Algorithmique Distribuée et Calculs Locaux

Yves Métivier

LaBRI - Université de Bordeaux
metivier@labri.fr

24 novembre 2009

Références

Thèse d'Emmanuel Godard (MdC à Marseille)
Titre : Réécritures de Graphes et Algorithmique Distribuée
Date : juin 2002.
emmanuel.godard@lif.univ-mrs.fr.

Thèse de Jérémie Chalopin (CR CNRS)
Titre : Algorithmique distribuée, calculs locaux et homomorphismes de graphes
Date : novembre 2006.
jeremie.chalopin@lif.univ-mrs.fr

Plan

Algorithmique distribuée

- *But :* Faire collaborer différents processus.
- *Difficulté :* Pas de coordination centralisée.
- Différents modèles pour exprimer les algorithmes distribués :
  - échanges de messages,
  - mémoire partagée,
  - calculs locaux.
- Différentes hypothèses possibles sur le réseau :
  - identifiants uniques ou réseaux anonymes,
  - systèmes fiables ou non-fiables.

Exemples de problèmes étudiés

Systèmes Anonymes :

Tous les processus sont initialement dans le même état et exécutent le même algorithme.

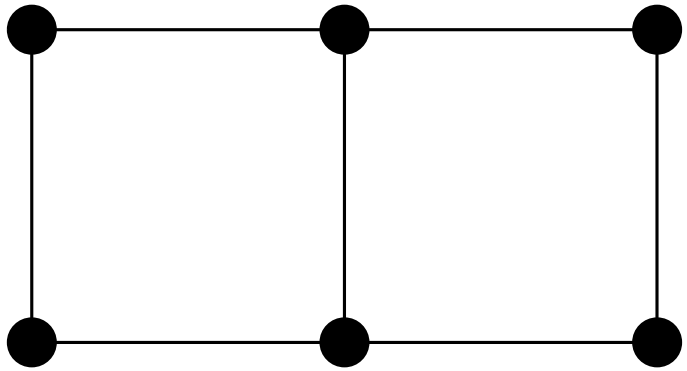
- **Élection :**
  Le but d'un algorithme d'élection est d'atteindre une configuration finale où
  - un processus est dans l'état ÉLU,
  - tous les autres processus sont dans l'état NON-ÉLU,
  - Élu et NON-ÉLU sont des états terminaux.

- **Détection de la terminaison :**
  Permettre à un sommet de savoir si il est dans son état final ou bien si le réseau a atteint son état final.

Représentation d'un réseau

Un réseau est représenté par un graphe simple où :

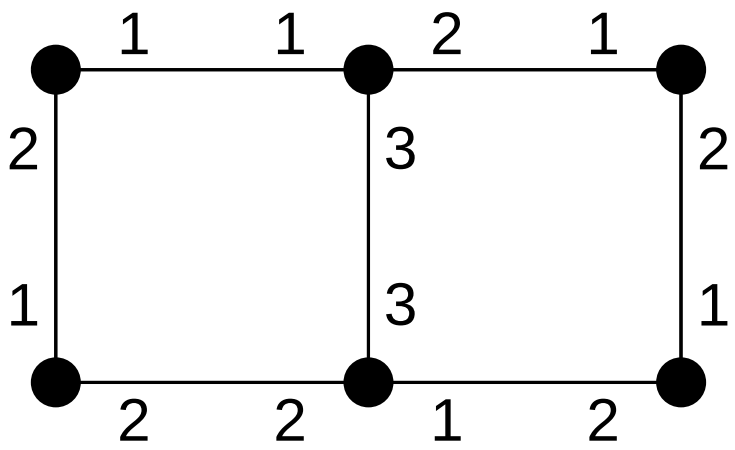
- les sommets représentent les processus,
- les arêtes représentent les liens de communication ou d'interaction,



Représentation d'un réseau

Un réseau est représenté par un graphe simple où :

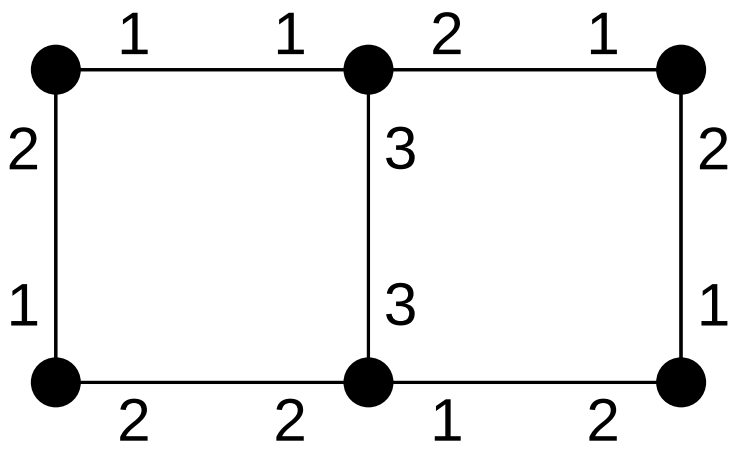
- les sommets représentent les processus,
- les arêtes représentent les liens de communication ou d'interaction,
- un étiquetage des ports peut exister pour permettre à chaque sommet de distinguer ses voisins.



Représentation d'un réseau

Un réseau est représenté par un graphe simple où :

- les sommets représentent les processus,
- les arêtes représentent les liens de communication ou d'interaction,
- un étiquetage des ports peut exister pour permettre à chaque sommet de distinguer ses voisins.



Un algorithme d'élection pour un graphe G doit fonctionner quel que soit l'étiquetage des ports.

Caractérisations existantes à propos de l'élection

- Mazurkiewicz '97
  - – Modèle où chaque pas de calcul nécessite une synchronisation entre un sommet et tous ses voisins puis lecture et modification des états de ces sommets.
  - \+ Caractérisation basée sur la notion de revêtements.
  - \+ Algorithme de nommage polynomial.

Caractérisations existantes à propos de l'élection

- Mazurkiewicz '97
  - – Modèle où chaque pas de calcul nécessite une synchronisation entre un sommet et tous ses voisins puis lecture et modification des états de ces sommets.
  - \+ Caractérisation basée sur la notion de revêtements.
  - \+ Algorithme de nommage polynomial.
- Yamashita et Kameda '96
  - \+ Modèle où les processus communiquent par échange de messages en mode asynchrone.
  - – Caractérisation basée sur la notion de vues.
  - – Algorithme d'élection nécessitant des messages de taille exponentielle.

Caractérisations existantes à propos de l'élection

- Mazurkiewicz '97
  - – Modèle où chaque pas de calcul nécessite une synchronisation entre un sommet et tous ses voisins puis lecture et modification des états de ces sommets.
  - \+ Caractérisation basée sur la notion de revêtements.
  - \+ Algorithme de nommage polynomial.
- Yamashita et Kameda '96
  - \+ Modèle où les processus communiquent par échange de messages en mode asynchrone.
  - – Caractérisation basée sur la notion de vues.
  - – Algorithme d'élection nécessitant des messages de taille exponentielle.
- Boldi *et al.* '96
  - \+ Modèle proche du modèle de Yamashita et Kameda.
  - \+ Caractérisation basée sur la notion de revêtements dirigés.
  - – Mêmes algorithmes que Yamashita et Kameda.

Objectifs

- Étudier des modèles intermédiaires entre les modèles de Mazurkiewicz et de Yamashita et Kameda.

Objectifs

- Étudier des modèles intermédiaires entre les modèles de Mazurkiewicz et de Yamashita et Kameda.
- Comparer les puissances de ces différents modèles.

Objectifs

- Étudier des modèles intermédiaires entre les modèles de Mazurkiewicz et de Yamashita et Kameda.
- Comparer les puissances de ces différents modèles.
- Obtenir un algorithme « à la Mazurkiewicz » dans le modèle de Yamashita et Kameda.

Objectifs

- Étudier des modèles intermédiaires entre les modèles de Mazurkiewicz et de Yamashita et Kameda.
- Comparer les puissances de ces différents modèles.
- Obtenir un algorithme « à la Mazurkiewicz » dans le modèle de Yamashita et Kameda.
- Mettre en évidence les outils combinatoires permettant d'étudier chaque modèle.

Plan

Calculs locaux

« What makes a system distributed is that each transition is only influenced by, and only influences, part of the configuration. »

Gerard Tel,
Introduction to Distributed Algorithms

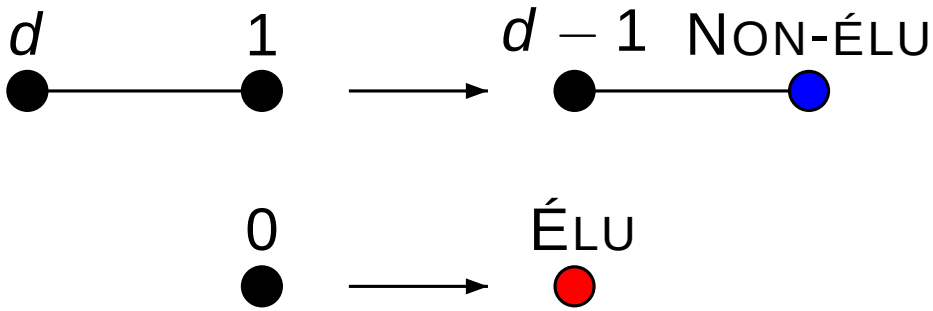
Le modèle des calculs locaux

- Modèle abstrait permettant de réfléchir sur des problèmes d'algorithmique distribuée indépendamment de la grande variété des modèles "concrets" utilisés pour représenter les systèmes distribués
- Comme les modèles classiques pour la programmation, il permet de construire et de prouver des systèmes complexes

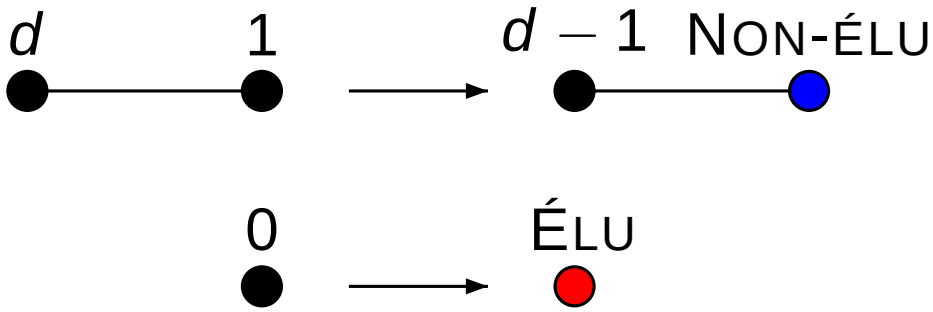
Calculs locaux (suite)

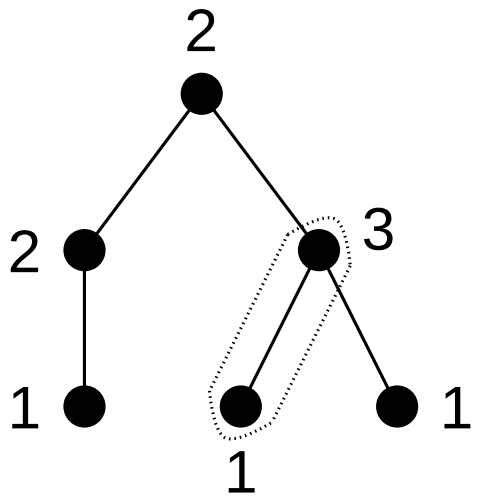
- Il est plus facile de comprendre et d'expliquer les problèmes pour obtenir des solutions ou des impossibilités
- Les Résultats d'impossibilité restent vrais dans des modèles plus faibles
- Toute solution dans ce modèle guide la recherche d'une solution dans des modèles plus faibles ou peut être implémenter par des procédures probabilistes
- Ce modèle permet de mettre en évidence les propriétés combinatoires des réseaux pour les problèmes considérés.

Un premier exemple

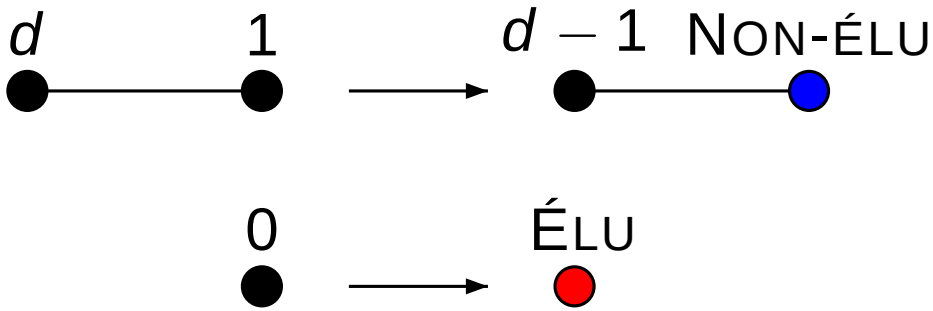


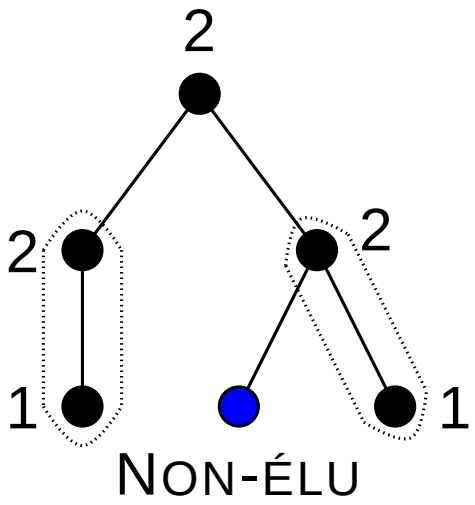
Un premier exemple



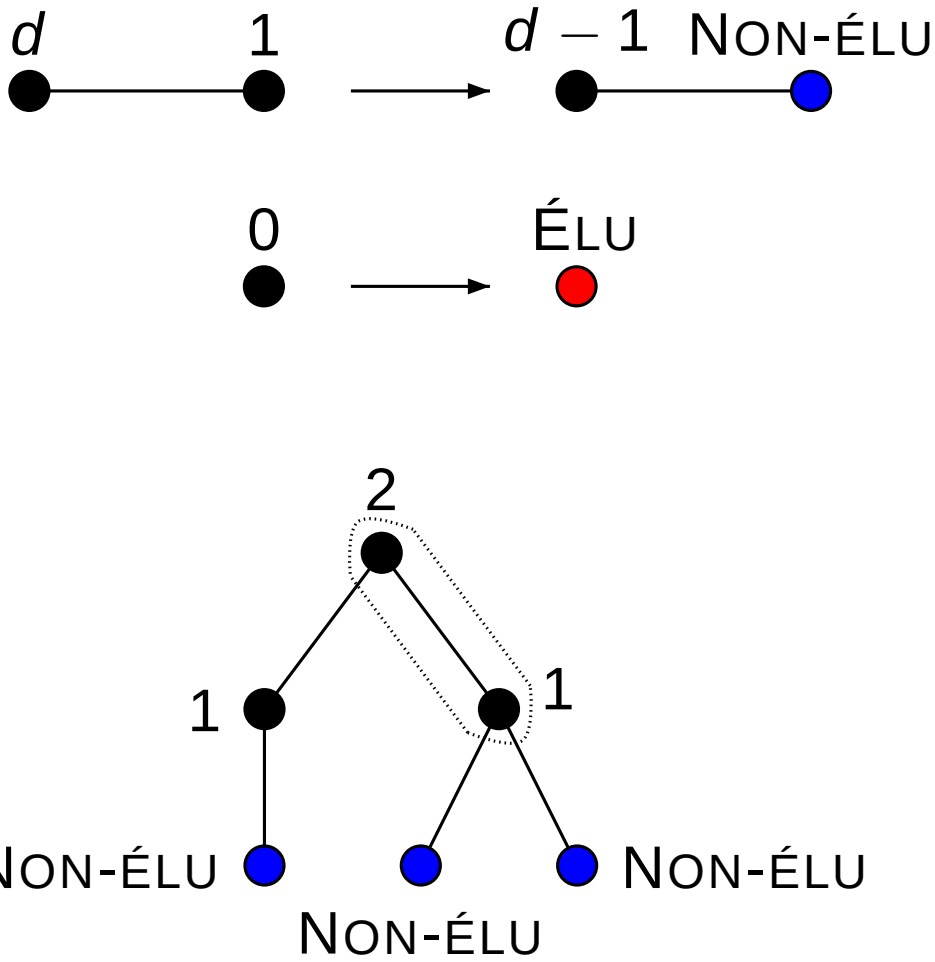


Un premier exemple

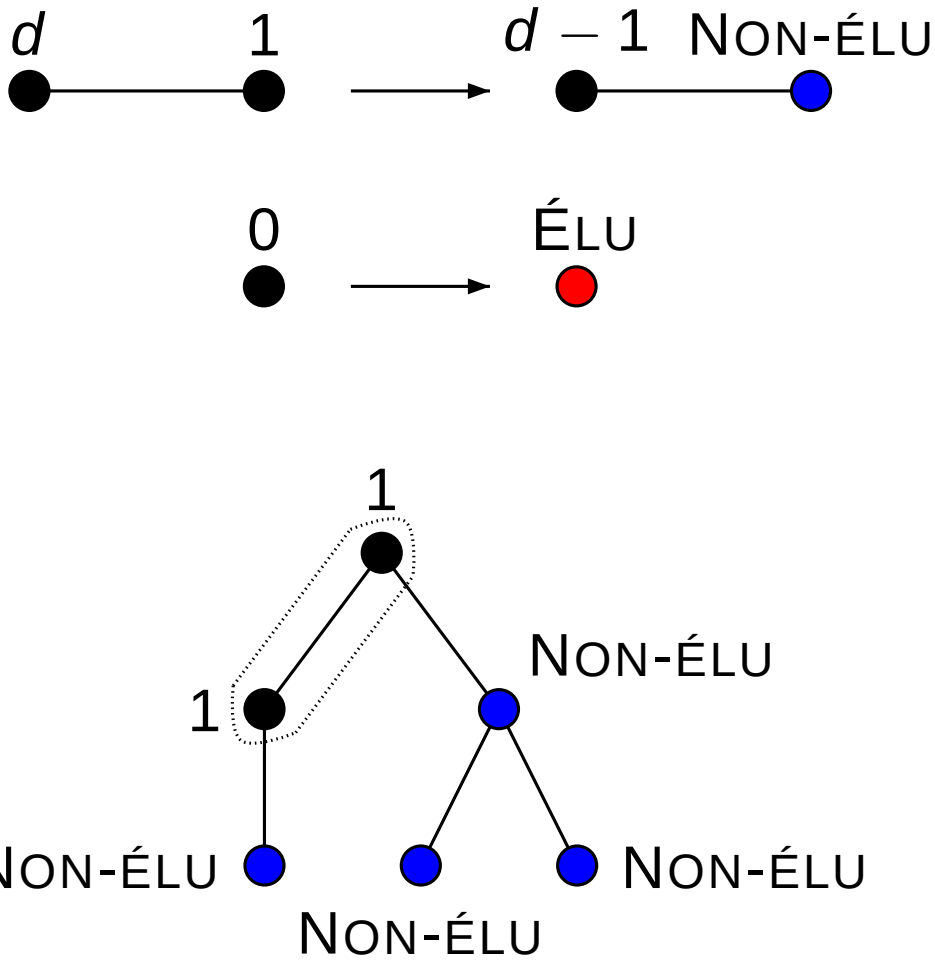




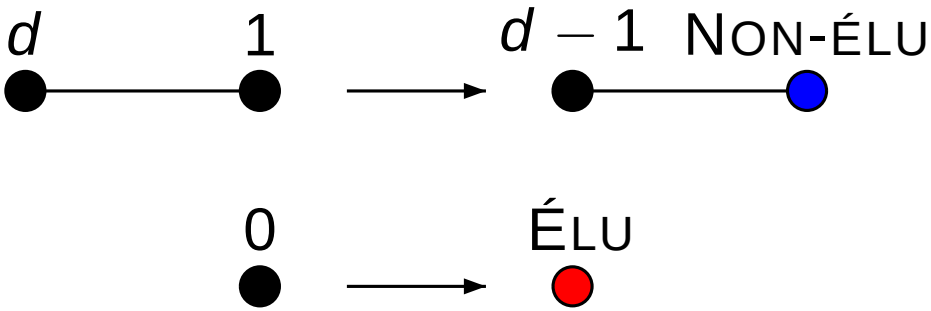
Un premier exemple

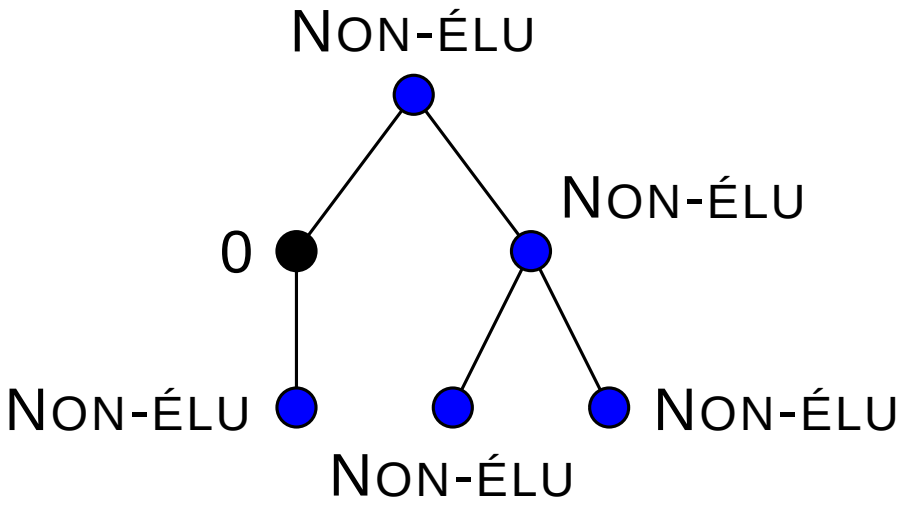


Un premier exemple

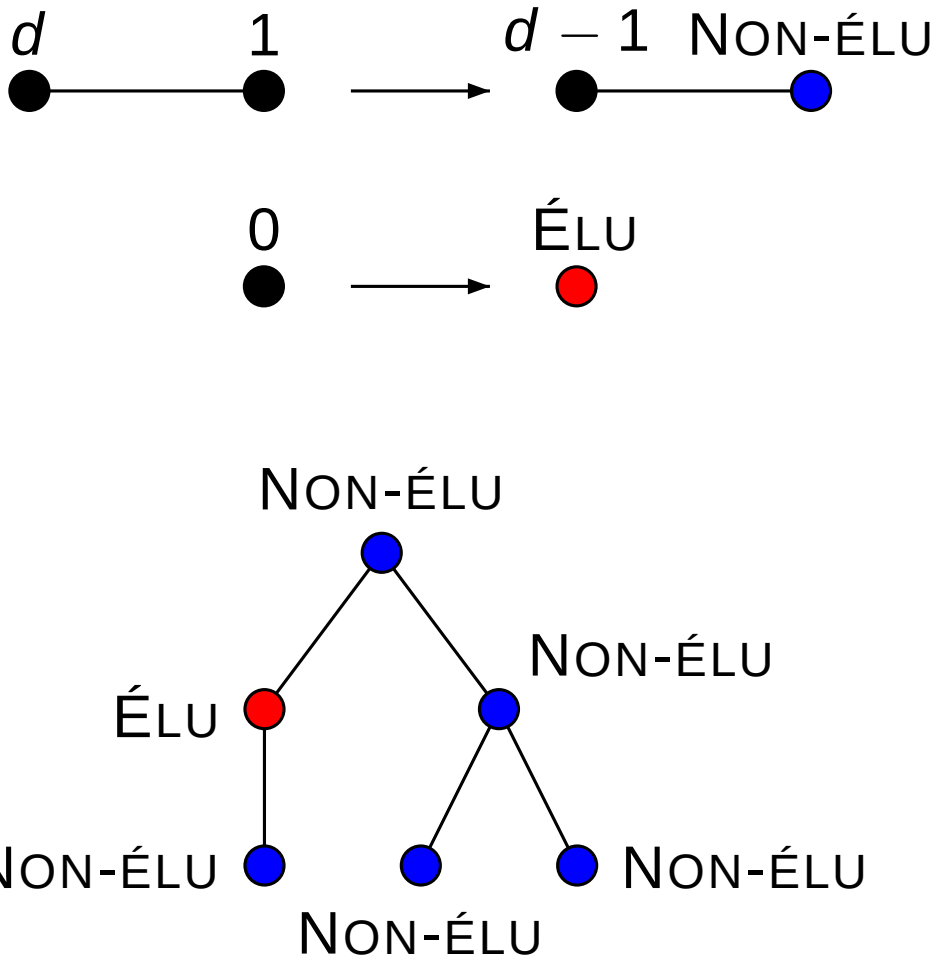


Un premier exemple

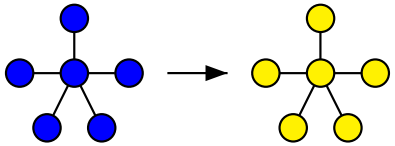




Un premier exemple

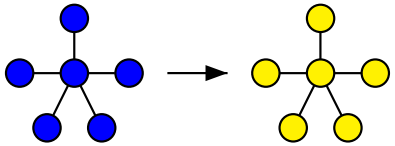


Modèles Considérés

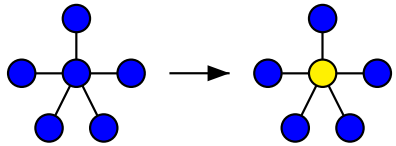


Mazurkiewicz ’97

Modèles Considérés

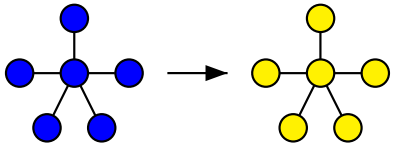


Mazurkiewicz ’97

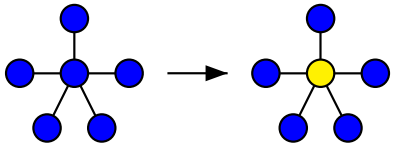


Boldi *et al.* ’96

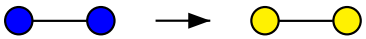
Modèles Considérés



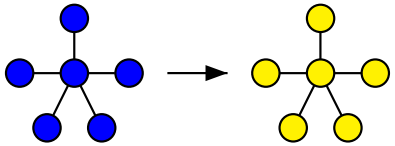
Mazurkiewicz ’97



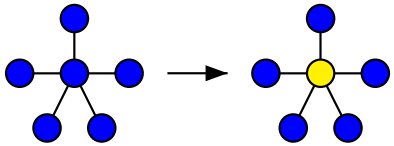
Boldi *et al.* ’96



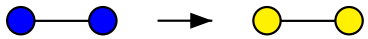
Modèles Considérés

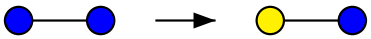


Mazurkiewicz ’97

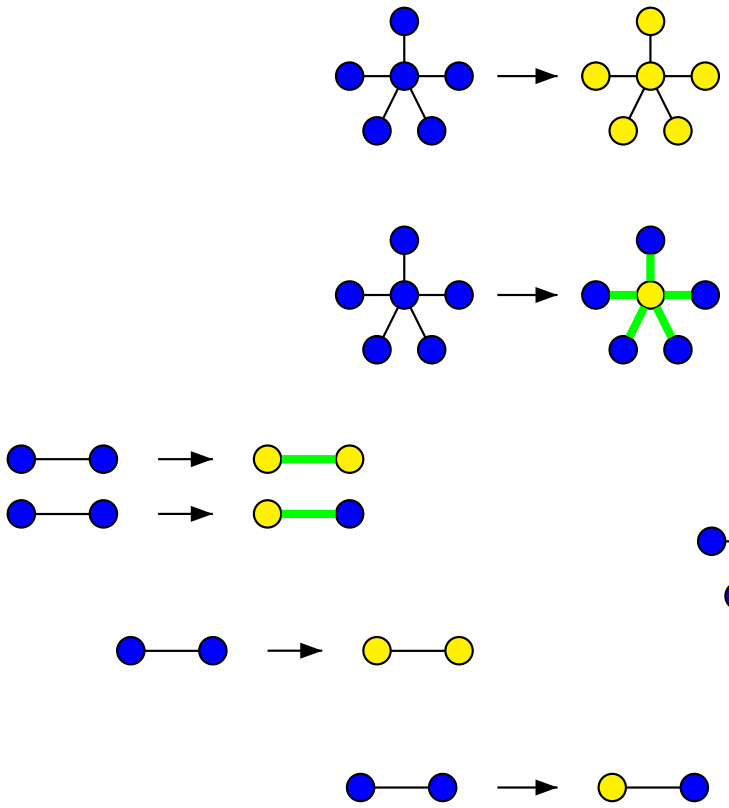


Boldi *et al.* ’96

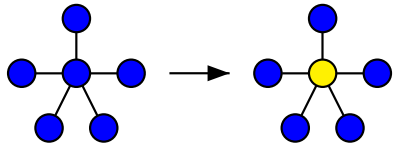




Modèles Considérés



Mazurkiewicz ’97

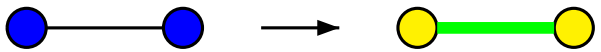


Boldi *et al.* ’96

Exemple des calculs locaux sur les arêtes étiquetées

Un pas de calcul effectué sur une arête $\{u, v\}$

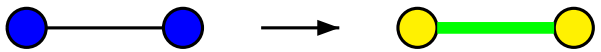
- modifie les étiquettes de u, de v et de $\{u, v\}$,
- ne dépend que des étiquettes de u, de v et de $\{u, v\}$.



Exemple des calculs locaux sur les arêtes étiquetées

Un pas de calcul effectué sur une arête $\{u, v\}$

- modifie les étiquettes de u, de v et de $\{u, v\}$,
- ne dépend que des étiquettes de u, de v et de $\{u, v\}$.

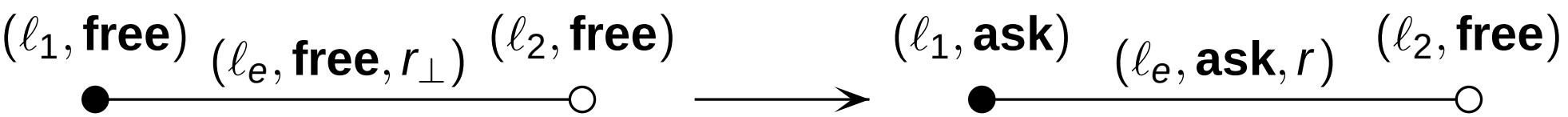


Théorème (Chalopin-Métivier '04)

Idées de la preuve

Soit $\mathcal{R}$ le système que l'on veut simuler.
Soit r une règle de $\mathcal{R}$.
À r on associe :

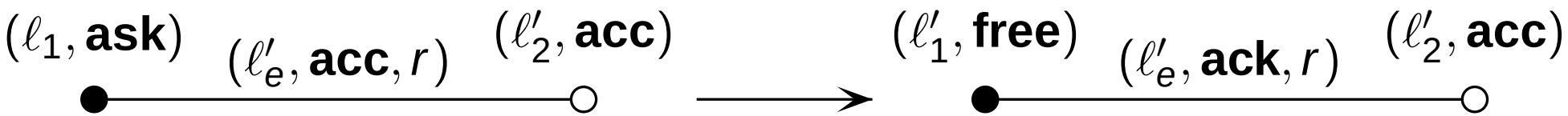
$\mathcal{S}_1(r)$:



If r=$(\ell_1, \ell_e, \ell_2, \ell'_1, \ell'_e, \ell'_2) \in \mathcal{R}$.

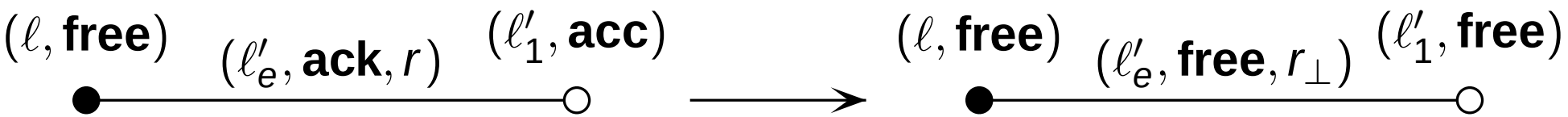
If $r = (\ell_1, \ell_e, \ell_2, \ell'_1, \ell'_e, \ell'_2)$.

$\mathcal{S}_3(r)$:



If $r = (\ell_1, \ell_e, \ell_2, \ell'_1, \ell'_e, \ell'_2)$.

$\mathcal{S}_4(r)$:



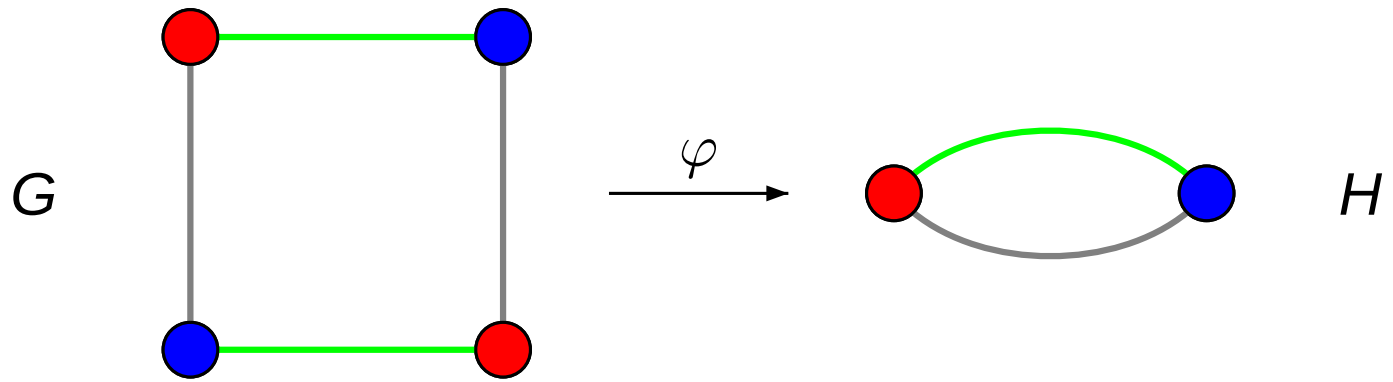
If $r = (\ell_1, \ell_e, \ell_2, \ell'_1, \ell'_e, \ell'_2)$.

If $r = (\ell_1, \ell_e, \ell_2, \ell'_1, \ell'_e, \ell'_2)$ and $\ell \neq \ell_2$.

L'outil combinatoire : Revêtement

Définition

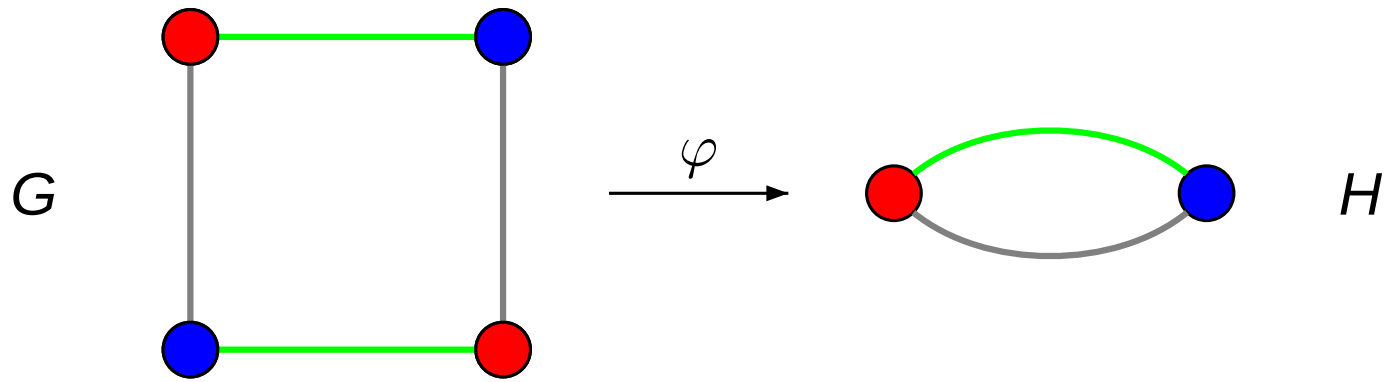
- G est un *revêtement* de H à travers φ si φ est un homomorphisme surjectif et localement bijectif.
- G est *minimal pour les revêtements* s'il n'existe pas $H \not\simeq G$ tel que G est un revêtement de H.



Revêtements

Proposition (Reidemeister '32)

Si G est un revêtement de H à travers φ,
$\forall v \in V(H), \forall f \in E(H), |\varphi^{-1}(v)| = |\varphi^{-1}(f)|$.



Revêtements

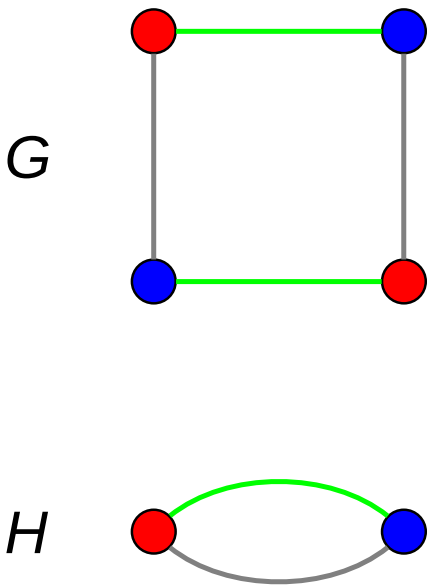
Proposition (Reidemeister '32)

Si G est un revêtement de H à travers φ,
$\forall v \in V(H), \forall f \in E(H), |\varphi^{-1}(v)| = |\varphi^{-1}(f)|$.

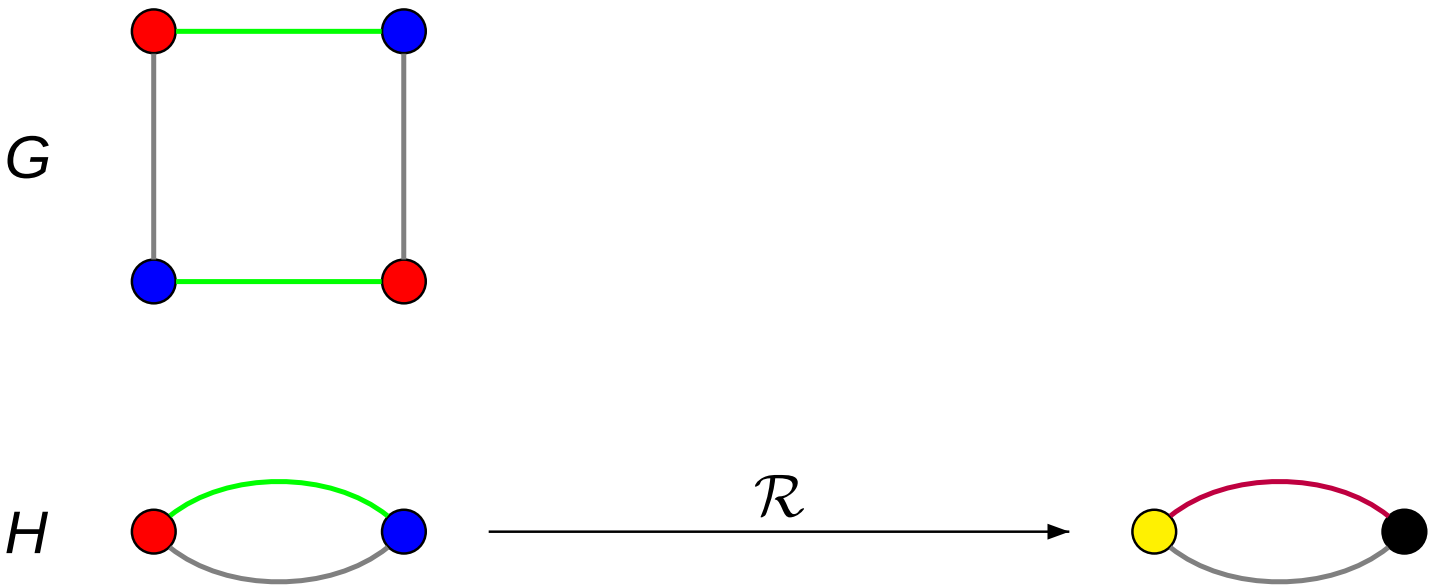
Exemples de graphes minimaux

- tout graphe G tel que $|E(G)| \wedge |V(G)| = 1$,
- les arbres,
- les anneaux de taille première.

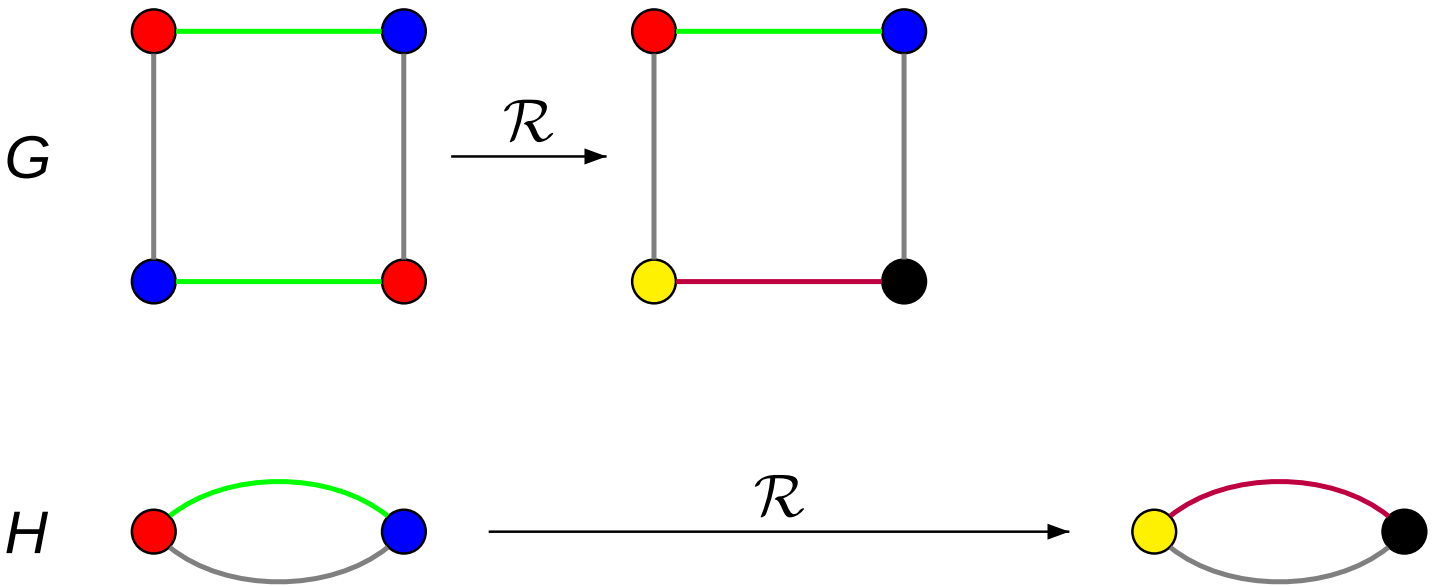
Lemme de relèvement (Angluin ’80)



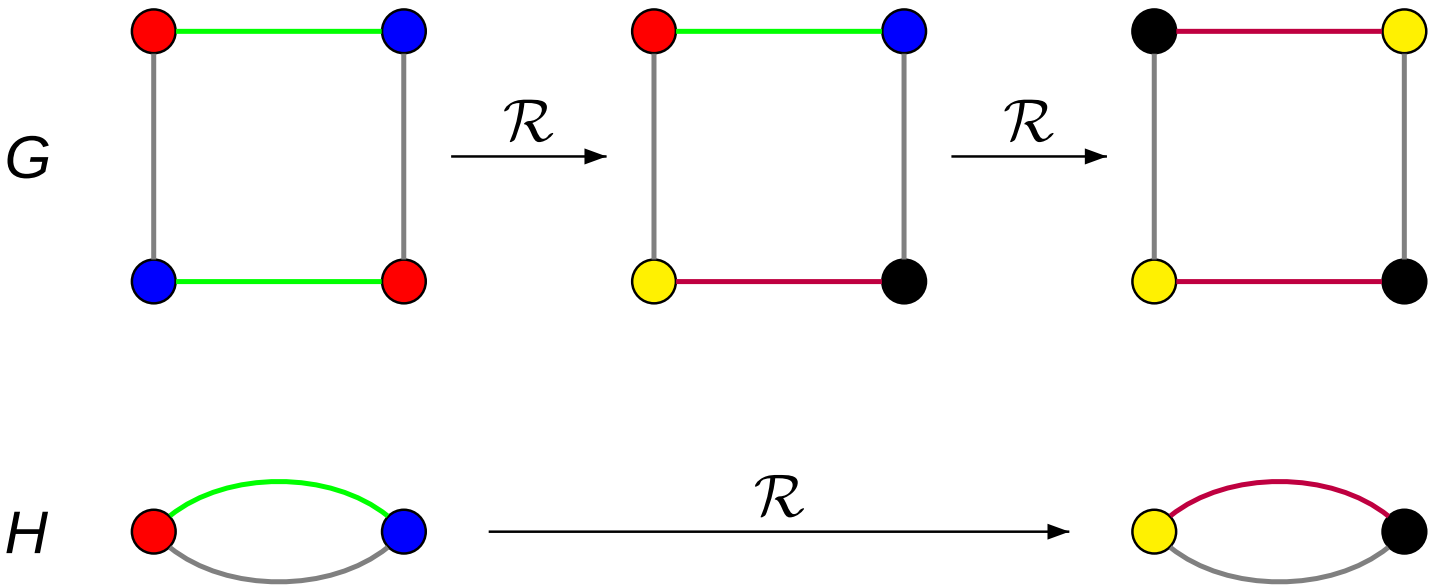
Lemme de relèvement (Angluin ’80)



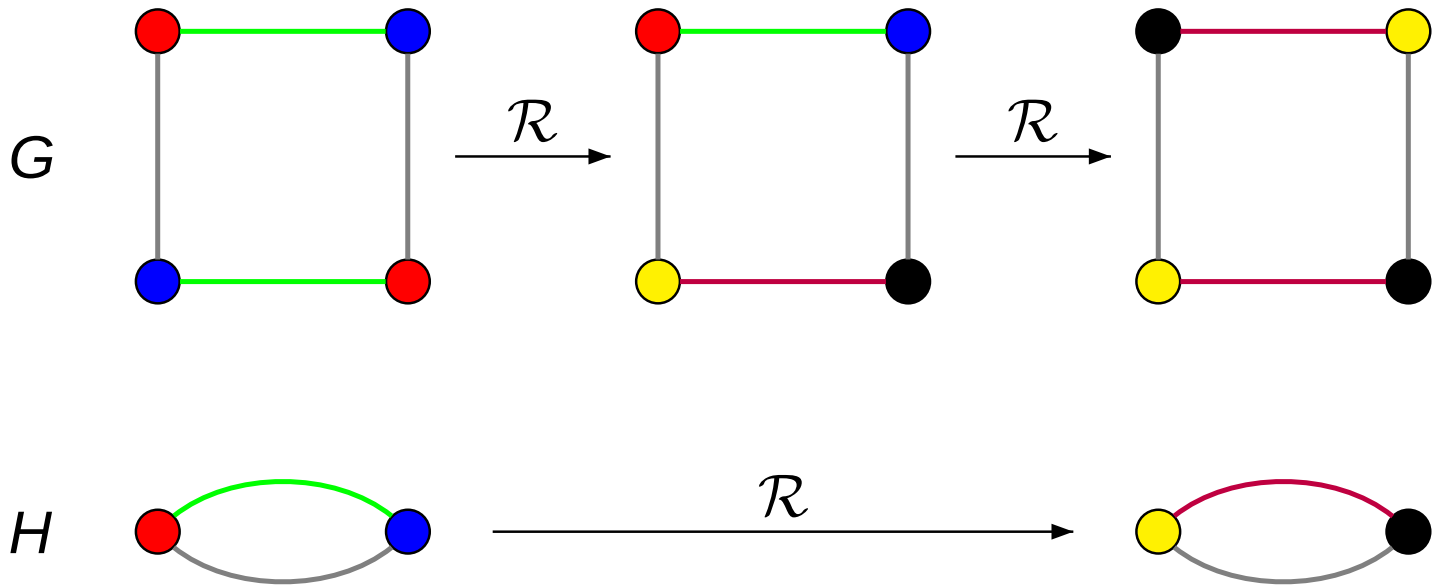
Lemme de relèvement (Angluin ’80)



Lemme de relèvement (Angluin ’80)



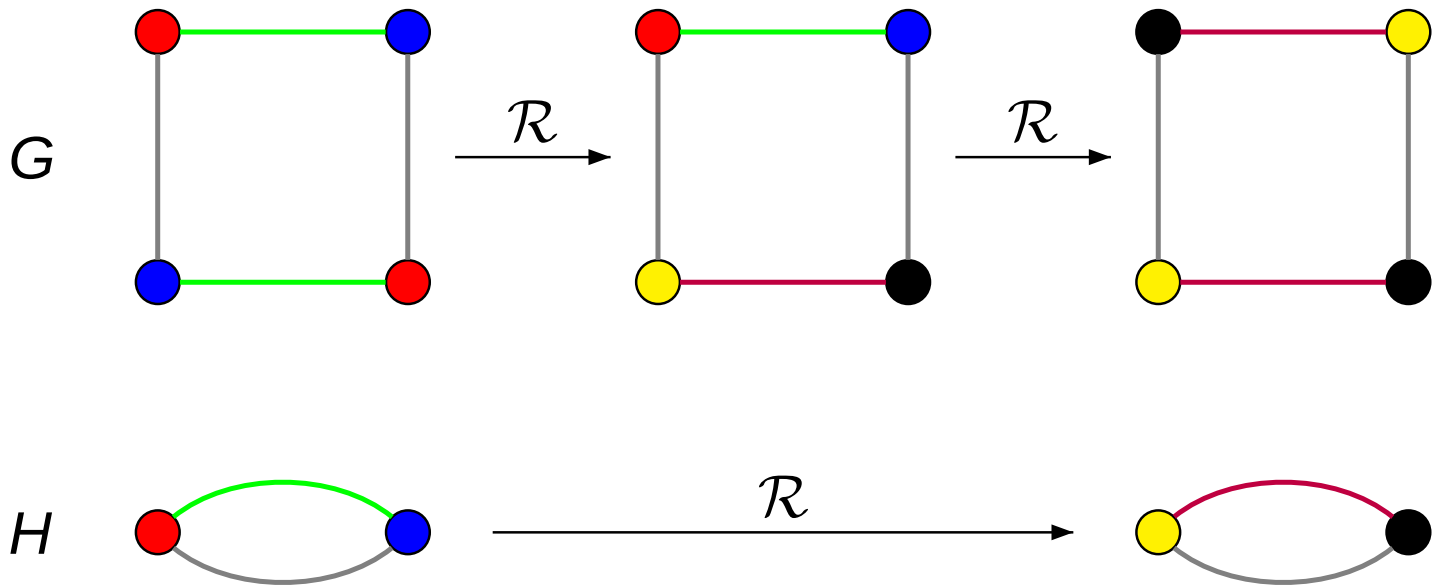
Lemme de relèvement (Angluin ’80)



Lemme (Angluin ’80)

Dans les modèles  *et*  , *un graphe G n’admet pas d’algorithme d’élection (ou de nommage) si G n’est pas minimal pour les revêtements.*

Lemme de relèvement (Angluin ’80)

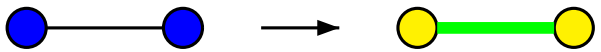


Théorème (Chalopin-Métivier ’04)

Dans les modèles  *et*  , *un graphe G admet un algorithme d’élection (et de nommage) si et seulement si G est minimal pour les revêtements.*

Un pas de calcul effectué sur une arête $\{u, v\}$

- modifie les étiquettes de u, de v et de $\{u, v\}$,
- ne dépend que des étiquettes de u, de v et de $\{u, v\}$.

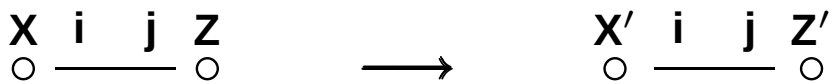


Plus précisément une règle est de la forme :

$$R : \mathbf{X} \circ\!\!-\!\!\!-\!\!\circ \mathbf{Y} \; \mathbf{Z} \longrightarrow \mathbf{X'} \; \mathbf{Y'} \circ\!\!-\!\!\!-\!\!\circ \mathbf{Z'}$$

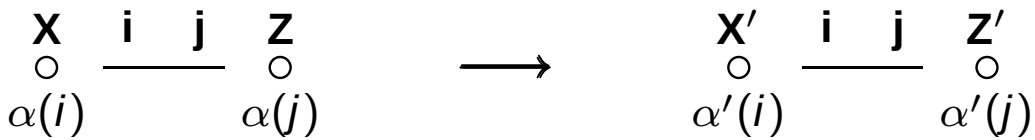
où $X' = f_1(X, Y, Z)$, $Y' = f_2(X, Y, Z) = f_2(Z, Y, X)$, $Z' = f_3(Z, Y, X)$, f_1, f_2 and f_3 sont des fonctions de transition sur des triplets d'états.

Un pas de calcul dans le modèle d'Angluin est du type :

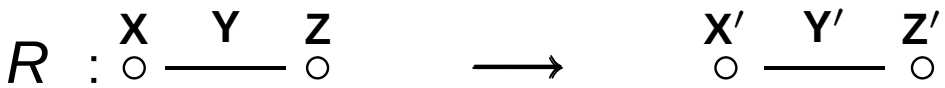


où $X' = f(X, Z, i, 0)$, $Z' = f(Z, X, j, 1)$, i et j sont les numéros de port et f est la fonction de transition.

The port numbers of the edges incident to a node may be used by the process located at that node to memorize the labels (states) associated to the corresponding incident edges. This fact implies that relabelling on edges with port numbering can simulate local computations on labelled edges, i.e., ;



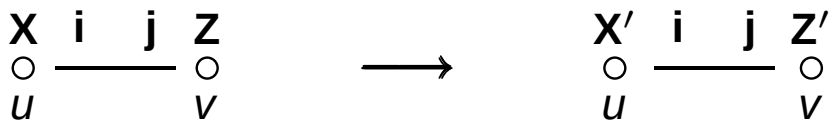
simulates :



with $\alpha(i) = \alpha(j) = Y$ and $\alpha'(i) = \alpha'(j) = Y'$.

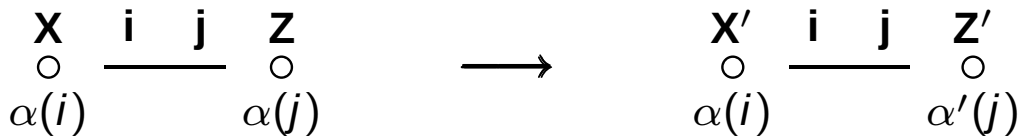
If we consider an asynchronous system with a port numbering and synchronous message passing, events may be encode by rules having the following form :

R_{Synch} :



where $X' = f_1(X, i), Z' = f_2(Z, X, j)$, i and j are the corresponding port numbers and f_1, f_2 are transition functions.

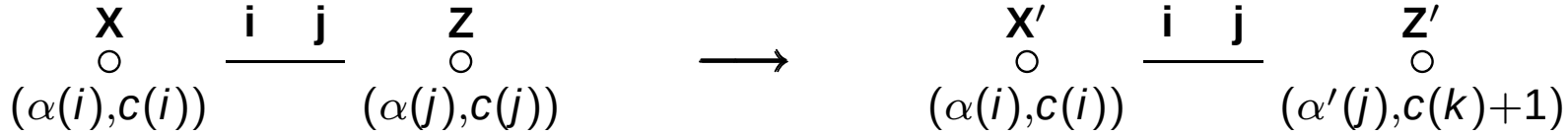
Consider the following notation :



(where $\alpha(i)$ and $\alpha(j)$ are states associated to the edge through ports i and j) the state up to date of the edge e is $\alpha'(j)$.

To each $\alpha(i)$ is associated a timestamp (counter) $c(i)$.

For the rule R_{Synch}, let k such that $c(k) = Max\{c(i), c(j)\}$, before the application of R_{Synch} the state of the edge is $\alpha(k)$. After the application of R_{Synch} the new state of e is $\alpha'(j)$ and its counter is $c(k) + 1$:

R_{Synch} : 

In fact we can solve this problem with only 3 values $\{0, 1, 2\}$ by applying the following method : if for a endvertex the counter is 0 and it is 1 for the other endvertex then the maximal counter is 1 ; if it is 1 and 2 then the maximal counter is 2 ; if it is 0 and 2 then the maximal counter is 0. If the state up to date of the edge (the state of the edge associated to the counter having the maximal value) before the application of the transformation is on the receiver vertex then the new counter associated to the new state of the edge is equal to the old ; if not, to obtain the value of the new counter associated with the new state of the edge, we increment modulo 3 the maximal counter before the transformation.

One can wonder whether the knowledge of the vertex-degrees makes local computations on labelled edges more powerful.

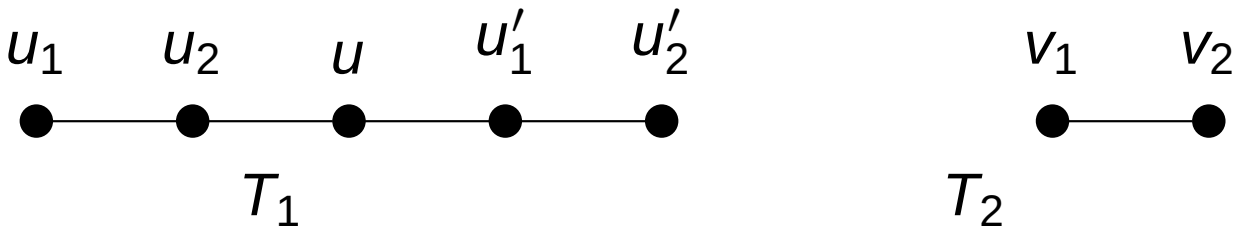


FIG.: The graphs T_1 and T_2 are minimal, nevertheless it does not exist local computations on labelled edges without the knowledge of the vertex-degrees which solves the election problem on the family $\{T_1, T_2\}$.

From the election algorithm given at the beginning :

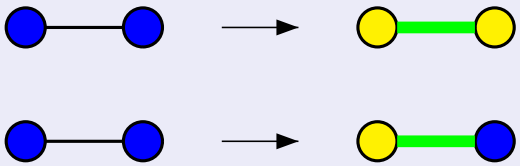
Lemme

There exists an election algorithm defined by local computations on labelled edges for the family of graphs having the knowledge of vertex-degrees.

Importance de la connaissance du degré

Théorème (Chalopin-Métivier ’04)

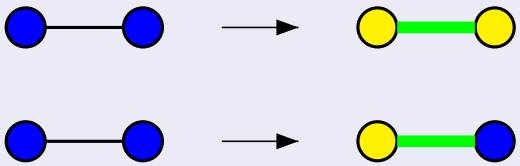
Les cinq modèles suivants sont équivalents :

-  *avec connaissance du degré,*
- *avec connaissance du degré,*

Importance de la connaissance du degré

Théorème (Chalopin-Métivier '04)

Les cinq modèles suivants sont équivalents :

-  *avec connaissance du degré,*
- *avec connaissance du degré,*
- *le modèle étudié par Angluin,*

Importance de la connaissance du degré

Théorème (Chalopin-Métivier '04)

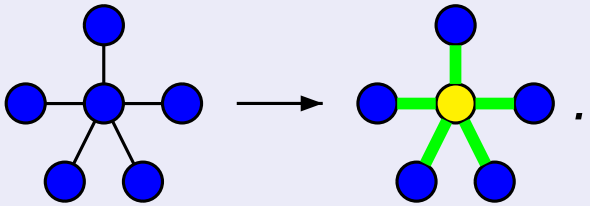
Les cinq modèles suivants sont équivalents :

-  *avec connaissance du degré,*
-  *avec connaissance du degré,*
- *le modèle étudié par Angluin,*
- *le modèle de communication synchrone,*

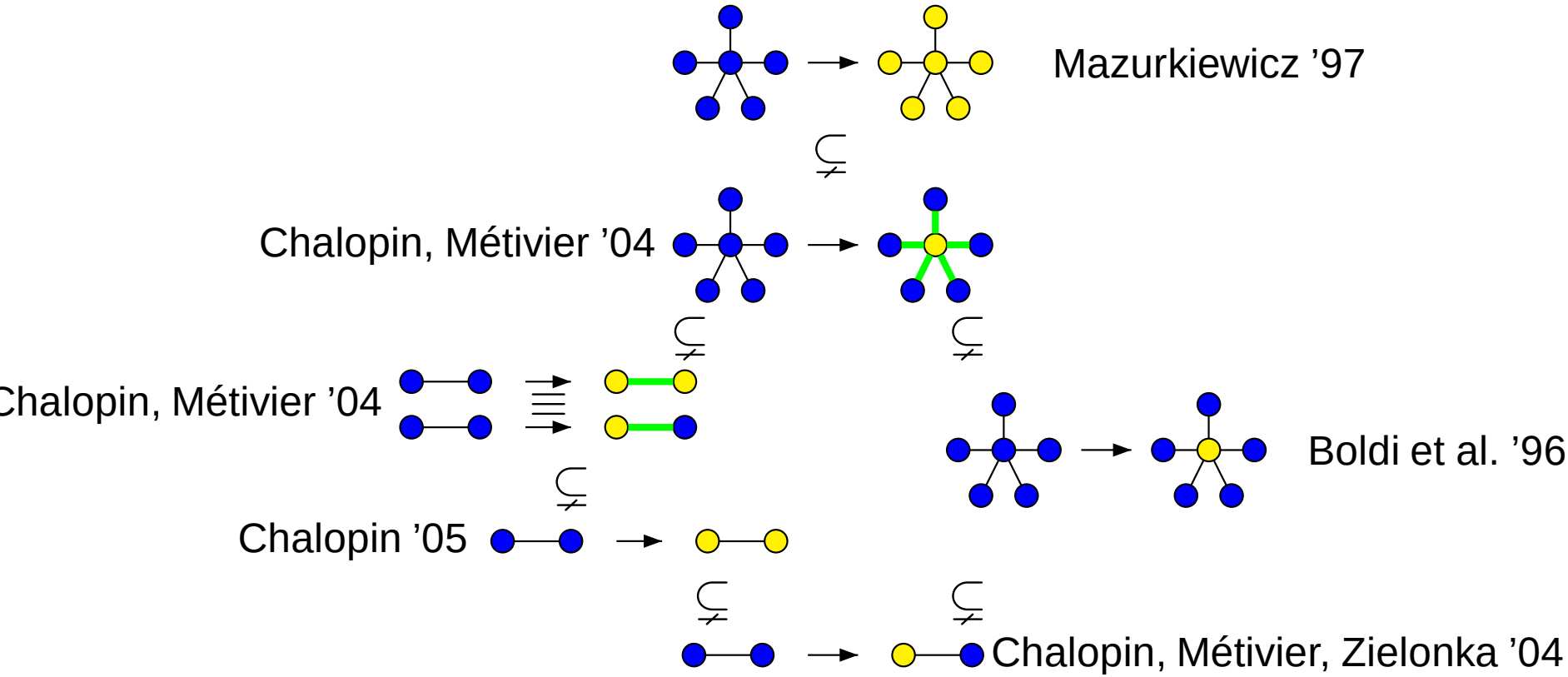
Importance de la connaissance du degré

Théorème (Chalopin-Métivier '04)

Les cinq modèles suivants sont équivalents :

-  *avec connaissance du degré,*
-  *avec connaissance du degré,*
- *le modèle étudié par Angluin,*
- *le modèle de communication synchrone,*
-  .

Hiérarchie

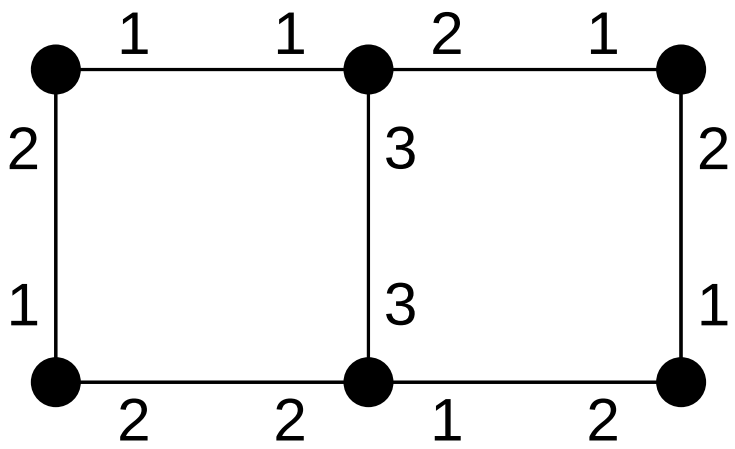

Mazurkiewicz ’97
Chalopin, Métivier ’04
Chalopin, Métivier ’04
Boldi et al. ’96
Chalopin ’05
Chalopin, Métivier, Zielonka ’04

Plan

Modèle

Un réseau est modélisé par un graphe G avec un étiquetage des ports δ. En un pas de calcul, chaque processus peut

- modifier son état,
- ou envoyer un message à travers un port p,
- ou recevoir un message à travers un port q.

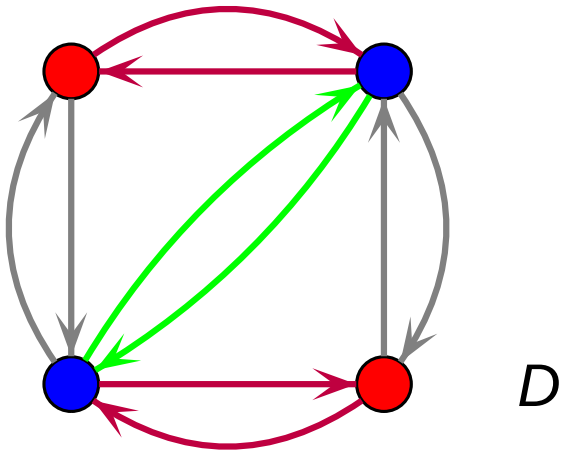


(G, δ)

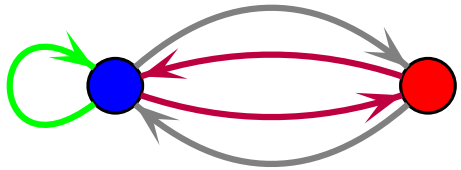
Revêtements dirigés symétriques

Définition

- D est un *revêtement dirigé symétrique* de D' à travers φ si φ est un homomorphisme surjectif et localement bijectif qui préserve la symétrie des arcs.
- D est *minimal pour les revêtements dirigés symétriques* s'il n'existe pas $D' \not\simeq D$ tel que D est un revêtement dirigé symétrique de D'.



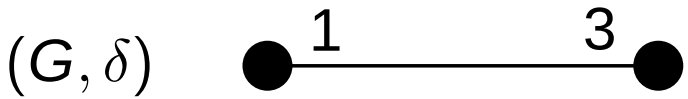
D

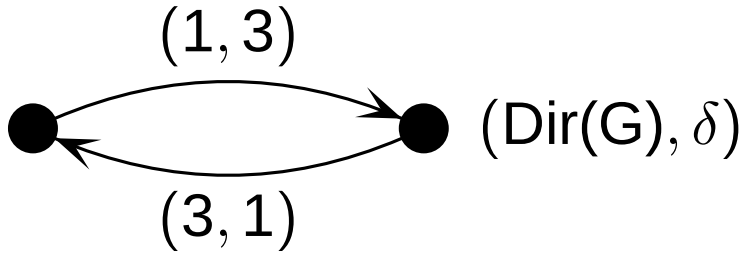


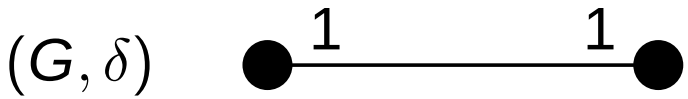
D'

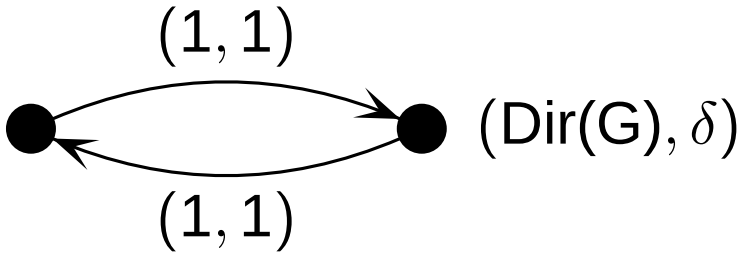
Codage du réseau

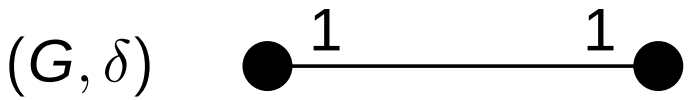
Codage du réseau

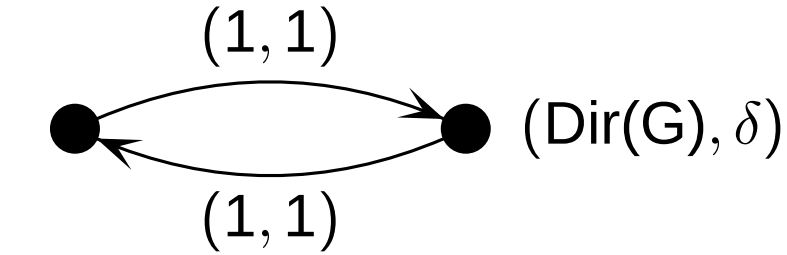


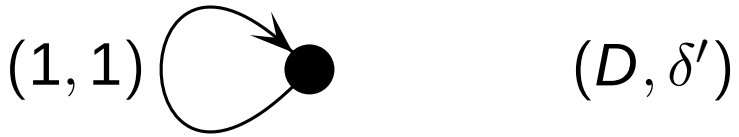



(G, δ)
1
1


(1, 1)
(Dir(G), δ)
(1, 1)


(G, δ)
1
1


(1, 1)
(Dir(G), δ)
(1, 1)


(1, 1)
(D, δ')

Caractérisation

Théorème (Chalopin, Métivier '05)

Un réseau (G, δ) *admet un algorithme d'élection ou de nommage*

$$\Longleftrightarrow$$

$(Dir(G), \delta)$ *est minimal pour les revêtements dirigés symétriques.*

Caractérisation

Théorème (Chalopin, Métivier '05)

Un réseau (G, δ) *admet un algorithme d'élection ou de nommage*

$\Longleftrightarrow$

$(Dir(G), \delta)$ *est minimal pour les revêtements dirigés symétriques.*

Théorème (Chalopin, Métivier '05)

Un graphe G admet un algorithme d'élection ou de nommage

$\Longleftrightarrow$

Dir(G) est minimal pour les revêtements dirigés symétriques.

Propriétés de l'algorithme

- Cet algorithme est totalement asynchrone.
- Il peut tolérer un déséquencement des messages dans les canaux de communication.

Complexité

Sur un graphe G avec n sommets, m arêtes dont le diamètre est D et dont le degré maximum est Δ.

	Notre algo.	algo. YK
Temps :	$O(Dn^2)$	$O(n)$
Nombre total de messages :	$O(m^2 n)$	$O(mn)$
Taille des messages :	$O(\Delta \log n)$	$\Delta^{O(n)}$
Mémoire des processus :	$O(\Delta n \log n)$	$\Delta^{O(n)}$

Un algorithme d'énumération

But : Donner à chaque sommet de G un numéro unique de telle sorte qu'il y ait une bijection entre $V(G)$ et $[1, |V(G)|]$.

Un algorithme d'énumération

But : Donner à chaque sommet de G un numéro unique de telle sorte qu'il y ait une bijection entre $V(G)$ et $[1, |V(G)|]$.

- Chaque sommet va choisir un numéro.

Un algorithme d'énumération

But : Donner à chaque sommet de G un numéro unique de telle sorte qu'il y ait une bijection entre $V(G)$ et $[1, |V(G)|]$.

- Chaque sommet va choisir un numéro.
- Il va échanger son numéro avec ses voisins et il va construire sa « vue locale » : $\{(n_1, p_1, q_1), \ldots, (n_d, p_d, q_d)\}$.
- Il va diffuser son numéro accompagné de sa « vue locale ».

Un algorithme d'énumération

But : Donner à chaque sommet de G un numéro unique de telle sorte qu'il y ait une bijection entre $V(G)$ et $[1, |V(G)|]$.

- Chaque sommet va choisir un numéro.
- Il va échanger son numéro avec ses voisins et il va construire sa « vue locale » : $\{(n_1, p_1, q_1), \ldots, (n_d, p_d, q_d)\}$.
- Il va diffuser son numéro accompagné de sa « vue locale ».
- Si un sommet remarque l'existence d'un autre sommet avec le même numéro que lui :
  - il compare sa « vue locale » avec celle de son « adversaire »,
  - s'il a la « vue locale » la plus faible, il change de numéro et le diffuse à nouveau.

Étiquettes

L'étiquette d'un sommet v est un triplet $(n(v), N(v), M(v))$,

Étiquettes

L'étiquette d'un sommet v est un triplet $(n(v), N(v), M(v))$,

- $n(v) \in \mathbb{N}$ est le numéro courant du sommet.

Étiquettes

L'étiquette d'un sommet v est un triplet $(n(v), N(v), M(v))$,

- $n(v) \in \mathbb{N}$ est le numéro courant du sommet.
- $N(v) = \{(n_1, p_1, q_1), \ldots, (n_d, p_d, q_d)\}$ contient les informations dont v dispose à propos de ses voisins.

Étiquettes

L'étiquette d'un sommet v est un triplet $(n(v), N(v), M(v))$,

- $n(v) \in \mathbb{N}$ est le numéro courant du sommet.
- $N(v) = \{(n_1, p_1, q_1), \ldots, (n_d, p_d, q_d)\}$ contient les informations dont v dispose à propos de ses voisins.
- $M(v) = \{(n_1, N_1), \ldots, (n_p, N_p)\}$ est la boîte aux lettres de v, i.e., les informations dont il dispose sur les autres sommets du graphe.

Étiquettes

L'étiquette d'un sommet v est un triplet $(n(v), N(v), M(v))$,

- $n(v) \in \mathbb{N}$ est le numéro courant du sommet.
- $N(v) = \{(n_1, p_1, q_1), \ldots, (n_d, p_d, q_d)\}$ contient les informations dont v dispose à propos de ses voisins.
- $M(v) = \{(n_1, N_1), \ldots, (n_p, N_p)\}$ est la boîte aux lettres de v, i.e., les informations dont il dispose sur les autres sommets du graphe.

Au départ, chaque sommet est étiqueté $(0, \emptyset, \emptyset)$.

Comment comparer les « vues locales » ?

$$N_1 \prec N_2 \iff \max_{<_{lex}}(N_1 \Delta N_2) \in N_2$$

$$N_1 = \{ \quad (9,1,4), \quad (7,2,6), \quad (1,4,2), \quad (1,3,1) \quad \}$$

$$N_2 = \{ \quad (9,1,4), \quad (7,3,2), \quad (6,2,1) \quad \}$$

Changer de numéro

si $n(v) = 0$ *ou* **alors**

Changer de numéro

si $n(v) = 0$ *ou* $\exists (n(v), N') \in M(v)$ *tel que* $N(v) \prec N'$ **alors**

Changer de numéro

si $n(v) = 0$ *ou* $\exists (n(v), N') \in M(v)$ *tel que* $N(v) \prec N'$ **alors**

$n_{old} := n(v)$;

Changer de numéro

si $n(v) = 0$ *ou* $\exists (n(v), N') \in M(v)$ *tel que* $N(v) \prec N'$ **alors**

$n_{old} := n(v)$;

$n(v) := 1 + \max\{n \in \mathbb{N} \mid \exists (n, N) \in M(v)\}$;

Changer de numéro

si $n(v) = 0$ *ou* $\exists (n(v), N') \in M(v)$ *tel que* $N(v) \prec N'$ **alors**

$n_{old} := n(v)$;
$n(v) := 1 + \max\{n \in \mathbb{N} \mid \exists (n, N) \in M(v)\}$;
$M(v) := M(v) \cup \{(n(v), N(v))\}$;

Changer de numéro

si $n(v) = 0$ *ou* $\exists (n(v), N') \in M(v)$ *tel que* $N(v) \prec N'$ **alors**

$n_{old} := n(v)$;
$n(v) := 1 + \max\{n \in \mathbb{N} \mid \exists (n, N) \in M(v)\}$;
$M(v) := M(v) \cup \{(n(v), N(v))\}$;
pour $q := 1$ ***à*** $\deg(v)$ **faire**
  envoyer $\langle (n(v), n_{old}, M(v)), q \rangle$ par le port q ;

Mettre à jour sa connaissance du réseau

si ***mes*** $= \langle (n', n'_{old}, M'), p \rangle$ *est arrivé par le port* q **alors**

$M_{old} := M(v)$;

Mettre à jour sa connaissance du réseau

si ***mes*** $= \langle (n', n'_{old}, M'), p \rangle$ *est arrivé par le port* q **alors**

$M_{old} := M(v)$;

$N(v) := N(v) \setminus \{(n'_{old}, p, q)\} \cup \{(n', p, q)\}$;

Mettre à jour sa connaissance du réseau

si ***mes*** $= \langle (n', n'_{old}, M'), p \rangle$ *est arrivé par le port* q **alors**

$M_{old} := M(v)$;
$N(v) := N(v) \setminus \{(n'_{old}, p, q)\} \cup \{(n', p, q)\}$;
$M(v) := M(v) \cup M' \cup \{(n(v), N(v))\}$;

Mettre à jour sa connaissance du réseau

si ***mes*** $= \langle (n', n'_{old}, M'), p \rangle$ *est arrivé par le port* q **alors**

$M_{old} := M(v)$;
$N(v) := N(v) \setminus \{(n'_{old}, p, q)\} \cup \{(n', p, q)\}$;
$M(v) := M(v) \cup M' \cup \{(n(v), N(v))\}$;
si $M(v) \neq M_{old}$ **alors**
  pour $q := 1$ ***à*** $\deg(v)$ **faire**
    envoyer $\langle (n(v), n(v), M(v)), q \rangle$ par le port q ;

Étiquetage final

Lemme

Dans la configuration finale, pour tous sommets v, v' :

- $M(v) = M(v')$,

Étiquetage final

Lemme

Dans la configuration finale, pour tous sommets v, v' :

- $M(v) = M(v')$,
- $n(v) = n(v') \implies N(v) = N(v')$,

Étiquetage final

Lemme

Dans la configuration finale, pour tous sommets v, v' :

- $M(v) = M(v')$,
- $n(v) = n(v') \implies N(v) = N(v')$,
- $(m, p, q) \in N(v) \iff \exists \{v, w\} \in E(G)$ *telle que*
  - $n(w) = m$,
  - $\delta_v(w) = q$ *et* $\delta_w(v) = p$.

Étiquetage final

Lemme

Dans la configuration finale, pour tous sommets v, v' :

- $M(v) = M(v')$,
- $n(v) = n(v') \implies N(v) = N(v')$,
- $(m, p, q) \in N(v) \iff \exists \{v, w\} \in E(G)$ *telle que*
  - $n(w) = m$,
  - $\delta_v(w) = q$ *et* $\delta_w(v) = p$.

Proposition

L'étiquetage final permet de construire (D, δ') *tel que* $(Dir(G), \delta)$ *est un revêtement dirigé symétrique de* (D, δ').

Travaux en cours

Terminaison implicite : l'algorithme est terminé mais aucun sommet ne peut le détecter.
Terminaison explicite : l'algorithme est terminé et un sommet peut le détecter.
Transformation d'algorithmes distribués avec terminaison implicite en algorithmes distribués avec terminaison explicite.
Idem pour la terminaison locale : situations où sommet peut détecter si son état est final.

Publications

- Calculs locaux
  - *Election and local computations on edges*
    J. Chalopin, Y. Métivier (*FoSSaCS'04*)
  - *Election, naming and cellular edge local computations*
    J. Chalopin, Y. Métivier, W. Zielonka (*ICGT'04*)
  - *Local computations on closed unlabelled edges : the election problem and the naming problem*
    J. Chalopin (*SOFSEM'05*)
  - *Graphs labelings derived from models in distributed computing*
    J. Chalopin, D. Paulusma (*WG'06*)
  - *Local computations in graphs : the case of cellular edge local computations*
    J. Chalopin, Y. Métivier, W. Zielonka (*Fundamenta Informaticae'06*)

Publications

- Échanges de messages
  - *A bridge between the asynchronous message passing model and local computations in graphs*
    J. Chalopin, Y. Métivier (*MFCS'05*)
  - *Groupings and pairings in anonymous networks*
    J. Chalopin, S. Das, N. Santoro (*DISC'06*)
  - *About the termination detection in the asynchronous message passing model*
    J. Chalopin, E. Godard, Y. Métivier, G. Tel (*SOFSEM'07*)
- Agents mobiles
  - *Election in the qualitative world*
    J. Chalopin (*SIROCCO'06*)
  - *Mobile agent algorithms versus message passing algorithms*
    J. Chalopin, E. Godard, Y. Métivier, R. Ossamy (*OPODIS'06*)

Plan