lemma 11 and Proposition 13 we obtain a more general result:

Theorem 15. *Let $\mathcal{I}$ be a class of connected labelled graphs closed under connected k-coverings and let R be a $k-$locally generated relabelling relation. Assume that a graph $G \in \mathcal{I}$ and an edge $e \in E(G)$ exist such that $(V(G), E(G) \setminus \{e\})$ is connected, but e belongs to no cycle of length at most $2k+1$. Let $r \leq k$.*

Then normal forms obtained from $\mathcal{I}$ cannot be $r-$locally characterized.

Recall that a graph G is a homeomorphic image of G' if G can be obtained from G' by subdivision of edges.

Corollary 16. *Let $\mathcal{I}$ be a class of connected labelled graphs closed under connected k-coverings and under homeomorphisms, and containing at least one non-tree graph. Let R be a $k-$locally generated relabelling relation and let $r \leq k$.*

Then normal forms obtained from $\mathcal{I}$ cannot be $r-$locally characterized.

Our result is quite powerful: local computations are very general, they include relabelling with an infinite number of labels and an infinite number of rules, provided that the diameter of the rules is bounded by some constant. The relabelling relation may be deterministic or not. For Proposition 13 and Theorem 15, T may be infinite provided that the diameter of the graphs is uniformly bounded. As an illustration, we give some concrete applications:

Corollary 17. *There is no local computation system allowing the local detection of termination which solves one of the following problems on uniformly labelled graphs:*

– *computing the size of a graph;*
– *computing the sum, product, minimum or maximum of the integers labelling the vertices of a graph;*
– *solving the majority problem, i.e. determining for a graph with nodes labelled by A or B whether the number of vertices labelled by A is greater than the number of vertices labelled by B.*

We note that these results were already known in the case of rings [17].

6 Quasi $k-$coverings and local detection of normal forms: the case of $T-$prime graphs

In this section we introduce the notion of quasi k-coverings, which allows to extend the results of the previous section to certain families of graphs, e.g. to T-prime graphs.

6.1 Quasi k-coverings

Definition 18. Let G, G' be two labelled graphs and let $\gamma\colon V(G) \to V(G')$ be a graph homomorphism. Let $k > 0$ be a positive integer.

Then G is a *quasi $k-$covering* of G' of size s if there exist a finite or infinite k-covering G_0 of G' via δ, vertices $v_0 \in V(G_0)$, $v \in V(G)$, and an integer $r > 0$ such that

1. $B_G(v,r)$ is isomorphic via φ to $B_{G_0}(v_0,r)$,
2. $\mathrm{card}(V(B_G(v,r))) \geq s$, and
3. $\gamma = \delta \circ \varphi$ when restricted to $V(B_G(v,r))$.

The question whether a graph has nontrivial finite or infinite k-coverings is recursively equivalent to the property of being a quasi k-covering. Hence, by [6] the property introduced above is in general undecidable. We can replace in the previous definition k-coverings by classical coverings, thus obtaining *quasi coverings*. The question whether a graph is a quasi covering of another graph becomes now decidable, it can be solved in NP.

The idea behind quasi k-coverings is to enable the simulation of local computations on a given graph in a restricted area of a larger graph, such that the simulation can lead to false conclusions. The restricted area where we can perform the simulation will shrink while the number of simulated steps increases.

Consider a quasi k-covering G of G' via γ. This means that a vertex $z \in V(G)$ and an integer $r > 0$ exist such that $B_G(z,r)$ is isomorphic to a subgraph of a k-covering G_0 of G'. More precisely, $B_G(z,r)$ is isomorphic via φ to $B_{G_0}(z_0,r)$, where G_0 is a k-covering of G' via δ. Moreover, $\mathrm{card}(V(B_G(z,r))) \geq s$ and $\gamma = \delta \circ \varphi$ on $V(B_G(z,r))$.

Fix now a spanning tree T of G', then $\delta^{-1}(T) \subseteq V(G_0)$ is a disjoint union of copies of T (see Remark 9). Let $\mathcal{J} = \{T_0, T_1, \dots, T_q\} \subseteq \gamma^{-1}(T) \subseteq V(G)$ be such that for all vertices $u \in V(T_i)$, $0 \leq i \leq q$, the ball $B_G(u,k)$ is included in $B_G(z,r)$. Suppose also w.l.o.g. that $z \in V(T_0)$.

We consider in the following the undirected graph $H = (\{0,\dots,q\},F)$ with $\{i,j\} \in F$ if and only if for some $x \in V(T_i)$, $y \in V(T_j)$ there is an edge $\{x,y\} \in E(G)$. By means of H we obtain a distance d on $\mathcal{J}$ given by $d(T_i,T_j) = d_H(i,j)$. Note that the degree of vertices of H is bounded by $\mathrm{card}(E(G')) - \mathrm{card}(V(G')) + 1$. Hence, for each $d \geq 1$ by choosing s sufficiently large (depending on G', k, d) we obtain $d(T_0,T_i) \geq d$ for some $T_i \in \mathcal{J}$.

Lemma 19. *Let $G, G', \gamma, \mathcal{J}, d$ be as above, with $d(T_0,T_i) \geq l$ for some $T_i \in \mathcal{J}$, $l \geq 2k$. Let R be a k-locally generated relabelling relation and suppose $G'RG'_1$. Moreover, assume that for every $T_i \in \mathcal{J}$ with $d(T_0,T_i) \leq l$ and for every vertex $u \in V(T_i)$ the labelled balls $B_G(u,k)$ and $B_{G'}(\gamma(u),k)$ are isomorphic via γ.*

Then a labelled graph G_1 exists such that GR^*G_1 holds. Moreover, for every $T_i \in \mathcal{J}$ with $d(T_0,T_i) \leq l-2k$ and for every vertex $v \in V(T_i)$ the labelled balls $B_{G_1}(v,k)$ and $B_{G'_1}(\gamma(v),k)$ are isomorphic via γ.

Proof. Let $G'RG'_1$ hold via a relabelling step which changes only the relabelling of $B_{G'}(v',k)$. We can simulate this step on each $v \in \gamma^{-1}(v')$ with $v \in V(T_i)$ and $d(T_0,T_i) \leq l$. Let G_1 denote the graph obtained in this way.

Suppose that $w \in V(T_i)$ and $d(T_0,T_i) \leq l-2k$ holds, and let $w' = \gamma(w)$. If $B_{G'}(v',k) \cap B_{G'}(w',k) = \emptyset$ then $B_G(v,k) \cap B_G(w,k) = \emptyset$ holds for all $v \in \gamma^{-1}(v') \cap \cup_{T \in \mathcal{J}} V(T)$, and the result holds by induction. Hence assume that $B_{G'}(v',k) \cap B_{G'}(w',k) \neq \emptyset$ and let v be the unique vertex in $\gamma^{-1}(v')$ such that $B_G(v,k) \cap B_G(w,k) \neq \emptyset$. Moreover, let $v \in V(T_j)$ and note that $d(T_0,T_j) \leq$

$d(T_0, T_i) + 2k \leq l$. Therefore, the labelled balls $B_{G_1}(w, k)$ and $B_{G_1}(w', k)$ are isomorphic.

6.2 Local detection of termination and quasi k-coverings

Lemma 19 yields a more general result on the impossibility of detecting termination locally.

Theorem 20. *Let $\mathcal{I}$ be a class of connected labelled graphs and let R be a k-locally generated relabelling relation. Suppose that some $G' \in \mathcal{I}$ has connected quasi k-coverings in $\mathcal{I}$ of arbitrary large size. Let $r \leq k$.*

Then normal forms obtained from $\mathcal{I}$ cannot be r-locally characterized.

Proof. Let C be a relabelling chain of length n on G', $C = (G' = G'_0, G'_1, \ldots, G'_n)$, such that G'_n is a normal form. Let G be a quasi k-covering of G' of size s. For s sufficiently large we have for some $T_i \in \mathcal{J}$ that $d(T_0, T_i) \geq 2k(n+1)$ (recall the definition of $\mathcal{J}$ and d from Lemma 19).

We can apply Lemma 19 with $l = 2k(n+1-m)$ for the mth relabelling step of C. We obtain thus a relabelling G_n of G with GR^*G_n such that G_n is a normal form. However, we have simulated no step of C on vertices belonging to $V(T_i)$ with $d(T_0, T_i) = 2k(n+1)$ (recall that such vertices still have balls of radius k isomorphic to their image by γ). Hence, this contradicts the fact that G_n is a normal form.

Clearly, we cannot use the results of the previous section for the family of T-prime graphs, because no connected non-isomorphic k-covering of a $T-$prime graph is $T-$prime. But we can apply Theorem 20. For this, suppose that G' is a connected T-prime graph containing an edge e' such that $(V(G'), E(G') \setminus \{e'\})$ is still connected, but e' belongs to no cycle of length at most $2k+1$. The construction from Lemma 11 can be easily modified such that we obtain a quasi k-covering of size at least $(q-2)\text{card}(V(G'))$ which is also T-prime. For this, it suffices to subdivide the edge $\{x_{q-1}, y_0\}$ until the size of the graph obtained is prime (hence, the graph obtained is T-prime).

As in the previous section it follows that

Corollary 21. *There is no local computation system with local detection of termination where all input graphs are uniformly labelled by the same label, which solves one of the following problems:*

- *computing the size of $T-$prime graphs;*
- *computing the sum, product, minimum or maximum of the integers labelling the vertices of a $T-$prime graph;*
- *solving the majority problem for the family of $T-$prime graphs.*

Remark 22. Recall that there is an election algorithm for T-prime graphs which uses the size of the graph as additional knowledge. The natural question which arises is whether this knowledge is necessary. Theorem 20 provides an indirect positive answer to this question. More precisely, suppose that the election problem could be solved with local detection of termination on uniformly labelled T-prime graphs (i.e. labelled by a fixed constant). Then we could compute after the election the size of the graph, thus contradicting the previous corollary.

Moreover, we note that the construction of Theorem 20 can be slightly modified in order to obtain the impossibility result for the election problem for T-prime graphs directly.

7 Final remarks

We consider the following three problems: the election problem (ELECT), the local detection of termination (LDT) and computing the size of the graph (SIZE).

We note that ELECT and LDT are equivalent with respect to local computations: if we can solve the election problem for a class of graphs $\mathcal{I}$, then we can also detect the termination of a local computation system on $\mathcal{I}$ locally. Conversely, if we have a class of uniformly labelled graphs $\mathcal{I}$ and a local computation system with local termination detection such that every element of $\mathcal{I}$ is reducible, then we can solve ELECT on $\mathcal{I}$.

The first assertion is easily seen by letting the elected vertex compute a spanning tree and observe whether a normal form has been reached. For the other direction assume that normal forms obtained from $\mathcal{I}$ with respect to a local computation system of radius k are characterized by a set of labelled graphs T. Moreover, suppose that every graph in $\mathcal{I}$ is reducible and let r be an upper bound for the radius of each element of T. Consider a normal form G obtained from $H \in \mathcal{I}$ and two vertices u, v such that both $B_G(u, r)$ and $B_G(v, r)$ contain a subgraph isomorphic to an element of T. Since T characterizes exactly normal forms the balls $B_G(u, r)$ and $B_G(v, r)$ contain each a subgraph from T due to the last step of the relabelling chain from H to G. Hence, the distance between u and v is at most $4r$. A simple relabelling system with forbidden contexts of radius $2r$ can now be used in order to elect one of the vertices u having the property that $B_G(u, r)$ contains a subgraph isomorphic to an element of T (thus labelled by T): every path of length less or equal $2r$ with extremities labelled by T changes the label of one of its extremities into N. A vertex labelled by T with no further neighbours labelled by T at distance less or equal $4r$ becomes elected.

On the other hand, there is also an easy reduction from SIZE to ELECT, since the size of the graph can be computed along a rooted spanning tree. However, ELECT is more difficult than SIZE, a fact which can be seen considering the class of hypercubes. Clearly, each vertex in a hypercube can compute locally its degree n, thus also the size 2^n of the graph. However, by symmetry arguments it can be easily shown that no local computation system can solve ELECT for the class of hypercubes. To see this, assume R is a relabelling system of radius k and let H_n be the hypercube with 2^n nodes, $n > 2k$. Then we can define a mapping f_{2k} on H_n by letting $f_{2k}(b_1 \dots b_n) = \bar{b}_1 \dots \bar{b}_{2k} b_{2k+1} \dots b_n$ ($b_i \in \{0, 1\}$). Clearly, for each vertex x the balls $B_{H_n}(x, k)$ and $B_{H_n}(f_{2k}(x), k)$ are disjoint and isomorphic. We can simulate each relabelling step of R on both balls in parallel. This simulation satisfies the condition that the labelled balls of radius k with center y, resp. $f_{2k}(y)$ are isomorphic via f_{2k}. We can summarize:

Proposition 23. *The election problem (ELECT) is equivalent to the termination problem (LDT). The size problem is reducible to the election problem and the election is not reducible to the size problem.*

References

1. D. Angluin, *Local and global properties in networks of processors,* Proceedings of the 12th Symposium on theory of computing, (1980), pp. 82–93.
2. C. Berge, *Graphes et Hypergraphes,* Dunod (1983).
3. H.-L. Bodlaender and J. Van Leeuwen, *Simulation of large networks on smaller networks,* Information and Control **71** (1986) 143–180.
4. H.-L. Bodlaender, *The classification of coverings of processor networks,* J. Parallel Distrib. Comput. **6** (1989) 166–182.
5. B. Courcelle and Y. Métivier, *Coverings and minors : application to local computations in graphs,* Europ. J. Combinatorics **15** (1994) 127–138.
6. F. Demichelis and W. Zielonka, *Decidability questions for graph k-coverings,* Technical Report, LaBRI, 1997.
7. M. J. Fisher, N. A. Lynch and M. Merritt, *Easy impossibility proofs for distributed consensus problems,* Distrib. Comput. **1** (1986) 26–29.
8. I. Litovsky and Y. Métivier, *Computing trees with graph rewriting systems with priorities,* In: Tree automata and languages, M. Nivat and A. Podelski (editors), Elsevier Science Publishers B.V., (1992) 115–139.
9. I. Litovsky, Y. Métivier and W. Zielonka, *On the recognition of families of graphs with local computations,* Information and Computation **118,** No 1 (1995) 110–119.
10. I. Litovsky, Y. Métivier and E. Sopena, *Different local controls for graph relabelling systems,* Math. Syst. Theory **28,** (1995) 41–65.
11. I. Litovsky, Y. Métivier and E. Sopena, *Checking global graph properties by means of local computations: the majority problem,* Electronic Notes in Theoretical Computer Science 1 (1995).
12. N. A. Lynch, *Distributed algorithms,* Morgan Kaufman Publishers (1996).
13. W. S. Massey, *A basic course in algebraic topology,* Graduate texts in mathematics, Springer-Verlag (1991).
14. A. Mazurkiewicz, *Solvability of asynchronous ranking problem,* Inform. Proc. Letters **28,** (1988) 221–224.
15. Y. Métivier, N. Saheb and P.-A. Wacrenier, *A distributed algorithm for computing a spanning tree,* Technical report, LaBRI **1152-96,** (1996).
16. M. Raynal, *Distributed algorithms and protocols,* Wiley and sons (1988).
17. G. Tel, *Introduction to distributed algorithms,* Cambridge University Press (1994).
18. M. Yamashita and T. Kameda, *Computing on anonymous networks: Part I - Characterizing the solvable cases,* IEEE Transactions on parallel and distributed systems **7,** No 1, (1996) 69–89.
19. M. Yamashita and T. Kameda, *Computing on anonymous networks: Part II - decision and membership problems,* IEEE Transactions on parallel and distributed systems **7,** No 1, (1996) 90–96.