



Homards, Araignées, Chenilles et autres arbres : Une dendrologie théorique des graphes

par

Hélène-Sarah Bécotte-Boutin

Sciences de la gestion
(Méthodes analytiques de gestion)

*Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M. Sc.)*

Mars, 2013

© Hélène-Sarah BÉCOTTE-BOUTIN, 2013

Résumé

Le nombre de graphes connus et étudiés est particulièrement imposant. L'objectif de ce mémoire est donc de classer les différentes familles de graphes. Étant une famille distincte, les arbres sont des graphes connexes, acycliques et finis. Afin de restreindre cette recherche, seules les sous-familles de cette catégorie ont été considérées. Le but ultime de cette recherche est de créer un outil, sous la forme de programme informatique, qui servira à identifier les graphes grâce à des caractéristiques discriminantes.

Plusieurs familles existent donc parmi les arbres et celles-ci ont des propriétés bien distinctes. Dans ce mémoire, celles-ci seront tout d'abord décrites grâce à une revue de littérature, qui prendra en compte les définitions les plus populaires utilisées pour décrire ces dernières et leurs caractéristiques connues. Par la suite, ces caractéristiques seront traduites grâce à des invariants, de manière à les rendre discriminantes. Dans cette même section, des preuves mathématiques formelles seront élaborées dans le but de prouver que ces caractéristiques sont discriminantes pour chaque espèce d'arbre.

Le mémoire démontrera de plus la filialité entre certaines familles et les différentes inclusions entre celles-ci. Tel un arbre généalogique, l'effet de parenté entre plusieurs familles n'est pas nécessairement exclusif. Une famille peut donc aisément avoir plus d'un parent et partager un enfant avec une autre espèce d'arbre. Ces inclusions seront évidemment démontrées et prouvées.

Ensuite, la structure du programme qui servira à identifier et reconnaître les différents arbres sera décrite. À différentes étapes du processus, des familles pourront être identifiées. Considérant l'inclusion d'une famille à l'intérieur d'une autre, ce programme prend en compte les sous-caractéristiques de manière à guider l'utilisateur pour four-

nir à ce dernier une catégorisation précise. Une suite de questions sera utilisée à cette fin.

Mots clés : Théorie des graphes, classification, arbres.

Abstract

The number of graphs actually known and studied is quite huge. The goal of this master thesis is to classify the families of graphs. Being a distinct family, the trees are acyclics, connected and finite graphs. To restrain this research, only the sub-families of this precise family were considered here. The ultimate goal of this research is to create a software which will identify graphs, using discriminant characteristics.

Whitin the trees, many sub-families exists and each of them have particular properties and characteristics. In this thesis, those families will be described using definitions and known properties from the literature. Thereafter, these characteristics will be translated using some invariants, to make them discriminant. Along with this description, formal mathematical proofs will be given in order to prove that those characteristics are discriminant for each tree family.

The filiation between some families will also be presented in the article. As a family tree, the kinship between some families are not necessarily exclusive. Therefore, a family can have more than a relative and can share a kid with another tree specie. Those inclusions will be described and proved.

The structure of the software will be shortly described. Considering the inclusions of some families within another ones, this tool will take into account the sub-characteristics, to guide the user and to provide a precise categorization.

Keywords : Graph Theory, Classification, Trees.

Table des matières

Résumé	iii
Abstract	iv
Lexique	xi
Remerciements	xiii
Avant-propos	xiv
1 Introduction	1
1.1 Introduction	1
1.2 Plan du mémoire	3
2 La Classification	4
2.1 La classification	4
2.1.1 La classification en théorie des graphes	6
3 Lobsters, Spiders, Caterpillars and Other Trees : A Graph Theoretical Dendrology	7
3.1 Introduction	11
3.2 Notation and Definitions	12
3.2.1 Main definitions	12
3.2.2 List of invariants	13
3.3 The Trees Families	15
3.3.1 Path	15
3.3.2 Star	16

3.3.3	Caterpillar	16
3.3.4	Claw	17
3.3.5	Brush	18
3.3.6	Comb	19
3.3.7	Spider	21
3.3.8	Spur	21
3.3.9	Y-Tree	23
3.3.10	Fork	24
3.3.11	Comet	25
3.3.12	Double Star	26
3.3.13	Double Comet	26
3.3.14	Bamboo	27
3.3.15	Banana	28
3.3.16	Double Spider	29
3.3.17	Cayley tree	29
3.3.18	Lobster	31
3.3.19	Firecracker	32
3.3.20	Superstar	33
3.3.21	Dandelion	34
3.3.22	Olive	35
3.3.23	Spraying Pipe	36
3.3.24	Some other kind of trees	37
3.4	Dendrology (The Tree's Taxonomy)	38
3.4.1	The Y-tree Family	38
3.4.2	The Symmetric Y-tree Family	39
3.4.3	The Caterpillar Family	39
3.4.4	The Spider Family	45
3.4.5	The Cayley Family	47
3.4.6	The Lobster Family	48
3.4.7	Intersections	49
3.5	Limits	50

4	Le logiciel Who's that Tree	51
4.1	Structure	51
4.2	Matrice d'adjacence	57
4.2.1	Algorithmes reliés aux invariants	57
5	Conclusion	59

Liste des figures

3.1	The Path	15
3.2	The Star	16
3.3	The Caterpillar	17
3.4	The Claw	17
3.5	The Brush (a)	18
3.6	The Brush (b)	18
3.7	The Comb a	19
3.8	The Comb b	20
3.9	The Spider	21
3.10	The Spur	22
3.11	The Y-tree	23
3.12	The Symmetric Y-tree	24
3.13	The Fork	24
3.14	The Comet	25
3.15	The Double Star	26
3.16	The Double Comet	27
3.17	The Bamboo	27
3.18	The Banana	28
3.19	The Double Spider	29
3.20	The Cayley tree	31
3.21	The Lobster	31
3.22	The Firecracker	32
3.23	The Superstar	33
3.24	The Dandelion	34

3.25 The Olive	35
3.26 The Spraying Pipe	36
3.27 The Dendrology	38
4.1 Structure du programme "Who's that tree"	52

Liste des tableaux

4.I	Invariants nécessaires pour l'identification des arbres	53
-----	---	----

Lexique

Français : Anglais

Graphe : graph

Arbre : tree

Invariant : invariant

Arête : edge

Sommet : vertex

Degré : degree

Diamètre : diameter

Distance : distance

Feuille : leaf

Matrice d'adjacence : adjacency matrix

Familles : families

Araignée : Spider

Arbre Cayley : Cayley tree

Arbre Y : Y-Tree

Arbre Y symétrique : Symmetric Y-Tree

Bambou : Bamboo

Bananier : Banana

Brosse a : Brush a

Brosse b : Brush b

Chemin : Path

Chenille : Caterpillar
Comète : Comet
Double Araignée : Double Spider
Double Comète : Double Comet
Double Étoile : Double Star
Éperon : Spur
Étoile : Star
Feu d'Artifice : Firecracker
Fourchette : Fork
Griffe : Claw
Homard : Lobster
Olivier : Olive
Peigne a : Comb a
Peigne b : Comb b
Pissenlit : Dandelion
Super Étoile : Superstar
Tuyau d'Arrosage : Spraying Pipe

Remerciements

Je tiens d'abord à remercier mes directeurs de recherche, Gilles Caporossi et Sylvain Perron, qui tout au long du processus de rédaction de ce mémoire, m'ont fait confiance, ont su m'encourager et me guider.

Un grand merci à mes parents qui chacun à leur manière m'ont appuyée et motivée. C'est grâce à eux que j'ai eu la chance de me rendre jusqu'ici et ils savent mieux que personne la valeur de leur présence.

Finalement, je remercie tous mes amis et collègues du GERAD, qui sont précieux tant pour leur aide que pour les moments moins sérieux.

Avant-propos

Le chapitre 3 de ce mémoire est constitué d'un article co-écrit avec Gilles Caporossi et Sylvain Perron (Lobsters, Spiders, Caterpillars and Other Trees : A Graph Theoretical Dendrology). Cet article est présentement en préparation. L'identification des familles, la caractérisation et les preuves présentées sont l'oeuvre principale d'Hélène-Sarah Bécotte-Boutin.

Chapitre 1

Introduction

1.1 Introduction

En 1976, Bondy and Murty [2, p. 1] décrivaient un graphe en faisant une analogie similaire à ce qu'est aujourd'hui le concept Facebook : l'amitié en tant que «réseau» social. Un graphe est une représentation visuelle qui est constituée de points éventuellement liés ensemble par des lignes. Dans cet exemple, les relations d'amitié sont faites de personnes, représentées par des points et l'amitié entre deux personnes est indiquée par la présence d'une ligne entre ces deux points.

Les graphes, qui peuvent représenter plusieurs situations réelles, sont utilisés dans différentes disciplines. Ceci explique qu'un grand nombre de graphes aient été étudiés jusqu'à ce jour. Que ce soit dû à une certaine prolifération ou simplement à un manque de rigueur, plusieurs de ces graphes sont connus sous plusieurs noms. À partir de ce simple constat, l'idée de répertorier et caractériser ceux-ci s'est imposée d'elle-même.

Lorsqu'un chercheur travaille sur un certain type de graphe, il est normal qu'il s'intéresse à savoir si des articles existent déjà sur le sujet et quelles seraient les propriétés déjà étudiées sur une structure donnée. Or, il peut être ardu, sans connaître à l'avance le nom d'un graphe, d'effectuer de telles recherches. Conséquemment, sans peut-être réussir à trouver de la documentation existante, un ou plusieurs auteurs ont pris la liberté de baptiser un graphe qui avait pourtant été préalablement nommé par d'autres auteurs. Par exemple, ce phénomène s'est produit dans le cas de El-Basil [13], qui traite

d'un arbre appelé Gutman en 1986 alors que Harary et Schwenk [28] avaient déjà abordé les propriétés d'un arbre ayant exactement les mêmes caractéristiques, en 1973, sous le nom de Chenille.

Ainsi, non seulement un graphe peut avoir plusieurs noms, mais la description que les auteurs en font est généralement arbitraire. Ces descriptions sont souvent basées sur des aspects visuels du graphe, ce qui les rend plutôt approximatives puisque non mathématiques. La littérature regorge de descriptions de toutes sortes mais rarement on y fait mention que les caractéristiques d'un graphe soient discriminantes pour ce dernier. Outre certaines propriétés mathématiques étudiées, il est donc presque impossible avec les connaissances actuelles de déterminer, avec les invariants d'un graphe, si celui-ci est membre d'une famille distincte.

Pour toutes ces raisons, une classification, puis un programme informatique permettant d'identifier un graphe donné serait particulièrement utile en recherche. Un tel logiciel agirait à titre d'anti-dictionnaire et aiderait non seulement à uniformiser la terminologie mais aussi à orienter les chercheurs dans leurs travaux.

Afin de pouvoir créer le logiciel, cet exercice de classification et de recherche approfondie s'est imposé comme étant essentiel. D'abord pour trouver quelle terminologie est utilisée en pratique, mais aussi pour déterminer de quelle manière les familles peuvent être caractérisées et finalement pour proposer des définitions rigoureuses et discriminantes. Toutes ces étapes sont primordiales afin d'établir des caractéristiques mathématiques pour chacune des familles, qui sont implantables dans le logiciel pour identifier les arbres.

Considérant le nombre élevé de familles de graphes, cette recherche se penchera seulement sur les arbres, une catégorie plus restreinte de graphes qui sont connexes, acycliques et finis. Ce mémoire jette donc les bases d'un projet d'envergure dont le produit final sera un programme permettant d'identifier des graphes grâce à leurs caractéristiques.

1.2 Plan du mémoire

Le chapitre 2 de ce mémoire traite de la classification et de son application en théorie des graphes, autant passée que dans le cadre de cette recherche. L'article fait office de chapitre 3 et est divisé comme suit : la section 3.1 est une brève introduction, la section 3.2 définit la notation ainsi que les concepts utilisés, la section 3.3 caractérise en détail les différentes familles d'arbres recensées tandis que la section 3.4 traite de la dendrologie donc des liens entre les différentes familles et finalement la section 3.5 conclut brièvement l'article en mentionnant les limites de celui-ci. Le chapitre 4 présente la motivation derrière cette recherche, soit un logiciel permettant l'identification des graphes grâce à leurs invariants. Pour terminer, le chapitre 5 sera une conclusion générale.

Chapitre 2

La Classification

2.1 La classification

La classification est une méthodologie qui consiste à regrouper des objets en un nombre limité de groupes appelés classes [60]. Lorsqu'il est question de classification dans la littérature, il s'agit généralement de classification automatique qui est une opération statistique visant à découvrir les différentes classes en cours de traitement des données. Dans un contexte classique, on relie la classification à l'analyse de larges bases de données. De manière plus générale la classification est utilisée dans divers domaines pour regrouper des objets similaires dans des ensembles plus sommaires [20, chapitre 1 p.2]. Les «objets» sont généralement décrits par un ensemble de variables [20, chapitre 1 p.2]. La discrimination des classes se fait donc à partir de ces variables. Ce sont elles qui déterminent la parenté entre objets.

Cette méthodologie est ainsi utilisée autant dans des domaines reliés à l'administration des affaires tels que le marketing [38] ou la gestion des opérations [36] que dans des branches toutes autres comme la géographie [51] ou la biologie [30]. Dans ces derniers domaines, il est d'ailleurs souvent question du terme «taxonomie» pour décrire les inclusions d'objets à l'intérieur de groupes distincts. En botanique, le terme «dendrologie» est utilisé plus spécifiquement pour parler de l'identification et la classification systématique des arbres [52], d'où provient le nom de ce mémoire.

Le produit d'une classification se résume en une partition d'ensembles d'objets en classes définies d'une telle manière que les objets qui sont dans une même classe sont similaires [20, chapitre 1 p.2]. Plusieurs types de classifications existent. Par exemple, dans la classification hiérarchique, il y a des relations entre différentes classes d'objets telles qu'une classe peut être entièrement comprise à l'intérieur d'une autre [50]. Le terme "taxonomie" est d'ailleurs repris dans ce type de classification particulier où les auteurs parlent plus précisément de taxonomie de classes [56]. Dans une telle classification, il existe donc une structure entre les classes, qui est souvent illustrée à l'aide d'un arbre. C'est d'ailleurs ces relations entre les différentes classes qui est sujet d'intérêt dans la classification hiérarchique [20, chapitre 4 p.1].

Il peut cependant être utile, voire nécessaire, d'avoir des classes qui partagent certains objets mais pas tous [20, chapitre 5 p.11]. S'inspirant donc du principe de classification hiérarchique, la classification "chevauchante" (overlapping classification) peut être décrite comme une classification hiérarchique, permettant cependant à une classe de partager plusieurs mais pas nécessairement tous ses objets avec une autre classe [20, chapitre 5 p.16].

Dans ces précédents types de classification cependant, ni les classes, ni leurs caractéristiques déterminantes, ni les objets qui les constituent ne sont connus d'avance [20, chapitre 1 p.3]. Or, dans la situation présente, les classes sont déjà connues, ce sont les familles. Il est également possible de générer des graphes et arbres constituant ces familles, donc les objets peuvent déjà être attributés à une classe. L'intérêt est donc de trouver les caractéristiques discriminantes, qui permettent de distinguer ces dites familles entre elles.

Dans l'objectif précis de cette recherche, il s'agira plutôt de classification supervisée. La définition de cette dernière est la détection et la caractérisation de classes (pattern) connues [23, p.179]. Autrement dit, le but de la classification supervisée est de faire correspondre un modèle ou un arbre de décision pour qu'il associe de manière juste des objets à des classes prédéfinies [16, chapitre 6 p.1]. Outre les arbres de décision, les modèles de prédiction suivants sont utilisés : l'analyse discriminante (DA), la régression linéaire, la régression logistique, les méthodes bayésiennes, les réseaux de neurones et les algorithmes génétiques, pour ne nommer que ceux-ci [60, p.289-549].

Plusieurs techniques de classification existent donc pour ce cas particulier. Cependant, tel que mentionné précédemment, la classification est utilisée lorsqu'un nombre imposant de données est à traiter. Puisque ce mémoire porte sur une sous-catégorie de graphes, soit les arbres et que leur nombre est relativement limité, il n'a pas été nécessaire de programmer des algorithmes afin de générer automatiquement des caractéristiques discriminantes entre objets et classes. La démonstration par preuve de ces caractéristiques discriminantes et de l'inclusion d'une famille à l'intérieur d'une autre a suffi pour effectuer cette classification.

2.1.1 La classification en théorie des graphes

Lorsqu'il est question de classification en théorie des graphes, il s'agit généralement d'une technique appliquée à une famille spécifique de graphes [35, 65, 34]. Puisque les graphes ont des applications concrètes, il arrive également que la classification soit utilisée dans un cas particulier, comme la reconnaissance de formes (pattern recognition) entre autres [11, 41].

Parallèlement, certains ouvrages répertorient différentes familles de graphes [4] mais souvent il s'agit d'une recherche dans un but particulier comme le "graph labeling" par exemple, où de multiples études ont été réalisées sur le sujet [18, 53].

Une lacune dans le domaine est que concrètement, il n'existe pas d'anti-dictionnaire permettant à un chercheur d'identifier un graphe et de déterminer ses caractéristiques de manière exhaustive. C'est pour cette raison que cette recherche vise à combler un manque actuel en théorie des graphes, soit les liens existant entre les différentes familles de graphes connues, en les classifiant.

Chapter 3

Lobsters, Spiders, Caterpillars and Other Trees : A Graph Theoretical Dendrology

Lobsters, Spiders, Caterpillars and other Trees : A Graph Theoretical Dendrology

H.-S. Bécotte-Boutin, G. Caporossi, S. Perron

March 21, 2013

Abstract

When working on a special kind of graph, a researcher often wants to know if some literature already exists on that topic. Indeed, it is not easy to search for some literature on a given family of graph without knowing its name, which is likely a reason why some families of graphs are already known with various names. To handle this problem, we propose a tool that could be used to find the name(s) of a graph given its description.

Because there are a lot of graphs, it would have been hard to characterize them all at the same time. For that reason, this article focusses on the characterization of a specific family, i.e., the Tree. The Lobster, the Caterpillar and the Spider are a few families of trees. In this paper, those families and some other ones will be presented and characterized.

Keywords: Graph Theory, Classification, Trees.

3.1 Introduction

Back in 1976, Bondy and Murty ([2] p.1) used to describe a graph with what could now be the Facebook concept : friendship as a - social - network. A graph is a visual representation made of points joined together with lines. For social network, the points represent the people while the line between two points represent the existence of a friendship relation between the two persons.

Because graphs can represent many other real world situations such as transportation and telecommunication networks, they have been largely studied. The litterature is quite prolific in this specific field. Depending of their characteristics, such as the number of points and lines, graphs can be generalized and classified in families. However, the researchers don't agree on a specific terminology to classify these families. Because there is continuous research in this field, a problem manifests when someone is doing research on a specific family but don't know exactly its name or that the literature is duplicated by the fact that several names are used to identify the same family.

The fact that sometimes there is a name duplication for the graphs happened for exemple in 1986, when El-Basil [13] wrote about Gutman trees while in 1973, Harary and Schwenk [28] already described and characterized a tree they named Caterpillar which is the exact same concept. Moreover, the definitions of the graphs found in the literature are often approximate because they are based on a description of the visual aspects of the graphs. This is mainly the reason why this exhaustive research is done and why this classification exercice is essential : to find which terms are used, how to characterized the graph families and propose rigorous definitions.

To resolve this problem, we propose the creation of a software, called Who's That Graph -Who's That Tree in a first version with the Tree families-, that can identify the family of a graph. This software would identify graph families according to the characteristics of this graph. Considering the fact that the number of the families of graph currently known is tremendous, this article focusses on the Tree family.

In the next section, the notation and definitions will be briefly tackled. We then turn to the Trees families, where the litterature will review the most common definitions used to describe these trees and the known characteristics while the mathematical part will

prove why these characteristics are discriminant for each tree species and why some families belong to other ones. The two last sections are the dendrology and the limits.

3.2 Notation and Definitions

This section presents the main definitions and the notation used in this article.

3.2.1 Main definitions

Several terminologies have been proposed in the literature for the components of a graph. The point is also called vertex or node and the line is an edge or an arc. We will use the most common terminology, i.e. vertex and edge.

Graph We denote $G = (V, E)$ a graph where V is the set of n vertices and where E is the set of m edges.

Vertex Bondy and Murty [2, p.2] simply explain that a "vertex is indicated by a point".

Edge An edge is "a line joining the points which represent its ends" [2, p.2].

Degree Harary [25, p.18] describe the degree of a vertex v in a graph G as the number of [edges] incident with v .

Distance "The distance $d(u, v)$ between two [vertices] u and v in G is the length of a shortest path joining them if any" [25, p.14].

Connected Graph A graph is said to be connected if "every pair of points are joined by a path" [25, p.13].

Cycle Bondy and Murty describe that "a walk is closed if it has positive length and its origin and terminus are the same. A closed trail whose origin and internal vertices are distinct is a cycle" [2, p.14].

Acyclic graph An acyclic graph is one that contains no cycles [2, p.25].

Leaf A leaf, also called pending vertex, is a vertex of degree 1.

Tree A tree is a connected acyclic graph [25, p.32]. An acyclic connected graph necessarily has a number m of edges equals to the number n of vertices minus 1 : $m = n - 1$ (and has at least two vertices of degree one).

In this research, the trees are considered to be finite.

Every proof in this paper will implicitly consider these characteristics. In the Who's That Tree Software, the fact that a tree is a graph with $m = n - 1$ is also implicit in the algorithms.

Invariant "An invariant of a graph G is a number associated with G which has the same value for any graph isomorphic of G . Thus the [number of vertices and edges] are certainly invariants" [25, p.11].

3.2.2 List of invariants

In this article, we decided to limit the number of invariants used and to privilege the most simple ones, to make the Who's that Tree project more accessible and user friendly. The following invariants are the only ones used to characterize the Trees families.

$n_i = \text{number of vertices of degree } i$ The symbol n is also used to talk about the total number of vertices in a tree T_n .

n_1 = **number of leaves**

$D(T_n)$ = **diameter** "The diameter ($D(T_n)$) is the value of the longest distance of a graph" [25, p.14].

$d(v)$ = **degree of a vertex v**

Δ = **maximum degree** The maximum degree (Δ) is the value of the maximal degree a vertex has in a graph ($\max_{v \in V} d(v)$).

m_{ij} The value of m_{ij} is the number of edges between a vertex of degree i and a vertex of degree j ($i \leq j$) [6].

Adjacency matrix The adjacency matrix (A) stores a graph as $n \times n$ matrix. The matrix has "a row and a column corresponding to every node, and its ij th entry equals 1 if $(i, j) \in E$ and equals 0 otherwise" [1, p. 33].

3.3 The Trees Families

In this section, all the listed trees will be described and characterized with discriminants invariants or with an algorithm. To facilitate the reading and comprehension, the presentation of each family will be similar to an index card. The structure of these index cards is uniform. First is the definition of the family, followed by its usual notation, a figure that illustrates the tree and finally, a mathematical definition. Recall that the goal of this characterisation is to create a software that will identify trees and graphs. Consequently, the description and characteristics used aim at ensuring that the criterion used by the software are consistent with the definition.

3.3.1 Path

Definition : *A Path is a tree with a sequence of vertices such that there is an edge between every two consecutive vertices [39]. So a Path is a tree with only two leaves.*

Notation : $P_n, n \geq 3$, denotes a Path with n vertices and $n - 1$ edges [24] [4, p.17]. An example of a Path containing seven vertices is depicted in the figure 3.1.



Figure 3.1: The Path

Formal definition : A Path is a tree with :

$$n_1 = 2 \text{ or} \tag{3.1}$$

$$D(T_n) = n - 1 \text{ or} \tag{3.2}$$

$$\Delta = 2. \tag{3.3}$$

3.3.2 Star

Definition : *A tree is a Star if it has exactly one vertex which is not a leaf [15, 63, 19].*

Notation : $K_{1,a}, a \geq 2, a = n - 1$, denotes the Star with n vertices and $n - 1$ leaves [15]. The notation S_n is also used [63].

An example of a Star containing nine vertices is depicted in the figure 3.2.

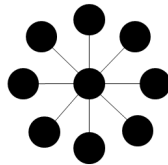


Figure 3.2: The Star

Formal definition : A Star is a tree with :

$$n_1 = n - 1 \text{ or} \tag{3.4}$$

$$D(T_n) = 2 \text{ or} \tag{3.5}$$

$$\Delta = n - 1. \tag{3.6}$$

3.3.3 Caterpillar

Definition : *The Caterpillar is "a tree which metamorphoses into a Path when its cocoon of endpoints is removed" [28].*

Said in a less poetic way, a Caterpillar is a tree in which the removal of all its leaves gives a Path [57, 19].

Caterpillar is the most commonly used term to describe that kind of tree. However we found in the literature that some authors also used the term "Gutman Trees" for the same kind of tree [13, 14].

An example of a Caterpillar containing 18 vertices is depicted in the figure 3.3.

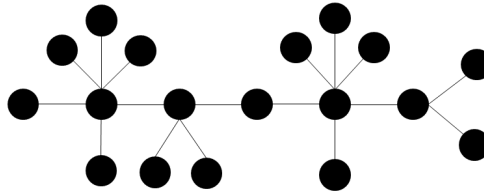


Figure 3.3: The Caterpillar

Formal definition : A Caterpillar is a tree with :

$$D(T_n) = n - n_1 + 1. \quad (3.7)$$

3.3.4 Claw

Definition : *The Claw is a tree that has exactly one vertex of degree three and three leaves [54].*

Notation : $K_{1,3}$ is the most common notation for the Claw [4, p.18]. It is then a Star on four vertices so the notation S_4 is also correct.

An example of a Claw is depicted in the figure 3.4.

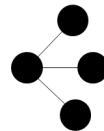


Figure 3.4: The Claw

Formal definition : A Claw is a tree with :

$$D(T_n) = 2 \quad (3.8)$$

$$n_1 = 3 \text{ or } \Delta = 3 \quad (3.9)$$

3.3.5 Brush

Because the Combs a and b are quite restrictive due to their maximum degree value, we propose an extension of these Combs which could be called Brushes.

Brush (a)

Definition : *The Brush (a) is a graph obtained by appending k leaves to each vertex of degree 2 of a Path.* An example of a Brush (a) containing 17 vertices is depicted in the figure 3.5.

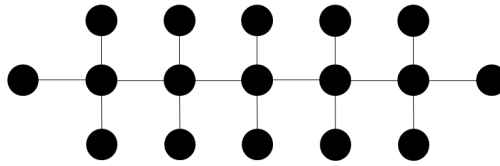


Figure 3.5: The Brush (a)

Formal definition : A Brush (a) is a tree with :

$$D(T_n) = n - n_1 + 1, \quad (3.10)$$

$$\Delta = \frac{n_1 - 2}{D(T_n) - 1} + 2. \quad (3.11)$$

Brush (b)

Definition : *The Brush (b) is a graph obtained by joining k leaves to each vertex of a Path.* An example of a Brush (b) containing 21 vertices is depicted in the figure 3.6.

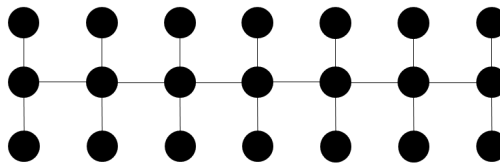


Figure 3.6: The Brush (b)

Formal definition : A Brush b is a tree such that $\Delta = k$ and

$$D(T_n) = \frac{n}{\Delta - 1} + 1, \quad (3.12)$$

$$n_1 = (\Delta - 2)(D(T_n) - 1), \quad (3.13)$$

$$m_{1(k-1)} = 2(k - 2), \quad (3.14)$$

$$m_{(k-1)k} = 2. \quad (3.15)$$

3.3.6 Comb

It is also called a Centipede [3, 37]. There are two contradictory definitions of this kind of tree.

Comb (a)

Definition : A *Comb (a)* is a graph obtained by appending a leaf to each vertex of degree two of a Path [3]. An example of a Comb (a) containing 12 vertices is depicted in the figure 3.7.

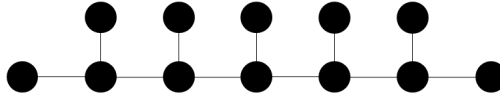


Figure 3.7: The Comb a

Formal definition : A Comb (a) is a tree with :

$$D(T_n) = n - n_1 + 1, \quad (3.16)$$

$$\Delta = \frac{n_1 - 2}{D(T_n) - 1} + 2 = 3. \quad (3.17)$$

Comb (b)

Gallian [18] and Levit and Mandrescu [37] have another definition of a Comb tree :

Definition : *A Comb (b) is a graph obtained by joining a single leaf to each vertex of a Path [18, 37].* An example of a Comb (b) containing 14 vertices is depicted in the figure 3.8.

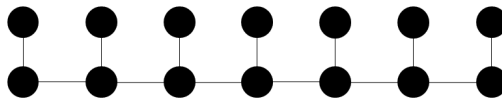


Figure 3.8: The Comb b

A Comb (b) is a Caterpillar whose vertices that are not leaves have degree three except the two that are adjacent to the leaves on its diameter that have degree two.

Formal definition : A Comb (b) is a tree with :

$$D(T_n) = \frac{n}{2} + 1, \quad (3.18)$$

$$n_1 = \frac{n}{2}, \quad (3.19)$$

$$\Delta = 3, \quad (3.20)$$

$$m_{12} = 2, \quad (3.21)$$

$$m_{23} = 2. \quad (3.22)$$

3.3.7 Spider

Definition : *A tree is a Spider if it has exactly one vertex of degree greater than two [27, 47, 44, 31].* An example of a Spider containing 23 vertices is depicted in the figure 3.9.

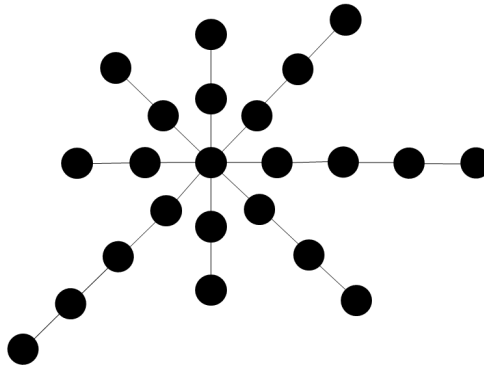


Figure 3.9: The Spider

Formal definition : A Spider is a tree with :

$$\Delta = n_1. \quad (3.23)$$

3.3.8 Spur

Definition : *A Spur is obtained from a Star, by attaching a pendant edge to non central vertices [31].*

Notation : "A [Spur] $S(n, k, r)$ with n vertices [...] is obtained from a Star with $k + r + 1$ vertices by attaching a pendant edge to k non central vertices" [5].

An example of a Spur containing 14 vertices is depicted in the figure 3.10.

Brualdi and Goldwasser [5] describe k and r as nonnegative integers that correspond to :

$$n = 2k + r + 1 \quad (3.24)$$

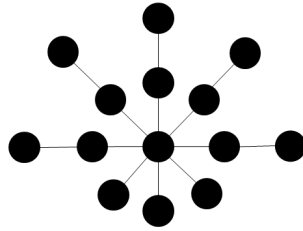


Figure 3.10: The Spur

The value of k then equals to :

$$k = n - n_1 - 1 \quad (3.25)$$

so

$$k = n_2 \quad (3.26)$$

And the value of r then equals to :

$$r = n_1 - k \quad (3.27)$$

However, the general characterization of the Spur is more simple.

Formal definition : A Spur is a tree with :

$$\Delta = n_1, \quad (3.28)$$

$$m_{22} = 0. \quad (3.29)$$

3.3.9 Y-Tree

General Y-tree

Definition : Sometimes called *T-Shape*, the *Y-tree* is a spider with maximum degree 3 [47]. An example of a Y-tree containing 11 vertices is depicted in the figure 3.11.

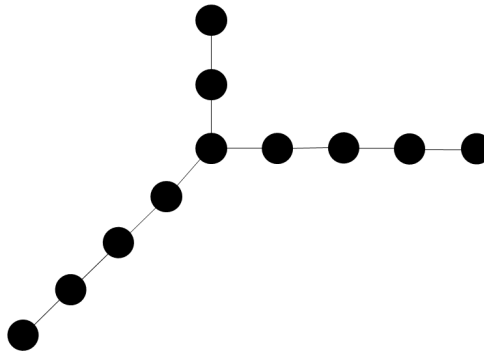


Figure 3.11: The Y-tree

Formal definition : A Y-tree is a tree with :

$$n_1 = 3. \quad (3.30)$$

Symmetric Y-tree

Definition : The characteristics of the *Symmetric Y-tree* are quite similar to the normal Y-tree except that every branch is of similar length [4, p.19].

Notation : T_k [4, p.19], where k represent the number of edges between the vertex of maximum degree and each of the leaves is the notation used for the Symmetric Y-tree.

An example of a Symmetric Y-tree containing 10 vertices is depicted in the figure 3.12.

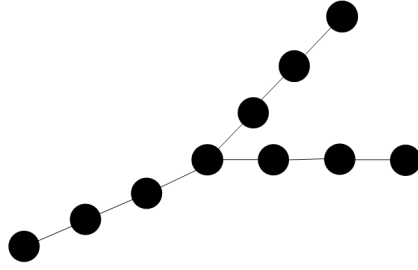


Figure 3.12: The Symmetric Y-tree

Formal definition : A Symmetric Y-tree is a tree with : T_n is a Symmetric Y-tree if and only if it has both these characteristics :

$$n_1 = 3, \quad (3.31)$$

$$D(T_n) = 2 * \left(\frac{n-1}{3}\right). \quad (3.32)$$

3.3.10 Fork

Definition : A Fork graph is a tree which consist of 5 vertices displayed this way : 3 leaves, one vertex of degree 2 and one vertex of degree 3 [58, 17].

The Fork graph is also known as the Chair graph [58] [4, p.18]. An example of a Fork is depicted in the figure 3.13.

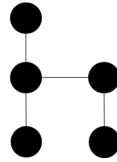


Figure 3.13: The Fork

Formal definition : A Fork is a tree with :

$$n_1 = 3, \quad (3.33)$$

$$n = 5. \quad (3.34)$$

3.3.11 Comet

Definition : A Comet is a tree that is constructed by connecting one of the two leaves of a Path to the only vertex of degree > 1 of a Star [5, 64, 10].

The Comet is also called a Broom.

Notation : Its notation is $B(n, k, r)$ [5] where $B(n, k, r)$ is obtained from the Path P_k with k vertices by attaching r pendant edges to a leaf u of P_k .

Yuan et al. [64] rather used the notation $B(n, i)$ ($2 \leq i \leq n - 1$) to describe the Broom of n vertices that is "obtained by joining one [leaf] of $K_{1,i}$ to one [leaf] of P_{n-i-1} by an edge.

An example of a Comet containing 13 vertices is depicted in the figure 3.14.

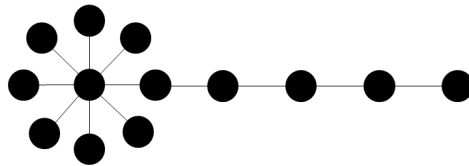


Figure 3.14: The Comet

Formal definition : A Comet is a tree which has one vertex t of degree $d(t)$, adjacent to $d(t) - 1$ leaves, the other vertices being of degree 1 or 2 and such that

$$D(T_n) = n - n_1 + 1, \quad (3.35)$$

$$\Delta = n_1, \quad (3.36)$$

$$m_{1\Delta} = \Delta - 1 = n_1 - 1. \quad (3.37)$$

3.3.12 Double Star

Definition : A Double Star is a tree which is formed from two Stars by connecting their maximum degree vertex with an edge [26, 39]. A Double Star can't be a Comet and its maximum degree $\Delta \leq n - 3$. An example of a Double Star containing 12 vertices is depicted in the figure 3.15.

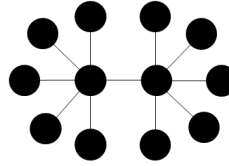


Figure 3.15: The Double Star

Formal definition : A Double Star is a tree with :

$$n_1 = n - 2 \text{ or } D(T_n) = 3, \quad (3.38)$$

$$D(T_n) \leq n - 3. \quad (3.39)$$

3.3.13 Double Comet

Definition : A Double Comet is a tree that is constructed by connecting each of the two leaves of a Path to the only vertex v of degree $d(v) > 1$ of two Stars that are not necessarily identical [8].

A Double Comet is also called a Double Broom [8, 9].

Notation : Its notation is $DC(n, r, s)$ [8] where " $DC(n, r, s)$ is obtained from two disjoint stars $K_{r-1,1}, K_{s-1,1}$ by adding a path of length $n - r - s + 1$ between their central vertices".

Dankelmann [9] rather used the notation $DB(n, d)$ to describe "a tree of order n obtained from a path on $d - 1$ vertices by appending $\lceil \frac{n-d+1}{2} \rceil$ and $\lfloor \frac{n-d+1}{2} \rfloor$ vertices, respectively, to its two ends". However, this notation suggests that the double comet is quasiequilibrated.

An example of a Double Comet containing 21 vertices is depicted in the figure 3.16.

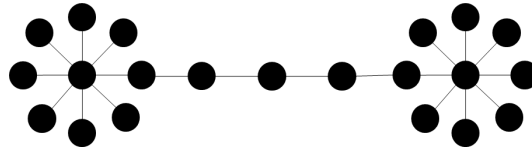


Figure 3.16: The Double Comet

Formal definition : A Double Comet is a tree which has one vertex t_1 of degree $d(t_1)$, adjacent to $d(t_1) - 1$ leaves, one vertex t_2 of degree $d(t_2)$, adjacent to $d(t_2) - 1$ leaves and the remaining vertices of degree 2 such that

$$m_{1d(t_1)} = d(t_1) - 1,$$

$$m_{1d(t_2)} = d(t_2) - 1,$$

and

$$n_1 = d(t_1) + d(t_2) - 2.$$

3.3.14 Bamboo

Definition : A Bamboo is a tree that is constructed by connecting the leaf adjacent to a vertex of degree 2 of $d(r)$ identical Comets to a same vertex r [55] (as referenced in [18, 53]). An example of a Bamboo containing 37 vertices is depicted in the figure 3.17.

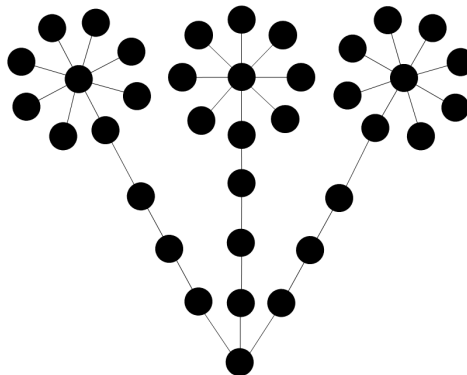


Figure 3.17: The Bamboo

Formal definition : A Bamboo is a tree which has one vertex of degree $d(r)$, $d(r)$ vertices t_i of degree $d(t)$, $(d(t) - 1) * d(r)$ leaves adjacent to vertices of degree $d(t)$ and a diameter such that

$$D(T_n) = 2\left[\frac{n-1}{d(r)} - d(t) + 2\right], \quad (3.40)$$

3.3.15 Banana

Definition : Let $G_1, G_2, \dots, G_n$ be a family of disjoint Stars. Let v be a new vertex. The tree obtained by joining v to one leaf of each Star is called a Banana tree [7] (as referenced in [59, 18, 32]). The Banana is a Bamboo with a diameter $D(T_n) = 6$.

An example of a Banana containing 31 vertices is depicted in the figure 3.18.

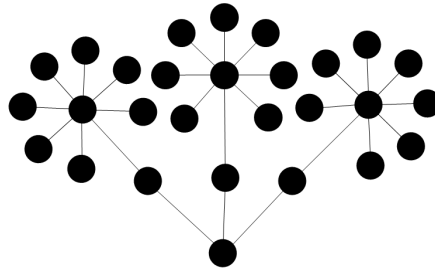


Figure 3.18: The Banana

Formal definition : A Banana is a tree which has one vertex of degree $d(r)$, $d(r)$ vertices t_i of degree $d(t)$, $d(r)$ vertices t_j of degree 2, $(d(t) - 1) * d(r)$ leaves and a diameter such that

$$D(T_n) = 2\left[\frac{n-1}{d(r)} - d(t) - 2\right] = 6. \quad (3.41)$$

3.3.16 Double Spider

Definition : A Double Spider tree is a subdivision of a Double Star in which the edge joining the two central vertices is not subdivided [26, 33, 44].

The Double Spider is also called a Double Starlike [26, 44].

An example of a Double Spider containing 30 vertices is depicted in the figure 3.19.

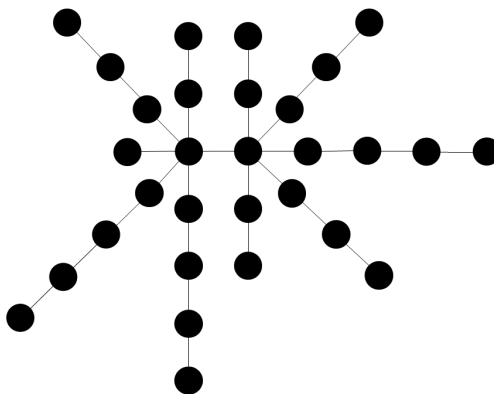


Figure 3.19: The Double Spider

Formal definition : A Double Spider is a tree which has $d(t_1) + d(t_2) - 2$ leaves, one vertex of degree $d(t_1)$, one vertex of degree $d(t_2)$ and the remaining vertices of degree 2 such that

$$n_1 = d(t_1) + d(t_2) - 2, \quad (3.42)$$

$$m_{d(t_1)d(t_2)} = 1, \quad (3.43)$$

$$d(t_1), d(t_2) > 2. \quad (3.44)$$

3.3.17 Cayley tree

The Cayley tree is sometimes called a Bethe Lattice. However, according to Ostilli [48], they are two slightly different concepts and that there "is a quite widespread confusion about their exact definition and use". The main difference between the two would be that "a Cayley is finite and that a Bethe Lattice is infinite". However, the two

terms are used in the literature.

In 1994, Gutman et al. [22] described the dendrimer which is an organic molecule that was previously presented by Newkome [45] in 1991. Adapting the Gutman et al.[22] notation to those used in this paper, these authors tackled more precisely an "highly branched tree $[T_{\frac{D(T_n)}{2}, \Delta}]$ which [they called] regular dendrimer graphs for which" $\frac{D(T_n)}{2} \geq 0$ and $\Delta \geq 3$. They described this tree as "for any $\Delta \geq 3$, $T_{0,\Delta}$ is the one-vertex graph and $T_{1,\Delta}$ is the star with $\Delta + 1$ vertices. Then for $\frac{D(T_n)}{2} = 2, 3, \dots$ and $\Delta \geq 3$, the tree $T_{\frac{D(T_n)}{2}, \Delta}$ is obtained by attaching $\Delta - 1$ new vertices of degree one to the vertices of degree one of $T_{(\frac{D(T_n)}{2}-1), \Delta}$. The number of vertices of the regular dendrimer graph $T_{\frac{D(T_n)}{2}, \Delta}$ is:

$$n = 1 + \frac{\Delta}{\Delta - 2} ((\Delta - 1)^{\frac{D(T_n)}{2}} - 1). \quad (3.45)$$

In 1995, Diudea and Parv [12] made the analogy between a regular dendrimer and a Bethe Lattice. A tree is an regular dendrimer "if all its radial chains (i.e., the chains starting from the core and ending in an external point) have the same length".

These descriptions are the same as the one used by Harary [29] to describe the Cayley:

Definition : *a tree is a Cayley if it has one central root of maximum degree and that each non-leaf vertex has a constant number of branches, also equal to the value of the maximum degree [29].* An example of a Cayley tree containing 53 vertices is depicted in the figure 3.20.

Formal definition : A Cayley tree is a tree with :

$$n_1 = \Delta(\Delta - 1)^{\frac{D(T_n)}{2}-1}. \quad (3.46)$$

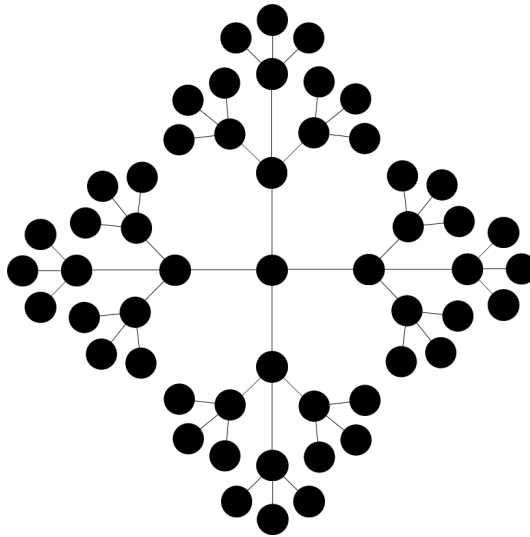


Figure 3.20: The Cayley tree

3.3.18 Lobster

Definition : *A Lobster is a tree that is made in such a way that if you remove all its leaves, then you have a Caterpillar. The distance between any vertex and the root Path is equal or less than two [61, 42, 43].*

Considering this definition, any tree with a diameter of four or less is a Lobster.

An example of a Lobster containing 48 vertices is depicted in the figure 3.21.

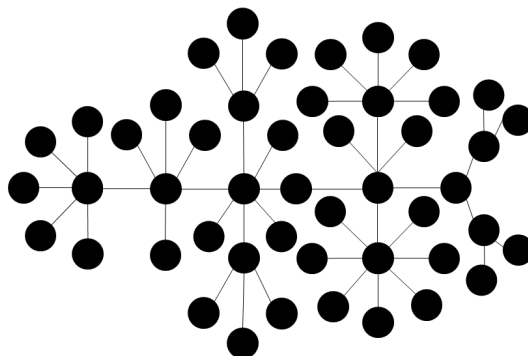


Figure 3.21: The Lobster

Formal definition : A Lobster is a tree T_n which has an induced Path P' such that any vertex $v \in T$ is at distance at most two from a vertex of P' .

To identify a Lobster that has a diameter greater than four, the use of the adjacency matrix is necessary. Neither of the leaves-maximum degree-diameter trio nor the m_{ij} can provide discriminant characteristics to describe the Lobster at this point.

Starting from the adjacency matrix, we developed an algorithm called isLobster that determines if a tree is a Lobster or not.

IsLobster algorithm

1. Identify a diameter of T_n as the root Path and note S its vertices.
2. Verify that for each vertex $v \in T_n, \exists u \in S$ such that $d(u, v) \leq 2$.
3. if 2. is not verified, T_n is not a Lobster.

3.3.19 Firecracker

Definition : A tree is a Firecracker if we join to a Path of k vertices k identical Stars by one of their leaves [7](as referenced in [18, 53].

An example of a Firecracker containing 36 vertices is depicted in the figure 3.22.

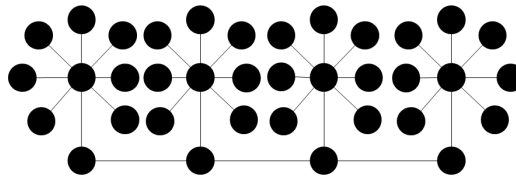


Figure 3.22: The Firecracker

Note that the firecracker is a Lobster. To identify this tree, the use of the IsLobster algorithm is necessary. A small addition after the last step is required.

IsFirecracker algorithm

1. Verify that the diameter is equal to $D(T_n) = \frac{n-n_1}{2} + 3$.
2. Verify that each of the leaves u that are on any diameter are at a distance of $d(u, v) = 3$ of the closest vertex v of degree $d(v) = 3$.

3. Verify that each of the leaves u that are on any diameter are at a distance of $d(u, z) = 2$ of a vertex z of degree $d(z) = 2$.
4. Verify that each of the leaves l that are not on any diameter are at a distance of $d(l, v) = 2$ of a vertex v of degree $d(v) = 3$.
5. Verify that there is only two vertices of degree $d(z) = 2$.
6. If true, then this is a Firecracker.

3.3.20 Superstar

Definition : A Superstar is obtained from a Star, by attaching pendant edges to non central vertices [63].

Notation : $S_{k,t,\pi}$ is used to describe the Superstar as "a tree on $1 + k + t$ vertices which has radius 2 about a vertex of degree k ". This tree has then a diameter $D(T_n)$ of 4 [63]. π is the repartition of the leaves at a radius 2 of the root, as shown in the figure 3.23 below where an example of a Superstar containing 15 vertices is depicted.

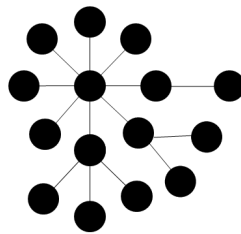


Figure 3.23: The Superstar

Formal definition : A Superstar is a tree T_n with a diameter of $D(T_n) = 4$.

3.3.21 Dandelion

Definition : A *Dandelion* is obtained from a *Star*, by attaching a *Path* to one of the non central vertices and pendant edges to other non central vertices [63].

Notation : $D_{z,t,\pi}$ is used to describe the Dandelion as a tree " constructed by identifying the [root] of the Superstar $S_{k,t,\pi}$ with an endpoint on a path on z vertices" [63]. As for the Superstar, π is the repartition of the leaves at a radius 2 of the root, as shown in the figure 3.24 below where an example of a Dandelion containing 18 vertices is depicted.

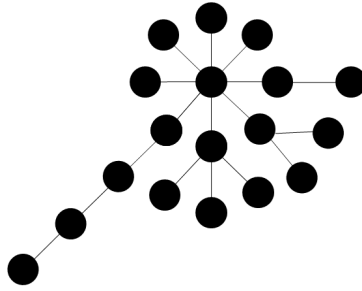


Figure 3.24: The Dandelion

Formal definition : Let T_n be a tree on n vertices and note $d^{k(u)}$ the number of vertices that are at a distance k from the vertex u .

A Dandelion is a tree that has a vertex u such that $d^{k(u)} = 1 \ \forall \ k \geq 3$.

To identify this tree, the use of the IsLobster algorithm is necessary. A small addition after the last step is required.

IsDandelion algorithm

1. One leaf v is at a distance of $d(v, r) = D(T_n) - 2$ of the root r .
2. All the other leaves p are at a distance $d(p, r) \leq 2$.
3. If these two conditions are satisfied, then this graph is a Dandelion.

3.3.22 Olive

Definition : An Olive tree is a rooted tree consisting of k branches, where the i th branch is a path of length i [49] (as referenced in [18, 53]). An example of an Olive containing 15 vertices is depicted in the figure 3.25.

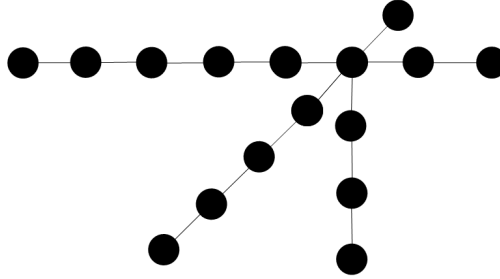


Figure 3.25: The Olive

Formal definition : An Olive is a tree with :

$$D(T_n) = 2n_1 - 1, \quad (3.47)$$

$$n = 0.5n_1^2 + 0.5n_1 + 1, \quad (3.48)$$

$$\Delta = n_1. \quad (3.49)$$

However, to identify an Olive tree, the use of an algorithm and the adjacency matrix is then necessary.

IsOlive algorithm

1. If the number of vertices is less than two, the algorithm ends. This can not be an Olive.
2. Identify that the number of leaves $n_1 = \Delta$.
3. Identify the distance $d(p, v)$ between any leaf p and the vertex v of degree $d(v) = \Delta$.
4. The distance between the i^{th} leaf and the vertex v of degree $d(v) = \Delta$ has to be $d(p_i, v) = i$.
5. Sort these leaves in an ascending order of $d(p, v)$.
6. If it is not the possible, this is not an Olive.

3.3.23 Spraying Pipe

Definition : A *Spraying Pipe tree* is a path $v_1, v_2, \dots, v_n$ such that each vertex v_i is joined to m_i paths at a leaf of each path, and all paths have the same length [7] (as referenced in [18]). An example of a Spraying Pipe containing 24 vertices is depicted in the figure 3.26.

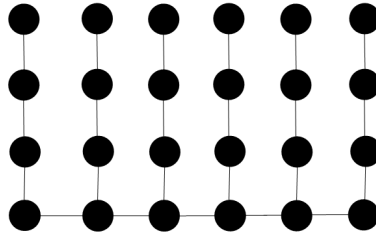


Figure 3.26: The Spraying Pipe

Formal definition : A Spraying Pipe is a tree with :

$$D(T_n) = 2\left(\frac{n}{n_1} - 1\right) + n_1 - 1, \quad (3.50)$$

$$\Delta = 3. \quad (3.51)$$

To identify a Spraying Pipe, the use of the adjacency matrix is necessary. Neither of the invariants could provide discriminant characteristics to describe the Spraying Pipe so far. However, these two characteristics have to met. The IsSprayingPipe algorithm is then only used if these conditions are satisfied.

IsSprayingPipe algorithm

1. Identify a diameter as the centered Path of the tree.
2. Identify the leaves of the graph.
3. Verify that each of the two leaves u_i that are on the centered Path are at a distance of $d(u_i, v) = (\frac{n}{n_1} - 1)$ of the closest vertex v of degree $d(v) = 3$.
4. Verify that the distance between the $n_1 - 2$ other leaves and the central path is $d(u, v) = (\frac{n}{n_1} - 2)$.

5. If true, then this is a Spraying Pipe.

3.3.24 Some other kind of trees

Some known and used families of trees have not been listed. It is the case of the Binary tree, the K-ary and all the fractals trees. In this article, only the unlabelled trees have been listed, which exclude all the rooted trees. A special nomenclature could be made eventually for those trees but not in this master thesis.

Binary Tree

Definition from the literature 1. *A Binary tree is a tree-like structure that is rooted and in which each vertex has at most two children and each child of a vertex is designated as its left or right child [46] [62, p.101] .*

By definition, the maximum degree in this tree is 3 and the root of the tree have to be either a vertex of degree 1 or 2 because of the fact that each vertex has at most two children.

K-ary Tree

The k-ary tree is also called M-ary [62, p.101] [21].

Definition from the literature 2. *A k-ary tree ($k \geq 2$) is a rooted tree in which every vertex has k or fewer children [62, p.101] [21].*

This tree is a generalization of the Binary tree.

3.4 Dendrology (The Tree's Taxonomy)

Dendrology is a science that identifies and makes a systematic classification of the trees [52]. The goal of this section is then to identify and prove the kinship of the different families of trees that have been presented in the previous section. The following figure illustrates the relationships between these families.

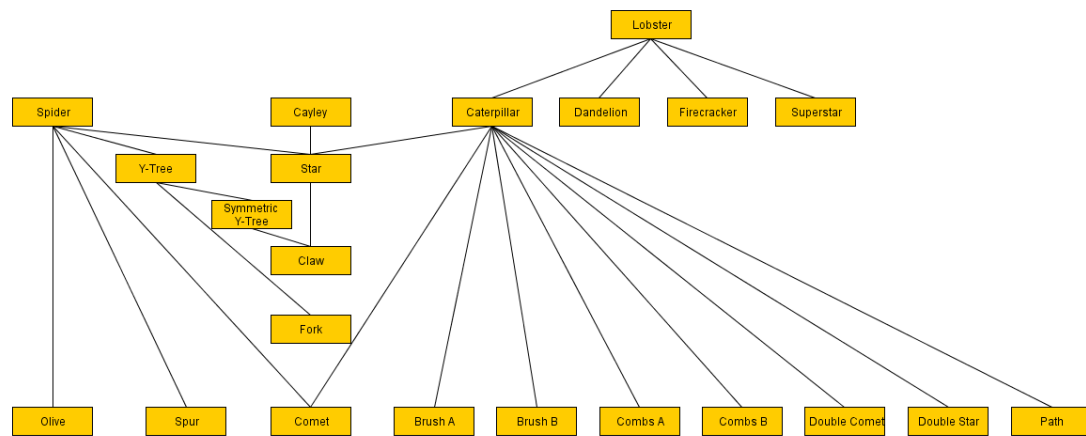


Figure 3.27: The Dendrology

3.4.1 The Y-tree Family

Recall that to be a Y-tree, a tree has to satisfy this condition :

$$n_1 = 3.$$

Fork

Theorem 1. *The Fork is a Y-tree.*

Proof :

With the Fork, we have $n_1 = 3$.

hence it is a Y-tree. □

3.4.2 The Symmetric Y-tree Family

Recall that to be a Symmetric Y-tree, a tree has to satisfy these conditions :

$$\begin{aligned} n_1 &= 3, \\ D(T_n) &= 2 * \left(\frac{n-1}{3} \right). \end{aligned}$$

Claw

Theorem 2. *The Claw is a Y-tree.*

Proof :

With the Claw, we have $D(T_n) = 2$, $n_1 = 3$ and $n = 4$.

$$\begin{aligned} D(T_n) &= 2, \\ &= 2 * \left(\frac{4-1}{3} \right) \end{aligned}$$

So the diameter of the Claw is

$$D(T_n) = 2 * \left(\frac{n-1}{3} \right)$$

and it has $n_1 = 3$ leaves,

hence it is a Symmetric Y-tree. □

3.4.3 The Caterpillar Family

Recall that to be a Caterpillar, a tree has to satisfy this condition :

$$D(T_n) = n - n_1 + 1.$$

According to Luo and Zhou [40], "obviously, all n -vertex trees with diameter 2, 3, $n - 2$ and $n - 1$ are Caterpillars".

Brush a

Proof : (that the Brush a is a Caterpillar).

The diameter of the Brush (a) is :

$$D(T_n) = n - n_1 + 1, \quad (3.52)$$

hence it is a Caterpillar. □

Brush b

Theorem 3. *The Brush b is a Caterpillar.*

Proof :

With the Brush (b), we have $D(T_n) = \frac{n}{\Delta-1} + 1$.

$$\begin{aligned} D(T_n) &= \frac{n}{\Delta-1} + 1, \\ &= n - n + 3, \\ &= n - n + 2 + 1, \\ &= n - (n - 2) + 1, \\ &= n - n_1 + 1. \end{aligned} \quad (3.53)$$

The diameter of the Brush (b) is :

$$D(T_n) = n - n_1 + 1, \quad (3.54)$$

hence it is a Caterpillar. □

□

Comb (a)

Theorem 4. *The Comb (a) is a Caterpillar.*

The diameter of the Comb (a) is :

$$D(T_n) = n - n_1 + 1, \quad (3.55)$$

hence it is a Caterpillar. □

Comb (b)

Theorem 5. *Let T_n be a tree which has n_1 leaves, 2 vertices that have a degree of 2 and $n - n_1 - 2$ vertices of degree 3.*

$$m_{12} = 2, \quad (3.56)$$

$$m_{23} = 2, \quad (3.57)$$

$$m_{13} = n_1 - 2, \quad (3.58)$$

$$m_{33} = n_1 - 3, \quad (3.59)$$

$$\text{and all the other } m_{ij} = 0. \quad (3.60)$$

Theorem 6. *The Comb (b) is a Caterpillar.*

Proof :

With the Comb (b), we have:

$$n_1 = \frac{n}{2}$$

which transforms into

$$n = 2n_1$$

and then we get

$$\begin{aligned} D(T_n) &= \frac{n}{2} + 1 \\ &= n_1 + 1 \\ &= 2n_1 - n_1 + 1 \\ &= n - n_1 + 1 \end{aligned} \tag{3.61}$$

The diameter of a Comb (b) is:

$$D(T_n) = \frac{n}{2} + 1 = n - n_1 + 1,$$

hence it is a Caterpillar. □

Comet

Theorem 7. *The Comet is a Caterpillar.*

Proof : The diameter of a Comet is:

$$D(T_n) = 3 = n - n_1 + 1,$$

hence it is a Caterpillar. □

Double Comet

Theorem 8. *The Double Comet is a Caterpillar.*

Proof :

The diameter of a Double Comet is:

$$D(T_n) = 3 = n - n_1 + 1,$$

hence it is a Caterpillar. □

Double Star

Theorem 9. *The Double Star is a Caterpillar.*

Proof :

With the Double Star, we have $D(T_n) = 3$ and $n_1 = n - 2$.

$$\begin{aligned} D(T_n) &= 3, \\ &= n - n + 3, \\ &= n - n + 2 + 1, \\ &= n - (n - 2) + 1, \\ &= n - n_1 + 1. \end{aligned} \tag{3.62}$$

The diameter of a Double Star :

$$D(T_n) = 3 = n - n_1 + 1,$$

hence it is a Caterpillar. □

Fork

Theorem 10. *The Fork is a Caterpillar.*

Proof :

With the Fork, we have:

$$D(T_n) = 3, \quad (3.63)$$

$$n_1 = 3, \quad (3.64)$$

$$n = 5. \quad (3.65)$$

The diameter of a Fork is:

$$D(T_n) = 3 = n - n_1 + 1,$$

hence it is a Caterpillar. □

Path

Theorem 11. *The Path is a Caterpillar.*

Proof :

With the Path, we have $D(T_n) = n - 1$.

$$\begin{aligned} D(T_n) &= n - 1, \\ &= n - 1 - 2 + 2, \\ &= n - 1 - n_1 + 2, \\ &= n - n_1 + 1. \end{aligned} \quad (3.66)$$

So the diameter of a Path is:

$$D(T_n) = n - 1 = n - n_1 + 1,$$

hence it is a Caterpillar. □

Star

Theorem 12. *The Star is a Caterpillar.*

Proof :

With the Star, we have $D(T_n) = 2$.

$$\begin{aligned}
 D(T_n) &= 2, \\
 &= n - n + 2, \\
 &= n - n + 1 + 1, \\
 &= n - (n - 1) + 1, \\
 &= n - n_1 + 1.
 \end{aligned} \tag{3.67}$$

So the diameter of a Star is :

$$D(T_n) = 2 = n - n_1 + 1,$$

hence "Stars are Caterpillars which are trees" [19]. □

3.4.4 The Spider Family

Recall that to be a Spider, a tree has to satisfy this condition :

$$\Delta = n_1.$$

Comet

Theorem 13. *The Comet is a Spider.*

Proof :

The maximum degree of the Comet is

$$\Delta = n_1, \tag{3.68}$$

hence it is a Spider. □

Olive

Theorem 14. *The Olive is a Spider.*

Proof :

With the Olive, we have :

$$\Delta = n_1, \quad (3.69)$$

hence it is a Spider. □

Spur

Theorem 15. *The Spur is a Spider.*

Proof :

With the Spur, we have :

$$\Delta = n_1, \quad (3.70)$$

hence it is a Spider. □

Star

Theorem 16. *The Star is a Spider.*

Proof :

Let T_n be a tree which has $n - 1$ leaves, one vertex of degree $n - 1$ and a maximum degree of $\Delta = n - 1$:

$$\begin{aligned} & \text{if } \Delta = n - 1, \\ & \text{and } n_1 = n - 1, \\ & \text{then } \Delta = n_1. \end{aligned} \quad (3.71)$$

So the maximum degree of the Star is

$$\Delta = n_1, \quad (3.72)$$

hence it is a Spider. $\square$

Y-Tree

Theorem 17. *The Y-tree is a Spider.*

Proof :

With the Y-tree, we have $\Delta = 3$.

$$\begin{aligned} &\text{if } \Delta = 3, \\ &\text{and } n_1 = 3, \\ &\text{then } \Delta = n_1. \end{aligned} \quad (3.73)$$

So the maximum degree of the Y-tree is

$$\Delta = n_1, \quad (3.74)$$

hence it is a Spider. $\square$

3.4.5 The Cayley Family

Recall that to be a Cayley, a tree has to satisfy this condition :

$$n_1 = \Delta(\Delta - 1)^{\frac{D(T_n)}{2} - 1}.$$

Star

Theorem 18. *The Star is a Cayley.*

Proof :

Let T_n be a tree which has $n - 1$ leaves, one vertex of degree $n - 1$ and a maximum degree of $\Delta = n - 1$:

$$\begin{array}{rcl}
 & & \text{if} \\
 \Delta & = & n - 1, \\
 D(T_n) & = & 2 \\
 & & \text{and} \\
 n_1 & = & n - 1. \\
 & & \text{then}
 \end{array}$$

$$\begin{aligned}
 n_1 &= n - 1 \\
 &= \Delta \\
 &= \Delta * 1 \\
 &= \Delta(\Delta - 1)^0 \\
 &= \Delta(\Delta - 1)^{\frac{2}{2}-1} \\
 &= \Delta(\Delta - 1)^{\frac{D(T_n)}{2}-1}
 \end{aligned} \tag{3.75}$$

So the number of leaves of a Star is

$$n_1 = \Delta(\Delta - 1)^{\frac{D(T_n)}{2}-1}, \tag{3.76}$$

hence it is a Cayley. □

3.4.6 The Lobster Family

Caterpillar

Theorem 19. *The Caterpillar is a Lobster.*

Proof :

If we remove all the leaves of a Caterpillar and we get a Path and that the Path is a Caterpillar, if a Lobster is a tree in which the removal of all its leaves leave a Caterpillar, then the Caterpillar is a Lobster.

Consequently, all the members of the Caterpillar family also are Lobsters.

Firecracker, Superstar and Dandelion

Theorem 20. *The Firecracker, the Superstar and the Dandelion are Lobsters.*

Proof :

As these three trees need the IsLobster algorithm, they are therefore part of the Lobster family. □

3.4.7 Intersections

Depending of the value of the invariants of a tree T_n , this tree can be included in other families than those previously described.

Here are some examples of these intersections :

- An Olive with a diameter $D(T_n) = 4$ is also a Superstar.
- A Y-tree with a diameter $D(T_n) = 5$ and $n = 7$ is also an Olive.
- A Spider with a distance of $d(u, v) \leq 2$ between its leaves and the vertex v of degree Δ is also a Superstar.

3.5 Limits

Due to the large amount of trees that have been repertoried and somewhere the lack of literature on specific families, it was impossible to include them all in this article. Besides, there is the fact that many articles that have been mainly referenced in the literature were impossible to find for instance [7, 55, 49].

As mentionned previously, this classification didn't take into account the labelled trees. It would be interesting to extend this classification exercise to those trees and more generally to the set of the graphs known.

Chapitre 4

Le logiciel Who's that Tree

Cette recherche, étant plus qu'un simple exercice de classification, a comme but d'être utile aux chercheurs. Pour atteindre cet objectif, la création d'un programme informatique est la solution proposée ici. En répondant à quelques simples questions à propos des invariants du graphe recherché, le programme serait en mesure de retourner à l'utilisateur le nom du graphe en question, sa définition ainsi que ses propriétés.

4.1 Structure

La figure suivante illustre la structure de l'outil en question.

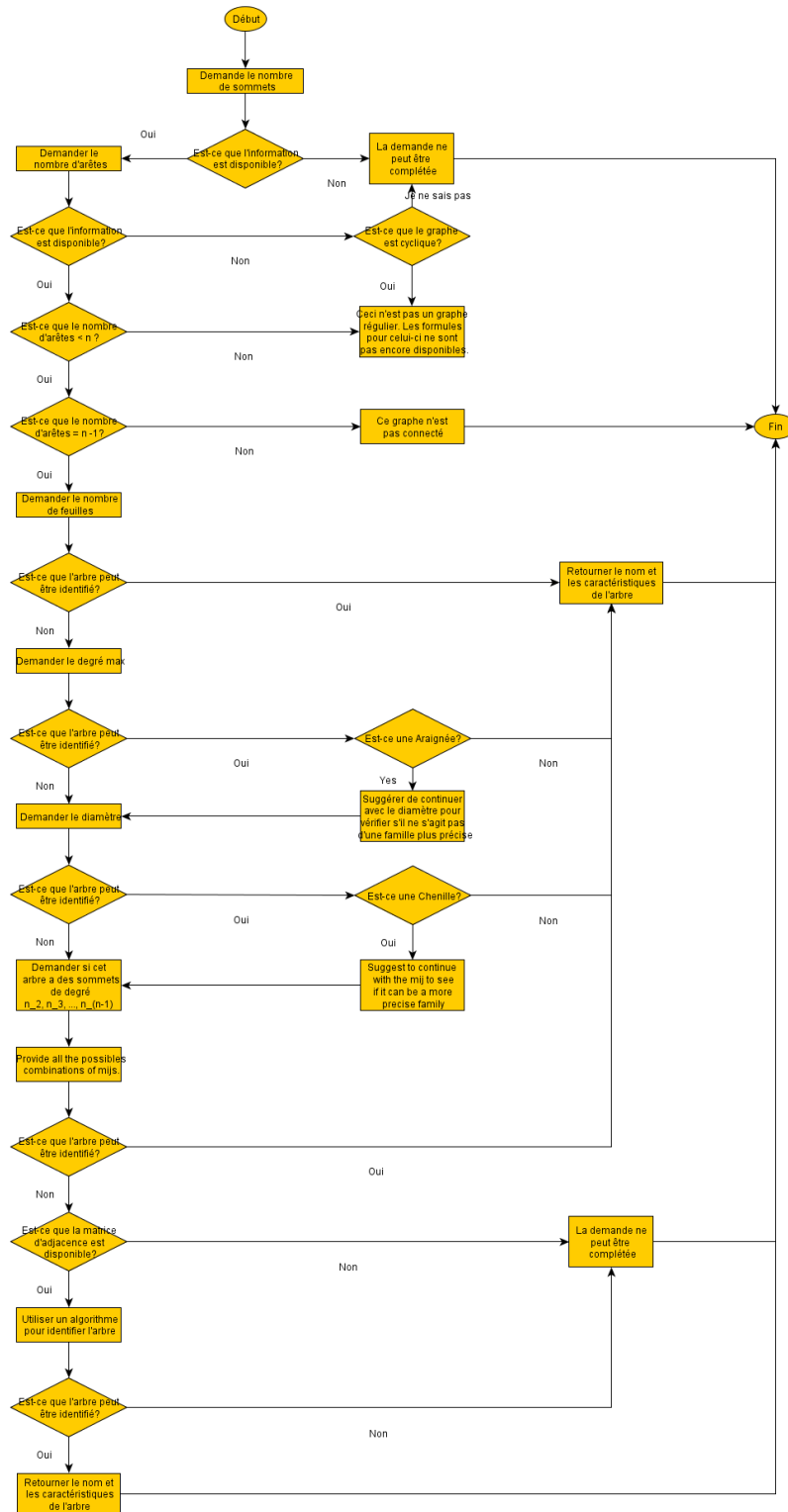


FIGURE 4.1 – Structure du programme "Who's that tree"

L'ordre des questions est optimal. Ceci signifie qu'à chaque étape, un nombre maximal d'arbres peut être identifié et ce, le plus rapidement possible. En raison du faible nombre de familles d'arbres, la recherche de l'optimalité de cet ordre n'a pas nécessité de méthode élaborée. La méthodologie est illustrée dans le tableau suivant.

Invariants nécessaires	Arbres identifiés	Nombre d'invariants	Nombre d'arbres
n, n_1	Chemin, Étoile, Griffes, Fourchette	2	4
n_1, Δ	Brosse a, Peigne a, Araignée, Arbre Y	2	2
n, n_1, Δ	Double Étoile, Brosse a, Peigne a, Araignée, Arbre Y	3	3
$n, n_1, D(T)$	Chenille	3	1
$n, n_1, D(T), \Delta$	Arbre Y Symétrique, Arbre Cayley	4	2

TABLE 4.I – Invariants nécessaires pour l'identification des arbres

Rapidement, il est facile d'identifier qu'avec le nombre d'invariants nécessaires, les arbres n'ayant besoin que de n, n_1 pour leur identification seront d'abord calculés puisqu'il est possible d'identifier quatre arbres grâce à ceux-ci, contrairement à deux dans le cas de n_1, Δ . Ensuite, deux groupes ont trois étapes mais l'un deux permet l'identification de davantage d'arbres, soit celui nécessitant les invariants n, n_1, Δ puisqu'il permet d'identifier cinq arbres, si on inclue les ceux nécessitant seulement les invariants n_1, Δ . Donc le troisième invariant à demander devrait être le degré maximum Δ . Finalement, en demandant le diamètre $D(T_n)$, la Chenille, l'arbre Y symétrique et l'arbre Cayley pourront à leur tour être identifiés.

La structure et la logique du programme sont décrites en détail ci-contre.

Demander le nombre de sommets et le nombre d'arêtes.

- Si le nombre de sommets n'est pas disponible, alors la requête ne peut pas être faite/complétée.
- Si le nombre d'arêtes n'est pas disponible, demander si le graphe est cyclique ou acyclique.

- Si la réponse à cette question est "Je ne sais pas", alors la requête ne peut pas être faite/complétée.
- Si la réponse à cette question est "cyclique" ou que le nombre d'arêtes $\geq n$, la réponse à la requête est qu'il s'agit d'un graphe général et que cette famille n'est pas encore disponible dans le programme.
- Si la réponse à cette question est "acyclique" ou que le nombre d'arêtes $< n$, alors passer aux prochaines questions.

Demander le nombre de feuilles (n_1).

- Certains arbres peuvent être identifiés uniquement avec le nombre de sommets et le nombre de feuilles.
- le Chemin, où $n_1 = 2$,
- l'Étoile, où $n_1 = n - 1$
- la Griffes, où $n_1 = 3$ and $n = 4$
- la Fourchette, où $n = 5$ and $n_1 = 3$

Demander le degré maximum (Δ).

- Plusieurs arbres peuvent être identifiés grâce au degré maximum, au nombre de sommets et de feuilles.
- la Double Étoile, où $n_1 = n - 2$ et $\Delta \leq n - 3$,
 - Si un arbre possède $n_1 = n - 2$ mais que $\Delta \leq n - 2$, alors il s'agit d'une petite Comète.
- la Brosse (a), où $\Delta = \frac{n_1-2}{n-n_1} + 2$,
- le Peigne (a), où $n_1 = \frac{n}{2} + 1$ et $\Delta = 3$,
- l'Araignée, où $\Delta = n_1$,
- l'arbre Y, où $\Delta = n_1$ et $n_1 = 3$

Demander le degré maximum (Δ).

- Plusieurs arbres peuvent être identifiés grâce au degré maximum, leur nombre de sommets et de feuilles :
 - la Brosse (a), où $\Delta = \frac{n_1-2}{n-n_1} + 2$,
 - le Peigne(a), où $n_1 = \frac{n}{2} + 1$ et $\Delta = 3$,
 - l'Araignée, où $\Delta = n_1$,
 - l'arbre Y, où $\Delta = n_1$ et $n_1 = 3$.

- Puisque plusieurs arbres font partie de la famille des Araignées, le programme suggère aux utilisateurs dont le graphe correspond à cette famille de continuer avec d'avantage de questions afin de préciser si l'arbre appartient à une famille plus spécifique.

Demander le diamètre ($D(T_n)$).

- Plusieurs arbres peuvent être identifiés grâce à leur diamètre, leur degré maximum, leur nombre de sommets et de feuilles :
 - la Chenille, où $D(T_n) = n - n_1 + 1$,
 - l'arbre Y symétrique, où $n_1 = 3, \Delta = 3$ et $D(T_n) = 2 * (\frac{n-1}{3}) - 1$,
 - l'arbre Cayley, où $n_1 = \Delta(\Delta - 1)^{\frac{D(T_n)}{2} - 1}$.
- Puisque plusieurs arbres font partie de la famille des Chenilles, le programme suggère aux utilisateurs dont le graphe correspond effectivement à cette famille de continuer avec d'avantage de questions afin de préciser si l'arbre appartient à une famille plus spécifique.

À l'exception des arbres faisant partie de la famille du Homard et du Bambou, tous les arbres auront été caractérisés d'une manière ou d'une autre grâce aux premières questions. La Comète, la Double Comète, la Brosse (b) et le Peigne (b) auront été considérés comme des Chenilles par le programme, tandis que l'Olivier et l'Éperon auront été classés parmi les Araignées. Cependant, ces arbres possèdent des caractéristiques non suffisantes à partir des invariants précédemment utilisés, qui peuvent aider à rediriger l'utilisateur vers la bonne sous-catégorie. En confirmant que ces caractéristiques non discriminantes sont rencontrées, le programme peut suggérer à l'utilisateur d'entrer les m_{ij} et/ou la matrice d'adjacence de l'arbre afin de confirmer si ce dernier appartient réellement à une famille plus spécifique.

C'est le cas pour :

- Les arbres nécessitant les m_{ij} et pouvant être sujets à une sous-catégorisation préliminaire grâce à leurs caractéristiques non-discriminantes :
 - la Brosse (b), où $D(T_n) = \frac{n}{\Delta-1} + 1$ et $n_1 = (\Delta - 2)(D(T_n) - 1)$,
 - le Peigne (b), où $D(T_n) = \frac{n}{2} + 1$ et $n_1 = \frac{n}{2}$,
 - la Comète, où $D(T_n) = n - n_1 + 1$ et $\Delta = n_1$,
 - l'Éperon, où $\Delta = n_1$ et $D(T_n) \leq 4$.

- La Double Comète, où $D(T_n) = n - n_1 + 1$, ne possède pas de caractéristiques non-discriminantes plus précises. Ce faisant, pour tous les arbres ayant un tel diamètre, donc étant des Chenilles et n’ayant pas encore été caractérisés dans une famille plus précise, suggérer à l’utilisateur de continuer avec les m_{ij} et/ou la matrice d’adjacence de l’arbre.
- Les arbres ayant besoin de la matrice d’adjacence et pouvant être sujets à une sous-catégorisation préliminaire grâce à leurs caractéristiques non-discriminantes :
 - l’Olivier, où $D(T_n) = 2n_1 - 1$, $n = 0.5n_1^2 + 0.5n_1 + 1$ et $\Delta = n_1$,
 - le Feu d’Artifice, où $D(T_n) = \frac{n-n_1}{2} + 3$,
 - le Tuyau d’Arrosage, où $D(T_n) = 2(\frac{n}{n_1} - 1) + n_1 - 1$ et $\Delta = 3$.

Si une des sous-caractéristiques d’un arbre pouvant être identifié par les m_{ij} est rencontrée, proposer à l’utilisateur d’identifier les m_{ij} .

- Demander à l’utilisateur quels sont les degrés des sommets de cet arbre : (1, 2, 3, ..., n).
- Alors, les m_{ij} potentiels apparaissent :
 - m_{12} ,
 - m_{13} ,
 - $m_{1...}$,
 - ...
 - $m_{...n}$.
- Donc l’utilisateur n’a qu’à entrer les valeurs correspondantes.
- Ceci contribue à éviter à l’utilisateur un oubli de m_{ij} par manque d’attention.
- Si l’une des sous-caractéristiques précédemment mentionnées n’est pas rencontrée ou si le programme ne peut pas reconnaître l’arbre en question, alors la seule possibilité qu’il reste à l’utilisateur pour voir son arbre identifié est de fournir la matrice d’adjacence de celui-ci.

Autrement, suggérer tout de même à l’utilisateur d’utiliser la matrice d’adjacence.

Tout au long de ce processus, l’utilisateur peut s’il le désire, fournir la matrice d’adjacence de son graphe, afin de rendre le programme plus flexible pour l’usager.

4.2 Matrice d'adjacence

Lorsqu'il reçoit la matrice d'adjacence, le programme va tout d'abord identifier si le graphe est bien un arbre. Par la suite, les invariants seront calculés, dans le même ordre que celui dicté par les questions posées à l'utilisateur, incluant les m_{ij} . Si l'arbre n'a pas pu être identifié avec ces derniers, alors le programme testera l'algorithme IsLobster, suivi de IsSuperstar et de IsDandelion si cela est nécessaire. Cet ordre est également optimal considérant l'inclusion des familles, puisque la Superétoile peut être un Pissenlit. L'algorithme IsOlive sera utilisé uniquement si l'arbre préalablement identifié correspond à une araignée et si les caractéristiques non discriminantes sont préalablement rencontrées, tel que décrit dans la section précédente. De la même manière, l'algorithme IsFirecracker et IsSprayingPipe seront seulement utilisés si les caractéristiques non discriminantes du Feu d'Artifice ou du Tuyau d'Arrosage sont rencontrées.

Il est cependant possible qu'un arbre ne soit membre d'aucune famille. Dans ce cas, le programme indiquera simplement à l'utilisateur que cet arbre ne peut pas être identifié.

4.2.1 Algorithmes reliés aux invariants

Nombre de feuilles Tous les sommets qui sont connectés à seulement un autre sommet sont des feuilles (dont le degré est égal à un). Leur nombre est n_1 .

Diamètre Il s'agit du chemin principal tiré de l'algorithme IsLobster.

Degré maximum (Δ ou $d(t_1)$ et $d(t_2)$)

- Identifier le vecteur de degrés, qui est la somme de chaque ligne de la matrice, afin de trouver la valeur du degré de chaque sommet.
- Vérifier si $n_1 + n_2 = n - 2$.
- Si faux, il y a seulement un sommet v considéré comme degré maximum Δ .
- Si vrai, il y a deux sommets principaux t_1 et t_2 tels que $t_1 > t_2$ ayant degrés $d(t_1)$ et $d(t_2)$ de plus grande valeur.

- Puisque $t_1 > t_2$, le sommet t_1 sera considéré comme celui ayant le degré maximum Δ dans les premières équations reliées à l'identification des arbres.
- Le sommet t_2 est celui qui possède un degré $d(t_2) > 2$.

Les m_{ij}

1. Identifier pour chaque sommet v de degré $d(v)$, le nombre de sommets voisins u de degré $d(u) \geq d(v)$ qu'il possède.
2. Faire la somme du nombre de voisins de degré $d(u) \geq d(v)$ de tous les sommets de degré $d(v)$.
3. Refaire les étapes 1 et 2 jusqu'à ce que tous les m_{ij} aient été calculés.

Chapitre 5

Conclusion

Cette recherche se voulait comme un exercice de classification, ayant comme but d'éclaircir les différentes définitions et familles d'arbres étudiées et possibles. Les familles ont donc été décrites puis leur filialité a été exposée. Finalement, la structure du programme "Who's That Tree" servant à identifier les familles, a été exposée.

Plusieurs limites sont cependant présentes. Bien que ce mémoire ait passé en revue un certain nombre d'arbres, certaines familles n'ont pas pu être répertoriées. Un manque de littérature des familles en particulier y a certainement contribué. De plus, plusieurs articles, cités par différents auteurs se sont avérés introuvables, c'est d'ailleurs le cas de [7, 55, 49], ce qui rend leur interprétation moins fiable. Dans un autre ordre d'idées, il a été clairement démontré que de simples invariants se sont avérés insuffisants pour classifier l'ensemble des familles de graphes. Le recours à des algorithmes s'est donc imposé, ce qui ne correspondait pas à l'idée première de vouloir caractériser de manière simple et efficace les arbres.

Ce mémoire, qui se voulait une introduction à une classification complète des graphes ouvre la porte à des recherches complémentaires sur le sujet, tant pour l'ajout de familles de graphes que dans les invariants utilisés pour classifier et distinguer les familles entre elles.

Bibliographie

- [1] R. Ahuja, R. Magnanti, and J. Orlin. *Network flows : Algorithms and applications*. Prentice Hall, 1993.
- [2] J.A. Bondy and U.S.R. Murty. *Graph theory with applications*. Elsevier Science Publishing Co., Inc., 1982.
- [3] R. Boulet. The centipede is determined by its laplacian spectrum. *Comptes Rendus Mathématique*, 346 :711–716, 2008.
- [4] A. Brandstadt, V. Bang Le, and J.P. Spinrad. *Graph classes : a survey*. Society for Industrial and Applied Mathematics, 1999.
- [5] R. A. Brualdi and J. L. Goldwasser. Permanent of the laplacian matrix of trees and bipartite graphs. *Discrete Mathematics*, 48 :1–21, 1984.
- [6] G. Caporossi, I. Gutman, and P. Hansen. Variable neighborhood search for extremal graphs iv : Chemical trees with extremal connectivity index. *Computers & Chemistry*, 23 :469–477, 1999.
- [7] W.C. Chen, H. I. Lu, and Y.N Yeh. Operations of interlaced trees and graceful trees. *Southeast Asian Bulletin of Mathematics*, 21 :337–348, 1997.
- [8] D. Cvetkovic, P. Rowlinson, and S. K. Simic. Eigenvalue bounds for the signless laplacian. *Publications de l'institut mathématique*, tome 81(95) :11–27, 2007.
- [9] P. Dankelmann. On the distance distribution of trees. *Discrete Applied Mathematics*, 159 :945–952, 2011.
- [10] P. Dankelmann, W. Goddard, and C. S. Stewart. The average eccentricity of a graph and its subgraphs. *Utilitas Mathematica*, 65 :41–51, 2004.
- [11] F. Depiero, M. Trivedi, and S. Serbin. Graph matching using a direct classification of node attendance. *Pattern Recognition*, 29 :2031–1048, 1996.
- [12] M. V. Diudea and B. Parv. Molecular topology. 25. hyper-wiener index of dendrimers. *Journal of Chemical Information and Computer Sciences*, 35 :1015–1018, 1995.
- [13] S. El-Basil. Gutman trees. *Journal of the Chemical Society - Faraday Transactions II*, 82 :299–316, 1986.
- [14] S. El-Basil. Caterpillar (gutman) trees in chemical graph-theory. *Topics in current chemistry*, 153 :273–289, 1990.
- [15] F. Esser and F. Harary. Digraphs with real and gaussian spectra. *Discrete Applied Mathematics*, 2 :113–124, 1980.

- [16] G. Fernandez. *Data Mining Using SAS Applications*. Chapman and Hall/CRC 2002, 2002.
- [17] J.-L. Fouquet and V. Giakoumakis. On semi- p_4 -sparse graphs. *Discrete Mathematics*, 165-166 :277–300, 1997.
- [18] J. A. Gallian. A dynamic survey of graph labeling. *The Electronic Journal of Combinatorics*, 18 :1–256, 2011.
- [19] D. Goncalves and P. Ochem. On star and caterpillar arboricity. *Discrete Mathematics*, 309 :3694–3702, 2009.
- [20] A. D. Gordon. *Classification, 2nd edition*. Number 272. Chapman and Hall/CRC, 1999.
- [21] J. L. Gross and J. Yellen. *Graph Theory and its application*. Discrete mathematics and its applications. Chapman & Hall/CRC, 2006.
- [22] I. Gutman, Y.-N. Yeh, S.-L. Lee, and J.-C. Chen. Wiener numbers of dendrimers. *Communications in Mathematical and in Computer Chemistry (MATCH)*, 30 :103–115, 1994.
- [23] M. F. Hancock. *Practical Data Mining*. Auerbach Publications, 2011.
- [24] F. Harary. The maximum connectivity of a grap. *Proceedings of the National Academy of Sciences of the United States of America*, 48(7) :1142–1146, 1962.
- [25] F. Harary. *Graph Theory*. Addison-Wesley Series In Mathematics. Addison-Wesley Publishing Company, Inc., 1969.
- [26] F. Harary and M. Lewinter. Spanning subgraphs of a hypercube ii : Double starlike trees. *Mathematical and Computer Modelling*, 11 :216–217, 1988.
- [27] F. Harary and M. Lewinter. The starlike trees which span a hypercube. *Computers and Mathematics with Applications*, 15(4) :299–302, 1988.
- [28] F. Harary and A. J. Schwenk. The number of caterpillars. *Discrete Mathematics*, 6 :359–365, 1973.
- [29] F. Harary and G. E. Uhlenbeck. On the number of husimi trees, i. *Proceedings of the National Academy of Sciences of the United States of America*, 39(4) :315–322, 1953.
- [30] J.J. Ibanez, M. Ruiz-Ramos, and A. M. Tarquis. Mathematical structures of biological and pedological taxonomies. *Geoderma*, 134 :360–372, 2006.
- [31] A. Ilic. Distance spectral radius of trees with given matching number. *Discrete Applied Mathematics*, 158 :1799–1806, 2010.
- [32] J. Jeba Jesintha and G. Sethuraman. All arbitrarily fixed generalized banana trees are graceful. *Mathematics in Computer Science*, 5 :51–62, 2011.
- [33] M. Kobeissi and M. Mollard. Spanning graphs of hypercubes : starlike and double starlike trees. *Discrete Mathematics*, 244 :231–239, 2002.
- [34] K. Kutnar and D. Marusic. A complete classification of cubic symmetric graphs of girth 6. *Journal of Combinatorial Theory*, 99 :162–184, 2009.

- [35] J. H. Kwak and Y. A. Kwon. Classification of nonorientable regular embeddings of complete bipartite graphs. *Journal of Combinatorial Theory*, 101 :191–205, 2011.
- [36] S. Lee and Y. Park. The classification and strategic management of services in e-commerce : Development of service taxonomy based on customer perception. *Expert Systems with Applications*, 36 :9618–9624, 2009.
- [37] V. E. Levit and E. Mandrescu. On the roots of independence polynomials of almost all very well-covered graphs. *Discrete Applied Mathematics*, 156 :478–491, 2008.
- [38] T. Li and S.T. Cavusgil. A classification assessment of research streams in international marketing. *International Business Review*, 4(3) :251–277, 1995.
- [39] X. Li and Y. Li. The asymptotic value of the randic index for trees. *Advances in Applied Mathematics*, 47 :365–378, 2011.
- [40] W. Luo and B. Zhou. On ordinary and reverse wiener indices of non-caterpillars. *Mathematical and Computer Modelling*, 50 :188–193, 2009.
- [41] A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, and M. Saeuens. Semi-supervised classification and betweenness computation on large, sparse, directed graphs. *Pattern Recognition*, 44 :1212–1224, 2011.
- [42] D. Mishra and P. Panigrahi. Some graceful lobsters with all three types of branches incident on the vertices of the central path. *Computers and Mathematics with Applications*, 56 :1382–1394, 2008.
- [43] D. Mishra and P. Panigrahi. Some new classes of graceful lobsters obtained from diameter four trees. *Mathematica Bohemica*, 135(3) :257–278, 2010.
- [44] C.M. Mynhardt. Trees with depression three. *Discrete Mathematics*, 308 :855–864, 2008.
- [45] G. R. Newkome, C. N. Moorefield, G. R. Baker, A. L. Johnson, and R. K. Behera. Alkane cascade polymers possessing micellar topology : Micellanoic acid derivatives. *Angewandte Chemie International Edition in English*, 30(9) :1176–1178, 1991.
- [46] J. Nievergelt, J. Pradels, C.K. Wong, and P.C. Yue. Bounds on the weighted path length of binary trees. *Information Processing Letters*, 1(6) :220–225, 1972.
- [47] G.R. Omid and K. Tajbakhsh. Starlike trees are determined by their laplacian spectrum. *Linear Algebra and its applications*, 422 :654–658, 2007.
- [48] M. Ostili. Cayley trees and bethe lattices : A concise analysis for mathematicians and physicists. *Physica A*, 391 :3417–3423, 2012.
- [49] A. M. Pastel and H. Raynaud. Numerotation gracieuse des oliviers. in *Colloq. Grenoble, Publications Université de Grenoble*, 55 :218–223, 1978.
- [50] C. Perruchet. Hierarchical classification of mathematical structures. *Statistics & Probability Letters*, 1 :61–67, 1982.
- [51] J. D. Phillips and D. A. Marion. Soil geomorphic classification, soil taxonomy and effects on soil richness assessments. *Geoderma*, 141 :89–97, 2007.
- [52] P. Robert. *Le Petit Robert : dictionnaire alphabétique et analogique de la langue française / texte remanié et amplifié sous la direction de Josette Rey-Debove et Alain Rey*. Dictionnaires Le Robert, 2004.

- [53] E. Robeva. An extensive survey of graceful trees. Master's thesis, Stanford University, June 2011.
- [54] Z. Ryjáček. On a closure concept in claw-free graphs. *Journal of Combinatorial Theory, Series B* 70 :217–224, 1997.
- [55] C. Sekar. *Studies in Graph Theory*. PhD thesis, Madurai Kamaraj University, 2002.
- [56] C. N. Silla and A. A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22 :31–72, 2011.
- [57] S.K. Simic, E.M.L. Marzi, and F. Belardo. On the index of caterpillars. *Discrete Mathematics*, 308 :324–330, 2008.
- [58] C. De Simone and A. Sassano. Stability number of bull- and chair-free graphs. *Discrete Applied Mathematics*, 41 :121–129, 1993.
- [59] V. Swaminathan and P. Jeyanthi. Super edge-magic strength of fire crackers, banana trees and unicyclic graphs. *Dis*, 306 :1624–1636, 2006.
- [60] S. Tuffery. *Data Mining et statistique décisionnelle - L'intelligence des données*, volume 705. 2010.
- [61] J.-G. Wang, D.-J. Jin, X.-G. Lu, and D. Zhang. The gracefulfulness of a class of lobster trees. *Mathematical and Computer Modelling*, 20(9) :105–110, 1994.
- [62] D. B. West. *Introduction to graph theory*. Prentice Hall, 2001.
- [63] P. Winkler. Mean distance in a tree. *Discrete Applied Mathematics*, 27 :179–185, 1990.
- [64] X.-Y. Yuan, H.-Y. Shan, and Y. Liu. On the laplacian spectral radii of trees. *Discrete Mathematics*, 309 :4241–4246, 2009.
- [65] S. Zhou. Classification of a family of symmetric graphs with complete 2-arc-transitive quotients. *Discrete Mathematics*, 309 :5404–5410, 2009.