

Université de Montréal

Symétrie et sens de l'orientation dans les graphes

par

Mohamad Awwad

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

novembre 1999

©Mohamad Awwad, 1999



QA

76

U54

2000

v.014

Université de Montréal

Document de référence de l'orientation des programmes

Document de référence

Document de référence de l'orientation des programmes

Document de référence de l'orientation des programmes

Document de référence de l'orientation des programmes

Document de référence de l'orientation des programmes

Document de référence de l'orientation des programmes

Document de référence de l'orientation des programmes



Document de référence de l'orientation des programmes

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

Symétrie et sens de l'orientation dans les graphes

présenté par:

Mohamad Awwad

a été évalué par un jury composé des personnes suivantes:

Patrice Marcotte

(président-rapporteur)

Geña Hahn

(directeur de recherche)

Paola Flocchini

(co-directrice)

Pierre McKenzie

(membre du jury)

Mémoire accepté le:

2000-02-18

Sommaire

Ce travail est une étude de la symétrie et du sens de l'orientation dans les graphes. Nous étudions différentes classes de graphes ayant une symétrie de sommets ou d'arêtes. Par exemple, nous considérons les graphes *sommet transitifs*, *arête transitifs*, *sommet uniformes*, *arête uniformes* et *chemin uniformes*. Nous explorons les relations entre ces classes en traçant une hiérarchie qui indique le degré de symétrie que ces graphes possèdent. Nous définissons le sens de l'orientation et nous exposons plusieurs de ses propriétés en faisant les liens avec les graphes symétriques et, en particulier, avec les graphes de Cayley.

Table des matières

Sommaire	iii
Liste des tableaux	vi
Liste des figures	viii
Remerciements	ix
1 Introduction	1
1.1 Un modèle de système distribué	1
1.2 L'apport du sens de l'orientation	2
1.3 L'apport de la symétrie	3
1.4 Notre contribution	5
1.5 Plan du mémoire	6
2 Définitions générales	8
3 Symétrie dans les graphes	12
3.1 Symétrie : définitions et relations	13
3.2 Graphes chemin uniformes (PR)	22
3.3 Indépendance de PR des autres propriétés	28
4 Symétrie dans les graphes étiquetés et graphes de Cayley	32
4.1 Notations et définitions de base	33

4.2	Symétrie de vue	34
4.3	Symétrie de voisinage	39
4.4	Graphes de Cayley	40
4.5	Relations entre différentes classes de symétrie	43
4.6	Importance de la symétrie	46
5	Sens de l'orientation	49
5.1	Définitions, exemples et propriétés	50
5.1.1	Définitions	51
5.1.2	Exemple	54
5.1.3	Propriétés	56
5.2	Application du sens de l'orientation	62
5.2.1	Influence sur la complexité	62
5.2.2	Influence sur la Calculabilité	66
6	Méthode pour décider le sens de l'orientation faible	71
6.1	Décision du sens de l'orientation faible	72
6.1.1	La méthode	72
6.1.2	L'algorithme	75
6.2	Décision des graphes de Cayley	77
6.2.1	L'algorithme	78
6.3	Remarques sur la construction d'un sens de l'orientation faible	78
	Conclusion	80
	Bibliographie	87

Liste des tableaux

5.1	Complexité des problèmes dans les graphes arbitraires.	64
5.2	Complexité des problèmes dans les graphes complets	65
5.3	Complexité des problèmes dans les hypercubes.	65

Liste des figures

3.1	Le graphe de Peterson est sommet transitif.	14
3.2	Une permutation de sommet dans le graphe de Peterson.	15
3.3	Un graphe arête transitif mais pas sommet transitif.	16
3.4	Un graphe sommet transitif mais pas arête transitif.	16
3.5	L'hypercube est distance transitif.	17
3.6	Relations entre différentes classes de symétrie.	19
3.7	Un graphe régulier sommet uniforme mais pas sommet transitif.	21
3.8	Autres relations entre différentes classes de symétrie.	22
3.9	Un graphe VPR.	24
3.10	Un graphe EPR.	24
3.11	Un graphe SPR.	25
3.12	Un graphe VPR et EPR mais pas SPR.	26
3.13	Relations entre certaines symétries et les graphes PR.	27
3.14	Quelques relations avec les graphes PR.	30
4.1	Une arête portant deux étiquettes.	32
4.2	Un graphe étiqueté et la 2-vue, $T_2(u)$, du sommet u	35
4.3	Un graphe V -symétrique et la 2-vue, $T_2(u)$ pour tout sommet $u \in V$	37
4.4	ST_3 et BS_3 générés par G_1 et G_2	42
4.5	MB_3 généré par G_3	43
4.6	Un graphe $\{1, 2\}$ -factorisable qui n'est pas de Cayley.	44
4.7	Résumé de relations entre différentes classes de symétrie.	47

5.1	Un graphe avec un sens de l'orientation.	51
5.2	Un graphe qui n'a pas de fonction de codage cohérente pour cet étiquetage.	52
5.3	Un graphe avec un sens de l'orientation faible.	53
5.4	Un graphe avec un sens de l'orientation cordal.	55
5.5	Un graphe qui n'est pas sommet uniforme, alors n'a pas de SOM.	58
5.6	Un graphe avec un sens de l'orientation minimal et son étiquetage n'est pas de Cayley.	60

Remerciements

Je remercie Geña Hahn, Paola Flocchini, Eve Marie Gingras, Fadia, Imad, Adel et Majed de m'avoir tous supporté d'une manière ou d'une autre.

à ceux qui m'ont appris à aimer...
à mes parents...

Chapitre 1

Introduction

1.1 Un modèle de système distribué

Un *système distribué* est une structure logicielle et matérielle distribuée en un réseau *point à point* d'entités communicantes. Ce système contient des entités (sites) et un ensemble de liens entre ces entités (ligne de communication). Chaque entité possède une mémoire, une horloge locale et une capacité de calcul. Une entité peut être, par exemple, un *processus* ou un *processeur*. Dans ce dernier cas, on parle d'un *système parallèle*. La mémoire locale est, typiquement, non partagée par les sites du système. Étant donné ce non partage des mémoires, nous considérons que la communication entre les entités du système se fait *uniquement* par l'échange (ou le passage) de messages. Une entité voulant communiquer avec une autre, envoie un message sur un lien, s'il existe, reliant ces deux entités. Ce message, une fois arrivé dans une file d'attente, peut être "lu" par l'entité destinataire. Chaque entité est dans un état quelconque. À l'arrivée d'un événement (par exemple, réception d'un message), une action se produit. Le *comportement* d'une entité est l'ensemble des règles décrivant l'action à suivre à partir d'un état et d'un événement particuliers.

Un *algorithme distribué* est, tout simplement, un ensemble des comportements de toutes les entités du système.

Dans ce travail, nous faisons référence au système distribué décrit ci-dessus. Nous considérons que la complexité d'un algorithme distribué est calculée en fonction de nombre de messages échangés entre les entités lors de l'exécution de cet algorithme. Ceci est tout simplement dû au fait que le temps que prennent les échanges de messages lors de l'exécution d'un algorithme distribué est dominant.

Il n'est pas difficile de voir que nous pouvons modéliser un système distribué par un graphe dont les sommets représentent les entités, et les arêtes les liens de communication entre les entités. Lorsque le lien de communication possède un nom (ce qui est souvent le cas) on parle d'un graphe étiqueté où l'étiquette sur une arête représente le nom du lien correspondant. Dans tout ce qui suit, nous considérons -sauf mention contraire- les termes *entités*, *processeurs*, *sites* et *sommets* comme équivalents, de même ceux de *lignes*, *liens* et *arête*, ainsi que ceux de *système*, *réseau* et *graphe*.

1.2 L'apport du sens de l'orientation

Nous avons, généralement, intérêt à écrire et à exécuter des algorithmes d'une manière efficace. Trouver un algorithme efficace pour résoudre un problème devrait être un objectif important en informatique. Mais lorsque nous n'arrivons pas à réduire la complexité du meilleur algorithme distribué existant résolvant un problème quelconque, que peut-on faire ?

Avant de poursuivre, regardons une notion récente mais importante dans le domaine des systèmes distribués : le *sens de l'orientation*. Le sens de l'orientation est un étiquetage particulier des graphes. Cet étiquetage permet à chaque sommet du graphe de *distinguer* non seulement ses voisins mais tous les sommets du graphe les uns des autres. Nous pouvons dire informellement que dans un graphe avec sens de l'orientation, chaque sommet peut savoir si deux chemins étiquetés mènent au même sommet ou non. En exploitant cette capacité, une entité qui

vient d'envoyer un message à une autre pourrait éviter un autre envoi inutile du même message lors de l'exécution d'un algorithme quelconque distribué. Prenons le problème suivant : l'*élection d'un leader* dans un système réparti consiste à choisir, parmi un ensemble d'entités candidates, une seule entité (qu'on appelle *leader*) qui sera dans un état particulier. Toutes les autres entités, qui connaîtront leur nouveau leader, seront dans un autre état.

Le meilleur algorithme distribué connu résolvant ce problème dans un graphe complet et étiqueté arbitrairement a une complexité (nombre de messages communiqués) dans $\Theta(n \log n)$ où n est le nombre de sommets du graphe. Si ce graphe a un sens de l'orientation, le même problème peut être résolu avec une complexité de $\Theta(n)$. Cette réduction de complexité est, décidément due à la manière dont on a évité l'envoi inutile des messages.

L'existence d'un sens de l'orientation dans les graphes peut *dans certains cas* réduire la complexité des algorithmes distribués. De plus, son existence pourrait augmenter *la puissance de calcul* dans un système distribué ; c'est à dire augmenter le nombre de problèmes résolubles dans un tel système. Nous explorons cet aspect plus loin dans le texte.

Ceci est une des possibilités d'améliorer l'efficacité des algorithmes et par conséquent, celle du système. L'exploration de cette possibilité pourrait être une réponse à la question posée ci-dessus mais elle ne garantit pas toujours une amélioration de l'efficacité. Ce fait nous amène à explorer une autre piste : la symétrie dans les graphes.

1.3 L'apport de la symétrie

La notion de symétrie est caractérisée, comme le nom l'indique, par une *régularité* de sommets ou d'arêtes. Par exemple, lorsqu'on parle d'une symétrie quelconque de sommets, cela veut dire qu'il existe une propriété vérifiée par tous les sommets du graphe. Prenons un graphe *sommet transitif* ; informellement, la

symétrie dans ce graphe réside dans le fait que tous ses sommets peuvent “voir” leur entourage exactement de la même manière. Il existe des problèmes distribués que nous pouvons résoudre efficacement en choisissant une “bonne” topologie de système (de graphe). Cela revient à modéliser le système par un graphe possédant une symétrie ou une autre. Étant donné qu’il existe différentes classes de symétrie, le choix d’un graphe appartenant à une classe ou une autre pour modéliser un système dépend souvent du type des problèmes à résoudre. À titre d’exemple, les graphes sommet transitifs représentent des bonnes topologies à la construction des chemins dans un réseau pour transmettre des messages. On appelle cette construction le problème de *routage*.

Rappelons que la symétrie et le sens de l’orientation peuvent être liés. Par exemple, les graphes possédant un sens de l’orientation minimal (utilisant un nombre d’étiquettes égal au degré du graphe), correspondent parfois à une classe de graphes hautement symétriques qu’on appelle les *graphes de Cayley*.

Nous venons de citer deux possibilités à l’amélioration de l’efficacité dans un système distribué : l’existence d’un sens de l’orientation dans le système et la modélisation du système par un graphe symétrique. Bien que, comme nous avons vu, l’exploration de ces deux notions peut être “rentable” du point de vue algorithmique, une précision s’impose : Si un problème est résoluble dans un système distribué avec un étiquetage arbitraire, il n’est pas garanti qu’il soit résoluble avec sens de l’orientation d’une manière plus efficace.

De même, il n’est pas garanti qu’un problème résoluble dans un réseau arbitraire soit résoluble dans un réseau symétrique d’une manière plus efficace. Par contre, nous connaissons des problèmes (comme c’est décrit plus haut) qui sont plus efficacement résolubles dans un réseau avec sens de l’orientation ou dans un réseau possédant une symétrie quelconque. Cette réalité constitue, pour nous, une motivation suffisante pour étudier ces deux notions. Ce travail est globalement consacré à une étude des classes de symétrie et du sens de l’orientation dans les graphes.

1.4 Notre contribution

Notre contribution dans ce travail consiste tout d’abord à présenter une synthèse dans laquelle différentes classes de symétrie sont formellement définies et les relations entre elles sont systématiquement étudiées. De plus, présenter le sens de l’orientation ainsi qu’étudier ses propriétés et les liens entre ces dernières et ceux de la symétrie dans les graphes constituent une partie majeure de ce travail.

Nous avons prêté une attention particulière aux définitions, elles devraient être complètes et surtout précises. Autrement dit, nous avons essayé de “nettoyer” plusieurs définitions données dans la littérature afin de les rendre précises. De plus, nous présentons autrement certaines définitions existantes. À titre d’exemple, nous définissons, d’une nouvelle manière, la *symétrie de vue*, la *symétrie de voisinage* et le *sens de l’orientation cordal*. Nous verrons ces définitions plus loin dans le texte.

Étant données les classes de graphes suivantes : *chemin uniforme*(PR), *sommet uniforme* et *sommet transitif* (expliquées en détails plus loin) et en se basant sur quelques résultats connexes, nous avons pu *prouver* que la classe de graphes chemin uniforme est indépendante des deux autres classes. Ainsi, à partir de certains résultats existants, nous avons *déduit* les relations entre les graphes qui ont une symétrie de voisinage et les autres classes de graphes symétriques. À titre d’exemple, nous prouvons que le degré de symétrie dans les graphes S_1 -*symétrique* est strictement plus élevé que celui des graphes *vue symétrique*. Par ailleurs, nous soumettons la *conjecture* que tout graphe sommet transitif peut être étiqueté de manière à ce qu’il devienne vue-symétrique.

Enfin, du point de vue d’application, nous avons *implanté* un algorithme (déjà existant) qui détecte l’existence (ou l’inexistence) d’un sens de l’orientation (faible) dans un graphe ; cet algorithme détecte aussi, dans certains cas, si

un graphe donné est un graphe de Cayley avec un étiquetage de Cayley.

Il faudrait mentionner que le travail de traduction de certains termes existants en anglais n'a pas toujours été facile. L'utilisation de ces termes est relativement récente dans le domaine et visiblement inexistante dans les textes français. Il nous fallait trouver des nouveaux termes représentatifs et précis. Les termes les plus importants qu'on a traduits sont : sens de l'orientation (sense of direction), sommet uniforme (vertex cycle regular ou cycle symmetry), arête uniforme (cycle regular), graphes chemin uniformes (path regular graphs), symétrie de vue (view symmetric) et symétrie de voisinage (surrounding symmetric). Remarquons que notre choix du mot "uniforme" au lieu du mot "régulier" nous a évité une confusion avec le terme "graphe régulier".

1.5 Plan du mémoire

Ce mémoire est conçu de la manière suivante : le chapitre 2 contient des définitions générales sur la théorie des graphes. Ces définitions peuvent servir comme références pour les chapitres qui suivent. Au troisième chapitre, nous définissons formellement différentes classes de symétrie, les relations entre elles, et à prouver l'indépendance entre la classe PR et les autres classes.

Au chapitre 4, nous étendons cette étude (de symétrie) aux graphes étiquetés. Dans ce chapitre, nous définissons formellement la symétrie de vue et la symétrie de voisinage afin d'établir des liens entre celles-ci et les classes de symétrie vues au chapitre 3. Il ne faut pas oublier qu'au chapitre 4, nous mettons aussi en relation les graphes de Cayley et les autres propriétés étudiées.

Le chapitre 5 sera consacré entièrement à l'étude du sens de l'orientation. Des définitions, exemples et propriétés sont d'abord présentés, et on termine ce chapitre en donnant un aperçu des applications du sens de l'orientation; fondamentalement ces applications portent sur la complexité et la calculabilité des

problèmes dans les systèmes distribués.

Le chapitre 6 nous permettra de présenter un algorithme (que nous avons implanté) qui décide le sens de l'orientation faible et, dans certains cas, les graphes de Cayley avec étiquetage de Cayley. Nous terminons ce travail par une conclusion où nous soulevons certaines questions connexes qui pourraient être le sujet d'un travail futur.

Nous donnons les références où nous pouvons trouver les preuves et les détails de tous les théorèmes, définitions (sauf les très générales) et autres hypothèses présentés dans le texte. Par contre, nous prouvons toujours tout nouvel énoncé qu'on propose.

Chapitre 2

Définitions générales

Si A est un ensemble, $P_i(A)$ désigne l'ensemble de tous les sous-ensembles de A de cardinalité i .

Définition 2.1 (Graphe)

Un graphe G est un couple $G = (V(G), E(G))$ où $V(G)$ est un ensemble et $E(G)$ un sous-ensemble de $P_2(V(G))$. Nous appellerons les éléments de $V(G)$ les sommets de G et ceux de $E(G)$, les arêtes de G .

Remarquons qu'avec la définition de graphe ci-dessus nous avons défini seulement les graphes simples et non orientés. Dans ce qui suit, nous remplacerons parfois $V(G)$ et $E(G)$ respectivement par V et E . Nous utilisons la notation (u, v) pour désigner l'arête qui relie les sommets u et v .

Définition 2.2 (Ordre de graphe)

L'ordre du graphe $G = (V, E)$ est $|V|$, la cardinalité de V .

Définition 2.3 (Graphe fini)

Un graphe fini est un graphe dont l'ensemble de sommets est fini.

Définition 2.4 Deux sommets u et v d'un graphe $G = (V, E)$ sont adjacents si et seulement s'il existe une arête (u, v) les reliant, c'est à dire si $(u, v) \in E$. L'arête (u, v) est dite incidente à u et à v .

Soit $G = (V, E)$ un graphe et soit $x \in V(G)$, nous utiliserons la notation suivante :

$$V_G(u) = \{v \in V(G) : (u, v) \in E(G)\}$$

et

$$E_G(u) = \{(u, v) \in E(G) : v \in V_G(u)\}$$

On désigne par $V_G(u)$ l'ensemble des sommets adjacents à u et $E_G(u)$ les arêtes incidentes à u . On appelle $V_G(u)$ le *voisinage* de u (dans G).

Définition 2.5 (*Degré et graphe régulier*)

Le degré d'un sommet $u \in V(G)$ dans un graphe $G = (V, E)$ est $d_G(u) = |E_G(u)| = |V_G(u)|$. Si $d_G(u) = d_G(v)$ pour tout $u, v \in V$, on appelle G un graphe régulier de degré $d = d_G(u)$. Le degré d'un tel graphe est noté par $d(G)$.

Définition 2.6 (*Parcours*)

Un parcours dans $G = (V, E)$ est une séquence de sommets telle que toute paire de sommets consécutifs forme une arête.

Définition 2.7 (*Chemin*)

Un chemin de longueur k dans un graphe G est une séquence $v_0v_1 \cdots v_k$ de sommets distincts où (v_i, v_{i+1}) est une arête dans G pour tout $0 \leq i \leq k-1$. La longueur d'un chemin est le nombre d'arêtes sur le chemin, k .

Les sommets v_0 et v_k sont les *extrémités* du chemin $v_0v_1 \cdots v_k$ et les autres sommets sont des *sommets intérieurs*. Remarquons que tout chemin de longueur supérieure ou égale à 1 contient une paire d'extrémités distinctes. Le chemin v_0 de longueur zéro contient une seule extrémité v_0 , et contient la "paire" d'extrémités non distinctes v_0, v_0 . Nous utilisons la notation $C_{(u,v)}$ pour désigner un chemin entre u et v et nous notons $|C_{(u,v)}|$ sa longueur.

La différence entre un chemin et un parcours est que ce dernier peut être une séquence de sommets *non distincts*.

Définition 2.8 (*Plus court chemin*)

Un chemin $C_{(u,v)}$ est un plus court chemin si pour tout autre chemin $C'_{(u,v)}$, $|C_{(u,v)}| \leq |C'_{(u,v)}|$.

Définition 2.9 (*Cycle*)

Un cycle de longueur k dans un graphe G est une séquence $v_0v_1 \cdots v_{k-1}$ de sommets distincts où $(v_i, v_{i+1(\text{mod } k)})$ est une arête dans G pour tout $0 \leq i \leq k-1$. La longueur d'un cycle est le nombre d'arêtes sur le cycle, k .

Définition 2.10 (*Grphe connexe*)

Un graphe $G = (V, E)$ est connexe si et seulement si $\forall u, v \in V(G)$, il existe un chemin $C_{(u,v)}$ dans G reliant u et v .

Définition 2.11 (*Distance*)

La distance $d(u, v)$ entre deux sommets u et v d'un graphe est la longueur d'un plus court chemin reliant u et v .

Définition 2.12 (*Diamètre*)

Le diamètre D d'un graphe $G = (V, E)$ est défini comme suit : $D = \max\{d(u, v), \forall u, v \in V\}$.

Définition 2.13 (*Maille*)

La maille d'un graphe G est la longueur d'un plus court cycle non trivial dans G .

Définition 2.14 (*Nombre chromatique*)

Un k -coloriage d'un graphe G est une assignation de k couleurs, $1, 2, \dots, k$ aux sommets de G . Ce coloriage est dit propre s'il n'existe pas deux sommets adjacents ayant la même couleur. Le graphe G est dit k -coloriable s'il existe un k -coloriage propre de G . Le nombre chromatique de G , $\chi(G)$, est défini de la manière suivante : $\chi(G) = \min\{k \in \mathbb{N} \mid G \text{ est } k\text{-coloriable}\}$.

Définition 2.15 (*Connectivité par sommet*)

La connectivité par sommet dans un graphe G est le nombre minimal, k , tel

que en enlevant k sommets de G , le graphe résultant devient non connexe. La robustesse du graphe G est égale à $k - 1$.

Définition 2.16 (*Connectivité par arête*)

La connectivité par arête dans un graphe G est le nombre minimal, k , tel que en enlevant k arêtes de G , le graphe résultant devient non connexe.

Définition 2.17 (*Grappe biparti*)

Un graphe $G = (V, E)$ est biparti s'il admet une partition de V en deux sous-ensembles V_1 et V_2 tels que toute arête de E a une de ces deux extrémités dans V_1 et l'autre extrémité dans V_2 . Si $|V_1| = m$ et $|V_2| = n$ et $\forall u \in V_1, v \in V_2, (u, v) \in E$, le graphe G sera noté par $K_{m,n}$.

Définition 2.18 (*Grappe complet*)

Un graphe simple G est complet si pour tout $u, v \in V$ on a $(u, v) \in E$. Si l'ordre de G est n , ce graphe sera noté par K_n .

Définition 2.19 (*{1, 2}-factorisable*)

Étant donné un graphe $G = (V, E)$, un graphe couvrant de G est un graphe $H = (V, F)$ tel que $F \subseteq E$. Un p -facteur de G est un sous-graphe couvrant p -régulier de G . Le graphe G est $\{1, 2\}$ -factorisable s'il existe un ensemble de 1- ou 2- facteurs F_1, \dots, F_k ($F_i = (V, E_i)$) tels que E_1, \dots, E_k constitue une partition de E ; c'est à dire les arêtes dans ces facteurs sont disjointes et $G = F_1 \cup F_2 \cup \dots \cup F_k$.

Dans ce travail, nous utilisons des graphes finis, simples, connexes et non orientés.

Chapitre 3

Symétrie dans les graphes

Le mot *symétrie*, d'origine grecque (de *sun* "avec", et *metron* "mesure"), veut dire "juste proportion". Ce mot est défini dans le Petit Robert comme une "régularité et harmonie dans les parties d'un objet ou dans la disposition d'objets semblables", ou encore comme une "correspondance exacte en forme, taille et position de parties opposées" . *

La symétrie dans les graphes peut être vue comme une régularité dans laquelle les sommets sont connectés. En général, on parle d'une symétrie parmi les sommets ou parmi les arêtes. Celle-ci peut être décrite comme une propriété qui donne aux sommets ou aux arêtes un certain "pouvoir" égal. Ce pouvoir peut être différent d'une symétrie à l'autre et différentes classes de symétrie existent. En informatique, nous avons grand intérêt à étudier ces différentes classes et à comprendre les "vertus" qui peuvent en découler. Nous savons déjà qu'un système distribué (ou réparti) peut être modélisé par un graphe. Nous cherchons à construire des systèmes (ou à relier ses entités) d'une manière répondant à une ou plusieurs exigences; par exemple nous voulons parfois construire un *système robuste* qui continue à fonctionner même si plusieurs de ses liens sont en panne. D'autres fois, nous exigeons un système dans lequel nous pouvons utiliser efficacement des

* "Il y a en architecture, comme en musique, des rythmes d'une symétrie harmonieuse qui charment l'oeil et l'oreille" (Gautier).

algorithmes distribués de routage. Dans ce chapitre, nous allons voir plusieurs classes de graphes dont deux en particulier : l'une qu'on appelle *sommet transitif* où il y a une symétrie de sommets et l'autre qu'on appelle *arête transitif* où la symétrie, cette fois, est liée aux arêtes. À titre d'exemple, les graphes sommet transitifs sont (comme c'est déjà mentionné) des modèles idéaux quand il s'agit d'utiliser des algorithmes distribués de routage, quant à ceux arêtes transitifs, ils garantissent un degré important de *robustesse*.

Nous explorons aussi dans ce chapitre des résultats qui décrivent les relations entre ces classes. Plus précisément, dans la deuxième section, nous étudions les graphes sommet transitifs, arête transitifs, arête uniformes, sommet uniformes, etc., et nous montrons à chaque fois "ce qui contient quoi", c'est à dire nous essayons d'établir une hiérarchie entre ces classes de symétrie. Dans la troisième section, nous exposons les définitions et quelques résultats existants dans la littérature sur une classe de graphe qu'on appelle PR vérifiant une propriété qu'on peut voir comme une symétrie de chemins dans le graphe. La quatrième section est une partie où nous *prouvons* que les graphes PR sous toutes leurs formes (VPR, EPR) sont complètement (ou presque) indépendants des graphes sommets transitifs et sommet uniformes.

3.1 Symétrie : définitions et relations

Définition 3.1 (*Automorphisme*)

Un automorphisme de graphe $G = (V, E)$ est une bijection $\vartheta : V \rightarrow V$ telle que $(\vartheta(u), \vartheta(v)) \in E$ si et seulement si $(u, v) \in E$.

Les automorphismes de G avec l'opération de composition forment un groupe, $Aut(G)$, qu'on appelle le *groupe d'automorphisme* de G .

Définition 3.2 (*Sommet transitif*)

Un graphe $G = (V, E)$ est sommet transitif (nous disons parfois sommet symétrique) si le groupe d'automorphisme de G , $Aut(G)$, est transitif sur V , c'est

à dire, si $\forall u, v \in V, \exists \alpha \in \text{Aut}(G)$ tel que $\alpha(u) = v$.

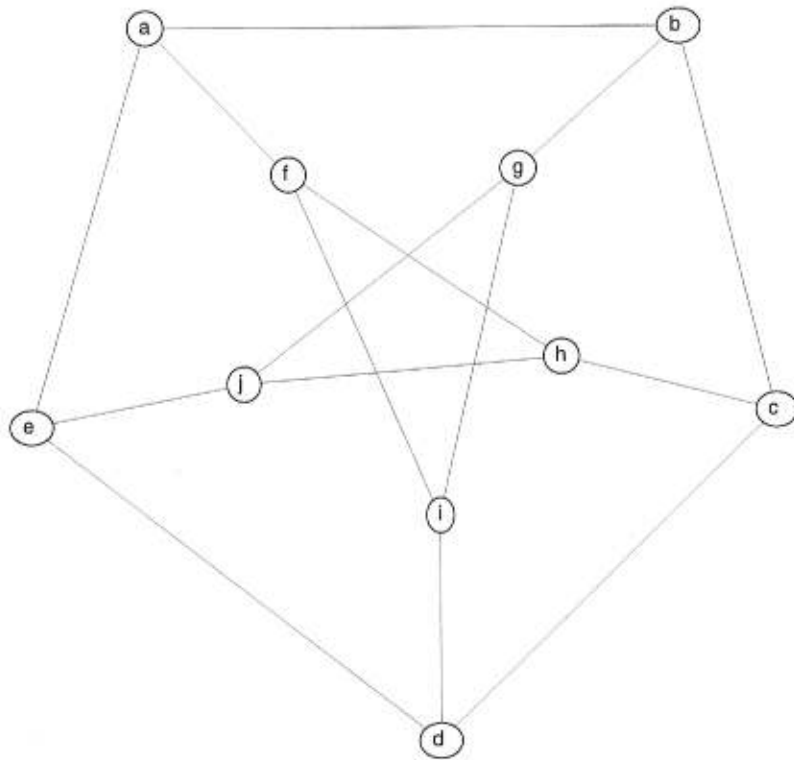


FIG. 3.1: Le graphe de Peterson est sommet transitif.

Par la définition même, nous concluons qu'un graphe sommet transitif est nécessairement régulier ; ceci est dû au fait qu'il n'existe pas d'automorphisme entre deux sommets de degrés différents qui préserve l'adjacence. Un exemple de graphe sommet transitif est donné par la figure 3.1 [4]. Nous pourrions voir ceci en considérant deux sous-ensembles de sommets dans ce graphe ; $S = \{a, b, c, d, e\}$ et $S' = \{f, g, h, i, j\}$. Si nous cherchons un automorphisme α tel que $\alpha(x) = y$ où $x, y \in S$ ou bien $x, y \in S'$, une simple rotation du graphe nous montre qu'un tel automorphisme existe. Par ailleurs, si on cherche un automorphisme α tel que $\alpha(x) = y$ où $x \in S'$ et $y \in S$, nous pouvons toujours le trouver en appliquant des simples rotations et en utilisant, sans perte de généralité, l'automorphisme α' (décrit par la figure 3.2) suivant : $\alpha'(a) = f, \alpha'(b) = i, \alpha'(c) = g, \alpha'(d) = j, \alpha'(e) = h, \alpha'(f) = a, \alpha'(g) = d, \alpha'(h) = b, \alpha'(i) = e, \alpha'(j) = c$.

Remarquons qu'avec α' , l'ensemble S sera à l'intérieur du graphe et S' sera à l'extérieur, ainsi un automorphisme comme $\alpha(x) = y$ où $x \in S$ et $y \in S'$ peut se faire d'une manière semblable.

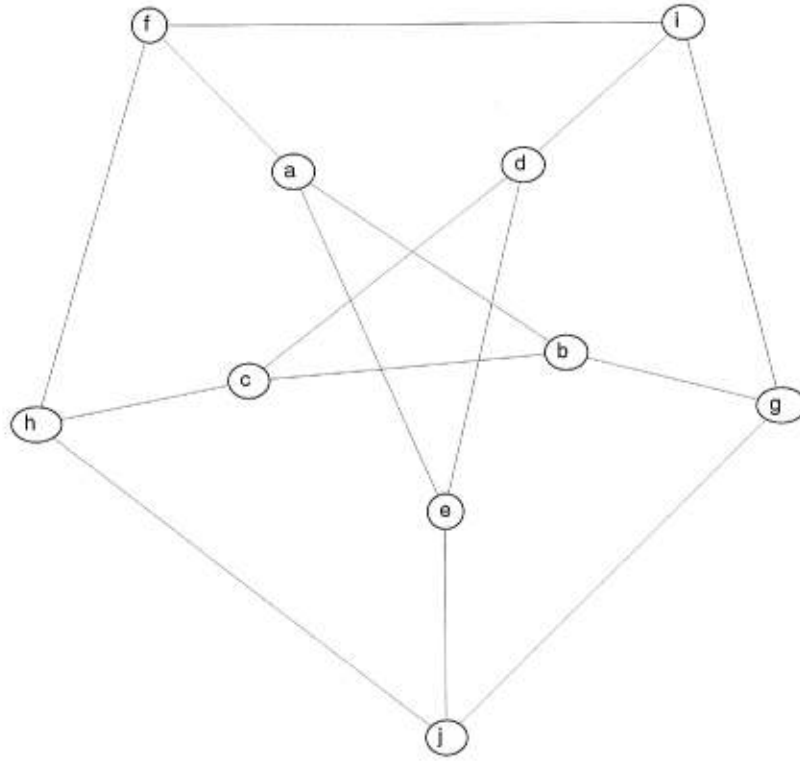


FIG. 3.2: Une permutation de sommet dans le graphe de Peterson.

Définition 3.3 (*Arête transitif*)

Un graphe $G = (V, E)$ est arête transitif (nous disons parfois arête symétrique) si le groupe d'automorphisme de G , $Aut(G)$, est transitif sur E , c'est à dire si $\forall (u, v), (u', v') \in E, \exists \alpha \in Aut(G)$ tel que $(\alpha(u), \alpha(v)) = (u', v')$.

Remarquons que cette fois les automorphismes sont liés aux arêtes et par conséquent un graphe arête transitif n'est pas toujours régulier. Le graphe de la figure 3.3 est arête transitif [4]. Il existe des graphes sommet transitifs sans qu'ils soient arête transitifs. Le graphe CCC(3) de la figure 3.4 en est un exemple [35]. L'arête étiquetée par a , dans ce graphe, appartient à un cycle de longueur 3, tandis que celle étiquetée par b n'y appartient pas et par conséquent un automorphisme

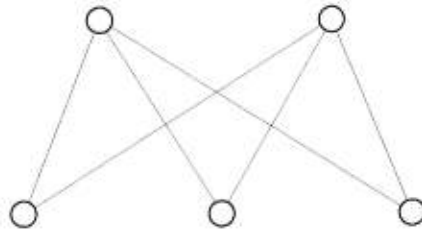


FIG. 3.3: Un graphe arête transitif mais pas sommet transitif.

qui conserve l'adjacence ne peut pas exister entre ces deux arêtes.

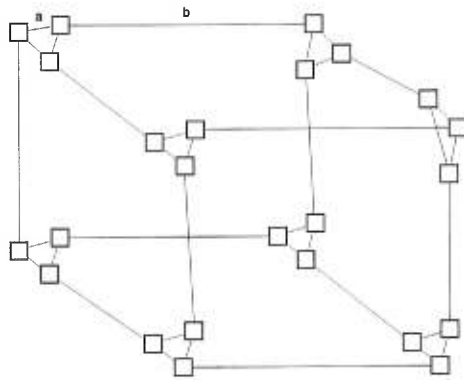


FIG. 3.4: Un graphe sommet transitif mais pas arête transitif.

Un graphe arête transitif n'est pas nécessairement sommet transitif non plus. Étant donné la régularité des graphes sommet transitifs, n'importe quel graphe arête transitif qui n'est pas régulier (nécessairement biparti) forme un exemple. Le graphe biparti $K_{2,3}$ de la figure 3.3 en est un.

Bien qu'il ne soit pas toujours évident de trouver des graphes réguliers qui sont arête transitifs mais pas sommet transitifs, quelques exemples de ce type de graphes sont donnés dans [11, 13].

Dans ce qui suit, nous allons définir deux classes de graphes qui possèdent des degrés de symétrie plus élevés que ceux de sommet et arête transitifs : les graphes *distance transitifs* et *arc transitifs*. Nous ne définissons pas ces deux classes pour les étudier en profondeur, mais pour mieux placer les classes entre elles en créant une hiérarchie plus complète décrivant le degré de symétrie de chaque classe par

rapport aux autres

Définition 3.4 [40] (*k*-distance transitif)

Soit D le diamètre d'un graphe $G = (V, E)$. Soit $0 \leq k \leq D$. On dit que G est *k*-distance transitif si pour tout $x, y, u, v \in V$ avec $d(x, y) = d(u, v) = k$, il existe un automorphisme α tel que $\alpha(x) = u$ et $\alpha(y) = v$.

Un graphe G est *distance transitif* s'il est *k*-distance transitif pour tout $0 \leq k \leq D$. La définition de 1-distance transitif correspond *exactement* à la définition d'un *graphe symétrique* donnée dans [4]. Dans ce texte nous utilisons le terme "graphe symétrique" pour désigner n'importe quel graphe ayant une symétrie quelconque. Ainsi, par abus de langage, quand nous utilisons des termes comme "classe de symétrie" ou "classe de graphes symétriques", c'est pour faire référence à n'importe quelles classes de graphes possédant une symétrie ou une autre.

L'hypercube de la figure 3.5 est un exemple de graphe distance transitif [40].

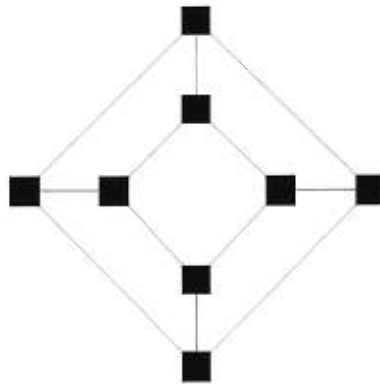


FIG. 3.5: L'hypercube est distance transitif.

Remarque 3.1 Un *graphe sommet transitif* est 0-distance transitif. Nous avons aussi que la propriété d'être 1-distance transitif implique celle d'être *arête transitif* mais la réciproque n'est pas nécessairement vraie (voir la figure 3.3). Par le fait même, un *graphe 1-distance transitif* est nécessairement *sommet transitif* ET *arête transitif*. Le graphe de la figure 3.5 est à la fois *sommet transitif* et *arête transitif*. Si un *graphe régulier*, G , est *arête transitif* et *sommet transitif* et si son

degré est impair, alors le graphe est 1-distance transitif [51]. Si, le degré est pair, alors la même conclusion n'est pas nécessairement vraie [12].

Définition 3.5 [40](*k*-arc transitif)

Un *k*-arc dans un graphe $G = (V, E)$ est une suite $u_0, u_1, u_2, \dots, u_k$ de $k + 1$ sommets de V tels que $(u_i, u_{i+1}) \in E$ pour $0 \leq i \leq k - 1$ et $u_{i-1} \neq u_{i+1}$ pour $1 \leq i \leq k - 1$. Le graphe G est dit *k*-arc-transitif pour $k \geq 1$ s'il existe un automorphisme α tel que pour tout $0 \leq j \leq k$ nous ayons $\alpha(w_j) = z_j$ où $w = w_0, w_1, w_2, \dots, w_k$ et $z = z_0, z_1, z_2, \dots, z_k$ sont deux *k*-arcs arbitraires dans G .

Remarque 3.2 Le graphe de Peterson est un exemple de graphe 3-arc transitif [4]. La propriété d'être *k*-arc transitif représente une condition plus forte que celle d'être 1-distance transitif : "*k*-arc transitif" implique "1-distance transitif" mais la réciproque n'est pas nécessairement vraie. Un graphe sommet transitif et arête transitif n'est pas toujours *k*-arc transitif. Un exemple d'un graphe régulier de degré 4 avec 27 sommets qui est sommet transitif et arête transitif mais qui n'est pas *k*-arc transitif a été donné dans [36].

Nous schématisons les relations entre les différentes classes de symétrie que nous venons de définir par la figure 3.6.

Dans les prochains paragraphes nous définissons deux autres classes de symétrie : *sommet uniforme* et *arête uniforme*. La première est liée à une certaine régularité du nombre de cycles auxquels les sommets appartiennent. La deuxième est liée aussi à cette même régularité mais entre les arêtes et non pas les sommets. Ces deux classes, bien que définies récemment, elles nous permettent d'explorer des nouveaux liens avec les autres classes dans la hiérarchie, et elles nous aident à caractériser certains types de sens de l'orientation étudiés au chapitre 5.

Étant donné un graphe $G = (V, E)$, nous avons pour tout sommet $x \in V$, $C_i(x)$ est le nombre de cycles de longueur i auxquels le sommet x appartient.

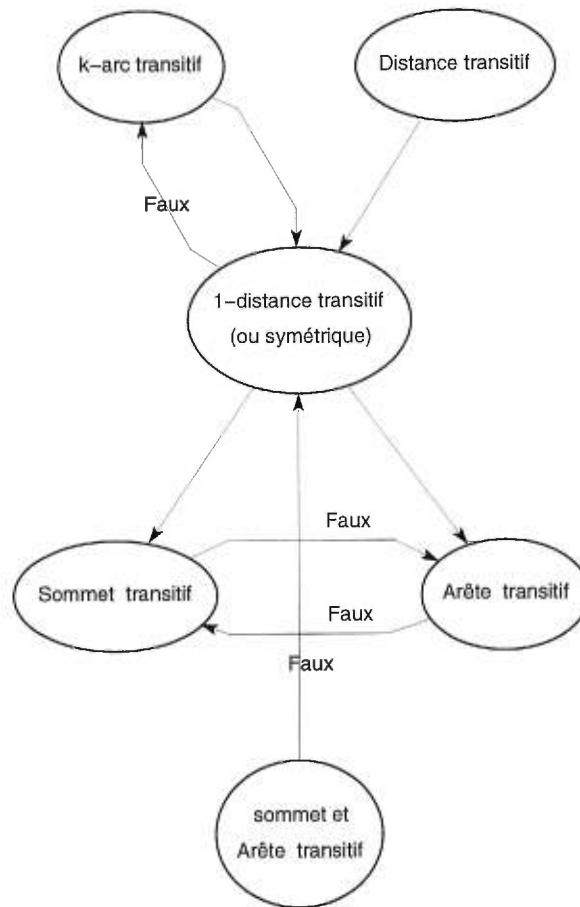


FIG. 3.6: Relations entre différentes classes de symétrie.

Définition 3.6 [14, 31] (*Sommet uniforme*)

Un graphe $G = (V, E)$ est sommet uniforme si et seulement si

$$\forall i \in \mathbb{N}, \forall x, y \in V, C_i(x) = C_i(y)$$

Cela veut dire tout simplement que tous les sommets du graphe appartiennent au même nombre de cycles de chaque longueur. Les cycles (C_n) , les graphes complets (K_n) et les hypercubes sont tous des exemples de graphes sommet uniformes.

Théorème 3.1 [14]

Tout graphe sommet transitif est sommet uniforme.

Nous allons voir plus loin l'importance de cette propriété quand il s'agit d'étudier certaines caractéristiques des graphes.

Question 3.1 [14]

Étant donné un graphe G régulier et sommet uniforme, est-ce que G est sommet transitif ?

Les auteurs de [31] ont montré récemment que la réponse est *Non*. Avant d'expliquer leur raisonnement, nous avons besoin de donner la définition suivante :

Définition 3.7 [31](*Arête uniforme*)

Soit $G = (V, E)$ un graphe avec n sommets et soit $e \in E$. Pour tout $3 \leq k \leq n$, $c_k(e)$ est le nombre de cycles de longueur k qui contiennent e . Le graphe G est arête uniforme si $c_k(e) = c_k(e')$ pour tout $e, e' \in E$.

Remarquons que la définition d'*arête uniforme* est semblable à celle de *sommet uniforme* avec la différence que la régularité réside cette fois dans le nombre de cycles auxquels les arêtes (et non le sommet) appartiennent. Les auteurs ont observé que tout graphe arête uniforme est forcément sommet uniforme. Cette implication est due au fait que chaque cycle de longueur k passant par un sommet u passe exactement par deux arêtes incidentes à u et par conséquent, ce cycle est compté deux fois dans la somme de $c_k(e)$ où e est une arête incidente à u . D'autre part, les auteurs observent qu'un graphe arête transitif est nécessairement arête uniforme, et par conséquent sommet uniforme. Par ailleurs, nous savons qu'il existe des graphes réguliers qui sont arête transitifs mais pas sommet transitifs. Par ce qui précède, n'importe lequel de ces graphes prouve que la réponse à la question 3.1 est *Non*.

Le graphe régulier et arête transitif donné par la figure 3.7 est un exemple. Ce graphe a été construit dans [11] comme exemple d'un graphe régulier qui est arête transitif mais pas sommet transitif.

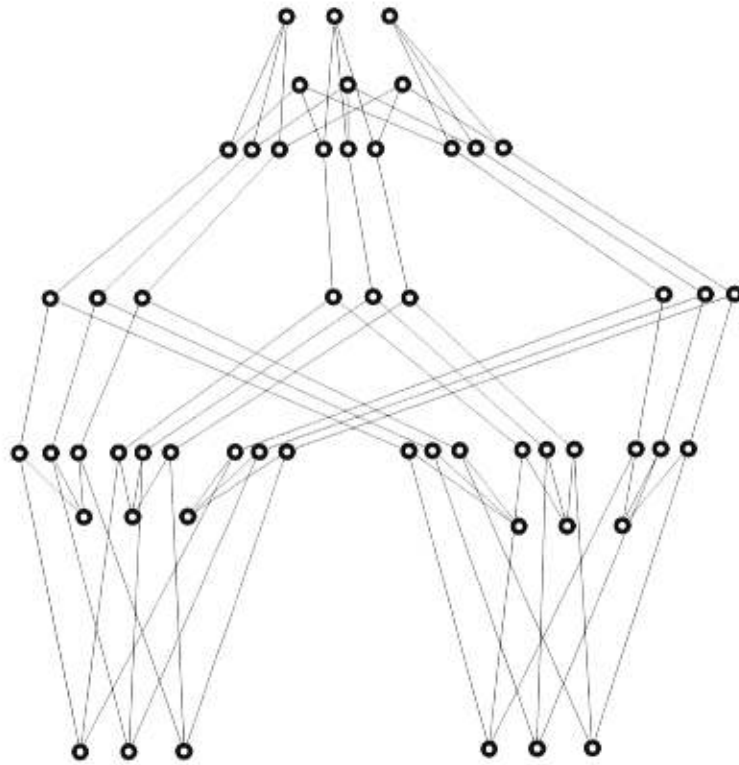


FIG. 3.7: Un graphe régulier sommet uniforme mais pas sommet transitif.

Nous reprenons, schématiquement, les relations entre toutes les propriétés que nous avons vues par la figure 3.8.

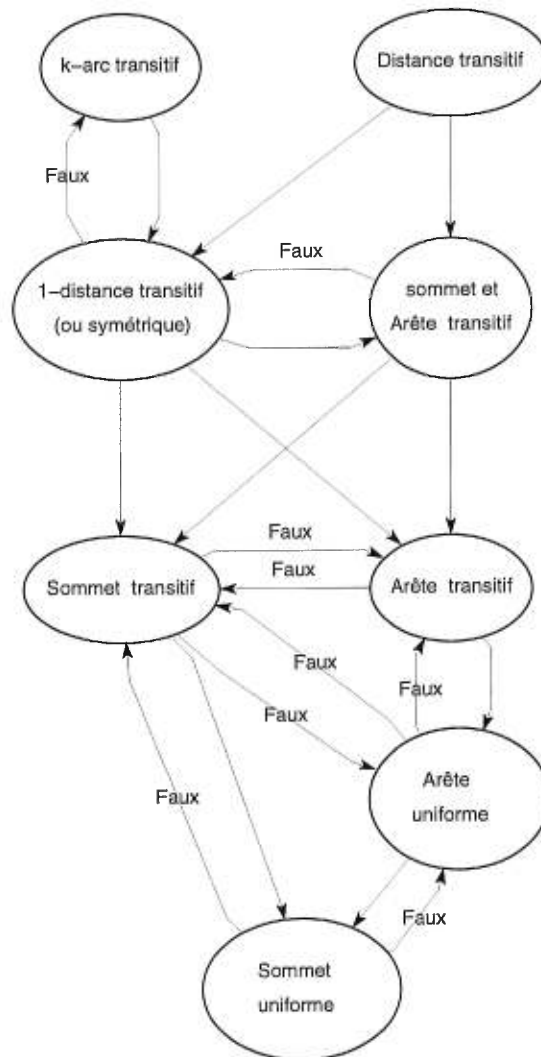


FIG. 3.8: Autres relations entre différentes classes de symétrie.

3.2 Graphes chemin uniforme (PR)

Le titre de cette section est une tentative de traduction du terme anglais "Path regular graphs". Dans ce qui suit nous allons, malheureusement, parler de ce type de graphes en gardant les termes abrégés de l'anglais. À cela, il y a au moins deux justifications : l'une est que nous ne connaissons pas de termes français équivalents, l'autre est pour ne pas alourdir le texte en essayant de traduire, imparfaitement, en français des termes de trois mots précédés par deux ou trois

paramètres. Donc, VPR et EPR sont des abréviations de “Vertex path regular” et “Edge path regular” respectivement.

Les graphes chemin uniformes possèdent une symétrie liée aux chemins les plus courts entre les sommets. Bien que nous n’ayons pas trouvé beaucoup de travaux portant sur l’étude de cette classe, les graphes chemin uniformes peuvent en théorie, comme nous le verrons plus loin, modéliser certains réseaux de communication.

Voici les définitions formelles de ce type de graphes :

Définition 3.8 [44] (k, m_v) -VPR et (k, m_e) -EPR

Un graphe $G = (V, E)$ est dit (k, m_v) -VPR (respectivement (k, m_e) -EPR) où les paramètres k , m_v et m_e sont des entiers naturels, si les conditions suivantes sont satisfaites :

- (i) Il existe une liste $\dagger \mathcal{L}$, associée au graphe, qui contient des chemins les plus courts, entre chaque paire de sommets dans G .
- (ii) Chaque paire de sommets forme les extrémités d’exactement $k \geq 1$ chemins dans \mathcal{L} .
- (iii) Chaque sommet (respectivement arête) apparaît exactement dans m_v (respectivement m_e) chemins de \mathcal{L} .

On utilise parfois VPR et EPR sans donner leurs paramètres quand la précision de ces derniers est inutile. On observe que si un graphe est (k, m_v) -VPR, il est $(ka, m_v a)$ -VPR pour tout entier $a \geq 1$ (la même remarque s’applique sur les graphes (k, m_e) -EPR). On dit alors qu’un graphe est VPR (EPR) s’il existe $k, m \in \mathbb{N}$ tels que ce graphe est (k, m) -VPR ((k, m) -EPR).

Le graphe $K_3 \times K_2$ de la figure 3.9 est un exemple d’un graphe VPR. Si nous considérons une liste \mathcal{L} contenant deux copies des plus courts chemins de longueur inférieure ou égale à 1, et une seule copie de chaque plus court chemin de longueur deux, nous constatons que c’est un graphe $(2, 14)$ -VPR. Par ailleurs,

\dagger Un multi-ensemble : des répétitions sont possibles.

il n'est pas difficile de voir que le graphe $K_{1,5}$ de la figure 3.10 est $(1, 5)$ -EPR.

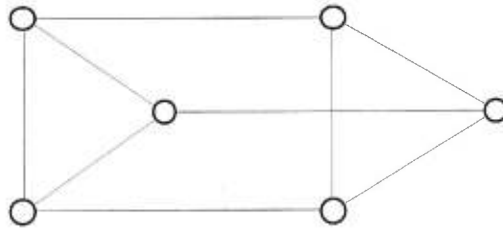


FIG. 3.9: Un graphe VPR.

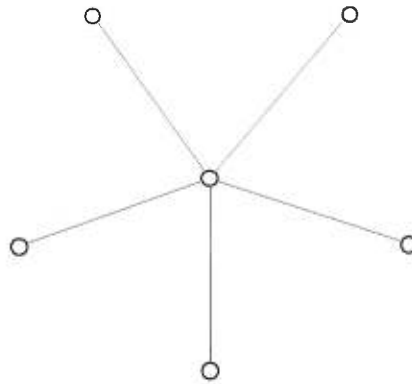


FIG. 3.10: Un graphe EPR.

Une classe de graphes possédant une symétrie encore plus forte que celle des graphes VPR et EPR est définie de la manière suivante :

Définition 3.9 [44] (k, m_v, m_e) -SPR

Un graphe G est dit (k, m_v, m_e) -SPR (pour "Strongly path regular") si les conditions suivantes sont satisfaites :

- (i) Il existe une liste, \mathcal{L} , associée aux plus courts chemins dans G .
- (ii) Le graphe G est à la fois (k, m_v) -VPR et (k, m_e) -EPR où les paramètres correspondent à la même liste \mathcal{L} .

Un exemple d'un graphe $(1, 6, 3)$ -SPR est donné par la figure 3.11.

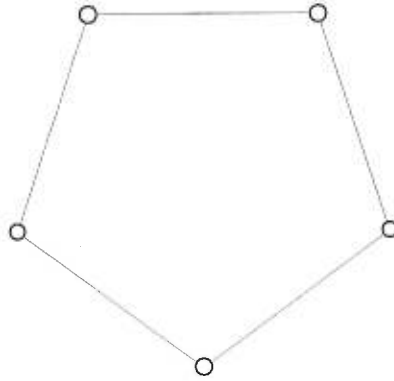


FIG. 3.11: Un graphe SPR.

Nous dirons qu'un graphe est PR (pour "Path regular") s'il est VPR ou EPR. Observons que dans un graphe (k, m_v) -VPR ou (k, m_e) -EPR ou même dans un graphe (k, m_v, m_e) -SPR, la liste associée au graphe doit contenir chaque sommet sous forme d'un chemin de longueur 0 avec une multiplicité k . De même, chaque paire de sommets adjacents sera considérée comme un chemin de longueur 1 avec multiplicité k .

Remarque 3.3 [44]

- Nous pouvons trouver des graphes qui sont à la fois VPR et EPR mais, puisque les paramètres ne correspondent pas à la même liste des chemins, ne sont pas SPR. Un exemple de tels graphes est donné par la figure 3.12. Ce graphe est $(5, 27)$ -VPR et $(4, 6)$ -EPR mais il n'est pas SPR.
- Pour conclure qu'un graphe quelconque est (k, m_e) -EPR il suffit de montrer que chaque arête existe dans exactement $(m_e - k)$ chemins de longueur supérieure ou égale à 2 dans la liste \mathcal{L} .
- Pour conclure qu'un graphe quelconque est (k, m_v) -VPR il suffit de montrer que chaque sommet existe comme sommet intérieur dans exactement $m_v - k|V(G)|$ chemins dans la liste \mathcal{L} .

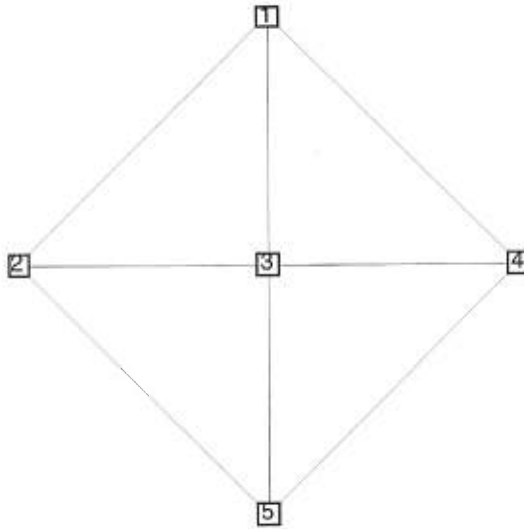


FIG. 3.12: Un graphe VPR et EPR mais pas SPR.

Remarquons que le graphe de la figure 3.12 n'est pas régulier et par conséquent la régularité des graphes n'est pas une condition nécessaire à l'existence de la propriété de VPR ou EPR. Il existe une méthode directe pour tester si un graphe donné est VPR ou EPR. Cette méthode, qui se trouve dans [44], est basée sur la programmation linéaire.

Sachant que tester si un graphe régulier est EPR est faisable par la méthode proposée dans [44], tester si le même graphe est SPR devient automatiquement faisable par le lemme suivant :

Lemme 3.1 [44] *Tout graphe qui est à la fois régulier et EPR est aussi SPR.*

Le théorème suivant nous montre des relations directes entre les graphes chemin uniformes et les graphes sommet transitifs et arêtes transitifs.

Théorème 3.2 [44]

- *Tout graphe connexe et sommet transitif est VPR mais la réciproque n'est pas nécessairement vraie.*
- *Tout graphe connexe et arête transitif est EPR mais la réciproque n'est pas nécessairement vraie.*

- *Tout graphe connexe et à la fois sommet transitif et arête transitif est SPR mais il existe des graphes SPR qui ne sont pas sommet transitifs et/ou arête transitifs.*

Un diagramme de relations représentant le théorème 3.2 est donné par la figure 3.13.

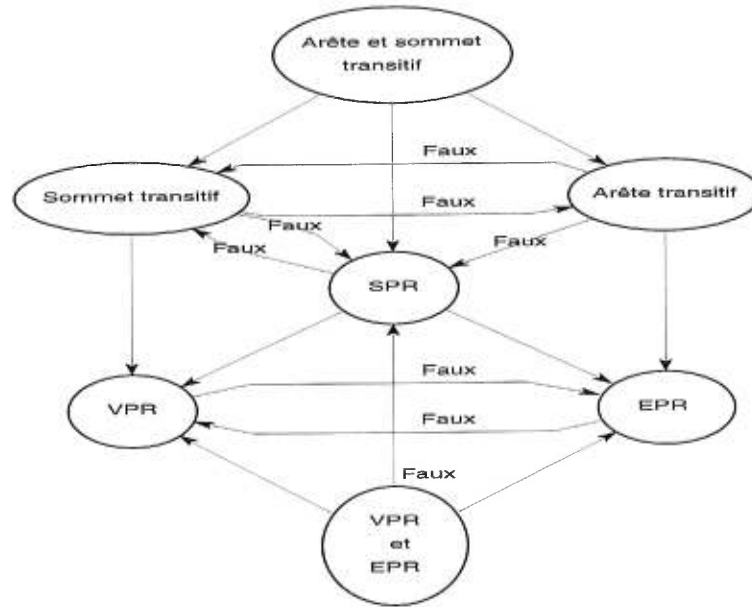


FIG. 3.13: Relations entre certaines symétries et les graphes PR.

Par le théorème que nous venons de voir, nous concluons que le degré de symétrie dans les graphes sommet transitifs et arête transitifs est plus élevé que celui dans les graphes chemin uniformes. Par contre, nous pouvons souvent trouver une certaine relation entre les classes contenant une régularité parmi les sommets ; par exemple les classes de graphes sommet transitifs, sommet uniformes et VPR. De même, les classes de graphes contenant une régularité parmi les arêtes sont normalement liées. Cependant, nous remarquons qu'en général la régularité parmi les sommets est indépendante de celle parmi les arêtes. Les deux corollaires suivants nous montrent cette indépendance entre les graphes sommets transitifs et les graphes EPR ainsi qu'entre ceux arête transitifs et VPR.

Corollaire 3.1 [44] *La propriété pour un graphe d'être EPR est indépendante*

de la propriété que ce graphe soit :

(i) Sommet transitif.

(ii) VPR.

(iii) Régulier.

Corollaire 3.2 [44] *La propriété pour un graphe d'être VPR est indépendante de la propriété que ce graphe soit :*

(i) Arête transitif.

(ii) EPR.

(iii) Régulier.

Pour résumer les résultats explorés jusqu'à maintenant, nous pouvons globalement dire que parmi les classes de graphes vues précédemment, les graphes arc et distance transitifs possèdent des degrés de symétrie supérieurs à ceux de tous les autres graphes. Par la suite, il y a les graphes sommet et arête transitifs qui ont tout de même des degrés de symétrie supérieurs à ceux des graphes sommet uniformes, arête uniformes ou les graphes chemin uniformes. Certaines nouvelles relations peuvent être déduites directement de celles qu'on a étudiées. La prochaine section sera consacrée à montrer quelques nouvelles relations.

3.3 Indépendance de PR des autres propriétés

Dans cette section, en nous basant sur différents résultats explorés dans ce chapitre, nous donnons une série d'énoncés avec leurs preuves montrant l'indépendance des graphes PR des autres classes de symétrie.

(i) Un graphe sommet transitif n'est pas nécessairement SPR.

Prenons le graphe $K_3 \times K_2$ de la figure 3.9. Ce graphe, comme c'est montré dans [44], est sommet transitif mais pas EPR. Étant donné qu'un graphe SPR est aussi EPR, le graphe en question n'est pas SPR.

- (ii) Un graphe sommet uniforme n'est pas nécessairement VPR.

Nous savons, par le Corollaire 3.2, que nous pouvons trouver des graphes arête transitifs qui ne sont pas VPR. De plus, nous savons par [31] qu'un graphe régulier et arête transitif *est nécessairement* sommet uniforme. N'importe quel graphe arête transitif qui n'est pas VPR constitue un exemple.

- (iii) Un graphe VPR n'est pas nécessairement sommet uniforme.

Prenons le graphe de la figure 3.12. Ce graphe, comme c'est montré dans [44], est (5, 27)-VPR. D'autre part, il n'est pas sommet uniforme car le sommet 3 appartient à exactement 4 cycles de longueur 3 et les autres sommets appartiennent à exactement 2 cycles de longueur 3.

- (iv) Un graphe sommet uniforme n'est pas nécessairement SPR.

Étant donné qu'un graphe sommet transitif est toujours sommet uniforme, le graphe donné dans la preuve de (i) prouve l'énoncé.

- (v) Un graphe EPR n'est pas nécessairement sommet uniforme.

Prenons le graphe de la figure 3.12. Ce graphe est (4, 6)-EPR tandis qu'il n'est pas sommet uniforme, comme c'est démontré dans la preuve de (iii).

- (vi) Un graphe sommet uniforme n'est pas nécessairement EPR.

Le graphe $K_3 \times K_2$ de la figure 3.9 est sommet transitif, alors il est sommet uniforme. Ce même graphe n'est pas EPR.

Étant donné qu'un graphe arête uniforme est toujours sommet uniforme, nous pouvons conclure que toute condition, dans les énoncés précédents, qui n'implique pas la propriété d'être sommet uniforme n'implique pas non plus celle d'être arête uniforme. La figure 3.14 rassemble les énoncés précédents.

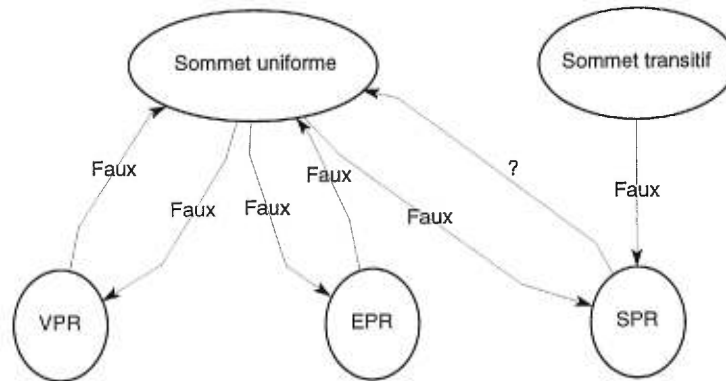


FIG. 3.14: Quelques relations avec les graphes PR.

Question 3.2 *Est-ce qu'un graphe SPR est toujours sommet uniforme ? (arête uniforme ?)*

Nous n'avons pas de réponse à cette question. Ceci dit, et étant donné les indépendances que nous venons de voir entre les graphes sommet uniformes et les graphes PR, nous avons tendance à croire qu'il n'y a pas de raison pour que la réponse à cette question soit positive. Remarquons que les graphes SPR que nous avons examinés étaient tous arête uniformes (et par conséquent sommet uniformes), par contre, quand nous considérons des graphes avec un assez grand nombre de sommets et d'arêtes, tester si ces graphes sont sommet uniformes (ou arête uniformes) devient rapidement difficile.

Question 3.3 *À quoi servent les graphes PR ?*

D'un point de vue pratique, Les graphes PR peuvent modéliser efficacement certains réseaux de communications. Supposons que les sommets d'un graphe (k, m_e) -EPR représentent des bases (ou des entités) de communications et les

arêtes des liens entre ses entités. Chaque arête est capable de supporter m_e canaux de communication concurrente. Le réseau modélisé par un graphe EPR permet d'établir k communications différentes entre toute paire de sommets d'une manière concurrente. L'utilisation d'un canal quelconque est assez efficace ; la communication est établie en passant par des chemins les plus courts et la capacité de chaque ligne (arête) est utilisée d'une manière maximale. [44]

Par ailleurs, les graphes PR peuvent garantir d'autres avantages sur la manière dont on veut construire un réseau précis de communication. Examinons, tout d'abord, le théorème suivant donné dans [44] :

Théorème 3.3 [44]

Il existe des graphes SPR ayant n'importe quel des paramètres donnés :

- (i) *Diamètre.*
- (ii) *Nombre chromatique.*
- (iii) *Maille.*
- (iv) *Connectivité par sommet.*
- (v) *Connectivité par arête.*

Avec ce théorème, il serait toujours possible de construire des graphes qui répondent à l'une ou l'autre des ces "exigences" (diamètre,...). Par exemple, le cycle C_n (qui est SPR) a la maille égale à n et le diamètre égal à $\lfloor n/2 \rfloor$. De même, les graphes complets K_n ont un nombre chromatique égal à n . Enfin, les graphes complets K_{n+1} , les graphes réguliers bipartis $K_{n,n}$ et les hypercubes de dimension n (qui sont tous SPR) ont une connectivité de sommet et une connectivité d'arête égale à n . Autrement dit, les graphes chemin uniformes peuvent non seulement modéliser certains réseaux de communication décrits ci-dessus, mais ils peuvent aussi répondre à certaines restrictions "numériques" dans le réseau, indiquées dans le théorème précédent.

Chapitre 4

Symétrie dans les graphes

étiquetés et graphes de Cayley

Dans un système distribué de communication, les liens entre les entités portent souvent des noms qu'on appellera *étiquettes*. En théorie des graphes, on peut modéliser un tel système par un graphe étiqueté où une arête (non orientée) entre deux sommets peut porter deux étiquettes (une de chaque côté de l'arête). À titre d'exemple, dans le graphe de la figure 4.1, le sommet a identifie l'arête (a, b) par l'étiquette e et le sommet b identifie la même arête par l'étiquette e' . Remarquons qu'on aurait pu utiliser, pour ce type de modélisation, des graphes orientés dont un arc porte une seule étiquette. Nous avons gardé les notations existantes dans la littérature et donc nous utilisons seulement des graphes non orientés (ou bidirectionnels) tout au long de ce chapitre.



FIG. 4.1: Une arête portant deux étiquettes.

Dans le chapitre précédent nous avons vu plusieurs classes de graphes (non étiquetés) possédant une certaine symétrie. Dans ce chapitre, nous nous intéressons

encore une fois à étudier la symétrie dans les graphes, mais cette fois nos graphes seront étiquetés. Après avoir précisé quelques notations et définitions de base, nous définissons deux types de symétrie pouvant exister dans les graphes étiquetés : la *symétrie de vue* et la *symétrie de voisinage*. Bien que dans les deux cas la régularité du graphe est nécessaire, la première symétrie (la vue) est liée à la capacité de chaque sommet de “voir” les mêmes étiquettes sur les arêtes incidentes. La symétrie de voisinage dans un graphe étiqueté est liée à une condition (que nous précisons ci-dessous) “d’uniformité” des parcours qui peuvent exister entre toute paire de sommets. Par la suite, nous introduisons des graphes basés sur (ou générés par) des groupes finis : les graphes de Cayley. Nous en donnons quelques exemples et nous soulignons quelques points qui montrent l’importance d’utiliser cette classe de graphes comme modèle dans les réseaux de communication. Par la suite, nous mettons en relation les graphes de Cayley, les graphes avec une symétrie de vue et les graphes avec une symétrie de voisinage pour arriver aussi à observer quelques liens entre ces graphes et d’autres vus au chapitre précédent. Enfin, dans la dernière section, nous exposons quelques points centraux sur l’importance de la symétrie dans les graphes.

4.1 Notations et définitions de base

Rappelons qu’étant donné un graphe régulier $G = (V, E)$, $E_G(x)$ désigne l’ensemble des arêtes incidentes au sommet $x \in V$ et $d(x) = |E(x)|$ le degré de x . Étant donné un ensemble d’étiquettes Σ , une *orientation locale* de $x \in V$ est une fonction injective $\lambda_x : E(x) \rightarrow \Sigma$ qui associe une étiquette distincte $l \in \Sigma$ à chaque arête $e \in E(x)$. Nous appellerons l’ensemble $\lambda = \{\lambda_x : x \in V\}$ de fonctions d’étiquetage locales, l’étiquetage de G . Le graphe étiqueté G sera désigné par (G, λ) . Notons que chaque arête (x, y) aura deux étiquettes, $\lambda_x((x, y))$ et $\lambda_y((x, y))$, qui peuvent être différentes. Nous dirons que λ est un *étiquetage minimal* s’il utilise un nombre d’étiquettes égal au degré du graphe. À noter que,

dans ce contexte, nous considérons seulement des graphes réguliers.

Un étiquetage λ est *symétrique* s'il existe une fonction bijective $\psi : \Sigma \rightarrow \Sigma$ telle que pour toute arête $(x, y) \in E$, $\lambda_y((x, y)) = \psi(\lambda_x((x, y)))$ [17, 21].

Autrement dit, le fait de connaître l'étiquette d'un côté de l'arête permet de déduire l'étiquette du deuxième côté de l'arête. Intuitivement, pour qu'un étiquetage soit symétrique, si une arête est étiquetée par \mathbf{a} d'un côté et par \mathbf{b} de l'autre côté, pour toutes les autres arêtes du graphe, si \mathbf{a} est une étiquette d'un côté de l'arête, \mathbf{b} doit être l'étiquette de l'autre côté.

4.2 Symétrie de vue

Il existe dans la littérature différentes manières de définir la *symétrie de vue*. Dans cette section nous aurons l'occasion de proposer une nouvelle manière (précise) de définir ce type de symétrie qui existe dans les graphes étiquetés. Intuitivement, la vue d'un sommet quelconque dans un graphe étiqueté est liée directement au nombre d'arêtes incidentes à ce sommet et aux étiquettes associées à ces arêtes. Voici une définition formelle de cette notion :

Définition 4.1 (*i*-Vue)

Étant donné un graphe étiqueté $(G = (V, E), \lambda)$, la *i*-vue du sommet $v \in V$ est une arborescence, $(T_i(v), \lambda')$ (parfois notée simplement par $T_i(v)$) définie récursivement de la manière suivante :

$$T_0(v) = (\{(v, \varepsilon, 0)\}, \emptyset)$$

$$T_i(v) = (V(T_{i-1}(v)) \cup X_i, E(T_{i-1}(v)) \cup Y_i)$$

où

$$X_i = \{(x, y, i) \mid (y, x) \in E(G) \text{ et } \exists y' \text{ tel que } (y, y', i-1) \in V(T_{i-1}(v))\},$$

$$Y_i = \{((x, y, i-1), (x', x, i)) \mid (x, y, i-1) \in V(T_{i-1}(v)) \text{ et } (x', x, i) \in X_i\}$$

avec les arêtes étiquetées par :

$$\lambda'_{(x,y,i-1)}((x,y,i-1), (x',x,i)) = \lambda_x(x,x')$$

et

$$\lambda'_{(x',x,i)}((x',x,i), (x,y,i-1)) = \lambda_{x'}(x',x).$$

On peut voir cette définition comme une manière de *construire*, à partir du graphe G , une arborescence qui sera la i -vue d'un sommet quelconque. Observons que $T_i(v)$ est un arbre de profondeur i . La racine de l'arbre correspond au sommet v (dont on cherche la vue) dans le graphe G et les enfants de chaque sommet de $T_i(v)$ correspondent aux sommets adjacents à ce sommet dans G . Chaque sommet de l'arbre est un triplet (x,y,i) où x correspond au sommet x du graphe, y correspond à un des voisins de x dans G et i est la profondeur du noeud (x,y,i) . Un exemple d'un graphe avec la 2-vue d'un sommet est donné par la figure 4.2. À noter que dans le triplet (x,y,i) , y correspond au "père" de x . Cette information et la profondeur du sommet dans l'arbre sont nécessaires pour distinguer les sommets les uns des autres ; ainsi les sommets de l'arbre sont tous uniques.

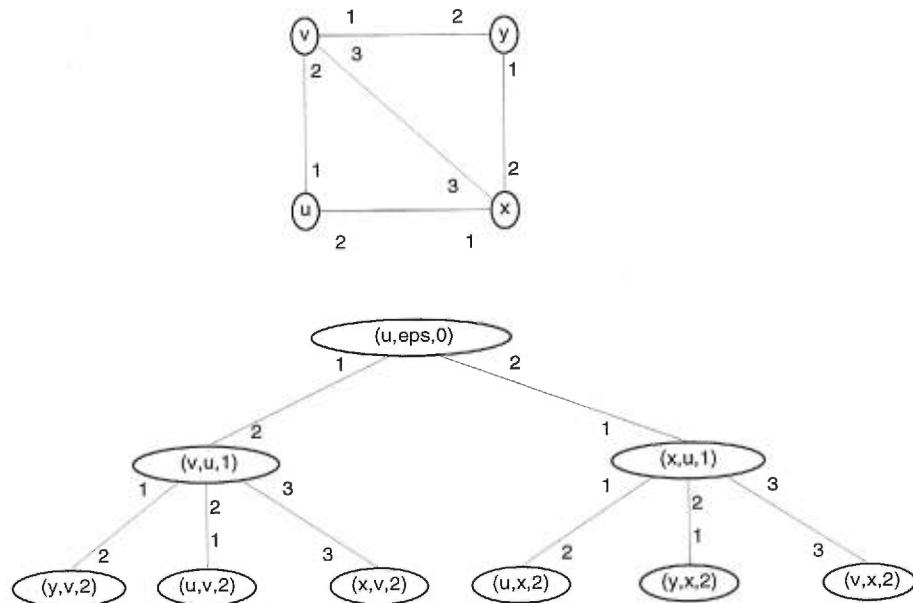


FIG. 4.2: Un graphe étiqueté et la 2-vue, $T_2(u)$, du sommet u .

On pourrait parfois se demander si deux sommets quelconques (et récursivement les sommets adjacents) ont le même nombre de liens étiquetés de la même manière. Ceci pourrait éventuellement être utile quand il s'agit d'exécuter certains algorithmes sur différents processeurs ayant "la même vue" dans un système distribué.

Avant de poursuivre, nous avons besoin d'établir la définition suivante :

Définition 4.2 (*Isomorphisme de graphes étiquetés*)

Soit $(G = (V, E), \lambda)$ et $(H = (V', E'), \lambda')$ deux graphes étiquetés. Un isomorphisme de graphe (G, λ) dans (H, λ') est une bijection $\beta : V \rightarrow V'$ telle que les deux conditions suivantes sont satisfaites :

- (i) $(\beta(u), \beta(v)) \in E'$ si et seulement si $(u, v) \in E$.
- (ii) $\lambda_u((u, v)) = \lambda'_{\beta(u)}(\beta(u), \beta(v))$.

On dit qu'un *automorphisme* de (G, λ) est un isomorphisme de (G, λ) dans (G, λ) .

Définissons maintenant formellement qu'est-ce que signifie une "même vue".

Définition 4.3 (*Vue*)

Nous disons que deux sommets u et v ont la même vue si et seulement si $\forall i \in \mathbb{N}$:

- (i) Il existe une bijection $\phi : V(T_i(u), \lambda) \rightarrow V(T_i(v), \lambda)$ telle que $(u, \varepsilon, 0) = \phi(v, \varepsilon, 0)$.
- (ii) $(T_i(u), \lambda)$ est isomorphe à $(T_i(v), \lambda)$.

Rappelons que les deux arborescences $T_i(u)$ et $T_i(v)$ sont des graphes étiquetés et le terme *isomorphe* dans ce contexte correspond à la définition d'isomorphisme dans les graphes étiquetés donnée ci-dessus.

En utilisant les deux dernières définitions, nous pouvons maintenant préciser ce qu'est une symétrie de vue.

Soit R une relation d'équivalence définie sur V par uRv si et seulement si u et v ont la même vue.

Définition 4.4 (V_k -symétrique)

Un graphe étiqueté (G, λ) est V_k -symétrique si ce graphe a une orientation locale et si R a k classes d'équivalence.

Quand $k = 1$, c'est à dire quand tous les sommets ont la même vue, le graphe est dit *vue symétrique*. Nous remplacerons parfois V_1 -Symétrique par tout simplement V -symétrique qu'on appellera la *symétrie de vue*. Un exemple d'un graphe V -symétrique est donné par la figure 4.3. Notons que sans la condition d'orientation locale, tout graphe régulier peut avoir un étiquetage qui le rend V_1 -Symétrique (il suffit d'étiqueter toutes les arêtes par la même étiquette).

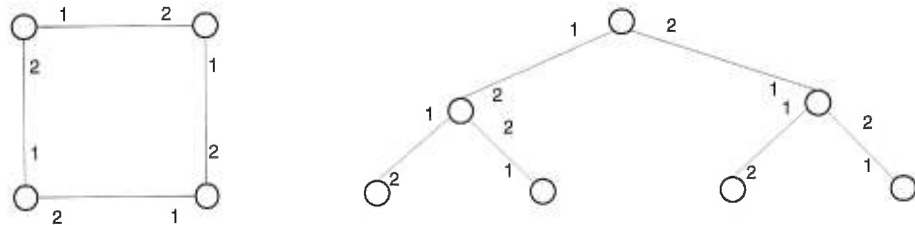


FIG. 4.3: Un graphe V -symétrique et la 2-vue, $T_2(u)$ pour tout sommet $u \in V$.

Avec la définition de vue ci-dessus, les profondeurs des arbres $T_i(u)$ et $T_i(v)$ peuvent être arbitrairement grandes. À partir de cela, nous pouvons naturellement nous demander à quelle profondeur il est possible d'arrêter quand nous voulons tester si deux sommets quelconques ont la même vue ou non. Les auteurs de [52] ont montré qu'une profondeur de n^2 suffit pour vérifier si deux arbres ont la même vue ou non. Un meilleur résultat donné par [45] prouve que pour tester si deux sommets ont la même vue ou non, il suffit de vérifier l'isomorphisme (de deux arbres correspondants) jusqu'à une profondeur de $n - 1$ seulement. Ceci est intuitivement justifié car il n'est pas nécessaire de vérifier plus loin que le diamètre du graphe.

Étant donné l'importance de cette classe de graphes dans le système distribué, nous allons caractériser ces graphes ; c'est à dire donner les conditions nécessaires

et suffisantes pour qu'un graphe ait une vue symétrique. Nous commençons par annoncer un théorème donné et prouvé dans [27].

Théorème 4.1 [27] *Un graphe étiqueté (G, λ) est V -symétrique si et seulement si G est régulier et λ est minimal et symétrique.*

Nous constatons par le théorème 4.1 qu'étant donné un graphe (G, λ) , pour tester si tous les sommets ont la même vue ou non, il suffit de tester si l'étiquetage est minimal et symétrique, ce qui peut se faire dans un temps linéaire en fonction de nombre d'arêtes.

Voici une autre caractérisation des graphes ayant une symétrie de vue basée sur les factorisations des graphes. Rappelons qu'une définition d'un graphe $\{1, 2\}$ -factorisable est déjà donnée au deuxième chapitre.

Théorème 4.2 [52] *Un graphe G est régulier et $\{1, 2\}$ -factorisable si et seulement s'il existe un étiquetage λ tel que (G, λ) est V -symétrique.*

En se basant sur les théorèmes 4.1 et 4.2, nous obtenons directement le théorème suivant :

Théorème 4.3 [27] *Un graphe régulier (G, λ) a un étiquetage minimal et symétrique si et seulement s'il est $\{1, 2\}$ -factorisable.*

Les caractérisations que nous venons de voir nous seront très utiles quand il s'agira d'étudier les relations entre les graphes possédant une symétrie de vue et les autres graphes. Plusieurs de ces relations (basées sur ces caractérisations) seront explorées plus loin dans ce chapitre.

Nous avons maintenant l'occasion d'établir une conjecture mettant en relation les graphes sommet transitifs et les graphes V -symétriques.

Conjecture 4.1 *Tout graphe G sommet transitif possède un étiquetage λ tel que le graphe (G, λ) soit vue symétrique.*

Remarquons que prouver cette conjecture revient à prouver que tout graphe sommet transitif est $\{1, 2\}$ -factorisable, ce que nous croyons vrai. On pourrait

aussi chercher une fonction qui étiquette n'importe quel graphe sommet transitif d'une manière minimale et symétrique.

4.3 Symétrie de voisinage

Dans cette section nous définissons la *symétrie de voisinage* d'une manière substantiellement différente de ce qui existe dans la littérature. Ce type de symétrie peut être intuitivement vu comme une symétrie de sommets (sommet transitif) qui préserve les étiquettes.

En se basant sur la définition d'automorphisme donnée antérieurement, on observe que les automorphismes de (G, λ) avec l'opération de composition forment un groupe, $Aut((G, \lambda))$, qu'on appellera le groupe d'automorphisme de (G, λ) . La relation R' définie sur V par $uR'v$ si et seulement si $\exists \alpha \in Aut(G, \lambda)$ tel que $\alpha(u) = v$ est une relation d'équivalence dont les classes s'appellent les *orbites* de $Aut((G, \lambda))$.

Définition 4.5 (*S_k -symétrique*)

Un graphe étiqueté (G, λ) est dit S_k -symétrique si $Aut((G, \lambda))$ a k orbites ; c'est à dire si R' a k classes d'équivalence.

On dit qu'un graphe S_1 -symétrique a une *symétrie de voisinage*. Différents exemples de ces graphes seront déduits dans la prochaine section où l'on montre que les graphes S_1 -symétriques sont équivalents à une autre classe de graphes qu'on appelle les graphes de Cayley.

Nous verrons plus loin dans ce chapitre que la symétrie de voisinage est strictement plus forte que la symétrie de vue. Nous déduisons aussi quelques relations entre ces types de symétrie et d'autres types vus précédemment. Avant de voir d'autres caractérisations de la symétrie de voisinage, nous avons besoin de définir les graphes de Cayley qui possèdent un degré élevé de symétrie et qui pourraient bien modéliser des réseaux distribués.

4.4 Graphes de Cayley

Les graphes de Cayley sont des graphes basés sur des groupes. Les sommets correspondent aux éléments du groupe et les arêtes correspondent à l'action des générateurs appartenant à un ensemble donné d'éléments du groupe. Ces graphes possèdent un haut degré de symétrie parmi les sommets. Ils nous permettront de mieux comprendre ou de "situer" certaines classes de graphes (en particulier les graphes S_1 -symétrique) dans la hiérarchie des graphes ayant une symétrie quelconque. Nous donnons une définition formelle de ces graphes afin d'établir, plus loin dans le texte, des nouvelles relations entre différents graphes. Il est à noter que l'objectif de cette section n'est pas d'approfondir l'étude de cette classe de graphes mais d'abord d'en définir une ayant un grand degré de symétrie en la mettant en relation avec les graphes vus précédemment. En plus, cela nous permet de caractériser, en fonction des graphes de Cayley, un certain type du sens de l'orientation étudié dans le prochain chapitre.

Définition 4.6 [35, 33](Graphes de Cayley)

Soit Γ un groupe et S est un sous-ensemble de Γ ne contenant pas l'élément neutre, fermé sous l'opération d'inverse, et l'élément g est dans S si et seulement si g^{-1} est dans S .

Un graphe de Cayley $\text{Cay}(\Gamma, S)$ est un graphe dont Γ est l'ensemble de sommets et deux sommets g et h sont reliés par une arête si et seulement si $g^{-1}h \in S$.

Définition 4.7 (Étiquetage de Cayley)

Soit $\text{Cay}(\Gamma, S)$ est un graphe de Cayley. Nous disons que λ est un étiquetage de Cayley si et seulement si pour toute arête (g, h) , $\lambda_g((g, h)) = g^{-1}h$ et $\lambda_h((g, h)) = h^{-1}g$.

Remarquons que le degré d'un graphe de Cayley est le nombre de générateurs ($|S|$). Par conséquent, son étiquetage est minimal. De plus, par la définition même, cet étiquetage est naturellement symétrique.

Un graphe de Cayley est toujours sommet transitif (voir, par exemple [1]). Cependant, tout graphe sommet transitif n'est pas nécessairement de Cayley ; le graphe de Peterson est sommet transitif mais n'est pas de Cayley. De plus, les graphes de Cayley sont indépendants des graphes arête transitifs. N'importe quel graphe arête transitif et non régulier n'est pas de Cayley non plus. Par contre, le graphe CCC(3) de la figure 3.4 est un graphe de Cayley mais il n'est pas arête transitif.

Le cycle est un exemple d'un graphe de Cayley avec un groupe cyclique. Les graphes complets, K_n , sont aussi des exemples des graphes de Cayley avec n'importe quel groupe Γ d'ordre n et $S = \Gamma \setminus \{e\}$ où e est l'élément neutre .

Nous donnons ici trois exemples simples des graphes de Cayley basés sur des groupe de permutations. Une *permutation* de l'ensemble X est une bijection $\alpha : X \rightarrow X$. Soit S_X l'ensemble de toutes les permutations de X (si $X = \{1, 2, \dots, n\}$, nous remplacerons S_X par S_n).

Si $G_1 = \{(1, i) \mid 2 \leq i \leq n\}$, $G_2 = \{(i, i + 1) \mid 1 \leq i \leq n\}$ et $G_3 = G_2 \cup \{(1, n)\}$ sont trois ensembles de générateurs, nous obtenons, à partir de cela, trois graphes de Cayley : $ST_n = (S_n, G_1)$, qu'on appelle l'étoile, $BS_n = (S_n, G_2)$, qu'on appelle le "Bubble sort", et $MB_n = (S_n, G_3)$, qui est une modification de BS_n . Remarquons que ces trois graphes utilisent le même groupe, mais il existe seulement un isomorphisme entre ST_n et BS_n . La figure 4.4 illustre les deux graphes de Cayley ST_3 et BS_3 , tandis que le graphe MB_3 est donné par la figure 4.5.

Nous pouvons trouver dans [35] différentes descriptions et caractéristiques des graphes de Cayley. Plusieurs classes des graphes de Cayley sont étudiées dans [40]. Dans cette étude, les relations avec les graphes arête transitifs et distance transitifs sont exploitées.

Question 4.1 *À quoi servent les graphes de Cayley ?*

Nous avons vu que les graphes de Cayley représentent un modèle théorique basé sur les groupes finis. Ce modèle peut servir, en informatique, à construire,

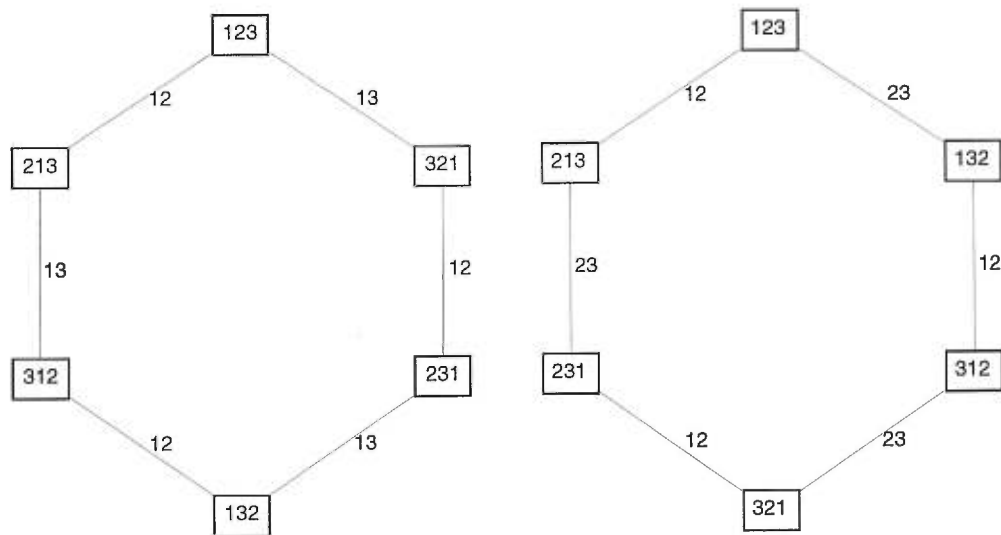
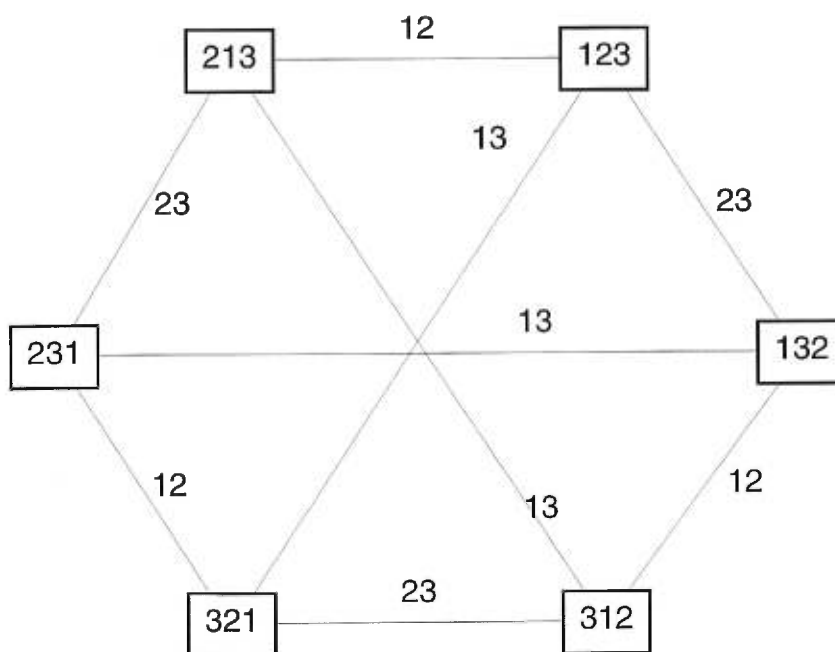


FIG. 4.4: ST_3 et BS_3 générés par G_1 et G_2

analyser et améliorer des réseaux *symétriques* d'interconnexion courants. Il est important de savoir que la plupart de réseaux symétriques d'interconnexion courants peuvent être représentés par des graphes de Cayley et *tout* réseau symétrique d'interconnexion peut être représenté par une simple extension de ce modèle [1]. Ceci nous permet de donner une représentation algébrique de chacun de ces réseaux. Plus important encore, ce modèle nous permet de construire, à partir d'un groupe fini, un réseau symétrique. Étant donné la vaste classe des groupes finis, nous pouvons construire une grande variété de nouveaux réseaux d'interconnexion.

Un autre avantage qui provient de l'analyse de ces réseaux de cette manière algébrique est le fait que plusieurs propriétés de ces réseaux peuvent être prouvées pour une *classe* de graphes au lieu de prouver de telles propriétés pour chaque graphe indépendamment. De plus, même pour des réseaux spécifiques construits en utilisant ce modèle, nous pouvons souvent trouver des nouvelles propriétés d'une manière algébrique et interpréter théoriquement les propriétés de ces graphes.

FIG. 4.5: MB_3 généré par G_3

4.5 Relations entre différentes classes de symétrie

Dans cette section nous mettons en relation les graphes ayant une symétrie de vue ou une symétrie de voisinage avec d'autres classes de graphes étudiés dans le chapitre précédent. Ces relations sont, en général, des *déductions* à partir des différents résultats vus précédemment. Sans donner les détails de la preuve, nous verrons que les graphes ayant une symétrie de voisinage sont équivalents aux graphes de Cayley.

Théorème 4.4 [27] *Un graphe étiqueté (G, λ) est S_1 -symétrique si et seulement si G est un graphe de Cayley et λ est un étiquetage de Cayley.*

Par ce théorème, nous déduisons que les graphes S_1 -symétrique “héritent” des propriétés des graphes de Cayley. Par exemple, tout graphe S_1 -symétrique sera aussi sommet transitif. Par conséquent, ce théorème nous permet d'identifier le degré de symétrie de ces graphes et donc de le mieux placer par rapport aux autres graphes qu'on étudie présentement. Notons que par ce même résultat, tous les

exemples que nous avons donnés sur les graphes de Cayley sont aussi des exemples des graphes S_1 -symétrique.

Le théorème suivant trace une relation entre les graphes S_1 -symétrique et ceux V -symétrique.

Théorème 4.5 [27] *Si un graphe (G, λ) est S_1 -symétrique, alors ce graphe est V -symétrique.*

La preuve de ce théorème est directe par la définition des graphes de Cayley dont les étiquetages sont toujours minimaux et symétriques.

Nous contribuons à prouver dans la prochaine proposition que le degré de symétrie dans les graphes S_1 -symétrique est strictement supérieur à celui dans les graphes V -symétrique.

Proposition 4.1 *Il existe des graphes étiquetés V -symétrique sans qu'ils soient S_1 -symétrique.*

Preuve Prenons n'importe quel graphe régulier et $\{1, 2\}$ -factorisable mais qui n'est pas un graphe de Cayley. Ce graphe est V -symétrique par le théorème 4.2 et il n'est pas S_1 -symétrique par le théorème 4.4. Le graphe de la figure 4.6 en est un exemple. Remarquons que ce graphe n'est pas sommet transitif et donc n'est pas de Cayley. \square

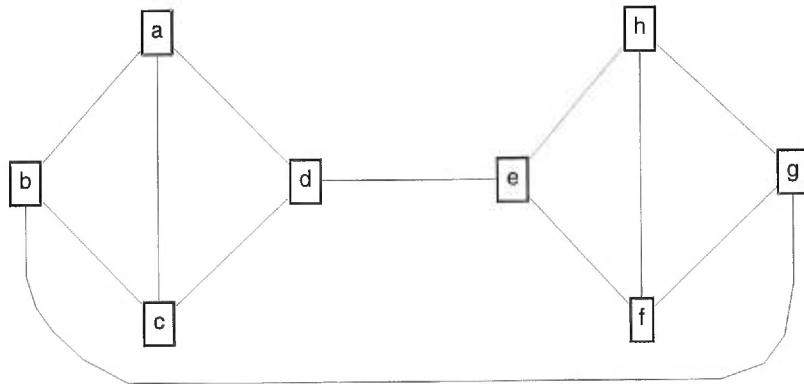


FIG. 4.6: Un graphe $\{1, 2\}$ -factorisable qui n'est pas de Cayley.

Nous allons maintenant annoncer une suite de résultats que nous avons déduits et qui sont basés fondamentalement sur plusieurs relations déjà établies. Ils mettront en relation les graphes S_1 -symétrique et V -symétrique avec ceux qu'on a vus antérieurement.

Proposition 4.2 *Si un graphe (G, λ) est S_1 -symétrique, alors le graphe G est sommet transitif.*

Preuve

Étant donné qu'un graphe de Cayley est toujours sommet transitif, par le théorème 4.4, la conclusion est directe.

□

Proposition 4.3 *Si un graphe (G, λ) est S_1 -symétrique, alors le graphe G est sommet uniforme.*

Preuve Étant donné qu'un graphe sommet transitif est sommet uniforme, par la proposition 4.2, la conclusion est directe.

□

Proposition 4.4 *Il existe des graphes étiquetés S_1 -symétrique sans qu'ils soient arête transitifs.*

Preuve Il suffit de considérer n'importe quel graphe de Cayley (nécessairement S_1 -symétrique) qui n'est pas arête transitif. Le graphe de la figure 3.4 en est un exemple.

□

Proposition 4.5 *Si un graphe (G, λ) est S_1 -symétrique, alors le graphe G est VPR.*

Preuve Directement par la proposition 4.2 et par le fait qu'un graphe sommet transitif est VPR.

□

Nous allons montrer, par les deux résultats suivants, l'indépendance entre les graphes ayant une vue symétrique et ceux qui sont arêtes transitifs.

Proposition 4.6 *Il existe des graphes étiquetés V -symétrique sans qu'ils soient arête transitifs.*

Preuve La preuve est exactement identique à celle de la proposition 4.4.

□

Proposition 4.7 *Il existe des graphes étiquetés arête transitifs sans qu'ils soient V -symétrique.*

Preuve

N'importe quel graphe arête transitif et non régulier constitue une preuve.

□

La figure 4.7 représente schématiquement la majorité de relations que nous avons vues jusqu'à maintenant. On rappelle que lorsqu'on parle de graphes V -symétrique, S_1 -symétrique ou graphes de Cayley, nous pensons à des graphes étiquetés (G, λ) , tandis que quand nous parlons de classes de graphes vus au chapitre précédent (comme le sommet ou arête transitif), il s'agit de graphes non étiquetés.

4.6 Importance de la symétrie

Maintenant, il n'est pas très difficile de se convaincre que l'utilisation des graphes possédant une ou plusieurs symétries comme modèle de système distribué peut avoir beaucoup d'avantages dont une influence positive sur la qualité et la performance du système.

Nous avons vu précédemment l'importance des graphes PR ainsi que quelques caractéristiques de graphes de Cayley qui peuvent avoir un niveau de symétrie très élevé (du moins au niveau de sommet transitif). Il a été mentionné dans [1] que la robustesse de certains graphes de Cayley (qu'on appelle "hiérarchique")

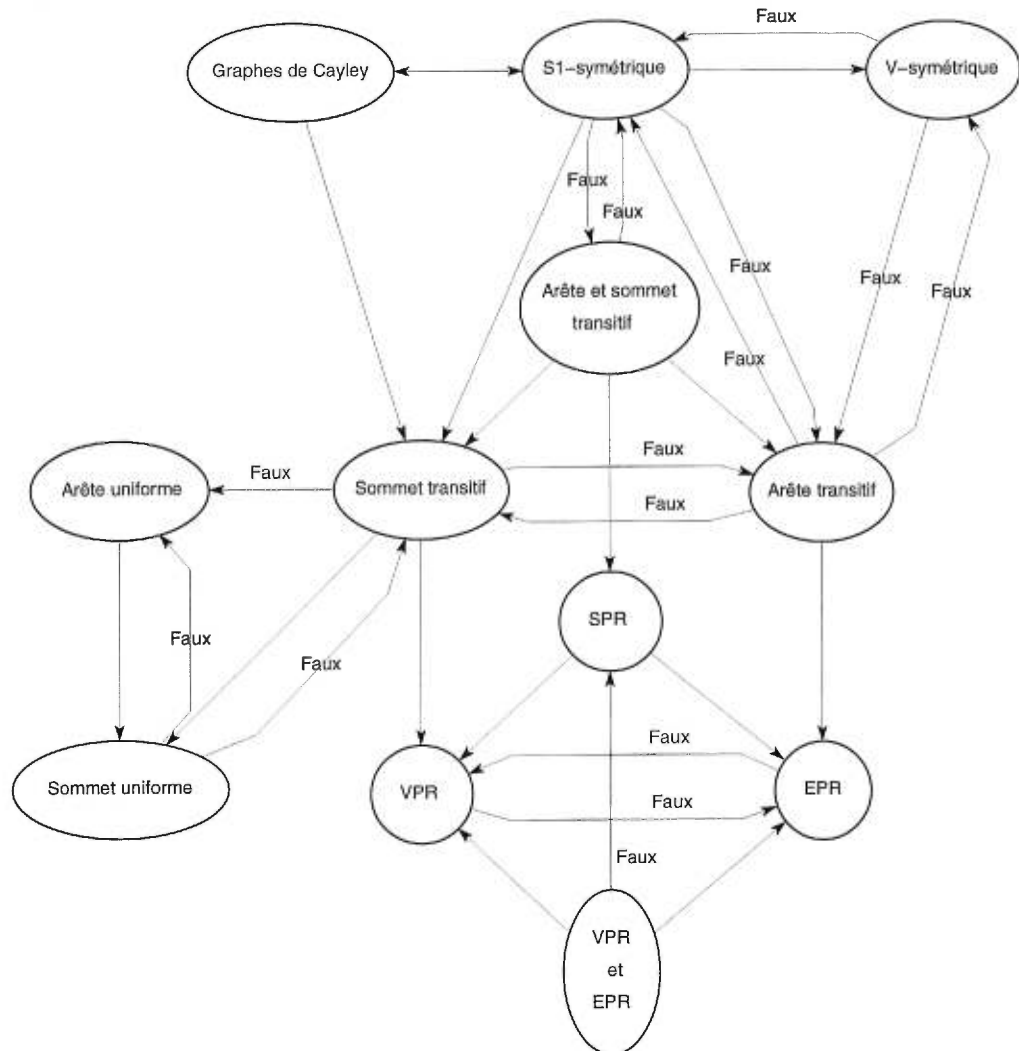


FIG. 4.7: Résumé de relations entre différentes classes de symétrie.

est *maximale* ; c'est à dire égale au degré du graphe moins un. Rappelons que la robustesse d'un graphe indique le nombre maximal de sommets qu'on peut enlever en gardant le graphe connexe. Ceci est un autre avantage qu'offre l'utilisation des graphes de Cayley comme modèle de réseau distribué.

Par ailleurs, les auteurs de [1] montrent que les graphes arête transitifs ont aussi un grand degré de robustesse, et les graphes sommet transitifs permettent la conception des algorithmes efficaces de routage. Nous avons vu que les graphes sommet transitifs ont la propriété que chaque sommet peut "voir" le graphe

exactement de la même manière. Par conséquent, le même algorithme de routage peut être utilisé sur chaque sommet (ou processeur). Notons que la symétrie dans ces graphes permet de minimiser la congestion, et la diffusion des informations peut être uniformément distribuée sur toutes les entités. Nous pouvons trouver facilement dans [48] d'autres arguments sur l'utilité des graphes de Cayley et dans [37] sur l'importance de certains réseaux symétriques.

Chapitre 5

Sens de l'orientation

Nous avons déjà mentionné que le sens de l'orientation est une notion largement utilisée dans le domaine des systèmes distribués. Cette notion est liée à la *manière* dont les arêtes du graphe sont étiquetées. Dans ce chapitre, nous allons :

- (i) Introduire le sujet en exposant les définitions fondamentales accompagnées de quelques exemples. Ainsi, nous allons étudier certaines propriétés du sens de l'orientation en mettant en relation ce dernier avec les graphes de Cayley et avec certaines classes de graphes vus précédemment ; en particulier, avec les graphes V -symétrique et S_1 -symétrique. Ceci fait l'objet de la première section.
- (ii) Présenter quelques résultats de la littérature qui nous montrent l'importance du sens de l'orientation dans les graphes quand il s'agit de réduire la complexité de certains algorithmes fondamentaux dans les systèmes distribués. Ainsi, nous exposons l'importance de cette notion quand il s'agit d'étudier la calculabilité des systèmes. La deuxième section sera consacrée à ces objectifs.

Nous suivons les notations et les définitions utilisées dans [23, 22]. Soient (G, λ) un graphe étiqueté, $P[x]$ l'ensemble de tous les parcours qui commencent

par $x \in V$ et $P[x, y]$ l'ensemble de tous les parcours qui commencent par x et qui se terminent par $y \in V$. Soient $\Lambda_x : P[x] \rightarrow \Sigma^+$ et $\Lambda = \{\Lambda_x : x \in V\}$ les extensions de λ_x et λ , respectivement, des arêtes aux parcours. Intuitivement, Λ_x est un ensemble contenant tous les étiquetages des parcours commençant par x , et Λ est l'ensemble qui contient tous les étiquetages de tous les parcours dans le graphe. Rappelons que Σ^+ est un ensemble infini qui contient tous les mots de l'ensemble d'étiquettes Σ sauf l'élément vide. L'étiquetage de chaque parcours dans le graphe est un élément de Σ^+ .

Nous définissons formellement Λ de la manière suivante :

$$\Lambda[x] = \{\Lambda_x(\pi) : \pi \in P[x]\} \text{ et } \Lambda[x, y] = \{\Lambda_x(\pi) : \pi \in P[x, y]\}.$$

Nous avons besoin de définir l'opérateur de concaténation “.”. Soient $\Sigma^k = \{a_1 \cdots a_k \mid a_i \in \Sigma, i = 1, \dots, k\}$ et $\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$. ($\Sigma^0 = \{\varepsilon\}$). Si $u = a_0 \cdots a_k$ et $v = b_0 \cdots b_l$, alors $u \cdot v = a_0 \cdots a_k b_0 \cdots b_l$. On écrit souvent uv à la place de $u \cdot v$.

Si $\pi \in P[x, y]$, alors :

$$\Lambda_x(\pi) = \lambda_x((x, x_0))\lambda_{x_0}((x_0, x_1)) \cdots \lambda_{x_k}((x_k, y))$$

où $\pi = (x, x_0)(x_0, x_1) \cdots (x_k, y)$.

5.1 Définitions, exemples et propriétés

Dans cette section, nous commençons par décrire informellement la notion du sens de l'orientation qui peut exister dans les graphes étiquetés pour arriver, ensuite, à une définition formelle de cette notion donnée pour la première fois dans [22]. Par la suite, nous présentons une manière avec laquelle nous pouvons étiqueter n'importe quel graphe. Cette manière garantit l'existence d'un sens de l'orientation qu'on appelle sens de l'orientation cordal. Enfin, la troisième sous-section est consacrée à l'étude de plusieurs propriétés surtout celles liées au sens

de l'orientation minimal.

5.1.1 Définitions

Description informelle

Supposons que les arêtes incidentes à un sommet ont chacune une étiquette distincte. Nous disons, de manière informelle, qu'un étiquetage dans un graphe (G, λ) est un sens de l'orientation (faible) s'il est possible de savoir, à partir des étiquettes associés aux arêtes, si plusieurs parcours partant d'un sommet quelconque x se terminent par le même sommet ou non. Ceci nous montre un lien entre l'étiquetage du graphe et la capacité de distinguer les parcours les uns des autres.

L'étiquetage du graphe donné par la figure 5.1 est un sens de l'orientation. Dans ce cas, on peut facilement savoir que, par exemple, les deux parcours partant du sommet A, $p_1 = [ouest, nord]$ et $p_2 = [est, nord, ouest, ouest]$ se terminent par le même sommet B.

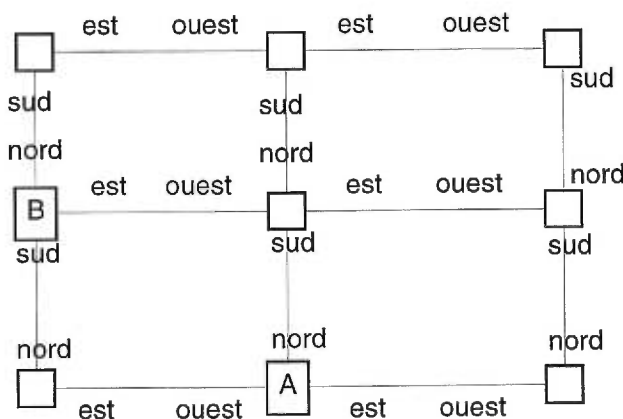


FIG. 5.1: Un graphe avec un sens de l'orientation.

Description formelle [22]

Une *fonction de codage cohérente* du graphe (G, λ) est une fonction $f : \Sigma^+ \rightarrow X$, où X est un ensemble quelconque, telle que :

$$\forall x, y, z \in V, \forall \pi_1 \in P[x, y], \pi_2 \in P[x, z] :$$

$f(\Lambda_x(\pi_1)) = f(\Lambda_x(\pi_2))$ si et seulement si $y = z$.

Par simplicité, nous allons remplacer, dans ce qui suit, le terme *fonction de codage cohérente* tout simplement par *fonction de codage*.

La figure 5.2 constitue un exemple d'un graphe étiqueté où il n'existe pas de fonction de codage. La preuve est tout simplement due au fait qu'en observant les parcours du sommet A au sommet C nous avons que $f(12) = f(3)$. Cependant, quand il s'agit d'analyser les parcours partant du sommet D , on conclut que $f(12) \neq f(3)$ et par conséquent une fonction de codage n'existe pas pour ce graphe étiqueté.

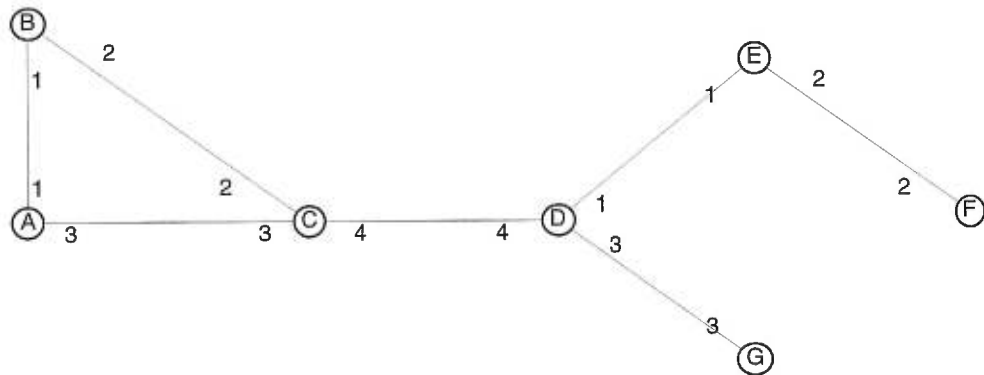


FIG. 5.2: Un graphe qui n'a pas de fonction de codage cohérente pour cet étiquetage.

Une fonction de codage est dite *homonyme* [22, 27, 26] quand :

$$\forall x, y \in V, \pi_1 \in P[x, x], \pi_2 \in P[y, y] : f(\Lambda_x(\pi_1)) = f(\Lambda_y(\pi_2))$$

Cela veut dire qu'une fonction de codage homonyme associe la même valeur à tous les cycles dans le graphe.

Définition 5.1 [22](Sens de l'orientation faible)

Un graphe (G, λ) a un sens de l'orientation faible si et seulement s'il existe une fonction de codage f pour ce système.

Le graphe de la figure 5.3 est un exemple de graphe avec sens de l'orientation faible. Par contre, le graphe donné par la figure 5.2 n'a pas de sens de l'orientation faible.

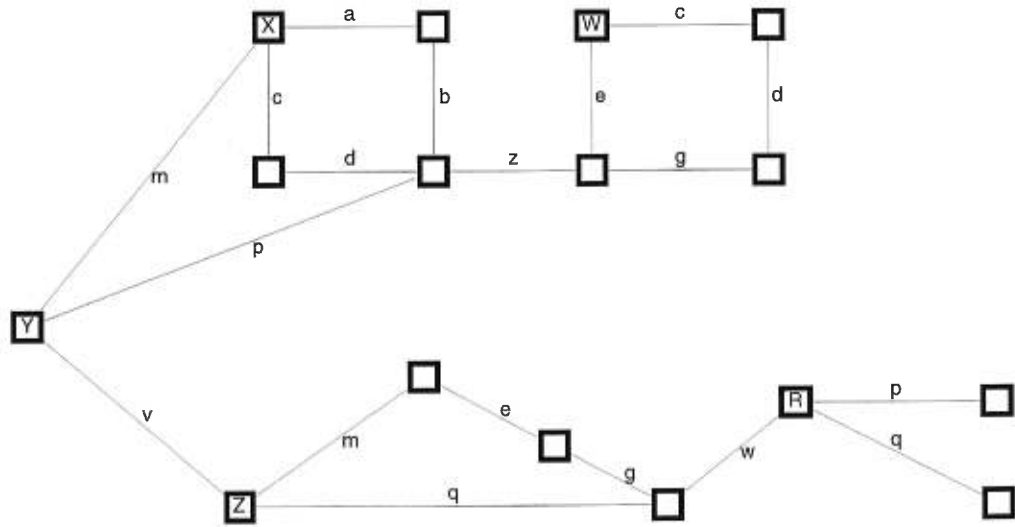


FIG. 5.3: Un graphe avec un sens de l'orientation faible.

Une *fonction de décodage cohérente*, h pour f , est une fonction qui, à partir d'une étiquette et d'un codage de parcours, retourne le résultat de l'application (qu'on appelle décodage) sur l'étiquette concaténée avec le parcours. Plus formellement, étant donné une fonction de codage f , une fonction de décodage $h : (\Sigma \times X) \rightarrow X$, où X est un ensemble quelconque, est définie comme suit :

$$h(\lambda_x((x, y)), f(\Lambda_y(\pi))) = f(\lambda_x((x, y)) \cdot \Lambda_y(\pi))$$

où $(x, y) \in E(x)$, $\pi \in P[y, z]$ et \cdot est l'opérateur de concaténation [22].

Nous allons remplacer, dans ce qui suit, le terme *fonction de décodage cohérente* tout simplement par *fonction de décodage*.

Définition 5.2 [22](Sens de l'orientation)

Un graphe (G, λ) a un sens de l'orientation si et seulement s'il existe, pour ce système, une fonction de codage f et une fonction de décodage h pour f .

Remarquons que l'existence du sens de l'orientation faible est une condition nécessaire, mais pas suffisante, pour qu'un graphe ait un sens de l'orientation. Autrement dit, tout graphe qui a un sens de l'orientation a nécessairement un sens de l'orientation faible. Cependant, nous pouvons trouver des graphes avec un sens de l'orientation faible sans un sens de l'orientation. Le graphe de la figure 5.3 en est un exemple. Si on suppose que f est une fonction de codage et h est une fonction de décodage, on voit que, dans ce graphe, à partir du sommet X on a $f((a, b)) = f((c, d))$ et à partir de W on a $f((c, d)) = f((e, g))$ alors $f((a, b)) = f((e, g))$. Par ailleurs, à partir du sommet Y , on a $h(m, f((a, b))) = f(p)$ alors $h(m, f((e, g))) = f(p)$. Maintenant, si on considère le sommet Z on aura $h(m, f((e, g))) = f(q)$ alors $f(p) = f(q)$. Ceci contredit le fait qu'à partir du sommet R on a $f(p) \neq f(q)$; par conséquent, une fonction de décodage h n'existe pas et le graphe n'a pas de sens de l'orientation.

5.1.2 Exemple

Il existe plusieurs instances de sens de l'orientation où différents modèles d'étiquetage sont utilisés. En général, ces instances sont regroupées dans cinq classes différentes : le sens de l'orientation **cartographique**, **cordal**, **contracté**, **avoisinant** et **dimensionnel**.

Dans cette section, nous allons définir seulement le sens de l'orientation* cordal sans parler des autres classes d'étiquetages dont nous pouvons trouver des descriptions dans [22].

Nous avons choisi de parler de ce type d'étiquetage pour au moins deux raisons.

*Dans ce qui suit, nous allons utiliser la notation SO pour désigner le sens de l'orientation et SOF pour désigner le sens de l'orientation faible.

L'une est que cette classe a été largement étudiée dans la littérature sur des topologies comme le graphe complet, l'anneau cordal et l'hypercube. La deuxième raison est due à la facilité de pouvoir appliquer cet étiquetage sur n'importe quel graphe. De plus, nous aurons l'occasion de définir le sens de l'orientation cordal d'une façon différente et plus compacte que ce qui existe dans la littérature.

Définition 5.3 (*SO cordal*)

Soit $G = (V, E)$ un graphe quelconque et soit $g : V \rightarrow Z_n$ une fonction bijective de numérotage. Un sens de l'orientation dans un graphe (G, λ) est dit cordal lorsque $\forall x, y \in V, \lambda_x((x, y)) = (g(y) - g(x)) \pmod n$.

Rappelons que dans Z_n nous écrivons souvent a pour $[a] = \{a + kn \mid k \in Z\}$. Le graphe de la figure 5.4 est un exemple d'un graphe avec un sens de l'orientation cordal. Ceci nous permet de supposer, sans perte de généralité, que $V = Z_n$ et que toute addition est modulo n .

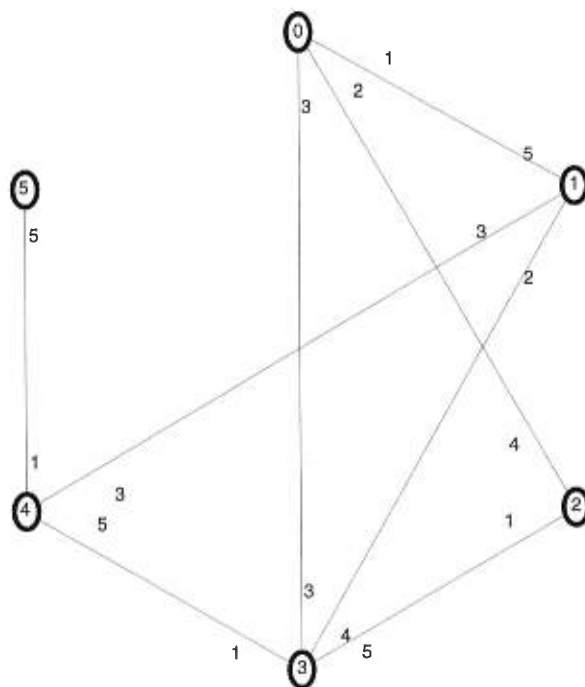


FIG. 5.4: Un graphe avec un sens de l'orientation cordal.

Il serait utile de définir la fonction de codage et la fonction de décodage pour λ , l'étiquetage du sens de l'orientation cordal.

Soit f une fonction de codage définie comme suit :

$$\forall \pi \in P[x_0], \pi = ((x_0, x_1), (x_1, x_2), \dots, (x_{m-1}, x_m))$$

$$f(\Lambda_{x_0}(\pi)) = f(\lambda_{x_0}((x_0, x_1)), \lambda_{x_1}((x_1, x_2)), \dots, \lambda_{x_{m-1}}((x_{m-1}, x_m)))$$

$$= \sum_{i=0}^{m-1} \lambda_{x_i}((x_i, x_{i+1}))$$

Et soit h une fonction de décodage définie de la manière suivante :

$$\forall (x_0, y_0) \in E(x_0), \forall \pi \in P[y_0]$$

$$h(\lambda_{x_0}((x_0, y_0)), f(\Lambda_{y_0}(\pi))) = \lambda_{x_0}((x_0, y_0)) + f(\Lambda_{y_0}(\pi))$$

Les fonction f et h sont, respectivement, celle du codage et du décodage pour le sens de l'orientation cordal. Si on choisit une paire de sommets (x, y) dans le graphe de la figure 5.4, on a que la somme (modulo n) des étiquettes sortantes est exactement la même pour tous les parcours qui commencent par x et qui se terminent par y ; on suppose qu'une étiquette sortante sur l'arête (x, y) est celle placée du côté x et n est le nombre des sommets dans le graphe. Ce fait est dû fondamentalement à la manière cyclique dont les sommets sont placés et à l'utilisation des étiquettes en fonction de la distance entre les sommets. Cette observation décrit intuitivement la fonction de codage. La fonction de décodage peut être décrite d'une manière similaire.

5.1.3 Propriétés

Un sens de l'orientation dans un graphe régulier est dit *minimal* (SOM) quand le nombre d'étiquettes (distinctes) utilisées est égal au degré du graphe. Dans cette section, nous verrons une condition nécessaire mais non suffisante, pour

qu'un sens de l'orientation minimal existe dans un graphe : ce graphe doit être sommet uniforme. Par la suite nous verrons que le fait qu'un graphe soit un graphe de Cayley avec étiquetage de Cayley ou qu'un graphe soit S_1 -symétrique constitue une condition *nécessaire et suffisante* pour l'existence de SOM. Nous nous intéressons particulièrement à étudier ce type d'étiquetage dans les graphes réguliers.

Bien que ces conditions représentent une caractérisation complète des graphes réguliers avec SOM et un étiquetage symétrique, ces mêmes conditions nous montrent des relations qui peuvent exister entre différentes classes de graphes vus antérieurement et le sens de l'orientation minimal.

Définition 5.4 [22, 14] (*Taille de l'étiquetage*)

La taille de l'étiquetage λ dans un graphe (G, λ) , est le nombre de différentes étiquettes utilisées dans Σ ; c'est à dire la taille est égale à $|\{\lambda_u(e) : e \in E(G) \text{ et } u \in V(G)\}|$.

Définition 5.5 [22, 14] (*SO minimal*)

Un sens de l'orientation dans un graphe régulier (G, λ) est minimal si et seulement si la taille de l'étiquetage λ est égale au degré du graphe G .

Condition nécessaire pour le SOM

Les deux théorèmes suivants décrivent une condition nécessaire, mais non suffisante de l'existence de SOM.

Théorème 5.1 [27, 14]

Si un graphe étiqueté et régulier a un sens de l'orientation minimal, alors ce graphe est sommet uniforme.

Le théorème 5.1 nous montre la possibilité de conclure l'inexistence de SOM quand le graphe n'est pas sommet uniforme. Rappelons qu'il est parfois plus facile de tester si le graphe est sommet uniforme que de tester le SO ou le SOF.

Examinons, à titre d'exemple, le graphe de la figure 5.5 . Il n'est pas difficile de voir que ce graphe n'est pas sommet uniforme (le sommets a appartient à un cycle de longueur 3 mais pas le sommet b). Par conséquent, nous concluons directement que ce même graphe ne peut pas avoir un SOM. Cela nous a évité de tester l'existence d'un SO ou d'un SOF quand un étiquetage est donné à ce graphe. Ceci dit, il n'est pas vrai que tester si un graphe est sommet uniforme est toujours facile surtout quand ce dernier contient un assez grand nombre de sommets.

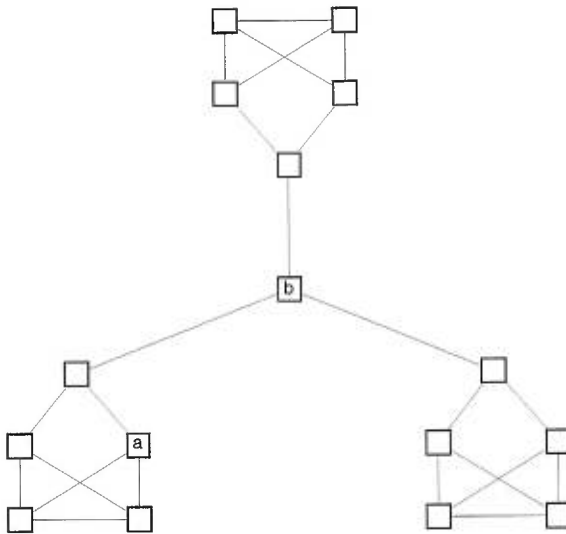


FIG. 5.5: Un graphe qui n'est pas sommet uniforme, alors n'a pas de SOM.

Théorème 5.2 [27, 14] *Un graphe sommet uniforme n'a pas nécessairement un sens de l'orientation minimal.*

Une preuve complète de ce théorème est donnée dans [14]. Le contre exemple qui a été donné est le graphe de Peterson. Ce graphe, bien qu'il soit sommet transitif et par conséquent sommet uniforme, n'a pas de sens de l'orientation minimal.

Condition nécessaire et suffisante pour le SOM

Un résultat très important qui caractérise le SOM en le mettant en relation avec les graphes S_1 -symétrique est donné dans le théorème suivant.

Théorème 5.3 [27, 14] *Si l'étiquetage λ est symétrique, alors le graphe (G, λ) a un SOM si et seulement s'il est S_1 -symétrique.*

Nous avons vu dans le troisième chapitre que dire qu'un graphe est S_1 -symétrique revient à dire que ce même graphe est un graphe de Cayley avec un étiquetage de Cayley. À partir de ce résultat, et en utilisant le théorème 5.3, nous obtenons directement le théorème suivant :

Théorème 5.4 [27, 14]

Si l'étiquetage λ est symétrique, le graphe (G, λ) a un SOM si et seulement s'il est un graphe de Cayley avec un étiquetage de Cayley.

Remarquons qu'un sens de l'orientation minimal existe seulement dans les graphes réguliers avec un étiquetage symétrique. À partir du théorème 5.3 et de la définition des graphes V -symétrique, nous avons directement le résultat suivant.

Théorème 5.5 [22, 14]

Soit (G, λ) un graphe avec un SOF. Le graphe (G, λ) est S_1 -symétrique si et seulement s'il est V -symétrique.

Le théorème suivant constitue un résultat important qui, sous certaines conditions d'étiquetage, identifie un lien entre les graphes de Cayley et le sens de l'orientation. Ce lien pourrait être très utile quand il s'agit de décider les graphes de Cayley avec un étiquetage de Cayley ou de tester si un graphe quelconque a un sens de l'orientation ou non (un chapitre ultérieur sera consacré à l'étude de ces problèmes de décision). En fait, ce théorème est une conséquence directe des trois théorèmes précédents. Rappelons que, comme nous avons vu dans le chapitre précédent, la propriété d'être V -symétrique est équivalente à celle d'avoir un étiquetage minimal et symétrique.

Théorème 5.6 [22, 27] Soit (G, λ) est un graphe régulier avec un étiquetage λ symétrique et minimal. Les énoncés suivants sont équivalents :

- Le graphe (G, λ) a un SOF.
- Le graphe (G, λ) a un SO.
- Le graphe (G, λ) est un graphe de Cayley avec un étiquetage de Cayley.

Nous mentionnons que la symétrie d'étiquetage est nécessaire pour établir le théorème 5.6 car il existe des graphes réguliers avec un sens de l'orientation minimal sans que cet étiquetage soit de Cayley. Le graphe de la figure 5.6 en est un exemple.

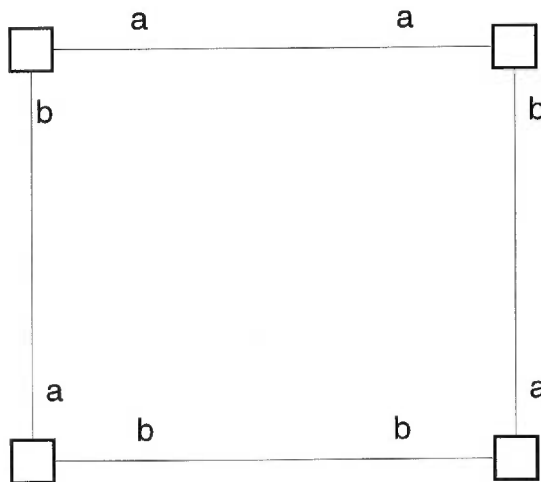


FIG. 5.6: Un graphe avec un sens de l'orientation minimal et son étiquetage n'est pas de Cayley.

Étant donné un graphe régulier avec un étiquetage minimal et symétrique, nous pouvons tester par le théorème 5.6, si ce graphe est un graphe de Cayley avec un étiquetage de Cayley. Ceci revient à tester l'existence d'une fonction de codage. De plus, par le même théorème, nous n'aurons pas à tester l'existence d'une fonction de décodage. Nous allons voir dans un chapitre ultérieur un algorithme

linéaire qui décide le SOF et dans certains cas (quand l'étiquetage est minimal et symétrique) testera si le graphe est un graphe de Cayley avec étiquetage de Cayley.

Le théorème suivant est une autre caractérisation du SOM. Cette caractérisation est liée à la manière dont les cycles dans le graphe sont étiquetés.

Théorème 5.7 [22, 27] *Soit (G, λ) un graphe régulier et étiqueté avec un SOM. L'étiquetage λ est symétrique si et seulement si la fonction de codage est homonyme.*

Le dernier résultat nous permet de remplacer dans le théorème 5.6 la condition de l'étiquetage symétrique par la condition d'homonymie définie précédemment. Le résultat du théorème 5.6 avec la condition d'homonymie a été indépendamment établi dans [9].

Problèmes ouverts

Il existe dans la littérature plusieurs problèmes liés au SOM qui demeurent ouverts. Nous en citons ici quelques uns [14] :

- Donner une caractérisation générale sur le SOM dans les graphes arbitraires.
- Étant donné un graphe régulier dans lequel il n'existe pas de SOM. Quel est le nombre minimal d'étiquettes nécessaires pour garantir l'existence de sens de l'orientation ? Est-il possible de trouver un SO avec $d(G) + 1$ étiquettes ?
- Étant donné un graphe G et un entier k . Peut-on trouver un SO pour G en utilisant un nombre d'étiquettes inférieur ou égal à k ?

5.2 Application du sens de l'orientation

Nous avons vu dans l'introduction de ce travail qu'un système distribué est un ensemble d'entités qui communiquent en échangeant des messages. Le comportement de chaque entité est dirigé par des événements à états finis. Une action se produit lorsqu'un événement arrive à une entité qui se trouve dans un état quelconque. Cette action peut être vue comme un nouvel événement dans le système avec la possibilité que l'entité en question change d'état.

Plus formellement, le comportement d'une entité est de la forme suivante : $\text{État} \times \text{Événement} \longrightarrow \text{Action}$. Rappelons que les ensembles des états et des événements sont finis. Nous considérons, dans ce système, que le coût le plus dominant dans l'exécution d'un algorithme est l'échange de messages. Par conséquent, la complexité de problèmes que nous allons étudier sera toujours calculée en fonction du nombre de messages échangés.

Dans la prochaine sous-section, nous allons montrer l'influence du sens de l'orientation sur la complexité de certains algorithmes. Par la suite, nous verrons que le sens de l'orientation a aussi un impact positif sur la calculabilité dans les systèmes distribués.

5.2.1 Influence sur la complexité

Il existe dans la littérature plusieurs exemples montrant que l'existence de sens de l'orientation dans les graphes peut aider à trouver des algorithmes (résolvant certains problèmes) plus efficaces que d'autres résolvant les mêmes problèmes en l'absence d'un sens de l'orientation. Dans cette section nous allons nous limiter à étudier la complexité de quatre problèmes fondamentaux dans les systèmes distribués. Nous comparerons le nombre de messages nécessaires à la résolution de chaque problème *avec* ou *sans* sens de l'orientation. Les définitions et les résultats dans cette section sont tirés directement de l'article [23]. Soient les problèmes suivants :

Diffusion : Ce problème consiste à envoyer, à partir d'une entité qu'on appelle initiateur, une information aux autres entités. À la fin de l'exécution, toutes les entités se trouvent dans le même état et possèdent l'information en question.

Parcours en profondeur : Il s'agit de parcourir en profondeur, tous les sommets du graphe à partir d'une entité particulière (initiateur).

Construction d'un arbre couvrant : Le problème consiste à construire à partir d'un ou de plusieurs initiateurs, un arbre couvrant (spanning tree). À la fin de l'exécution, toutes les entités se trouvent dans le même état et connaissent les liens incidents qui font partie de l'arbre.

Élection : Le problème consiste à élire parmi plusieurs entités candidates, une seule qui s'appellera Leader. À la fin de l'exécution, le Leader sera dans un état particulier (et unique) et les autres entités, qui connaissent leur Leader, seront dans un autre état. Le problème d'élire une arête au lieu d'un sommet est similaire.

Nous allons maintenant explorer la complexité de certaines solutions (aux problèmes mentionnés ci-dessus) exécutées dans un graphe (ou système) arbitraires, dans un graphe complet ou dans un hypercube. Tous ces résultats sont basés sur [23] où nous pouvons trouver des références détaillées aux solutions proposées. De plus, nous trouvons dans [5, 20, 43, 15, 47, 46, 42] différentes études des problèmes qui peuvent être résolus efficacement quand il s'agit d'utiliser des graphes ayant un sens de l'orientation.

Graphes arbitraires

Dans les graphes arbitraires (dont nous ne connaissons pas la topologie) les problèmes de diffusion et de parcours en profondeur nécessitent $\Omega(e)$ messages

même avec une orientation locale et avec des entités ayant des identificateurs distincts, où e est le nombre d'arêtes dans le graphe. Ces mêmes problèmes peuvent être résolus avec $\Theta(n)$ messages quand un sens de l'orientation avoisinant ou cordal existe dans ces graphes, où n est le nombre de sommets dans le graphe. De plus, ces problèmes nécessiteront $O(n)$ avec n'importe quel sens de l'orientation. Par ailleurs, dans les graphes arbitrairement étiquetés (mais avec une orientation locale), et avec l'absence de toute connaissance de la topologie, les problèmes d'élection et de l'arbre de recouvrement nécessitent $\Omega(e + n \log n)$ messages. Un nombre de messages dans $O(n \log n)$ suffit pour résoudre ces problèmes quand un sens de l'orientation avoisinant ou cordal existe, tandis que ces problèmes sont dans $O(n \log n)$ avec n'importe quel sens de l'orientation. La table 5.1 résume ces résultats.

	Diffusion Parcours en profondeur	Élection Arbre couvrant
Orientation locale	$\Omega(e)$	$\Omega(e + n \log n)$
SO avoisinant	$\Theta(n)$	$O(n \log n)$
SO cordal	$\Theta(n)$	$O(n \log n)$
Tout SO	$2n - 2$	$3n \log n + O(n)$

TAB. 5.1: Complexité des problèmes dans les graphes arbitraires.

Graphes complets

Dans ces graphes, et avec l'absence de sens de l'orientation, le problème de parcours en profondeur se résout avec $O(n^2)$ messages même avec une orientation locale et avec la connaissance de la topologie. Ce même problème nécessite seulement $O(n)$ messages avec n'importe quel sens de l'orientation même si le graphe est anonyme (c'est à dire si les sommets ont des identificateurs non distincts). Par contre, le problème d'élection prend $\Theta(n \log n)$ messages sans sens

de l'orientation et seulement $\Theta(n)$ quand un sens de l'orientation cordal existe. Nous rassemblons ces résultats dans la table 5.2.

	Parcours en profondeur	Élection
Orientation locale avec connaissance de la topologie	$O(n^2)$	$\Theta(n \log n)$
SO cordal	$\Theta(n)$	$\Theta(n)$
Tout SO	$\Theta(n)$?

TAB. 5.2: Complexité des problèmes dans les graphes complets

Hypercubes

Le problème de diffusion nous montre que l'existence de sens de l'orientation dans les hypercubes ne réduit pas le nombre de messages nécessaires à la résolution de ce problème. Plus précisément, ce problème nécessite $O(n)$ messages avec ou sans l'existence de sens de l'orientation. Par contre, le problème d'élection qui prend $O(n \log \log n)$ messages en présence d'une orientation locale et en l'absence de sens de l'orientation, peut être résolu avec $\Theta(n)$ messages avec un sens de l'orientation dimensionnel et seulement avec $O(1)$ messages quand le sens de l'orientation est cordal. La table 5.3 donne un résumé de ces résultats.

	Diffusion	Élection
Orientation locale	$O(n)$	$O(n \log \log n)$
SO avoisinant	$O(n)$	$O(n \log \log n)$
SO dimensionnel	$O(n)$	$\Theta(n)$
SO cordal	$O(n)$	$O(1)$

TAB. 5.3: Complexité des problèmes dans les hypercubes.

Problèmes ouverts

Un problème ouvert important est de caractériser les problèmes efficacement résolubles avec n'importe quel sens de l'orientation dans un graphe donné. De cela, peut découler la question de savoir s'il est possible de résoudre le problème d'élection dans les graphes complets avec n'importe quel sens de l'orientation et avec une complexité de $O(n)$ messages. De même, cette question peut se poser similairement quand il s'agit des hypercubes. Enfin, la question de prouver (ou de contredire) qu'une solution linéaire au problème d'élection existe avec l'absence de sens de l'orientation est toujours ouverte. Tous ces problèmes ont été posés dans [23].

5.2.2 Influence sur la Calculabilité

L'étude de la calculabilité dans un système anonyme a pris l'attention de certains chercheurs dans le domaine et par conséquent une grande partie de leurs recherches y était consacrée. Étudier la calculabilité dans un système X revient à identifier les problèmes que nous pouvons résoudre dans X [2, 38, 39, 52].

Typiquement, résoudre un problème dépend de plusieurs facteurs comme les propriétés qui décrivent la structure du système ou certaines connaissances structurelles qui sont disponibles aux entités du système.

Qu'est-ce qu'une "connaissance structurelle" ?

Une connaissance structurelle fait référence à toute connaissance sur la structure du système. En général, nous distinguons quatre types de connaissance [23] : connaissance de la topologie (T), connaissance de l'étiquetage (L), connaissance des données (D) et connaissance des entités et du système (S).

Le sens de l'orientation est de type L. Normalement, son existence est affectée par la disponibilité des autres types de connaissance.

Il existe plusieurs formes de connaissance de type T. Nous en citons ici quelques unes :

- Information métrique : représente de l’information “numérique” sur le système : le nombre d’arêtes, le degré maximal, la longueur du diamètre, etc.
- Propriétés topologiques : Information sur quelques propriétés de la topologie ; de savoir, par exemple, si le graphe est régulier, de Cayley, sommet ou arête transitif, etc.
- Classes topologiques : Information sur la classe des graphes à laquelle le graphe appartient ; de savoir, par exemple, si le graphe est un cycle (C_n), torus, hypercube, etc.
- Connaissance topologique : Information équivalente à la connaissance de la matrice d’adjacence qui décrit le graphe.
- Topologie complète : chaque sommet connaît la matrice d’adjacence et sa position dans cette matrice[23].

Des exemples de connaissances de type D sont :

Identificateur-unique : Tous les identificateurs (de sommet, par exemple) sont distincts.

Réseau anonyme : tous les identificateurs ont la même valeur.

Taille : le nombre de valeurs distinctes.

Un exemple de connaissance de type S est : *système avec un leader* : il y a une seule entité dans un état “leader” et toutes les autres entités sont dans le même état.

La puissance de calcul du sens de l’orientation dans un système anonyme a été étudiée dans [24] où on montre que cette puissance est liée à la notion de *la symétrie de vue*[52] et à la *Symétrie de voisinage* [27] étudiées dans un chapitre

précédent.

Étant donné un graphe G , soit \mathcal{G}_G la classe de tous les graphes étiquetés (G, λ) . Soit \mathcal{W}_G et \mathcal{D}_G les ensembles de tous les graphes étiquetés avec, respectivement, SOF et SO.

Nous disons qu'un problème P est *résolvable* dans \mathcal{G}_G s'il est résolvable dans G avec n'importe quel étiquetage. De la même manière, nous disons que P est résolvable dans \mathcal{W}_G s'il est résolvable dans G avec n'importe quel étiquetage qui admet un SOF.

Notons qu'il existe des problèmes qui, étant donné un graphe G , sont résolvables seulement dans un sous-ensemble de \mathcal{G}_G ; par contre, il est possible qu'un problème P , en l'absence de toute *connaissance*, soit résolvable seulement dans un sous-ensemble $\mathcal{G}'_G \subset \mathcal{G}_G$, mais, quand une certaine connaissance K sera disponible, le problème P sera résolvable dans \mathcal{G}_G .

Résultats généraux

Un résultat fondamental établi dans [24] est le suivant : Si un problème P est résolvable dans \mathcal{G}_G avec certaine connaissance de type T , alors P est résolvable dans \mathcal{W}_G sans aucune connaissance. Autrement dit, avec un sens de l'orientation, aucune autre connaissance n'est nécessaire à la résolution du problème. Le sens de l'orientation faible, alors, représente une propriété "plus forte" que toute information topologique.

Une implication importante de ce résultat est que, avec SOF il est possible de faire un routage par de plus courts chemins même si le système est anonyme.

Résultats sur des problèmes spécifiques

Les auteurs de [24], en étudiant la puissance de calcul d'un système avec SOF, ont considéré plusieurs problèmes typiques : l'élection d'un leader \mathcal{LE} , l'élection

d'une arête \mathcal{EE} , construire l'arbre de recouvrement \mathcal{SPT} , reconnaissance de topologie \mathcal{T} , reconnaissance complète de topologie \mathcal{CT} . Les auteurs ont caractérisé les classes de graphes dans lesquelles ces problèmes sont résolubles. En particulier, ils ont obtenu les résultats suivants :

- Le problème de l'élection d'un leader est résoluble dans $(G, \lambda) \in \mathcal{W}_G$, pour tout λ quand chaque sommet a une *vue* différente.
- Le problème de l'élection d'une arête et de construction d'un arbre de recouvrement sont résolubles dans $(G, \lambda) \in \mathcal{W}_G$ pour tout λ tel que le nombre d'orbites définis par la relation R' (voir la section "symétrie de voisinage") est ≤ 2 .
- Les problèmes \mathcal{T} et \mathcal{CT} sont résolubles dans \mathcal{W}_G , pour tout graphe G .

Si nous appelons ces classes des graphes, respectivement, par $\mathcal{W}_G(\mathcal{LE})$, $\mathcal{W}_G(\mathcal{EE})$, $\mathcal{W}_G(\mathcal{SPT})$, $\mathcal{W}_G(\mathcal{T})$ et $\mathcal{W}_G(\mathcal{CT})$, nous avons, pour tout graphe G :

$$\mathcal{W}_G(\mathcal{LE}) \subset \mathcal{W}_G(\mathcal{EE}) = \mathcal{W}_G(\mathcal{SPT}) \subset \mathcal{W}_G(\mathcal{T}) \subset \mathcal{W}_G(\mathcal{CT}) = \mathcal{W}_G$$

Résultats sur des problèmes de Connaissance

Nous avons vu que les auteurs de [24] ont montré que le SOF est une propriété aussi puissante que la plus forte connaissance de type T . De plus, les auteurs ont tenté d'exploiter les relations entre le SOF et les connaissances (de type T) dans le but de vérifier s'il existe une connaissance \mathcal{K} (différente de \mathcal{CT}) qui, ajoutée à un système, est aussi puissante que le SO. Ils ont montré que le SOF est strictement plus puissant que la connaissance topologique. En fait, il existe des graphes G et des problèmes qui sont résolubles dans \mathcal{W}_G mais qui ne le sont pas dans \mathcal{G}_G même avec la connaissance de la topologie. Un exemple de ces problèmes est donné dans [23].

Problème ouvert

Comme c'est mentionné précédemment, une question qui a été posé dans [24, 23] est : *Existe-t-il une connaissance \mathcal{K} qui, ajoutée à un système, est aussi puissante que le SO ?*

Chapitre 6

Méthode pour décider le sens de l'orientation faible

Nous avons vu dans le chapitre précédent que l'existence du sens de l'orientation (faible) dans les graphes peut avoir un grand impact positif sur la complexité de calcul de certains problèmes. Il serait donc utile de vérifier ou de décider si un graphe (G, λ) quelconque a un $\text{SO}(\mathbb{F})$ ou non. Une méthode pour décider le SOF a été donnée dans [7]. L'objectif principal de ce chapitre est d'exposer cette méthode et les étapes détaillées à exécuter afin de résoudre ce problème. Rappelons que nous avons implanté cette méthode de manière séquentielle afin de vérifier si un graphe étiqueté quelconque (n'ayant pas un très grand nombre de sommets) a un sens de l'orientation faible ou non. Nous analysons la complexité de calcul séquentiel et parallèle de cet algorithme et nous montrons comment, en utilisant cette même méthode, nous pouvons vérifier si un graphe donné est un graphe de Cayley avec un étiquetage de Cayley. Rappelons que les mêmes auteurs ont fourni une méthode (dont nous ne parlons pas dans ce travail) pour décider le sens de l'orientation. Bien que la complexité de cette solution est polynômiale, elle demeure très inefficace dans la pratique. Nous discutons brièvement dans la dernière section de ce chapitre du problème qui consiste à construire un sens de l'orientation dans les graphes et nous fournissons les références qui traitent de ce

sujet. Pour des raisons de compatibilité avec les notations utilisées dans [7], nous ferons deux exceptions dans ce chapitre : les graphes sont orientés, et la notation (x, y) représente la paire de sommets x et y qui peuvent être non adjacents.

6.1 Décision du sens de l'orientation faible

Les auteurs dans [7] ont proposé un algorithme qui décide le SOF. Ils ont démontré que décider le sens de l'orientation faible peut se faire très efficacement en parallèle ; plus précisément, la décision de ce problème est, en utilisant un modèle CRCW PRAM de calcul, dans AC^1 pour n'importe quel graphe, où AC^1 est la classe de problèmes résolubles en parallèle en un temps dans $O(\log n)$ et en utilisant un nombre polynômial de processeurs (dans ce cas c'est n^6 processeurs).

L'idée de la solution est basée sur une caractérisation combinatoire du SOF. Le sens de l'orientation faible a été montré équivalent à une condition combinatoire dans les graphes étiquetés appelée *uniformité*. Décider cette même condition est, comme c'est prouvé dans le même travail, dans AC^1 .

6.1.1 La méthode

Soit $G = (V, E)$ un graphe orienté, un étiquetage du graphe G est une fonction $\lambda : E \rightarrow \Sigma$ où Σ est un ensemble fini d'étiquettes. Soit $P[x, y] \subseteq E^*$ l'ensemble des parcours qui commencent par x et qui se terminent par y . Soit $\pi \in P[x, y]$ si $\pi = x_0x_1 \cdots x_n$ et $x = x_0$ et $y = x_n$, on définit $\lambda^*(\pi) = \lambda(x_0) \cdots \lambda(x_n)$.

Définissons :

$$L(x, y) = \{\lambda^*(\pi) \mid \pi \in P[x, y]\}$$

où $\lambda^* : E^* \rightarrow \Sigma^*$ et pour tout $I \subseteq V^2$ soit

$$L_I = \bigcup_{(x,y) \in I} L(x, y)$$

Remarquons que Σ^* est l'ensemble de toutes les suites d'étiquettes, et $\lambda^*(\pi)$ est l'étiquetage du parcours π . Par conséquent, $L(x, y)$ est intuitivement l'ensemble (ou le langage) des étiquetages de tous les parcours entre x et y . Par ailleurs, si I est un ensemble de paires de sommets, le langage de I , (L_I) , sera la réunion de tous les langages de ses éléments.

L'objectif principal de la prochaine proposition est de montrer qu'à partir d'un graphe étiqueté, il est possible de construire des classes d'équivalence contenant des paires de sommets telles que deux paires de sommets (x, y) et (z, w) appartiendront à la même classe si et seulement s'il existe deux parcours P_1 et P_2 tels que l'étiquetage de $P_1[x, y]$ est égal à l'étiquetage de $P_2[z, w]$.

Proposition 6.1 [7]

Soit T le sous-ensemble minimal de 2^{V^2} tel que :

Pour tout $x, y \in V$, $\{(x, y)\} \in T$, et

$$\text{Si } I, J \in T \text{ et } L_I \cap L_J \neq \emptyset \text{ alors } I \cup J \in T. \quad (6.1)$$

Soit T^{\max} l'ensemble des éléments maximaux de T avec l'ordre partiel défini par l'inclusion.

On a :

- (i) T et T^{\max} peuvent se construire d'une manière effective ;*
- (ii) T^{\max} est une partition de V^2 ;*
- (iii) La famille des ensembles $\{L_I \mid I \in T^{\max} \wedge L_I \neq \emptyset\}$ est une partition de L_{V^2} .*

Remarquons que la condition 6.1 revient à regrouper les deux ensembles (contenant des paires de sommets) I et J s'il existe un $(x, y) \in I$, $(z, w) \in J$ et deux parcours P_1 et P_2 tels que l'étiquetage de $P_1[x, y]$ est égal à l'étiquetage de $P_2[z, w]$. De même, le fait de préciser que T^{max} contient seulement les éléments maximaux de T de l'ordre partiel par l'inclusion, revient à dire qu'un sous-ensemble $I \in T$ n'apparaît pas dans T^{max} s'il existe un $J \in T$ tel que $I \subset J$. Enfin, il n'est pas très difficile de voir que T^{max} contient les classes d'équivalence dont on a parlé dans l'introduction de la proposition.

Définition 6.1 (*monodrome*)[7]

Étant donné un ensemble X , une partition Π de X^2 est dite monodrome si et seulement si ses éléments sont des graphes des fonctions partielles de X dans X . Autrement dit, si :

$$\forall I \in \Pi, (x, y), (x, z) \in I \implies y = z.$$

Définition 6.2 (*Étiquetage uniforme*)[7]

Un étiquetage d'un graphe (G, λ) est uniforme si et seulement si T^{max} (qui correspond à ce graphe) est monodrome.

Nous remarquons que la définition de monodrome ci-dessus correspond à la fonction de codage donnée dans le chapitre précédent. Par conséquent, vérifier l'existence d'un sens de l'orientation faible dans un graphe revient à vérifier si son étiquetage est uniforme. Voici une formalisation de ce résultat dont nous ne donnons pas la preuve :

Corollaire 6.1 [7]

Si un graphe (G, λ) a un sens de l'orientation faible, alors l'étiquetage λ est uniforme.

Corollaire 6.2 [7]

Si l'étiquetage d'un graphe (G, λ) est uniforme, alors le graphe (G, λ) a un sens de l'orientation faible.

Théorème 6.1 [7]

Soit (G, λ) un graphe étiqueté et M une matrice booléenne $V^2 \times V^2$ définie comme suit :

$$M((x, x'), (y, y')) = 1 \iff (x, y), (x', y') \in E \wedge \lambda((x, y)) = \lambda((x', y')).$$

Alors,

$$L(x, y) \cap L(x', y') \neq \emptyset \iff M^*((x, x'), (y, y')) = 1.$$

Ce théorème est nécessaire à la construction d'une matrice d'adjacence (M) qui décrit les relations d'étiquetages entre toutes les paires de sommets. Ces relations correspondent à la définition de l'ensemble T décrit ci-dessus.

6.1.2 L'algorithme

Dans cette section, en se basant sur les définitions et les notations évoquées ci-dessus, nous allons décrire les étapes nécessaires pour décider le sens de l'orientation faible.

Problème 1 SENS DE L'ORIENTATION FAIBLE

Donnée Un graphe (G, λ) .

Question Est-ce que (G, λ) a un sens de l'orientation faible ?

Pour résoudre ce problème, il suffit de vérifier si l'étiquetage du graphe en question est uniforme (corollaire 6.2). Pour ce faire, nous pouvons exécuter les étapes suivantes [7] :

- Construire la matrice M .
- Construire M^* , la fermeture transitive de M .
- Construire la matrice N :

$$N((x, y), (x', y')) = M^*((x, x'), (y, y')).$$

- Construire $T^{max} = N^*$, la fermeture transitive de la matrice N .
- Vérifier si T^{max} est monodrome.
- Si T^{max} est monodrome, nous concluons que le graphe (G, λ) a un sens de l'orientation faible. Sinon, alors ce graphe n'a pas de tel sens de l'orientation.

On remarque que la manière dont on construit la matrice N correspond exactement à la définition de l'ensemble T . Par conséquent, la fermeture transitive de N correspondra à l'ensemble T^{max} dont il faut vérifier la “monodromie”. Cette condition, comme nous l'avons vu, correspond à la fonction de codage qui doit exister pour avoir un sens de l'orientation faible.

Complexité séquentielle de calcul

Nous allons donner une brève analyse de la complexité en fonction de temps de l'algorithme que nous venons de voir, exécuté de manière *séquentielle*.

Étant donné que la dimension de la matrice M est $n^2 \times n^2$, le calcul, qui consiste à créer n^4 éléments de cette matrice, prendra un temps dans $\Theta(n^4)$.

Ensuite, nous pouvons utiliser un algorithme (comme celui de *Warshall*) qui calcule la fermeture transitive d'une matrice de dimension $n \times n$ avec un temps dans $\Theta(n^3)$. Par conséquent, le temps de calcul nécessaire de la fermeture transitive de $M(n^2)(n^2)$ est dans $\Theta(n^6)$.

De la même manière, la construction de la matrice N et sa fermeture transitive prendront un temps, respectivement, dans $\Theta(n^4)$ et $\Theta(n^6)$. Enfin, la vérification de la condition de monodrome ou de l'uniformité nécessitera un temps dans $\Theta(n^2)$. Globalement, la complexité en temps pour calculer toutes les étapes décrites ci-dessus est dans $\Theta(n^6)$ (le terme dominant).

Complexité parallèle de calcul

Si nous voulons exécuter en *parallèle* les étapes qui décident le problème 1, nous obtiendrons la complexité suivante : La construction de la matrice M peut

se faire en temps constant en utilisant n^4 processeurs. La fermeture transitive de M peut se faire en temps *logarithmique* en utilisant n^6 processeurs. Ensuite, la construction de N et sa fermeture transitive, N^* , peuvent s'achever, comme M et M^* , dans un temps, respectivement, constant et logarithmique. Enfin, la vérification de monodrome prendra un temps constant. Au total, la complexité parallèle de cet algorithme est dans AC^1 , la classe de problèmes résolubles en parallèle en un temps logarithmique en utilisant un nombre *polynômial* de processeurs (dans notre cas c'est n^6 processeurs).

6.2 Décision des graphes de Cayley

Nous montrons dans cette section comment, à partir de la solution du problème 1 décrite précédemment, nous pouvons décider les graphes de Cayley avec un étiquetage de Cayley. la solution est basée sur celle du problème 1 et le théorème suivant que nous avons déjà vu dans un chapitre précédent.

Théorème 6.2 [22, 27] *Soit (G, λ) un graphe avec étiquetage minimal et symétrique. Les énoncés suivants sont équivalents :*

- (i) *Le graphe (G, λ) a un sens de l'orientation faible.*
- (ii) *Le graphe (G, λ) a un sens de l'orientation.*
- (iii) *Le graphe (G, λ) est un graphe de Cayley avec un étiquetage de Cayley.*

Problème 2 GRAPHE DE CAYLEY

Donnée Un graphe (G, λ) .

Question Est-ce que le graphe (G, λ) est un graphe de Cayley avec étiquetage de Cayley?

6.2.1 L'algorithme

En utilisant la même solution pour répondre à la question 1 et en se basant sur le théorème 6.2 nous décidons le problème 2 de la manière suivante :

- Si ce graphe est régulier *et* si l'étiquetage λ est minimal et symétrique, alors passer à la prochaine étape. Sinon, le graphe (G, λ) n'est pas un graphe de Cayley ou bien l'étiquetage donné ne correspond pas à un étiquetage de Cayley pour ce graphe.
- Si le graphe (G, λ) a un sens de l'orientation faible (vérifiable par la solution 1), alors ce graphe est un graphe de Cayley avec un étiquetage de Cayley. Sinon, ce graphe n'est pas un graphe de Cayley ou bien l'étiquetage donné ne correspond pas à un étiquetage de Cayley.

Notons que la complexité de calcul pour résoudre le problème 2 est essentiellement identique à celle du problème 1. Ceci est simplement dû au fait que la partie la plus dominante dans l'exécution de cet algorithme est celle qui décide le SOF. Ainsi, pour une exécution séquentielle de l'algorithme, la complexité est dans $\Theta(n^6)$ où n est le nombre de sommets du graphe. Pour une exécution parallèle, ce problème appartiendra à AC^1 .

6.3 Remarques sur la construction d'un sens de l'orientation faible

Nous avons vu que l'existence du sens de l'orientation a un impact positif sur la complexité de communication entre différents processeurs dans un réseau. Nous avons intérêt à réfléchir sur les possibilités, au lieu de tester si un étiquetage donné est un SO, de *construire* explicitement dans les graphes non étiquetés un étiquetage qui garantit l'existence d'un SO .

Ce problème a été étudié dans [50, 49] où on montre que tout algorithme qui cherche un SO dans les cliques, hypercubes ou les graphes arbitraires nécessite

un échange de messages dont la complexité est dans $\Omega(e - \frac{1}{2}n)$ où le réseau contient e arêtes et n sommets.

Ce résultat n'est pas très efficace sur des graphes assez denses comme les cliques, par contre, il est assez intéressant quand il s'agit de graphes avec un nombre linéaire d'arêtes comme le torus ou le cycle cordal avec un degré constant. Plus de détails sur la complexité des messages nécessaires pour construire un sens de l'orientation dans certains graphes peuvent être trouvés dans [23].

Une construction optimale du SO est donnée dans [8] où les auteurs s'intéressaient à construire un SO en utilisant un nombre minimal d'étiquettes. Ce nombre varie entre le degré du graphe et le nombre de sommets. En utilisant des techniques algébriques et géométriques, les auteurs ont montré la possibilité de construire un SO dans des graphes assez grands (en fonction du nombre de sommets) avec un nombre d'étiquettes assez petit. L'idée est de construire un SO sur des graphes assez grands en connectant ensemble, d'une manière appropriée, des graphes plus petits. Avec cette technique de construction primitive, ils ont obtenu un sens de l'orientation minimal pour les graphes de Cayley.

Conclusion

Ce travail est une synthèse de travaux sur les graphes symétriques et les relations entre eux, ainsi que sur les propriétés et l'utilité du sens de l'orientation. Nous avons tenté de donner des définitions exactes et précises et de redéfinir certaines notions dont les définitions étaient imprécises dans la littérature. On a pu tracer des nouveaux liens entre certaines classes de graphes et on ne s'est pas privé de soumettre des remarques et des questions qu'on a jugées pertinentes.

Parmi les points faisant partie de la continuité de ce travail, nous citerons d'abord le travail d'élargir la hiérarchie des graphes possédant une symétrie quelconque. Bien qu'élargir cette hiérarchie consiste à étudier des nouvelles symétries de graphes, cela impliquerait une étude connexe à laquelle nous nous sommes intéressés tout au long de ce travail : donner des nouvelles caractérisations de ces classes de graphes et découvrir de nouvelles relations entre elles. Ceci constitue pour nous un travail enrichissant sur le plan théorique et nécessaire en vue de l'application surtout quand il s'agit de choisir un graphe comme modèle de réseau de communication.

Nous pensons aussi à des questions étroitement liées à l'étude de ces relations entre les graphes. Par exemple, étant donné qu'un graphe appartient en même temps à deux classes de symétries, C_1 et C_2 , est-ce qu'il appartient aussi à une autre classe particulière C_3 ? À titre d'exemple, est-ce qu'un graphe sommet transitif et arête uniforme est aussi arête transitif?

Une autre question qui pourrait être étudiée est de caractériser, à partir d'un graphe G , transitif (par exemple, arête ou distance transitif) et non étiqueté, les

graphes (G, λ) pour lesquels des automorphismes préservent aussi les étiquettes. Nous avons vu que si cette condition est vérifiée pour un graphe sommet transitif G , le graphe (G, λ) sera un graphe de Cayley.

Du côté de l'application, il serait utile de trouver des algorithmes efficaces pour tester si un graphe est sommet uniforme ou arête uniforme. Rappelons qu'un graphe sommet transitif (ou arête transitif) est forcément sommet uniforme (ou respectivement arête uniforme). Cependant, tester si un graphe est sommet uniforme peut s'avérer utile quant à l'étude du sens de l'orientation surtout quand on sait qu'un graphe ne peut avoir un sens de l'orientation minimal s'il n'est pas sommet uniforme.

Par ailleurs, étant donné que la notion du sens de l'orientation est très récente et souvent mal connue, nous croyons qu'il serait utile de redéfinir cette notion en terme d'*automates* (qu'on peut voir comme des graphes orientés et étiquetés). Cette redéfinition impliquerait la redéfinition des notions connexes vues dans ce travail comme la symétrie de vue et la symétrie de voisinage. Ainsi, les propriétés de toutes ces notions seront exploitées en fonction de la théorie des automates. L'objectif principal de cette proposition serait de mieux comprendre et caractériser le sens de l'orientation ou tout simplement de le voir sous un autre angle. Cela pourrait enrichir notre compréhension du sujet et permettrait éventuellement de tracer des nouveaux liens entre les propriétés du sens de l'orientation et celles des graphes symétriques.

Bibliographie

- [1] S.B. Akers et B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4) :555–566, 1989.
- [2] H. Attiya, M. Snir et M.K. Warmuth. Computing on an anonymous ring. *Journal of the A.C.M.*, 35(4) :845–875, 1988.
- [3] N.L. Biggs. *Finite groups of automorphisms*. Cambridge : Cambridge University Press, 1971.
- [4] N.L. Biggs. *Algebraic graph theory*. Cambridge : Cambridge University Press, 1974.
- [5] G. v. Bochmann, P. Flocchini et D. Ramazani. Distributed objects with sense of direction. Dans *1st Workshop on Distributed Data and Structures*, pages 1–15, Orlando, 1998.
- [6] F. Boesch et R. Tindell. Connectivity and symmetry in graphs. Dans F. Harary et J.S. Maybee, éditeurs, *Graphs and Applications*, pages 53–67. John Wiley and Sons, 1984.
- [7] P. Boldi et S. Vigna. On the complexity of deciding sense of direction. Dans *Proc. of 2nd Colloquium on Structural Information and Communication Complexity*, pages 39–52, Olympia, 1995.
- [8] P. Boldi et S. Vigna. On some constructions which preserve sense of direction. Dans *3rd International Colloquium on Structural Information and Communication Complexity*, number 6, pages 47–57, Siena, 1996.

- [9] P. Boldi et S. Vigna. Minimal sense of direction and decision problems for Cayley graphs. *Information Processing Letters*, 64 :299–303, 1997.
- [10] J.A. Bondy et U.S.R. Murthy. *Graph theory with applications*. New York, American Elsevier, 1976.
- [11] I.Z. Bouwer. An edge but not vertex transitive cubic graph. *Canad.Math.Bull.*, 11(4) :533–535, 1968.
- [12] I.Z. Bouwer. Vertex and edge transitive, but not 1-transitive graphs. *Canad.Math.Bull.*, 13 :231–237, 1970.
- [13] I.Z. Bouwer. On edge but not vertex transitive regular graphs. *J. Combin. Theory*, (B) 12 :32–40, 1972.
- [14] P. Flocchini. Minimal sense of direction in regular networks. *Information Processing Letters*, 61 :331–338, 1997.
- [15] P. Flocchini et B. Mans. Senses of direction in hypercube networks. Dans *Proc. of 6th International Conference on Parallel and Distributed Computing and Systems*, pages 109–112, Washington, 1994.
- [16] P. Flocchini, B. Mans, A. Roncato et N. Santoro. Levels of sense of direction in distributed systems. Dans *1st International Conference on Principles of Distributed Systems*, pages 319–331, Chantilly, 1997.
- [17] P. Flocchini, B. Mans et N. Santoro. Sense of direction : formal definition and properties. Dans *Proc. of 1st Colloquium on Structural Information and Communication Complexity*, International Informatic Series 1, pages 9–34, Ottawa, Canada, 1994. Carleton University Press.
- [18] P. Flocchini, B. Mans et N. Santoro. Sense of direction, associativity, and symmetricity in distributed systems. Dans *Proc. of 26th SE Conference on Combinatorics, Graph Theory and Computing*, *Congressus Numerantium* 110, pages 49–63, 1995.

- [19] P. Flocchini, B. Mans et N. Santoro. Translation properties of sense of direction. Dans *Proc. of 2nd Colloquium on Structural Information and Communication Complexity*, pages 13–26, Olympia, 1995.
- [20] P. Flocchini, B. Mans et N. Santoro. On the impact of sense of direction on message complexity. *Information Processing Letters*, 63(1) :23–31, 1997.
- [21] P. Flocchini, B. Mans et N. Santoro. Sense of direction : definition, properties and classes, 1997. To appear in *Networks*. Preliminary version in *Proc. of 1st Colloquium on Structural Information and Communication Complexity*, 9-34,1994.
- [22] P. Flocchini, B. Mans et N. Santoro. Sense of direction : definition, properties and classes. *Networks*, 32(3) :165–180, 1998.
- [23] P. Flocchini, B. Mans et N. Santoro. Sense of direction in distributed computing. Dans *Proc. of 12th International Symposium on Distributed Computing (DISC 98)*, pages 1–15, Andros, 1998.
- [24] P. Flocchini, A. Roncato et N. Santoro. Computing on anonymous networks with sense of direction. *Theoretical Computer Science*. To appear.
- [25] P. Flocchini, A. Roncato et N. Santoro. Complete symmetries and minimal sense of direction in labeled graphs. Dans *Proc. of 27th SE Conference on Combinatorics, Graph Theory and Computing*, volume 121 of *Congressus Numerantium*, pages 3–18, 1996.
- [26] P. Flocchini, A. Roncato et N. Santoro. Symmetries and sense of direction in labeled graphs. Technical Report TR-96-14, School of Computer Science, Carleton University, Ottawa, K1S 5B6 Canada, 1996.
- [27] P. Flocchini, A. Roncato et N. Santoro. Symmetries and sense of direction in labeled graphs. *Discrete Applied Mathematics*, 87 :99–115, 1998.
- [28] P. Flocchini, A. Roncato et N. Santoro. Backward consistency and sense of direction in advanced distributed systems. Dans *Proc. of the 18th ACM Symposium on Principles of Distributed Computing*, 1999.

- [29] P. Flocchini et N. Santoro. Proper coloring and sense of direction. Submitted.
- [30] P. Flocchini et N. Santoro. Topological constraints for sense of direction. *International Journal on Foundations of Computer Science*, 9(2) :179–198, 1998.
- [31] J-L Fouquet et G. Hahn. Cycle regular graphs need not be transitive. Manuscript.
- [32] E. Fuller et B. Krishnamurthy. Symmetries in graphs : An annotated bibliography. Technical Report CR-86-03, Tektronix Lab., 1986.
- [33] G. Hahn. Graph homomorphisms : structure and symmetry. *Dans : Graph symmetry, Kluwer Academic Publishers, Netherlands, Eds. : G. Hahn et G.Sabidussi*, pages 107–166, 1997.
- [34] Z. Harpaz et S. Zaks. Sense of direction in complete distributed networks. Technical Report TR-453, Département d’informatique, Technion, Haifa 32000, Israel, Avril 1987.
- [35] M.C. Heydemann. Cayley graphs and interconnection networks. *Dans : Graph symmetry, Kluwer Academic Publishers, Netherlands, Eds. : G. Hahn et G.Sabidussi*, pages 167–224, 1997.
- [36] D.F. Holt. A graph which is edge transitive but not arc transitive. *J. Graph theory*, 5 :201–204, 1981.
- [37] L.M. Huisman et S.Kundu. Highly reliable symmetric networks. *IEEE Transaction on Parallel and Distributed Systems*, 5(1), 1994.
- [38] E. Kranakis et D. Krizanc. Distributed computing on anonymous hypercubes. *Journal of Algorithms*, 23 :32–50, 1997.
- [39] E. Kranakis et N. Santoro. Distributed computing on anonymous hypercubes with faulty components. *Dans Proc. of 6th Int. Workshop of Distributed Algorithms*, Lecture Notes in Computer Science 647, pages 253–263. Springer-Verlag, 1992.

- [40] S. Lakshmivarahan, Jung-Sing Jwo et S.K. Dhall. Symmetry in interconnection networks based on Cayley graphs of permutation groups : a survey. *Parallel Computing*, 19 :361–407, 1993.
- [41] F. Thomson Leighton. *Introduction to parallel algorithms and architectures ; arrays, trees, hypercubes*. M.I.T. Press, 1992.
- [42] M.C. Loui, T.A. Matsushita et D.B. West. Election in complete networks with a sense of direction. *Information Processing Letters*, 22 :185–187, 1986. see also *Information Processing Letters*, vol.28, p.327, 1988.
- [43] B. Mans et N. Santoro. On the impact of sense of direction in arbitrary networks. Dans *Proc. of 14th International Conference on Distributed Computing Systems*, pages 258–265, Poznan, 1994.
- [44] D. Matula et D. Dolev. Path regular graphs. Technical Report STAN-CS-80-807, Stanford department of computer sciences, Stanford University, 1980.
- [45] N. Norris. Universal covers of graphs : isomorphism to depth $n-1$ implies isomorphism to all depths. *Discrete Applied Mathematics*, 56 :pages 61–74, 1995.
- [46] N. Santoro. Sense of direction, topological awareness and communication complexity. *SIGACT NEWS*, 2(16) :50–56, Summer 1984.
- [47] N. Santoro, J. Urrutia et S. Zaks. Sense of direction and communication complexity in distributed networks. Dans *Proc. of the 1st International Workshop on Distributed Algorithms*, pages 123–132, Ottawa, 1985. Carleton University Press.
- [48] S.T. Schibell et R.M. Stafford. Processor interconnection networks from Cayley graphs. *Discrete Applied Mathematics*, 40 :333–357, 1992.
- [49] G. Tel. Network orientation. *International Journal of Foundations of Computer Science*, 5(1) :1–41, 1994.

- [50] G. Tel. Sense of direction in processor networks. Dans *Proc. of Conference on Theory and Practice of Informatics*, Lecture Notes in Computer Science 1012, pages 50–82. Springer-Verlag, 1995.
- [51] W.T.Tutte. *Connectivity in Graphs*. University of Toronto Press, 1966.
- [52] M. Yamashita et T. Kameda. Computing on anonymous networks, part I : characterizing the solvable cases. *IEEE Transaction on Parallel and Distributed Computing*, 7(1) :69–89, 1996.