

A Characterization of Families of Graphs in Which Election Is Possible

(*Extended Abstract*)

Emmanuel Godard and Yves Métivier

LaBRI, Université Bordeaux I, ENSEIRB,
351 cours de la Libération
33405 Talence, France
{godard,metivier}@labri.fr

1 Introduction

The Model. We consider networks of processors with arbitrary topology. A network is represented as a connected, undirected graph where vertices denote processors and edges denote direct communication links. Labels (or states) are attached to vertices and edges. Labels are modified locally, that is, on a subgraph of fixed radius 1 of the given graph, according to certain rules depending on the subgraph only (*local computations*). The relabelling is performed until no more transformation is possible, *i.e.*, until a normal form is obtained.

The Problem. The election problem is one of the paradigms of the theory of distributed computing. Considering a network of processors the election problem is to arrive at a configuration where exactly one process is in the state *elected* and all other processes are in the state *non-elected* see [Tel00]. The elected vertex is used to make decisions, to centralize or to broadcast some information.

Known Results. Graphs where election is possible were already studied but the algorithms usually involved some particular knowledge. Solving the problem for different knowledge has been investigated for some particular cases including (see [Tel00] for details): - the network is known to be a tree - the network is known to be complete - the network is known to be a grid - the nodes have different identification numbers - the network is known to be a ring and has a known prime number of vertices.

The classical proof techniques used for showing the non-existence of election algorithm are based on coverings [Ang80], which is a notion known from algebraic topology [Mas91]. A graph G is a covering of a graph H if there is a surjective morphism from G to H which is locally bijective. The general idea is as follows. If G and H are two graphs such that G covers H and $G \neq H$, then every local computation on H induces a local computation on G and every label which appears in H appears at least twice in G . Thus using H it is always possible to build a computation in G such that the label *elected* appears twice. By this way it is proved that there is no election algorithm for G and H [Ang80].

A graph G is called covering-minimal if every covering from G to some H is a bijection. Mazurkiewicz has proved that, knowing the size of graphs, there exists

an election algorithm for the class of covering-minimal graphs [Maz97]. This distributed algorithm, applied to a graph of size n , assigns bijectively numbers of $[1..n]$ to vertices of G .

In [MMW97] the notion of quasi-covering has been introduced to study the problem of termination detection. A graph G is a quasi-covering of a graph H if G is locally a covering of H (locally means that there is a vertex v of G and a positive integer k such that the ball centered on v of radius k is a covering of a ball of H).

The Main Result. In this paper, using some techniques of [MT00] developed for the termination detection problem, we characterize which knowledge is necessary and sufficient to have an election algorithm, or equivalently, what is the general condition for a class of graphs to admit an election algorithm. More precisely we prove the following theorem (Theorem 15):

There is an election algorithm for a family \mathcal{I} of graphs if and only if graphs of \mathcal{I} are minimal for the covering relation and every graph G of \mathcal{I} has quasi-coverings of bounded radius in \mathcal{I} .

Sufficient conditions given below are just special cases of criteria of Theorem 15.

We explain new parts in this theorem. It is well known (see above) that the existence of an election algorithm needs graphs minimal for the covering relation. Analogously to [MMW97], we prove in this paper that if a graph is minimal for the covering relation and admits quasi-coverings of arbitrary large size in the family there is no election algorithm. This part can be illustrated by the family of prime rings. Indeed, prime rings are minimal for the covering relation nevertheless there is no election algorithm for this family: without the knowledge of the size, a ring admits quasi-covering prime rings of arbitrary large size.

These two results prove one direction of Theorem 15. To prove the converse:

- we extend the Mazurkiewicz algorithm to labelled graphs;
- we prove that the Mazurkiewicz algorithm applied in a graph G enables the reconstruction, on each node of G , of a graph K such that G is a quasi-covering of K ; and when the computation is terminated G is a covering of K ;
- we use an extension of an algorithm by Szymanski, Shi and Prywes [SSP85] which enables the distributed detection of stable properties in a graph;
- we prove that the bounded size of quasi-coverings of a given graph enables to each node v to detect the termination of the Mazurkiewicz algorithm and finally each node can decide if it has obtained the maximum number among numbers computed by the Mazurkiewicz algorithm.

Related Works. Among models related to our model there are local computation systems as defined by Rosenstiehl et al. [FHR72], Angluin [Ang80], Yamashita and Kameda [KY96] and Boldi and Vigna [BV99]. In [FHR72] a synchronous model is considered, where vertices represent (identical) deterministic finite automata. The basic computation step is to compute the next state of each processor according to its state and the states of its neighbours. In [Ang80]

an asynchronous model is considered. A basic computation step means that two adjacent vertices exchange their labels and then compute new ones. In [KY96] an asynchronous model is studied where a basic computation step means that a processor either changes its state and sends a message or it receives a message. In [BV99] networks are directed graphs coloured on their arcs; each processor changes its state depending on its previous state and on the states of its in-neighbours. Activation of processors may be synchronous, asynchronous or interleaved.

2 Graphs, Labelled Graphs and Coverings

We only consider finite, undirected and connected graphs without multiple edges and self-loops. If G is a graph, $V(G)$ denotes the set of vertices and $E(G)$ denotes the set of edges. Let v be a vertex, we denote by $B_G(v, k)$, or briefly $B(v, k)$, the centered ball of radius k with center v . The set of neighbours of a vertex v in G is denoted by $N_G(v)$.

A homomorphism between two graphs G and H is a mapping $\gamma: V(G) \rightarrow V(H)$ 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 γ^{-1} is also a homomorphism. By $G \simeq G'$ we mean 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. The class of all graphs will be denoted \mathcal{G} .

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 $\lambda: V(G) \cup E(G) \rightarrow L$ is the labelling function. 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}_L .

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 φ is a homomorphism from (G, λ) to (G', λ') if φ is a graph homomorphism from G to G' which preserves the labelling, i.e. such that $\lambda'(\varphi(x)) = \lambda(x)$ holds for every $v \in V(G)$ and if $\{u, v\}$ is an edge of G then $\lambda(\{u, v\}) = \lambda(\{\varphi(u), \varphi(v)\})$. A labelled graph will be designed by a bold letter like \mathbf{G} , \mathbf{H} etc ... If \mathbf{G} is a labelled graph, G denotes the underlying graph. An *occurrence* of (G, λ) in (G', λ') is an isomorphism φ between (G, λ) and a subgraph (H, η) of (G', λ') . 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 $B_G(v, 1)$ is a bijection onto $B_H(\gamma(v), 1)$. The covering is proper if G and H are not isomorphic. It is called connected if G (and thus also H) is connected. We extend the notion of covering to labelled graphs in an obvious way. The labelled graph (H, λ') is

covered by (G, λ) via γ , if γ is a homomorphism from (G, λ) to (H, λ') whose restriction to $B_G(v, 1)$ is an isomorphism from $(B_G(v, 1), \lambda)$ to $(B_H(\gamma(v), 1), \lambda')$.

A graph G is called *covering-minimal* if every covering from G to some H is a bijection. Graphs with prime size (or with prime number of edges), trees, labelled graphs with a distinguished vertex, graphs with nodes having different identification numbers are examples of covering-minimal graphs.

Let H be a connected graph and let G be a covering of H via γ . Then there exists an integer q such that, for every $v \in V(H)$, we have $\text{card}(\gamma^{-1}(v)) = q$. The integer q is called the number of *sheets* of the covering.

3 Local Computations in Graphs

Graph relabelling systems and more generally local computations satisfy the following constraints which seem to be natural when describing distributed computations with a decentralized control:

- (C1) they do not change the underlying graph but only the labelling of its components (edges and/or vertices), the final labelling being the result of the computation,
- (C2) they are *local*, that is, each relabelling step changes only a connected subgraph of a fixed size in the underlying graph,
- (C3) they are *locally generated*, that is, the application condition of the relabelling only depends on the *local context* of the relabelled subgraph.

3.1 Local Computations

Local computations as considered here can be described in the following general framework. Let \mathcal{G}_L be the class of L -labelled graphs and let $\mathcal{R} \subseteq \mathcal{G}_L \times \mathcal{G}_L$ be a binary relation on \mathcal{G}_L . Then \mathcal{R} will denote a graph rewriting relation. We assume that \mathcal{R} is closed by isomorphism, i.e., whenever $\mathbf{G}\mathcal{R}\mathbf{G}'$ if $\mathbf{H} \simeq \mathbf{G}$ then $\mathbf{H}\mathcal{R}\mathbf{H}'$ for some labelled graph $\mathbf{H}' \simeq \mathbf{G}'$. In the remainder of this paper \mathcal{R}^* stands for the reflexive and transitive closure of \mathcal{R} . The labelled graph \mathbf{G} is \mathcal{R} -*irreducible* if there is no \mathbf{G}' such that $\mathbf{G}\mathcal{R}\mathbf{G}'$. Let $\mathbf{G} \in \mathcal{G}_L$, then $\text{Irred}_{\mathcal{R}}(\mathbf{G})$ denotes the set of \mathcal{R} -irreducible graphs (or just irreducible if \mathcal{R} is fixed) which can be obtained from \mathbf{G} using \mathcal{R} . The relation \mathcal{R} is noetherian if there is no infinite relabelling chain $\mathbf{G}_1\mathcal{R}\mathbf{G}_2\mathcal{R}\dots$.

Definition 1. Let $\mathcal{R} \subseteq \mathcal{G}_L \times \mathcal{G}_L$ be a graph rewriting relation. 1. \mathcal{R} is a relabelling relation if whenever two labelled graphs are in relation then the underlying graphs are equal i.e.: $\mathbf{G}\mathcal{R}\mathbf{H} \implies G = H$. 2. \mathcal{R} is local if only labels of a ball of radius 1 may be changed by \mathcal{R} , i.e., $(G, \lambda)\mathcal{R}(G, \lambda')$ implies that there exists a vertex $v \in V(G)$ such that $\lambda(x) = \lambda'(x)$ for every $x \notin V(B_G(v, 1)) \cup E(B_G(v, 1))$.

The next definition states that a local relabelling relation \mathcal{R} is *locally generated* if its restriction on centered balls of radius 1 determines its computation on any graph.

Definition 2. Let \mathcal{R} be a local relabelling relation. Then \mathcal{R} is locally generated if the following is satisfied: 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 $B_G(v, 1)$ and $B_H(w, 1)$ are isomorphic via $\varphi: V(B_G(v, 1)) \rightarrow V(B_H(w, 1))$ and $\varphi(v) = w$, the following three conditions

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

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

3.2 Local Computations and Coverings

The fundamental lemma which connects coverings and locally generated relabelling relations states that whenever \mathbf{G} is a covering of \mathbf{H} , every local computation in \mathbf{H} can be lifted to a local computation in \mathbf{G} which is compatible with the covering relation. This is expressed in the following diagram:

$$\begin{array}{ccc} \mathbf{G} & \xrightarrow{R^*} & \mathbf{G}' \\ \downarrow \text{covering} & & \downarrow \text{covering} \\ \mathbf{H} & \xrightarrow{R^*} & \mathbf{H}' \end{array}$$

Lemma 3 (Lifting Lemma). Let \mathcal{R} be a locally generated relabelling relation and let \mathbf{G} be a covering of \mathbf{H} via γ . Moreover, let $\mathbf{H}\mathcal{R}\mathbf{H}'$. Then there exists \mathbf{G}' such that $\mathbf{G}\mathcal{R}\mathbf{G}'$ and \mathbf{G}' is a covering of \mathbf{H}' .

3.3 Local Computations and Quasi-coverings

Definition 4. Let \mathbf{G}, \mathbf{H} be two labelled graphs and let γ be a partial function on $V(G)$ that assigns to each element of a subset of $V(G)$ exactly one element of $V(H)$. Then \mathbf{G} is a quasi-covering of \mathbf{H} via γ if there exists a finite or infinite covering \mathbf{G}_0 of \mathbf{H} via δ , vertices $z_0 \in V(G_0)$, $z \in V(G)$, and an integer $r > 0$ such that:

1. $B_{\mathbf{G}}(z, r)$ is isomorphic via φ to $B_{\mathbf{G}_0}(z_0, r)$,
2. the domain of definition of γ contains $B_G(z, r)$, and
3. $\gamma = \delta \circ \varphi$ when restricted to $V(B_G(z, r))$.

The integer r is called the radius of the quasi-covering, $\text{card}(V(B_G(z, r)))$ is called the size of the quasi-covering, and z the center. The graph \mathbf{G}_0 is the associated covering of the quasi-covering.

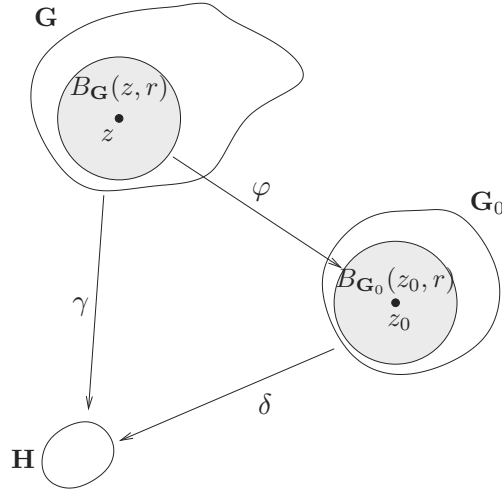


Fig. 1. \mathbf{G} is a quasi-covering of \mathbf{H} of radius r

The idea behind quasi-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. They have been introduced to study the problem of the detection of the termination [MMW97]. The following lemma makes precise the shrinking of the radius when one step of simulation is performed :

Lemma 5 (Quasi-Lifting Lemma). *Let \mathcal{R} be a locally generated relabelling relation and let \mathbf{G} be a quasi-covering of \mathbf{H} of radius r via γ . Moreover, let $\mathbf{H}\mathcal{R}\mathbf{H}'$. Then there exists \mathbf{G}' such that $\mathbf{G}\mathcal{R}^*\mathbf{G}'$ and \mathbf{G}' is a quasi-covering of radius $r - 2$ of \mathbf{H}' .*

Definition 6. *We define the number of sheets q of a quasi-covering to be the minimal cardinality of the sets of preimages of vertices of \mathbf{H} which are in the ball: $q = \min_{v \in V(H)} |\{w \in \delta^{-1}(v) \mid B_{\mathbf{G}}(w, 1) \subset B_{\mathbf{G}}(z, r)\}|$.*

Using the notation of the definition of a quasi-covering, we say that a quasi-covering is *strict* if $B_{\mathbf{G}}(z, r - 1)$ is not equal to G . Note that any non-strict quasi-covering is a covering. We have:

Lemma 7. *Let \mathbf{G} be a strict quasi-covering of \mathbf{H} of radius r via γ . Then, for any $q \in \mathbb{N}$, if $r \geq q|V(H)|$ then γ has at least q sheets.*

4 Two Fundamental Algorithms

4.1 The Mazurkiewicz Enumeration Algorithm

A distributed enumeration algorithm on a graph \mathbf{G} is a distributed algorithm such that the result of any computation is a labelling of the vertices that is a

bijection from $V(G)$ to $\{1, 2, \dots, |V(G)|\}$. In particular, an enumeration of the vertices where vertices know whether the algorithm has terminated solves the election problem. In [Maz97] Mazurkiewicz presents a distributed enumeration algorithm for the class of graphs minimal for the covering relation. In [Maz97], the computation model consists exactly in relabelling balls of radius 1. The Mazurkiewicz algorithm will be denoted \mathcal{M} . In the following we give a description of the Mazurkiewicz algorithm including its extension to labelled graphs.

Description. We first give a general description of the algorithm \mathcal{M} applied to the graph \mathbf{G} . Let $\mathbf{G} = (G, \lambda)$, let v_0 be a vertex of G , let $\{v_1, \dots, v_d\}$ be the set of neighbours of v_0 .

The label of the vertex v_0 used by \mathcal{M} is the couple $(\lambda(v_0), c(v_0))$ where $c(v_0)$ is a triple $(n(v_0), N(v_0), M(v_0))$ representing the following information during the computation (formal definitions are given below) :

- $n(v_0) \in \mathbb{N}$ is the number of the vertex v_0 computed by the algorithm
- $N(v_0) \in \mathcal{N}$ is the local view of v_0 , it is the set of triples defined by :

$$\{(n(v_i), \lambda(v_i), \lambda(\{v_0, v_i\})) \mid 1 \leq i \leq d\}$$

- $M(v_0) \subset \mathbb{N} \times L \times \mathcal{N}$ is the mailbox of v_0 and contains all the information received by v_0 at this step of the computation.

Every vertex v attempts to get its own number $n(v)$, which shall be an integer between 1 and $|V(G)|$. A vertex chooses a number and broadcasts it together with its label and its labelled neighbourhood all over the network. If a vertex u discovers the existence of another vertex v with the same number, then it compares its label and its *local view*, *i.e.*, its number-labelled ball, with the local view of its rival v . If the label of v or the local view of v is “stronger”, then u chooses another number. Each new number, with its local view, is broadcasted again over the network. At the end of the computation it is not guaranteed that every node has a unique number, unless the graph is covering-minimal. However, all nodes with the same number will have the same label and the same *local view*.

The crucial property of the algorithm is based on a total order on local views such that the local view of any vertex cannot decrease during the computation. We assume that the set of labels L is equipped with a total order denoted $>$. Let v_0 be a vertex and let $\{v_1, \dots, v_d\}$ the neighbours of v_0 we assume that: $n(v_1) \geq n(v_2) \geq \dots \geq n(v_d)$, if $n(v_i) = n(v_{i+1})$ then $\lambda(v_i) \geq \lambda(v_{i+1})$, if $n(v_i) = n(v_{i+1})$ and $\lambda(v_i) = \lambda(v_{i+1})$ then $\lambda(\{v_0, v_i\}) \geq \lambda(\{v_0, v_{i+1}\})$. Then to $N(v)$, the local view, we associate the d -tuple $((n(v_1), \lambda(v_1), \lambda(\{v_0, v_1\})), \dots, (n(v_d), \lambda(v_d), \lambda(\{v_0, v_d\})))$.

Let $\mathcal{N}_>$ be the set of such ordered tuples. We define a total order, \prec , on $\mathcal{N}_>$ by comparing the numbers, then the vertex labels and finally the edge labels. Formally, let $((n_1, l_1, e_1), \dots, (n_d, l_d, e_d))$ and $((n'_1, l'_1, e'_1), \dots, (n'_d, l'_d, e'_d))$ be two elements of $\mathcal{N}_>$ then

$$((n'_1, l'_1, e'_1), \dots, (n'_d, l'_d, e'_d)) \prec ((n_1, l_1, e_1), \dots, (n_d, l_d, e_d))$$

if one of the following holds

1. $n_1 = n'_1, \dots, n_{i-1} = n'_{i-1}$ and $n'_i < n_i$ for some i or
2. $d' < d$ and $n_1 = n'_1, \dots, n_{d'} = n'_{d'}$ or
3. $d = d'$, $n_1 = n'_1, \dots, n_d = n'_d$ and $l_1 = l'_1, \dots, l_{i-1} = l'_{i-1}$ and $l'_i <_L l_i$ for some i or
4. $d = d'$ and $n_1 = n'_1, \dots, n_d = n'_d$ and $l_1 = l'_1, \dots, l_d = l'_d$ and $e_1 = e'_1, \dots, e_{i-1} = e'_{i-1}$ and $e'_i <_L e_i$ for some i .

The initial labelling of the vertex v_0 is $(\lambda(v_0), (0, \emptyset, \emptyset))$. The rules are described below for a given centered ball $B = B(v_0, 1)$ with center v_0 . The vertices v of B have labels $(\lambda(v), (n(v), N(v), M(v)))$. The labels obtained after applying a rule are $(\lambda(v), (n'(v), N'(v), M'(v)))$. To make the rules easier to be read, we omit labels that are left unchanged.

\mathcal{M} -1 : Diffusion rule

Precondition :

- $\exists v \in B(v_0, 1), M(v) \neq M(v_0)$

Relabelling :

- For all $v \in B(v_0, 1)$, $M'(v) := \bigcup_{w \in B} M(w)$

\mathcal{M} -2 : Renaming rule

Precondition :

- $\forall v, M(v) = M(v_0)$
 - 1. $n(v_0) = 0$
- or
2. $(n(v_0) > 0$ and $\exists (n(v_0), l, N) \in M(v_0)$
such that $((\lambda(v_0) < l)$ or
 $((\lambda(v_0) = l)$ and $(N(v_0) \prec N)))$)

Relabelling :

- $n'(v_0) = 1 + \max\{n \in \mathbb{N} \mid (n, l', N) \in M(v_0)\}$
- $\forall v \in B(v_0, 1)$, $N'(v)$ is obtained from $N(v)$ by replacing the value of $n(v_0)$ by $n'(v_0)$.
- $\forall v \in B(v_0, 1)$, the mailbox contents $M(v)$ changes to $M'(v) = M(v) \cup \bigcup_{w \in B} \{(n'(w), \lambda(w), N'(w))\}$.

Let \mathbf{G} be a labelled graph, if v is a vertex of G , the label of v after the run ρ of the Mazurkiewicz algorithm is denoted $(\lambda(v), c_\rho(v))$ with $c_\rho(v) = (n_\rho(v), N_\rho(v), M_\rho(v))$ and (λ, c_ρ) denotes the final labelling.

Similar to [Maz97], for covering-minimal graphs the algorithm computes a one-to-one correspondance n_ρ between the set of vertices of G and the set of integers $\{1, \dots, |V(G)|\}$.

We interpret the final labelling as a graph that each vertex could compute. For a mailbox M , we define for each integer n in $\{1, \dots, |V(G)|\}$ the maximal mail box element of the form (n, l, N) :

$$F(M) = \{(n, l, N) \in M \mid \text{there is no } (n, l', N') \in M \text{ verifying}$$

$$l' > l \text{ or } (l = l' \text{ and } N < N').$$

For a given M , we define the graph \mathbf{G}_M as the following graph:

$$\begin{aligned} V(G_M) &= \{n \mid \exists N, l, (n, l, N) \in F(M)\} \\ E(G_M) &= \{\{n, n'\} \mid (n, l, N) \in F(M), \text{ and } N = (\dots, (n', l', l''), \dots)\} \end{aligned}$$

We also define a labelling on G_M as follows, $\lambda_M(n) = l$, such that there exists N , $(n, l, N) \in F(M)$. Note that uniqueness of l comes from the definition of $F(M)$. Let ρ be a run of \mathcal{M} , then $(G_{M_\rho(u)}, \lambda_{M_\rho(u)})$ does not depend on u . We then define $\rho(\mathbf{G}) = (G_{M_\rho(u)}, \lambda_{M_\rho(u)})$. As $F(M_\rho(u))$ represents the final numbers and neighbourhoods, we have the following.

Proposition 8. *For a given execution ρ of the Mazurkiewicz algorithm, we have $V(\rho(\mathbf{G})) = \{n_\rho(v) \mid v \in V(G)\}$ and $E(\rho(\mathbf{G})) = \{\{n_\rho(v), n_\rho(w)\} \mid \{v, w\} \in E(G)\}$.*

The proposition means that $\rho(\mathbf{G})$ is the quotient graph of \mathbf{G} by n_ρ . Before we emphasize the role of $\rho(\mathbf{G})$, note that $\rho(\mathbf{G})$ can be locally computed by every vertex, and that the graph depends only on the label M_ρ .

If $\mathbf{G} = (G, \lambda)$ is a labelled graph $\rho(\mathbf{G})$ denotes the labelled graph defined on $(V(G_\rho), E(G_\rho))$ by the labelling such that the label of $n_\rho(v)$ is $\lambda(v)$.

The next proposition states that we can see a run of \mathcal{M} as computing a graph such that:

Proposition 9. *Let \mathbf{G} be a labelled graph. For all runs ρ of \mathcal{M} , \mathbf{G} is a covering of $\rho(\mathbf{G})$. For all \mathbf{H} such that \mathbf{G} is a covering of \mathbf{H} , there exists a run ρ such that $\mathbf{H} \simeq \rho(\mathbf{G})$ (fairness).*

4.2 An Algorithm to Detect Stable Properties

In this section we describe in our framework the algorithm by Szymanski, Shi and Prywes (the SSP algorithm for short) [SSP85]. We consider a distributed algorithm which terminates when all processes reach their local termination conditions. Each process is able to determine only its own termination condition. The SSP algorithm detects an instant in which the entire computation is achieved. Let G be a graph, to each node v is associated a predicate $P(v)$ and an integer $a(v)$. Initially $P(v)$ is false and $a(v)$ is equal to -1 . Transformations of the value of $a(v)$ are defined by the following rules. Each local computation acts on the integer $a(v_0)$ associated to the vertex v_0 ; the new value of $a(v_0)$ depends on values associated to vertices of $B_G(v_0, 1)$. More precisely, let v_0 be a vertex and let $\{v_1, \dots, v_d\}$ be the set of vertices adjacent to v_0 . If $P(v_0) = \text{false}$ then $a(v_0) = -1$; if $P(v_0) = \text{true}$ then $a(v_0) = 1 + \text{Min}\{a(v_k) \mid 0 \leq k \leq d\}$. In [SSP85] the following assumption is considered: for each node v the value of $P(v)$ eventually becomes true and remains true for ever.

In this paper we use a variant of the SSP algorithm presented in [MT00]. For each node v the value of $P(v)$ eventually becomes true and either remains true for ever or becomes false and remains false for ever. Thus the predicate $P(v)$ may

change twice: initially $P(v)$ is false and $a(v)$ is equal to -1 , $P(v)$ may become true and then $a(v)$ is equal to 0 , if $P(v)$ is true it may become false, in this case it remains false for ever and $a(v)$ is equal to -1 . We denote this variant by VSSP algorithm. We will use the following notation. Let $(\mathbf{G}_i)_{0 \leq i}$ be a relabelling chain associated to the VSSP algorithm. We denote by $a_i(v)$ (resp. $P_i(v)$) the integer (resp. the boolean) associated to the vertex v of \mathbf{G}_i . The fundamental property is:

Proposition 10. *Let $(\mathbf{G}_i)_{0 \leq i}$ be a relabelling chain associated to the VSSP algorithm. Let $v \in V(G)$, $i > 0$ such that $a_i(v) > 0$. Let $p > 0$, we suppose that for some integer $j > i$: $a_j(v) = a_i(v) + p$. Let $k < \text{Min}(a_i(v), \frac{p}{2})$. Then : $\forall w \in B(v, k) \quad a_i(w) \geq 0$.*

5 The Mazurkiewicz Algorithm + the VSSP Algorithm = Reconstruction Algorithm with Local Agreement

Following [MT00], the main idea developed in this section is to use the VSSP algorithm for computing the radius of stability of \mathcal{M} in a relabelling chain at a given step and at a given vertex. In other words, any vertex will know until which distance all vertices agree with its topology reconstruction.

Let $\mathbf{G} = (G, \lambda)$ be a labelled graph, let $(\mathbf{G}_i)_{0 \leq i}$ be a relabelling chain associated to a run of the Mazurkiewicz algorithm on the graph \mathbf{G} . To the node v of \mathbf{G}_i is associated the label $(\lambda(v), (n_i(v), N_i(v), M_i(v)))$. Using the interpretation of Subsection 4.2 by defining $F(M_i)$, this label enables in some cases the reconstruction of a graph.

We introduce on the node v of the graph \mathbf{G}_i the predicate $p_{\mathbf{H}}(v)$, that will be true if label of v in \mathbf{G}_i enables the reconstruction of $\mathbf{H} = \mathbf{G}_{M_i}(v)$ and $(n_i(v), \lambda(v), N_i(v)) \in F(M_i(v))$. The associated value $a_{\mathbf{H}}(v)$ is computed by the VSSP algorithm. We note \mathcal{Q} the merging of the two algorithms.

First, we formalize the output of \mathcal{Q} . We note

$$\mathbf{H}(v) = \begin{cases} \mathbf{G}_{M_i(v)} & \text{if it is defined and } (n_i(v), \lambda(v), N_i(v)) \in F(M_i(v)) \\ \perp & \text{otherwise.} \end{cases} \quad (1)$$

$$a(v) = \begin{cases} -1 & \text{if } \mathbf{H}(v) = \perp \\ a_{\mathbf{H}(v)}(v) & \text{otherwise.} \end{cases} \quad (2)$$

Note that, while $\mathbf{H}(v) = \perp$, the node knows that it is in a non final state. The output of \mathcal{Q} is now, on each vertex v , $\langle \mathbf{H}(v), a(v) \rangle$.

For a run of the Mazurkiewicz algorithm on \mathbf{G} , we note $(\mathbf{G}^{(i)})_{(0 \leq i)}$ the labelled graph obtained by adding on each node v of \mathbf{G} the number $n_i(v)$ computed by the Mazurkiewicz algorithm at step i on the vertex v . The main property of the computations is now:

Proposition 11 (quasi-covering progression). *At all step j , for all vertex v , the output of \mathcal{Q} on v is a couple $\langle \mathbf{H}, a \rangle$ such that if $\mathbf{H} \neq \perp$, then there exists a previous step $i < j$, such that $\mathbf{G}^{(i)}$ is a quasi-covering of \mathbf{H} of center v and of radius $\lfloor \frac{a}{3} \rfloor$.*

6 Main Result: A Characterization of Families of Graphs in Which Election Is Possible

6.1 Impossibility

Considering a labelled graph, we say that a given vertex v has been elected when the graph is in a global state such that exactly one vertex has the label *elected* and all other vertices have the label *non-elected*.

Let \mathcal{I} be a class of connected labelled graphs. Let \mathcal{R} be a locally generated relabelling relation, we say that \mathcal{R} is an election algorithm for the class \mathcal{I} if \mathcal{R} is noetherian and for any graph \mathbf{G} of \mathcal{I} and for any normal form \mathbf{G}' obtained from \mathbf{G} , $\mathbf{G}\mathcal{R}^*\mathbf{G}'$, there exists exactly one vertex with the label *elected* all other vertices have the label *non-elected*.

Angluin showed that no election algorithm exists for a family of graphs containing a graph H and a proper covering of H [Ang80]. In fact there is no election algorithm for a not covering-minimal labelled graph. Furthermore if we consider a class \mathcal{I} of labelled graphs containing a graph and quasi-coverings of arbitrary large size there is no election algorithm for this family. More precisely:

Proposition 12. *Let \mathbf{G} be a labelled graph which is not covering-minimal. Then there is no election algorithm for \mathbf{G} .*

Proposition 13 (Necessary condition). *Let \mathcal{I} be a class of connected covering-minimal labelled graphs such that there is an algorithm of election for this class. Then there exists a computable function $\tau : \mathcal{I} \rightarrow \mathbb{N}$ such that for all graph \mathbf{H} of \mathcal{I} , there is no quasi-covering of \mathbf{H} , distinct of \mathbf{H} , of radius greater than $\tau(\mathbf{H})$ in \mathcal{I} .*

Proof. Let \mathcal{R} denote an algorithm of election on \mathcal{I} . For a graph $\mathbf{H} \in \mathcal{I}$, define $\tau(\mathbf{H}) = 2|V(\mathbf{H})| + 2n$ where n is the number of steps of an entire execution of \mathcal{R} on \mathbf{H} . Then τ has the desired property.

We prove this by contradiction. Let $\mathbf{H} \in \mathcal{I}$. Let \mathcal{C} be the relabelling chain of length n on \mathbf{H} used for the definition of $\tau(\mathbf{H})$. $\mathcal{C} = (\mathbf{H} = \mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_n)$ such that \mathbf{H}_n is a normal form. By hypothesis the label *elected* appears exactly once in \mathbf{H}_n .

Let \mathbf{G} be a quasi-covering of \mathbf{H} of radius $\tau(\mathbf{H})$, distinct of H . By iteration of Lem. 5, we get \mathbf{G}' such that $\mathbf{G}\mathcal{R}^*\mathbf{G}'$ and \mathbf{G}' is a quasi-covering of \mathbf{H}_n of radius $\tau(\mathbf{H}) - 2n = 2|V(\mathbf{H})|$. The graph \mathbf{G} being covering minimal and distinct of \mathbf{H} , the quasi-covering \mathbf{G}' of \mathbf{H}_n is strict. Hence, by Lem. 7, the label *elected* appears at least twice in \mathbf{G}' . A contradiction.

6.2 Possibility

Let \mathcal{R} denote a locally generated relabelling relation and let \mathbf{G} denote a labelled graph. Let \mathcal{I} be a class of labelled graphs, terminal configurations obtained from \mathcal{I} are said to be locally characterized if there exists a set F of labels such that for any $\mathbf{G} \in \mathcal{I}$ and for any \mathbf{G}' , with $\mathbf{G}\mathcal{R}^*\mathbf{G}'$, \mathbf{G}' is a terminal configuration if and only if there exists a vertex v of \mathbf{G}' having its label in F . In this case termination is said to be explicit. First we have the following theorem :

Proposition 14 (Graph reconstruction). *Let \mathcal{I} be a class of connected covering-minimal labelled graphs. Suppose that there exists a computable function $\tau : \mathcal{I} \rightarrow \mathbb{N}$ such that for all graphs \mathbf{G} of \mathcal{I} , there is no quasi-covering of \mathbf{G} of radius greater than $\tau(\mathbf{G})$ in \mathcal{I} , except \mathbf{G} . Then there exists a locally generated relabelling relation with explicit termination which computes for any graph \mathbf{G} in \mathcal{I} and for any vertex v in $V(G)$ the graph \mathbf{G} .*

Proof. We use \mathcal{Q} with the termination condition on the output $\mathbf{H} \in \mathcal{I}$ and $a \geq 3\tau(\mathbf{H})$. Termination is a corollary of the fact that \mathcal{M} will eventually terminate and then the counter a will increase indefinitely. When this termination condition is fulfilled, by Prop. 11, we have that there exists a previous step n such that $\mathbf{G}^{(n)}$ is a quasi-covering of radius $\tau(\mathbf{H}(v))$ of $\mathbf{H}(v)$ and by hypothesis on τ , we deduce that $\mathbf{G}^{(n)} = \mathbf{H}(v)$.

The knowledge of this reconstructed graph implies that each node knows if its number is the maximal or not among numbers of the graph. Finally, each node can decide if it is elected or not. The result of this subsection and of the previous are summarized in the the following theorem.

Theorem 15. *Let \mathcal{I} be a class of connected labelled graphs. There exists an election algorithm for \mathcal{I} if and only if graphs of \mathcal{I} are minimal for the covering relation, and there exists a computable function $\tau : \mathcal{I} \rightarrow \mathbb{N}$ such that for all graph \mathbf{G} of \mathcal{I} , there is no quasi-covering of \mathbf{G} of radius greater than $\tau(\mathbf{G})$ in \mathcal{I} , except \mathbf{G} itself.*

6.3 Applications

Known results appears now as simple corollaries of Th. 15, for possibility:

- [Maz97] Covering minimal networks where the size is known ;
- Trees, complete graphs, grids, networks with identities: those families contains no q -sheeted quasi-covering of a given graph for $q \geq 2$, hence the τ function can be twice the size of the graph, see Lem. 7.

We also get some new possibility results, in particular for covering-minimal graphs with at least 1 and at most k distinguished vertices or for covering-minimal graphs where a bound of the size is known (it is new and cannot be directly derived from [Maz97]). On the impossibility side, it is a corollary of Th. 15 that there is no election algorithm for the family of covering-minimal tori (prime rings are a particular case).

We also notice that the complexity of the algorithm of Mazurkiewicz, computed in [Godar], is often worse than the optimal known bounds. That lack of optimality is the counterpart of the universality of the algorithm.

References

- Ang80. D. Angluin. Local and global properties in networks of processors. In *Proceedings of the 12th Symposium on theory of computing*, pages 82–93, 1980.

- BV99. P. Boldi and S. Vigna. Computing anonymously with arbitrary knowledge. In *Proceedings of the 18th ACM Symposium on principles of distributed computing*, pages 181–188. ACM Press, 1999.
- FHR72. J.-R. Fiksel, A. Holliger, and P. Rosenstiehl. Intelligent graphs. In R. Read, editor, *Graph theory and computing*, pages 219–265. Academic Press (New York), 1972.
- Godar. E. Godard. A self-stabilizing enumeration algorithm. *Inform. Proc. Letters*, to appear.
- KY96. T. Kameda and M. Yamashita. Computing on anonymous networks: Part i - characterizing the solvable cases. *IEEE Transactions on parallel and distributed systems*, 7(1):69–89, 1996.
- Mas91. W. S. Massey. *A basic course in algebraic topology*. Springer-Verlag, 1991. Graduate texts in mathematics.
- Maz97. A. Mazurkiewicz. Distributed enumeration. *Inf. Processing Letters*, 61:233–239, 1997.
- MMW97. Y. Métivier, A. Muscholl, and P.-A. Wacrenier. About the local detection of termination of local computations in graphs. In *International Colloquium on structural information and communication complexity*, pages 188–200, 1997.
- MT00. Y. Métivier and G. Tel. Termination detection and universal graph reconstruction. In *International Colloquium on structural information and communication complexity*, pages 237–251, 2000.
- SSP85. B. Szymanski, Y. Shy, and N. Prywes. Terminating iterative solutions of simultaneous equations in distributed message passing systems. In *Proceedings of the 4th Symposium on Principles of Distributed computing*, pages 287–292, 1985.
- Tel00. G. Tel. *Introduction to distributed algorithms*. Cambridge University Press, 2000.