

2007

Forbidding and enforcing of formal languages, graphs, and partially ordered sets

Daniela Genova

University of South Florida

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>



Part of the [American Studies Commons](#)

Scholar Commons Citation

Genova, Daniela, "Forbidding and enforcing of formal languages, graphs, and partially ordered sets" (2007). *Graduate Theses and Dissertations*.

<http://scholarcommons.usf.edu/etd/2185>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Forbidding and Enforcing of Formal Languages, Graphs, and Partially Ordered
Sets

by

Daniela Genova

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Mathematics
College of Arts and Sciences
University of South Florida

Major Professor: Nataša Jonoska, Ph.D.
Gregory McColm, Ph.D.
Masahico Saito, Ph.D.
Stephen Suen, Ph.D.

Date of Approval:
June 14, 2007

Keywords: Classes of formal languages, Language families, Subwords, Subgraphs,
Posets, DNA computing

©Copyright 2007, Daniela Genova

Dedication

To my parents

Acknowledgements

This work and my research experience would not have been possible without the ideas, suggestions, monitoring, guidance, and support from my advisor Nataša Jonoska. Her expertise in language theory, theory of computation, topology, graph theory and DNA computing, among other research areas, made it all possible for me to attempt such an endeavor. I am honored to have had such a distinguished advisor and am extremely fortunate to have been her student. I am forever indebted to her!

I would like to thank all members of my dissertation committee: G. L. McColm, M. Saito, and S. Suen for their time and effort in reviewing previous drafts of this dissertation and providing valuable suggestions, as well as, great ideas for future work. I am very proud to have such outstanding scholars on my dissertation committee!

My special thanks go to all my mathematics professors who nourished my love for mathematics and encouraged me to pursue challenging ideas. I thank N. Jonoska, G. McColm, S. Suen, M. Saito, M. Ismail, J. Liang, E. Clark, B. Curtin, and B. Shekhtman for that. In addition, this journey would not have been possible without the help from the entire Mathematics Department at USF led by M. McWaters and S. Rimbey and the support staff Jim, Aya, Sarina, Evelyn, Beverly, Denise, Mary Ann, Nancy, Frances, Barbara. I thank them all!

Two people were very instrumental to enhancing my teaching experience: F.

Zerla and D. Kerr. They taught me the importance of service to the mathematical community and introduced me to the MAA. My research experience was much enriched by my friend and collaborator M. Cavaliere who taught me the basics of Membrane Computing. I am so fortunate to have have met them all!

The constant support, encouragement, and love from family and friends was invaluable. I thank Philip, Stella, Laura, my parents, Zornitza, Laura K., John, Karen, Elena, and Peter for always believing in me and providing purpose.

Last but not least, I would like to recognize the following institutions for their financial support. I thank the Department of Mathematics at the University of South Florida for awarding me a Graduate Teaching Assistantship; Eckerd College for providing me with a Visiting Assistant Professorship for two years, while being a graduate student; the GSBO for awarding me travel grants to present my research at local, state, and international conferences; University of Utah and the National Science Foundation for the travel grant to present my research in Sydney, Australia; the Association of Women in Mathematics for sponsoring my trip to the JMM in New Orleans and making it possible for me to participate in the AWM workshop and present my research there; the Departments of Mathematics at Indiana-Purdue University at Fort Wayne and University of North Florida for inviting me to present my work there. I can't help but recognize how fortunate I am to have had their support!

Table of Contents

List of Figures	iii
Abstract	iv
1 Introduction	1
2 Forbidding and Enforcing of Formal Languages	6
2.1 Definitions	6
2.2 Minimal Normal Forms	9
2.3 Maximal Languages	12
2.4 Extended Forbidding Sets	20
2.5 Generated Languages	23
3 Topological Properties of fe -Families of Languages	28
3.1 The Cantor Space $\mathcal{P}(A^*)$	28
3.2 Continuous Functions	32
3.3 Chomsky Families as Subspaces of $\mathcal{P}(A^*)$	34
3.4 Topological Properties of fe -Families	35
4 Morphisms and fe -Families of Languages	41
4.1 Morphic Maps and fe -Families	41
4.2 Characterizing Morphic Images as fe -Families	44
5 Forbidding and Enforcing of Graphs	49
5.1 Definitions	49

5.2	Connecting Graphs	51
5.3	Graph fe -Systems and Their Properties	64
5.4	Forbidding through Enforcing	68
5.5	Characterizations of Some Classes of Graphs by fe -Systems	69
6	Normal Forms for Graph fe -Systems	75
6.1	Normal Forms for Forbidding Sets	75
6.2	Normal Forms for Enforcing Sets	88
7	Forbidding and Enforcing on Partially Ordered Sets	94
7.1	fe -Families as Sets of Subposets	95
7.2	fe -Systems Defining a Single Subposet	103
7.3	Upper Bounds	108
7.4	Normal Forms for Forbidding Sets	111
7.5	Normal Forms for Enforcing Sets	119
8	Computing with fe -Systems	123
8.1	Modeling Molecular Bonding and Splicing Systems	124
8.2	Information Processing by fe -Systems	127
	References	132
	About the Author	End Page

List of Figures

2.1	Tree associated with the language A^*	17
2.2	Trees associated with the maximal languages from Example 2.3.2	19
3.1	Continuous functions	33
5.1	The graph $S_{C_3P_4C_4}$ from Example 5.2.3	52
5.2	Minimal connecting graphs of F from Example 5.2.5	53
5.3	Graphs related to Example 5.2.20	56
5.4	Graphs related to Example 5.2.29	60
5.5	The graph $G_{\mathcal{U}}$ from Definition 5.2.30	61
5.6	A minimal connecting graph of \tilde{X} in G from Example 5.2.34	63
5.7	Graph related to Example 5.3.6	65
5.8	Tree related to Proposition 5.5.1	70
5.9	Graphs related to \mathcal{E} in Corollary 5.5.8	74
6.1	Graphs related to \mathcal{E} in Example 6.2.8	90
8.1	(a) Cutting with restriction enzymes (b) DNA recombination	125
8.2	Missing nucleotides: (a) diagonally (b) in one strand.	126

Forbidding and Enforcing of Formal Languages, Graphs and Partially Ordered Sets

Daniela Genova

ABSTRACT

Forbidding and enforcing systems (*fe*-systems) provide a new way of defining classes of structures based on boundary conditions. Forbidding and enforcing systems on *formal languages* were inspired by molecular reactions and DNA computing. Initially, they were used to define new classes of languages (*fe*-families) based on forbidden subwords and enforced words. This paper considers a metric on languages and proves that the metric space obtained is homeomorphic to the Cantor space. This work studies Chomsky classes of families as subspaces and shows they are neither closed nor open. The paper investigates the *fe*-families as subspaces and proves the necessary and sufficient conditions for the *fe*-families to be open. Consequently, this proves that *fe*-systems define classes of languages different than Chomsky hierarchy. This work shows a characterization of continuous functions through *fe*-systems and includes results about homomorphic images of *fe*-families. This paper introduces a new notion of connecting graphs and a new way to study classes of *graphs*. Forbidding-enforcing systems on graphs define classes of graphs based on forbidden subgraphs and enforced subgraphs. Using *fe*-systems, the paper characterizes known classes of graphs, such as paths, cycles, trees, complete graphs and *k*-regular graphs. Several normal forms for forbidding and enforced sets are stated and proved. This work introduces the notion of forbidding and enforcing to *posets* where *fe*-systems are used to define families of subsets of a given poset,

which in some sense generalizes language fe -systems. Poset fe -systems are, also, used to define a single subset of elements satisfying the forbidding and enforcing constraints. The latter generalizes graph fe -systems to an extent, but defines new classes of structures based on weak enforcing. Some properties of poset fe -systems are investigated. A series of normal forms for forbidding and enforcing sets is presented. This work ends with examples illustrating the computational potential of fe -systems. The process of cutting DNA by an enzyme and ligating is modeled in the setting of language fe -systems. The potential for use of fe -systems in information processing is illustrated by defining the solutions to the k -colorability problem.

Chapter 1

Introduction

The constant attempt to improve computational capabilities has led scientists to investigation of unconventional computational tools. The authors in [33] provide a comprehensive discussion of new computing paradigms. Many computational models in DNA computing, self-assembly, and membrane computing have been proposed (see [30, 32, 35, 41]). Introduced in [30] by Gh. Păun, P systems employ nested membranes as a computational tool. A set of evolution rules acts on a multiset of objects that are placed in the regions enclosed by the membranes. The objects evolve according to these rules and can pass through the membranes, whereas the rules never leave the regions they are in. All objects (words) collected outside the skin membrane form a language and thus determine the computational power of the P system. Many variants of P systems have been proposed (see [31]). In [3, 4] M. Cavaliere and the author propose one such variant called CR P systems, where objects remain in their regions and never leave them, whereas evolution rules can pass through the membranes and act on them. A variant of CR P systems is studied in [14].

Chemical properties of DNA and actions of restriction enzymes (see [37, 42]) have inspired many DNA computing models like [1, 22, 24, 41]. Encoding the problem using DNA molecules involves avoiding undesirable hybridization of DNA strands. DNA coding properties needed to properly encode a problem have been introduced in [21] and widely studied in recent years (see for ex. [23, 28]).

All of these models are based on classical formal language theory (see [20, 36])

where grammars and automata define languages deterministically. In [8] Rozenberg and Ehrenfeucht proposed a new way of defining classes of languages based on two types of boundary conditions that captures the non-determinism of biomolecular reactions. Forbidding conditions exclude certain combinations of words (molecules) in a language and enforcing conditions state that the presence of some words (molecules) will trigger other words (molecules) in the language (solution). Computation evolves according to the rules proscribed by the enforcing sets while avoiding the formation of forbidden combinations of words prescribed by the forbidding sets. Forbidding and enforcing systems (*fe*-systems) on formal languages were, also, studied in [9, 10, 40]. In [16], Jonoska and the author investigated the topological properties of *fe*-systems and their morphic images. In [5], Jonoska and Cavaliere used *fe*-systems to define new variants of P systems. In [13] *fe*-systems were used to define properties of DNA graphs. In [15], Jonoska and the author presented a generalized way to define classes of structures through the boundary constraints of forbidding and enforcing. The paper showed that *fe*-systems on formal languages can be used to model DNA splicing, to define new classes of graphs, and for information processing.

Chapters 2, 3, and 4 are an extended version of [16]. The basic topological notions are assumed and can be found in [29]. The definitions of *fe*-systems are stated along with some of their properties, including new normal forms. Theorem 2.4.3 provides a characterization of extended *f*-families. The minimal generated languages extend the notion of \mathcal{E} -extensions introduced in [8]. The word metric 3.1.1 on the space of formal languages is the same as the one implicitly used in [10, 40] and follows a similar approach as in [26, 27]. This paper shows that the language space equipped with the word metric is homeomorphic to the Cantor space. A characterization of the continuous morphisms on that space is provided. Theorem 3.2.1 states that a morphism is continuous if and only if it is λ -free. This corresponds to the characterization of continuous maps on infinite sequences in [39]. In addition, examples of other continuous functions are presented that come from well known operations on languages such as taking products of languages with a fixed language,

Kleene star operation, intersection or union with a fixed language, etc. Considering the *the language space* as a topological space only comes to interest when studying *fe*-systems, since as it is stated in Theorem 3.3.2 none of the Chomsky families corresponds to an open or a closed set in that space. *fe*-systems define families that are closed sets [8, 40]. Theorem 3.4.9 states the necessary and sufficient conditions for *fe*-systems to define open families of languages. Namely, *fe*-systems with empty forbidders and finitely many enforcers define nontrivial open subspaces. Hence, *fe*-systems provide a new way to classify formal languages, different from Chomsky's hierarchy. In Proposition 3.4.7 the notion of generated languages is used to prove that infinite enforcing sets define non-open families of languages. Chapter 4 contains observations about morphisms that map *fe*-families into *fe*-families. This work shows that morphisms map an *f*-family to an extended *f*-family if and only if the morphism is induced by a symbol-to-symbol map. On the other hand, if an *e*-family is mapped into an *e*-family, the morphism is necessarily surjective.

In Chapter 5 an entirely new way of defining classes of graphs is presented based on boundary conditions. The chapter introduces a new notion called *connecting graphs*. Given a set of connected graphs, a connecting graph is a graph which contains each graph from the set as a subgraph. Minimal connecting graphs are defined with respect to subgraphs. This work shows that even small sets of graphs have an infinite number of minimal connecting graphs. This chapter defines forbidding sets of graphs as a collection of finite sets of connected graphs. The *f*-families are defined as all graphs that do not have forbidden combinations of subgraphs. Historically, forbidden graphs have only been defined as "strict" forbidding sets, where each forbidders is a singleton. Such definition has only been used in the case of induced subgraphs rather than subgraphs. In [7, 12, 17]) the main objective of these forbidden graphs is to prove hamiltonicity. Forbidden graphs were, also, used in extremal graph theory in Turan type problems. For a comprehensive list of papers see [2, 18]. In this work, *e*-families are defined as graphs in which certain subgraphs are required to be "enclosed" in larger subgraphs. A new tool for representing the set of connected graphs called the $G_{\mathcal{U}}$ graph is introduced. It is used to represent

and study certain properties of forbidding and enforcing sets. Using fe -systems, this paper characterizes some well known classes of graphs such as trees, bi-partite graphs, paths and cycles, complete graphs, and k -regular graphs. This work contains an investigation of redundant sets and several normal forms for forbidding sets and for enforcing sets are stated and proved.

Chapter 7 extends the forbidding and enforcing paradigm to partially ordered sets. In the first section the fe -family is defined as a set of subposets and in this respect, it generalizes the language fe -systems. Several normal forms for forbidding sets and enforcing sets from [8, 9, 40] are generalized to posets, including the minimal normal form and finitary normal form. The remaining sections of Chapter 7 define and study the fe -family as a single poset and in this respect generalizes the graph model from Chapter 5 to an extent. Forbidding sets are a generalization of the graph forbidding systems, but the enforcing sets present a new way to define classes of structures using the concept of “weak” enforcing. Examples of different types of posets, such as the natural numbers with divisibility and words with subword order are presented. Such fe -systems are used to characterize some familiar classes of structures. In the case of words, the fe -systems define a single language, as opposed to the language fe -systems discussed in Chapter 2, where a family of languages is obtained. Such approach of defining languages is entirely new compared to the traditional ways to define a language using a grammar or an automaton see [36]. Again, some normal forms for forbidding sets and enforcing sets are presented.

Chapter 8 is devoted to the motivation for and examples of applications of the forbidding and enforcing theory. The first section discusses an example of splicing with an enzyme and ligating DNA strands, which proves that fe -systems can provide an equivalent definition of splicing, originally defined in [19]. The second example shows how fe -systems can be used for information processing. Here, defining the solutions to the k -colorability problem is used as an example.

This paper is organized as follows. Chapter 2 defines language fe -systems and presents some of their properties. It investigates normal forms for fe -systems, maximal languages in forbidding families, extended forbidding sets, and generated sets

in enforcing families. Chapter 3 investigates the topological space of formal languages under the standard language metric and fe -families as subspaces. It shows this space is homeomorphic to the Cantor space and that based on topological properties fe -systems define classes of formal languages different than Chomsky classes. The continuous morphisms are, also, in this chapter, as well as, the characterization of fe -families that define open subspaces. Morphisms that map fe -families are investigated in Chapter 4. Chapter 5 introduces fe -systems in graph theory as a new way to define classes of graphs and presents some of their properties. It defines and investigates a new concept of connecting graphs of a set and presents fe -systems characterizations of familiar classes of graphs like trees, paths and cycles, bipartite, complete, and k -regular graphs. Chapter 6 presents normal forms for graph forbidding sets and enforcing sets. Chapter 7 defines fe -systems on partially ordered sets and investigates two generalized models. Chapter 8 presents two examples describing the computational capabilities of fe -systems.

Chapter 2

Forbidding and Enforcing of Formal Languages

Forbidding and enforcing systems were first defined on formal languages. Inspired by molecular reactions, the authors in [8] use boundary conditions of forbidding and enforcing to define classes of languages (*fe*-families). Properties of *fe*-systems are discussed in [8, 9, 10, 40]. The properties of *fe*-families discussed in this chapter are mainly from [16] and include maximal languages in *f*-families and generated languages in *e*-families.

2.1 Definitions

This section recalls the definitions of *fe*-systems from [8, 40] and discusses some of the basic properties of forbidding-enforcing systems. A finite set of symbols A is called an *alphabet* and the set of all *words* (finite strings) over A forms a free monoid A^* . A subset of A^* is called a *language* and $\mathcal{P}(A^*)$ (named thereafter the *language set*) is the set of all languages over A . The *length* of a word $w \in A^*$ is denoted by $|w|$ and A^m ($A^{\leq m}$) is the set of all words over A of length m ($\leq m$). The empty word, denoted by λ has length 0. The set of all words over A with positive length is denoted by A^+ and forms a free semigroup. For a language $L \neq A^*$, $L^{\leq m}$ denotes the set of all words in L with length $\leq m$. For $a \in A$, a^* is the language containing all finite concatenations of a 's, i.e., $a^* = \{a^n \mid n \geq 0\}$. On the other hand, an infinite string of symbols from A is called an ω -*word* and the set of all infinite words over A is denoted by A^ω . So, $A^\omega = \{\xi_0\xi_1\xi_2\dots \mid \xi_i \in A, i \geq 0\}$.

The word $y \in A^*$ is a *subword (factor)* of the word $x \in A^*$ denoted with $y \underline{sub} x$, if there exist $s, t \in A^*$ such that $x = syt$. The set of all subwords of a word x is denoted by $\underline{sub}(x)$, i.e., $\underline{sub}(x) = \{y \mid y \underline{sub} x\}$ and the set of all subwords of words in a language L by $\underline{sub}(L)$. Clearly, $\underline{sub}(L) = \cup_{x \in L} \underline{sub}(x)$.

The definitions for forbidding and enforcing systems that follow are from [8, 40] with the exception of extended forbidding sets.

Definition 2.1.1 A *forbidding set* \mathcal{F} is a (possibly infinite) family of finite nonempty subsets of A^+ ; each element of a forbidding set is called a *forbidder*.

A forbidding set $\hat{\mathcal{F}}$ is called *extended* if its forbidders are not necessarily finite.

A language L is said to be *consistent with a forbidder* F , denoted by $L \underline{con} F$, if $F \not\subseteq \underline{sub}(L)$. A language L is *consistent with a forbidding set* \mathcal{F} denoted by $L \underline{con} \mathcal{F}$, if $L \underline{con} F$ for all $F \in \mathcal{F}$. If L is not consistent with \mathcal{F} , the notation is $L \underline{ncon} \mathcal{F}$. In other words, $L \underline{ncon} \mathcal{F}$ if there is an $F \in \mathcal{F}$ that *forbids* L , i.e., $F \subseteq \underline{sub}(L)$.

For a forbidding set \mathcal{F} , the family of \mathcal{F} -consistent languages (the \mathcal{F} -family) is $\mathcal{L}(\mathcal{F}) = \{L \mid L \underline{con} \mathcal{F}\}$.

The family $\mathcal{L}(\mathcal{F})$ is said to be *defined* by the forbidding set \mathcal{F} . A family of languages \mathcal{L} is a *forbidding family (f-family)*, if there is a forbidding set \mathcal{F} such that $\mathcal{L} = \mathcal{L}(\mathcal{F})$. Two forbidding sets are *equivalent* if they define the same family of languages. The equivalence relation is denoted by \sim . In other words, $\mathcal{F} \sim \mathcal{F}'$ if and only if $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{F}')$.

Example 2.1.2 Let $\mathcal{F} = \{\{aa, bb\}, \{ab\}\}$. Then, the \mathcal{F} -family consists of all languages that are subsets of either the language $ba^* \cup a^*$ or the language $b^*a \cup b^*$.

Note that the empty language \emptyset and $\{\lambda\}$ are in $\mathcal{L}(\mathcal{F})$ for every \mathcal{F} (see [40]). The next remark simply says that if nothing is forbidden, then everything is allowed and vice versa.

Remark 2.1.3 $\mathcal{L}(\mathcal{F}) = \mathcal{P}(A^*)$ if and only if \mathcal{F} is empty.

Proof. If the forbidding set is empty, then for all $F \in \mathcal{F}$, $F \not\subseteq \underline{sub}(L)$ for all $L \in \mathcal{P}(A^*)$ trivially. Conversely, suppose that $\mathcal{L}(\mathcal{F}) = \mathcal{P}(A^*)$. If \mathcal{F} is not empty, then there is a forbidders F and any language L for which $F \subseteq \underline{sub}(L)$ is not consistent with \mathcal{F} . In particular, $F \underline{ncon} \mathcal{F}$. ■

Definition 2.1.4 An *enforcing set* \mathcal{E} is a (possibly infinite) family of ordered pairs (X, Y) , where X and Y are finite languages and $Y \neq \emptyset$. Such a pair (X, Y) is called an *enforcer*.

A language L *satisfies an enforcer* (X, Y) , denoted $L \underline{sat} (X, Y)$, if $X \subseteq L$ implies $Y \cap L \neq \emptyset$. A language L *satisfies an enforcing set* \mathcal{E} , denoted $L \underline{sat} \mathcal{E}$, if L satisfies every enforcer in \mathcal{E} . If L does not satisfy \mathcal{E} , the notation is $L \underline{nsat} \mathcal{E}$.

For an enforcing set \mathcal{E} , the family of \mathcal{E} -*satisfying* languages is $\mathcal{L}(\mathcal{E})$. The family of languages *defined* by the enforcing set \mathcal{E} is $\mathcal{L}(\mathcal{E}) = \{L \mid L \underline{sat} \mathcal{E}\}$. A family of languages \mathcal{L} is called an *enforcing family* (*e-family*) if there is an \mathcal{E} such that $\mathcal{L} = \mathcal{L}(\mathcal{E})$. Two enforcing sets are equivalent if they define the same family of languages, i.e., $\mathcal{E} \sim \mathcal{E}'$ if and only if $\mathcal{L}(\mathcal{E}) = \mathcal{L}(\mathcal{E}')$.

Observe that every $\mathcal{L}(\mathcal{E})$ contains the language A^* .

In both forbidding and enforcing, it is assumed that the languages under consideration contain words over a fixed finite alphabet A . If the alphabet is not specified, then it is assumed that A is the set of all symbols that appear in the words of the set of forbidders and/or in the enforcing set.

The definition for enforcers allows the set X in an enforcer (X, Y) to be empty. In this case, for every language L , $L \underline{sat} (\emptyset, Y)$ implies that $L \cap Y \neq \emptyset$. Such an enforcer is called *brute enforcer* [8, 40]. A language L satisfies the enforcer (X, Y) trivially, if $X \not\subseteq L$.

Example 2.1.5 Let $\mathcal{E} = \{(\emptyset, \{a\}), (\{a\}, \{a^2\}), \dots\}$, where a is some symbol in the alphabet A . This enforcing set requires that every language that satisfies it contains a^* as a subset, i.e., $\mathcal{L}(\mathcal{E}) = \{L \mid a^* \subseteq L\}$.

If the enforcing set is empty, then the premise “for every enforcer in \mathcal{E} ” is false. Hence, every language satisfies the enforcing set. On the other hand, if $\mathcal{L}(\mathcal{E}) = \mathcal{P}(A^*)$ and $\mathcal{E} \neq \emptyset$, $(X, Y) \in \mathcal{E}$ implies that $X \cap Y \neq \emptyset$. An enforcer (X, Y) such that $X \cap Y \neq \emptyset$ is called *trivial* and such enforcer is satisfied by every language. Hence, the following remark.

Remark 2.1.6 $\mathcal{L}(\mathcal{E}) = \mathcal{P}(A^*)$ if and only if \mathcal{E} is empty or \mathcal{E} contains trivial enforcers only.

The enforcing set \mathcal{E} is *trivial* if \mathcal{E} is empty or \mathcal{E} contains trivial enforcers only and non-trivial otherwise. Thus, \mathcal{E} is non-trivial if and only if $\mathcal{L}(\mathcal{E}) \neq \mathcal{P}(A^*)$. In what follows, unless otherwise stated, all enforcers are non-trivial.

Definition 2.1.7 Given an alphabet A , a *forbidding-enforcing system* (*fe-system*) is a pair $\Gamma = (\mathcal{F}, \mathcal{E})$, where \mathcal{F} is a forbidding set over A and \mathcal{E} is an enforcing set over A . The corresponding *forbidding-enforcing family* (*fe-family*) of languages, denoted $\mathcal{L}(\mathcal{F}, \mathcal{E})$, consists of all languages that are both consistent with \mathcal{F} , and satisfy \mathcal{E} . Hence, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{E})$.

Remark 2.1.8 The language set is an *fe-family*, i.e., $\mathcal{P}(A^*) = \mathcal{L}(\mathcal{F}, \mathcal{E})$, if and only if both \mathcal{F} and \mathcal{E} are empty.

Example 2.1.9 Let $(\mathcal{F}, \mathcal{E})$ be a *fe-system*, such that \mathcal{F} is as in Example 2.1.2, and \mathcal{E} is as in Example 2.1.5. Then, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \{L \mid L \subseteq ba^* \cup a^* \text{ and } a^* \subseteq L\}$.

2.2 Minimal Normal Forms

This section presents some immediate properties of *fe-systems*.

From the definitions and observations in the previous section it follows that there is no forbidding set \mathcal{F} , for which $\mathcal{L}(\mathcal{F})$ is empty. Also, there is no enforcing set \mathcal{E} , such that $\mathcal{L}(\mathcal{E}) = \emptyset$. The next remark is used in the proof of Theorem 3.4.9.

Remark 2.2.1 There exist *fe-systems* with empty *fe-families*.

The following examples consider fe -systems that define empty fe -families. The first example considers a fe -system with an infinite forbidding set and a finite enforcing set and the second example - a fe -system with a finite forbidding set and an infinite enforcing set.

Example 2.2.2 Consider $\mathcal{F} = \{\{w\} \mid w \in A^+\}$ and $\mathcal{E} = \{(\emptyset, \{\lambda\}), (\{\lambda\}, \{w\})\}$, where $w \in A^+$. Then, $\mathcal{L}(\mathcal{F}) = \{\emptyset, \{\lambda\}\}$, but the languages \emptyset and $\{\lambda\}$ are not in $\mathcal{L}(\mathcal{E})$. Hence, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \emptyset$.

Example 2.2.3 Consider $\mathcal{F} = \{\{a\}\}$ and $\mathcal{E} = \{(\emptyset, \{a\}), (\{a\}, \{a^2\}), \dots\}$, where $a \in A$. Then, $\mathcal{L}(\mathcal{F})$ contains only languages that do not contain a in their subwords, whereas $\mathcal{L}(\mathcal{E})$ contains only languages that contain a^* . Hence, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \emptyset$.

The following proposition characterizes the languages L , for which there exists a nontrivial forbidding set \mathcal{F} such that $L \in \mathcal{L}(\mathcal{F})$.

Proposition 2.2.4 *A language L is an element of $\mathcal{L}(\mathcal{F})$ for some $\mathcal{F} \neq \emptyset$ if and only if $\underline{sub}(L) \neq A^*$.*

Proof. If $\underline{sub}(L) = A^*$ then no word can be forbidden, i.e., $L \in \mathcal{L}(\mathcal{F})$ if and only if $\mathcal{F} = \emptyset$. Conversely, if $\underline{sub}(L) \neq A^*$ there is a $w \in A^*$ such that $w \notin \underline{sub}(L)$ and $\mathcal{F} = \{\{w\}\}$ is such that $L \in \mathcal{L}(\mathcal{F})$. ■

Note that for every language $L \neq \emptyset$ there is a nontrivial \mathcal{E} such that $L \in \mathcal{L}(\mathcal{E})$. For example, let $\mathcal{E} = \{(\emptyset, \{w\}) \mid w \in L\}$.

A forbidding set \mathcal{F} is said to be in *minimal normal form* if \mathcal{F} is subword free, i.e., a word in a forbidders cannot be a subword of another word in the same forbidders, and subword incomparable, i.e., for any two forbidders F_1 and F_2 , $\underline{sub}(F_1) \not\subseteq \underline{sub}(F_2)$ and $\underline{sub}(F_2) \not\subseteq \underline{sub}(F_1)$. The following theorem is from [9].

Theorem 2.2.5 *For every forbidding set \mathcal{F} there exists a unique equivalent forbidding set \mathcal{F}' in minimal normal form.*

Let \mathcal{E} be an enforcing set. Define $\mathcal{E}^{(1)} = \{X \mid (X, Y) \in \mathcal{E}\}$.

An enforcing set \mathcal{E} is said to be *finitary*, if for each $X \in \mathcal{E}^{(1)}$, there is a finite number of enforcers (X, Y_i) in \mathcal{E} . The following theorem is from [8].

Theorem 2.2.6 *For every enforcing set there exists an equivalent finitary enforcing set.*

The author of [40] observes that if $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$, $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$, and $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}', \mathcal{E}')$. Also, for any two $\mathcal{F}, \mathcal{F}'$, and any two $\mathcal{E}, \mathcal{E}'$, $\mathcal{L}(\mathcal{F} \cup \mathcal{F}', \mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{F}, \mathcal{E}) \cap \mathcal{L}(\mathcal{F}', \mathcal{E}')$ holds. The following remark is a direct corollary of these properties.

Remark 2.2.7 For any $\mathcal{F}, \mathcal{E}' \subseteq \mathcal{E}$ implies $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E}')$. Similarly, if $\mathcal{F}' \subseteq \mathcal{F}$ then $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}', \mathcal{E})$ for any \mathcal{E} .

Papers [8, 9, 40] discuss normal forms of forbidding sets and normal forms of enforcing sets. It turns out that even though a forbidding set may be given in minimal normal form and an enforcing set may be given in a finitary normal form, the *fe*-system as a whole may still be redundant. The remainder of this section presents such observations.

Proposition 2.2.8 *Let $(\mathcal{F}, \mathcal{E})$ be given. If there is $F \in \mathcal{F}$ and $(X, Y) \in \mathcal{E}$ such that $F \subseteq X$, then $(\mathcal{F}, \mathcal{E}) \sim (\mathcal{F}, \mathcal{E}')$ where $\mathcal{E}' = \mathcal{E} \setminus \{(X, Y)\}$.*

Proof. Let $(\mathcal{F}, \mathcal{E})$ be given. Let $F \in \mathcal{F}$ and $(X, Y) \in \mathcal{E}$ be such that $F \subseteq X$. Since $\mathcal{E}' \subseteq \mathcal{E}$, from Remark 2.2.7 it follows that $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E}')$. Let $L \in \mathcal{L}(\mathcal{F}, \mathcal{E}')$. Since $L \text{ con } \mathcal{F}$, it follows that $F \not\subseteq \text{sub}(L)$. Hence, $F \not\subseteq L$, which implies that $X \not\subseteq L$. Therefore, $\mathcal{L}(\mathcal{F}, \mathcal{E}') \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E})$. ■

The corollary below points out that certain enforcing sets may be removed from the *fe*-system without changing the *fe*-family.

Corollary 2.2.9 *Let $(\mathcal{F}, \mathcal{E})$ be given. If for every $(X, Y) \in \mathcal{E}$ there is a $F \in \mathcal{F}$ with $F \subseteq X$, then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{E})$, i.e., $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F})$.*

Proof. Let $L \in \mathcal{L}(\mathcal{F})$ and $(X, Y) \in \mathcal{E}$. Suppose $X \subseteq L$. Then, there is an $F \in \mathcal{F}$, such that $F \subseteq X \subseteq L$. Hence, $F \subseteq \underline{sub}(L)$ which contradicts the fact that $L \underline{con} F$. Therefore, $X \not\subseteq L$, i.e., $L \underline{sat}(X, Y)$. Consequently, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{E})$ and $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F})$. ■

Definition 2.2.10 Let L be any language and K be a finite language. Define $L_K = \{w \in L \mid K \subseteq \underline{sub}(w)\}$.

Proposition 2.2.11 Let $(\mathcal{F}, \mathcal{E})$ be given and let \mathcal{E}' be obtained from \mathcal{E} as follows. For every $(X, Y) \in \mathcal{E}$ the enforcer $(X, Y') \in \mathcal{E}'$, where $Y' = Y \setminus (\cup_{F \in \mathcal{F}} Y_F)$. Then, $(\mathcal{F}, \mathcal{E}) \sim (\mathcal{F}, \mathcal{E}')$.

Proof. Let $(\mathcal{F}, \mathcal{E})$ be given and let \mathcal{E}' be defined as in the conditions of the proposition. It is clear that $\mathcal{L}(\mathcal{F}, \mathcal{E}') \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E})$. Assume $L \in \mathcal{L}(\mathcal{F}, \mathcal{E})$ and let $(X, Y') \in \mathcal{E}'$. Then, there is an $(X, Y) \in \mathcal{E}$ such that $Y' = Y \setminus (\cup_{F \in \mathcal{F}} Y_F)$ and $L \underline{sat}(X, Y)$. Then, either $X \not\subseteq L$, in which case $L \underline{sat} \mathcal{E}'$ and $L \in \mathcal{L}(\mathcal{F}, \mathcal{E}')$, or $X \subseteq L$ and there is a $y \in Y$, such that $y \in L$. Since $F \subseteq \underline{sub}(y)$ implies $F \subseteq \underline{sub}(L)$, it follows that $y \notin Y_F$ for every $F \in \mathcal{F}$. Hence, $y \in Y'$ and $L \underline{sat} \mathcal{E}'$. Consequently, $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E}')$. ■

Proposition 2.2.12 Let $(\mathcal{F}, \mathcal{E})$ be given. Let $\mathcal{E}' \subseteq \mathcal{E}$ such that $(X, Y) \in \mathcal{E}'$ if and only if for every $F \in \mathcal{F}$, $X_F = \emptyset$. Then, $(\mathcal{F}, \mathcal{E}) \sim (\mathcal{F}, \mathcal{E}')$.

Proof. Obviously, $\mathcal{L}(\mathcal{F}, \mathcal{E}') \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E})$. Let $L \in \mathcal{L}(\mathcal{F}, \mathcal{E})$ and let $(X, Y) \in \mathcal{E} \setminus \mathcal{E}'$. Then, there is an $F \in \mathcal{F}$ such that $X_F \neq \emptyset$. Since $F \not\subseteq \underline{sub}(L)$, it follows that $X_F \cap L = \emptyset$. Hence, $X \not\subseteq L$. Consequently, $\mathcal{L}(\mathcal{F}, \mathcal{E}') \subseteq \mathcal{L}(\mathcal{F}, \mathcal{E})$. ■

2.3 Maximal Languages

In this section, maximal languages for f -families with respect to inclusion are defined and it is shown that they are essential for characterizing f -families.

Definition 2.3.1 A language L in a family of languages \mathcal{L} is called *maximal* for \mathcal{L} if for every language $L' \in \mathcal{L}$, $L \subseteq L'$ implies that $L = L'$.

The set $\mathcal{P}(A^*)$ with inclusion of languages forms a partially ordered set denoted by $(\mathcal{P}(A^*), \subseteq)$. Every chain \mathcal{C} in $\mathcal{P}(A^*)$ contains an upper bound, namely A^* , and by Zorn's lemma $\mathcal{P}(A^*)$ has a maximal element. For a family of languages \mathcal{L} , the set of its maximal languages is denoted by $\mathcal{M}(\mathcal{L})$. The set of maximal languages of an f -family $\mathcal{L}(\mathcal{F})$ is denoted by $\mathcal{M}(\mathcal{F})$. Note that every chain in $\mathcal{L}(\mathcal{F})$, e.g., $L_1 \subseteq L_2 \subseteq \dots$ is bounded by $\cup_{i \geq 1} L_i$ (see [8]), which implies that for every $L \in \mathcal{F}$ there is a $L_{max} \in \mathcal{L}(\mathcal{F})$ such that $L \subseteq L_{max}$.

The example below has been considered in [8, 9, 40].

Example 2.3.2 Let $A = \{a, b\}$. Consider the forbidding set $\mathcal{F} = \{\{aa, bb\}, \{ab, ba\}\}$. There are four maximal languages in $\mathcal{M}(\mathcal{F})$: $L_1 = a^*b \cup a^*$, $L_2 = ba^* \cup a^*$, $L_3 = b^*a \cup b^*$, and $L_4 = ab^* \cup b^*$.

Remark 2.3.3 Let \mathcal{F} be a forbidding set and $L \in \mathcal{L}(\mathcal{F})$. Then, $w \in L$ implies that $\underline{sub}(w) \underline{con} \mathcal{F}$. Moreover, $\underline{sub}(L) \underline{con} \mathcal{F}$ (see [40]), i.e., $\underline{sub}(L) \in \mathcal{L}(\mathcal{F})$.

A language is called *factorial (closed by its factors)* if it contains all of its factors (subwords) (see [36]). In general, $L \subseteq \underline{sub}(L)$, but for a factorial language L it holds that $\underline{sub}(L) = L$. These observations prove the following couple of lemmas.

Lemma 2.3.4 Let \mathcal{F} be a forbidding set and let L be a maximal language in $\mathcal{L}(\mathcal{F})$. Then:

- (i) L is factorial.
- (ii) every L' such that $L' \subseteq L$ is in $\mathcal{L}(\mathcal{F})$.

Proof. Let \mathcal{F} be a forbidding set and let L be a maximal language in $\mathcal{L}(\mathcal{F})$. (i) It is clear that $L \subseteq \underline{sub}(L)$. Since $L \in \mathcal{L}(\mathcal{F})$, $F \not\subseteq \underline{sub}(L)$ for every $F \in \mathcal{F}$. This implies that $\underline{sub}(L) \in \mathcal{L}(\mathcal{F})$ (see Remark 2.3.3). Since L is maximal, it follows that

$L = \underline{sub}(L)$. Hence, L is factorial. (ii) Follows from the fact that $F \not\subseteq \underline{sub}(L)$ implies that $F \not\subseteq \underline{sub}(L')$ for every L' with $L' \subseteq L$ (see [8, 9, 40]).

■

Remark 2.3.5 Note that the above lemma also holds for extended forbidding families.

Lemma 2.3.6 *Given two (extended) forbidding sets \mathcal{F} and \mathcal{F}' the following holds:*

- (i) *If $\mathcal{M}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$, then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$.*
- (ii) *$\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{F}')$ if and only if $\mathcal{M}(\mathcal{F}) = \mathcal{M}(\mathcal{F}')$.*

Proof. (i) Assume that $\mathcal{M}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Let $L \in \mathcal{L}(\mathcal{F})$. Then, there is a maximal language $L_{max} \in \mathcal{L}(\mathcal{F})$, i.e., $L_{max} \in \mathcal{M}(\mathcal{F})$, such that $L \subseteq L_{max}$. Since $L_{max} \in \mathcal{L}(\mathcal{F}')$, from Lemma 2.3.4 it follows that $L \in \mathcal{L}(\mathcal{F}')$. Hence, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. (ii) Assume $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{F}')$. Then, if $L \in \mathcal{M}(\mathcal{F})$ it follows that $L \in \mathcal{L}(\mathcal{F})$, which implies that $L \in \mathcal{L}(\mathcal{F}')$. Hence, there is a $L_{max} \in \mathcal{M}(\mathcal{F}')$ such that $L \subseteq L_{max}$. Since L_{max} is also in $\mathcal{L}(\mathcal{F})$ and $L \in \mathcal{M}(\mathcal{F})$, it follows that $L = L_{max}$. Hence, $\mathcal{M}(\mathcal{F}) \subseteq \mathcal{M}(\mathcal{F}')$. Similarly, $\mathcal{M}(\mathcal{F}') \subseteq \mathcal{M}(\mathcal{F})$; therefore, $\mathcal{M}(\mathcal{F}) = \mathcal{M}(\mathcal{F}')$. Conversely, assume that $\mathcal{M}(\mathcal{F}) = \mathcal{M}(\mathcal{F}')$ and let $L \in \mathcal{L}(\mathcal{F})$. Then, there is a $L_{max} \in \mathcal{M}(\mathcal{F})$ with $L \subseteq L_{max}$. Since $L_{max} \in \mathcal{M}(\mathcal{F}')$ (respectively $L_{max} \in \mathcal{L}(\mathcal{F}')$), from Lemma 2.3.4 it follows that $L \in \mathcal{L}(\mathcal{F}')$. Hence, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Similarly, $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F})$; therefore, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{F}')$.

■

Lemma 2.3.7 *Let \mathcal{F} be given in minimal normal form and let $F \in \mathcal{F}$. Then for each $w \in F$ there exists $L_w \in \mathcal{M}(\mathcal{F})$ such that $w \notin \underline{sub}(L_w)$ and $(F \setminus \{w\}) \subset \underline{sub}(L_w)$.*

Proof. Let \mathcal{F} be given in minimal normal form and let F be a forbider in \mathcal{F} . If $F = \{w\}$, then the lemma holds. Assume, $|F| \geq 2$ and let $w \in F$. Let $\mathcal{F}' = (\mathcal{F} \setminus \{F\}) \cup \{F'\}$, where $F' = F \setminus \{w\}$. If for all $L \in \mathcal{M}(\mathcal{F})$ the word $w \in \underline{sub}(L)$,

then $L \underline{con} F$ implies $L \underline{con} F'$. So, $\mathcal{M}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$ and by Lemma 2.3.6 (i), $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Hence, $\mathcal{F} \sim \mathcal{F}'$, which contradicts the minimality of \mathcal{F} . Therefore, there exists a language $L \in \mathcal{M}(\mathcal{F})$, such that $w \notin \underline{sub}(L)$. Let \mathcal{L} be the set of all such L . Suppose that for each $L \in \mathcal{L}$, $(F \setminus \{w\}) \not\subseteq \underline{sub}(L)$. Then, $\mathcal{L} \subseteq \mathcal{L}(\mathcal{F}')$. Let $K \in \mathcal{M}(\mathcal{F}) \setminus \mathcal{L}$. Since $K \underline{con} F$ there is a $v \in F$ such that $v \notin \underline{sub}(K)$ and since $v \neq w$ it holds that $K \underline{con} F'$. Therefore, $\mathcal{M}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$ and again by Lemma 2.3.6 (i) $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. This implies that $\mathcal{F} \sim \mathcal{F}'$ contradicting the minimality of \mathcal{F} . Hence, there is $L \in \mathcal{L}$ such that $(F \setminus \{w\}) \subseteq \underline{sub}(L)$. ■

The following example shows that there is a \mathcal{F} , with $F, H \in \mathcal{F}$ and $w \in F$ for which $L_w \neq L_v$ for every $v \in H$.

Example 2.3.8 Let $\mathcal{F} = \{\{aa, bb\}, \{ab, baa, aaa\}\}$. Obviously, \mathcal{F} is in minimal normal form. Let L be the language consisting of all words u that do not have two consecutive a 's. Then, $aa \notin \underline{sub}(L)$, which implies that $baa, aaa \notin \underline{sub}(L)$. Hence, $L \in \mathcal{L}(\mathcal{F})$. The language L is maximal, since if any other word is added to L , L will no longer be consistent with the first forbiddler. However, both baa and aaa are not in $\underline{sub}(L)$.

In the above example, there is a word in one forbiddler (aa) which is in the subwords of words from another forbiddler (baa and aaa). The following lemma generalizes this example.

Lemma 2.3.9 *Let \mathcal{F} be given in minimal normal form. If there is $L \in \mathcal{M}(\mathcal{F})$ such that there is $F \in \mathcal{F}$ where at least two words $w, v \in F$ ($w \neq v$) are not in $\underline{sub}(L)$, then there is $H \neq F$, $H \in \mathcal{F}$ for which there is $u \in H$ such that either $u \in \underline{sub}(w)$ or $u \in \underline{sub}(v)$.*

Proof. Let $L \in \mathcal{M}(\mathcal{F})$ and let $F \in \mathcal{F}$ such that there is $w, v \in F$ with $w \neq v$ and $\{w, v\} \cap \underline{sub}(L) = \emptyset$. It is clear that such \mathcal{F} is not a singleton. Suppose that for every $H \neq F$, $H \in \mathcal{F}$ and for every $u \in H$ it holds that $u \notin \underline{sub}(\{w, v\})$. Consider

$L' = L \cup \{v\}$. Obviously, $L' \text{ con } F$. Let $H \in \mathcal{F}$ and $H \neq F$. If $H \subseteq \underline{\text{sub}}(L')$, it follows that there is a $x \in H$ such that $x \in \underline{\text{sub}}(v)$, which contradicts the assumption that $x \notin \underline{\text{sub}}(v)$. Hence, $H \not\subseteq \underline{\text{sub}}(L')$ and $L' \text{ con } \mathcal{F}$, which contradicts the maximality of L . Thus, there is $H \in \mathcal{F}$ which has a word $u \in H$, such that either $u \in \underline{\text{sub}}(w)$, or $u \in \underline{\text{sub}}(v)$. ■

Definition 2.3.10 For a forbidding set \mathcal{F} denote with $\mathcal{W}(\mathcal{F})$ all words that appear in forbidders in \mathcal{F} , i.e., $\mathcal{W}(\mathcal{F}) = \cup_{F \in \mathcal{F}} F$. The language $\mathcal{W}(\mathcal{F})$ is called *subword free* if for every pair of words $w, v \in \mathcal{W}(\mathcal{F})$ it holds that $w \notin \underline{\text{sub}}(v)$ and $v \notin \underline{\text{sub}}(w)$.

Remark 2.3.11 Note that given \mathcal{F} , if $\mathcal{W}(\mathcal{F})$ is subword free, then \mathcal{F} is subword free. However, the converse does not hold. In addition, a subword free $\mathcal{W}(\mathcal{F})$ implies that \mathcal{F} is subword incomparable. Hence, if $\mathcal{W}(\mathcal{F})$ is subword free then, \mathcal{F} is in minimal normal form.

Lemma 2.3.12 *Let \mathcal{F} be given and $\mathcal{W}(\mathcal{F})$ be subword free. Then for every $L \in \mathcal{M}(\mathcal{F})$, it holds that for each $F \in \mathcal{F}$, there is a $w \in F$ such that $w \notin \underline{\text{sub}}(L)$ and $(F \setminus \{w\}) \subseteq \underline{\text{sub}}(L)$.*

Proof. Let \mathcal{F} be given as in the conditions of the lemma. Let $L \in \mathcal{M}(\mathcal{F})$ and let $F \in \mathcal{F}$. Then, there is a $w \in F$ such that $w \notin \underline{\text{sub}}(L)$. If $F = \{w\}$, then the lemma holds. Suppose there is a $v \neq w, v \in F$ such that $v \notin \underline{\text{sub}}(L)$. By Lemma 2.3.9 there is $H \neq F, H \in \mathcal{F}$, which contains u , such that either $u \in \underline{\text{sub}}(w)$ or $u \in \underline{\text{sub}}(v)$. This contradicts the assumption that $\mathcal{W}(\mathcal{F})$ is subword free. Therefore, $v \in \underline{\text{sub}}(L)$ for every $v \in (F \setminus \{w\})$. ■

It may be the case that $\mathcal{W}(\mathcal{F})$ is not subword free and the subwords of every language from $\mathcal{M}(\mathcal{F})$ contain all but one word from each forbinder. The following example shows that the converse of Lemma 2.3.12 does not hold.

Example 2.3.13 Let $\mathcal{F} = \{\{aa, bb\}, \{ab, baa\}\}$. Then every maximal language L has the property that $\underline{\text{sub}}(L)$ does not contain exactly one word from each forbinder

in its subwords. Thus, $\mathcal{M}(\mathcal{F}) = \{L_1, L_2, L_3\}$ where $L_1 = \{w \mid aa \notin \underline{\text{sub}}(w)\}$, $L_2 = \{w \mid bb \notin \underline{\text{sub}}(w)\} \cap \{w \mid ab \notin \underline{\text{sub}}(w)\}$, and $L_3 = \{w \mid bb \notin \underline{\text{sub}}(w)\} \cap \{w \mid baa \notin \underline{\text{sub}}(w)\}$.

Given A , a finitely branching infinite tree T_{A^*} rooted at λ is associated with A^* . Figure 2.1 depicts the tree T_{A^*} for $A = \{a, b\}$.

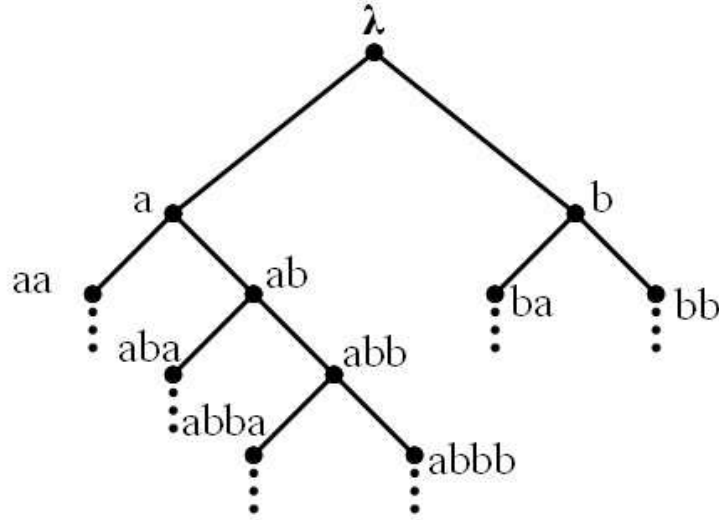


Figure 2.1: Tree associated with the language A^*

Definition 2.3.14 Let A be given. The tree T_{A^*} is the tree rooted at λ and constructed as follows. For every $a \in A$ there is a directed edge (λ, a) in the tree. For every vertex in the tree v and for every $a \in A$, there is a directed edge (v, va) , where v is called the *parent* of va and va is called a *child* of v .

Since every vertex has exactly $|A|$ children, the tree is finitely branching. Because for every vertex v there is a child va for $a \in A$, the tree is infinite. Also, note that for every vertex v there is a unique path from λ to v . In fact, the following correspondence holds.

Remark 2.3.15 Given A , there is a one-to-one correspondence between the words in A^* and the vertices of T_{A^*} .

For example, for $A = \{a, b\}$ the path associated with $w = aaba$ is $\lambda, a, aa, aab, aaba$. Since every vertex in the tree has a unique predecessor, there is a unique path from $aaba$ to λ .

The tree T_{A^*} can be used to obtain the trees for the maximal languages in an f -family.

Definition 2.3.16 Let A be given and $\mathcal{F} = \{F_1, F_2, \dots\}$ be a forbidding set over A given in minimal normal form. The tree T is said to be obtained from T_{A^*} through *branch cutting relative to \mathcal{F}* , if T is obtained through some ordering of \mathcal{F} in the following way. Order the forbidders in \mathcal{F} . Starting with forbidding $F_1 = \{w_1, \dots, w_n\}$ consider n copies of T_{A^*} and for the copy T_i “cut” the incoming edge to w_i removing w_i and all of its decedents from the tree and cut the incoming edge to any word x that has w_i in its subwords and remove subsequent branches. Next, move to $F_2 = \{u_1, \dots, u_m\}$ and for each of the T_i ’s make m copies of them $\{T_{i1}, \dots, T_{im}\}$. From each such tree T_{ik} cut (remove) the branches for u_k and all words that contain it as a subword. Use this procedure to get from F_s to F_{s+1} . Denote with $\mathcal{T}_{\mathcal{F}}$ all trees that can be obtained through branch cutting from T_{A^*} .

For a tree T , denote the language defined by its vertices by L_T . All languages L , such that there is a tree $T \in \mathcal{T}_{\mathcal{F}}$ with $L = L_T$ are denoted by $\mathcal{L}(\mathcal{T}_{\mathcal{F}})$. Observe that $L_T \underline{con} \mathcal{F}$ for every $T \in \mathcal{T}_{\mathcal{F}}$.

Note that if $w \in F$ for some $F \in \mathcal{F}$ is such that it is removed from T along with all words that contain w as a subword, then the rest of $v \in F$ (if any) are not necessarily in L_T . For example, if $w, v \in F$ for some $F \in \mathcal{F}$ with $w \neq v$ and there is a $H \in \mathcal{F}$ with $H \neq F$ such that $v \in H$, then there will be tree in which both w and v will be removed.

Consider Figure 2.2, which illustrates the trees for the four maximal languages in $\mathcal{M}(\mathcal{F})$ from Example 2.3.2 obtained from the branch cutting procedure.

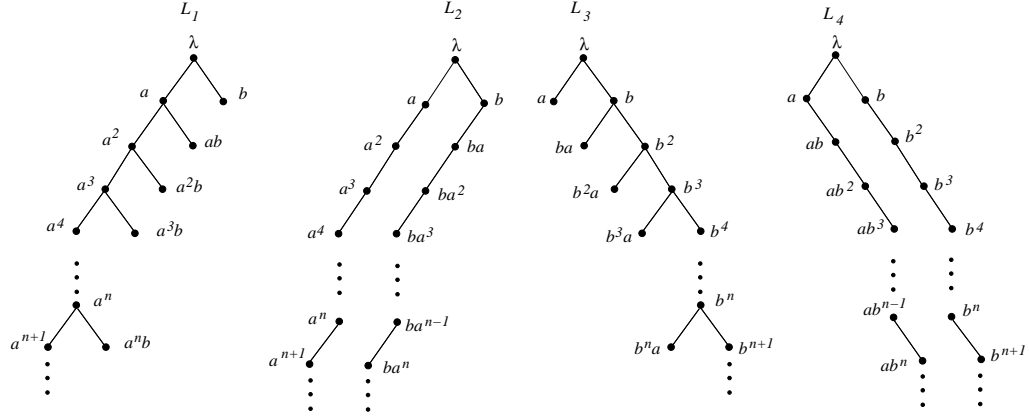


Figure 2.2: Trees associated with the maximal languages from Example 2.3.2

If $\mathcal{W}(\mathcal{F})$ is subword free, then the trees in $\mathcal{T}_{\mathcal{F}}$ represent exactly the maximal languages in $\mathcal{M}(\mathcal{F})$.

Proposition 2.3.17 *Let \mathcal{F} be given and $\mathcal{W}(\mathcal{F})$ be subword free. Then, $\mathcal{M}(\mathcal{F}) = \mathcal{L}(\mathcal{T}_{\mathcal{F}})$.*

Proof. Let $L \in \mathcal{M}(\mathcal{F})$. Then, from Lemma 2.3.12 it follows that for every $F \in \mathcal{F}$ there is a $w \in F$ such that $w \notin \underline{\text{sub}}(L)$ and $(F \setminus \{w\}) \subseteq \underline{\text{sub}}(L)$. Let W be the set of all such w , i.e., W contains exactly one word from each forbidding in \mathcal{F} . Order the forbidings, (respectively the w 's), i.e., $W = \{w_1, w_2, \dots\}$. By employing branch cutting a tree T can be obtained from T_{A^*} by taking the copy that cuts w_1 and subsequent branches and then removing w_2 from it and so on. The resulting $T \in \mathcal{T}_{\mathcal{F}}$ is such that $W \cap \underline{\text{sub}}(L_T) = \emptyset$. Since $\mathcal{W}(\mathcal{F})$ is subword free, no other words from the forbidings are removed from T . Hence, $L \in \mathcal{L}(\mathcal{T}_{\mathcal{F}})$. Conversely, let $L_T \in \mathcal{L}(\mathcal{T}_{\mathcal{F}})$. By construction, $L_T \underline{\text{con}} \mathcal{F}$. Since $\mathcal{W}(\mathcal{F})$ is subword free for every $F \in \mathcal{F}$ there is $w \in F$ such that $w \notin \underline{\text{sub}}(L_T)$, but $(F \setminus \{w\}) \subseteq \underline{\text{sub}}(L_T)$. If $x \notin L_T$, there is $w_i \in \underline{\text{sub}}(x)$ with $w_i \in F_i$ for some forbidding F_i . This means that the language $L' = L_T \cup \{x\}$ is non-consistent with \mathcal{F} . Therefore, L_T is maximal. ■

2.4 Extended Forbidding Sets

The extended forbidding sets are from [16]. Given an f -family, a construction of an extended forbidding set that defines this family is presented. This section contains a characterization of extended f -families.

Let \mathcal{F} be a forbidding set in minimal normal form. With every language $L \in \mathcal{M}(\mathcal{F})$ a tree rooted at λ is associated. A node u at the tree of L has a child ua for $a \in A$ if and only if $ua \in L$. Clearly, the trees might be infinite, but each node has at most cardinality of the alphabet number of children. Denote the tree for L with T_L .

An equivalent (extended) forbidding set to \mathcal{F} is constructed in the following way. An arbitrary symbol from the alphabet set is denoted with a . Let $w = w'a$ be a word such that w' is a node in T_L for some $L \in \mathcal{M}(\mathcal{F})$, but w is not a node in any T_L , $L \in \mathcal{M}(\mathcal{F})$. Then w is forbidden by \mathcal{F} in every language of $\mathcal{L}(\mathcal{F})$, i.e., w is *strictly forbidden* in $\mathcal{L}(\mathcal{F})$. Set $G_w = \{w\}$ and define

$$\mathcal{C} = \{G_w \mid w \text{ is strictly forbidden in } \mathcal{L}(\mathcal{F})\}.$$

Order \mathcal{C} with $G_w \leq G_{w'}$ if $w \in \underline{\text{sub}}(w')$. Let \mathcal{G} be the set of minimal elements of \mathcal{C} with respect to “ \leq ”. Then, include \mathcal{G} in the new extended forbidding set.

Now consider a word v which is a node in T_L , but va is not. In addition, va is a node of some other tree $T_{L'}$. Then, there must be a node u in $T_{L'}$ such that ua' is not a node in $T_{L'}$, but it is a node in T_L . Otherwise, $L \subset L'$ and L would not be maximal. All words like va and ua' are called *non-strictly forbidden* for $\mathcal{L}(\mathcal{F})$. Consider

$$\mathcal{P}' = \{va \mid va \text{ is non-strictly forbidden for } \mathcal{L}(\mathcal{F})\}.$$

Let $\mathcal{P} = \{w \in \mathcal{P}' \mid \underline{\text{sub}}(w) \cap \mathcal{P}' = \{w\}\}$. For $w = va \in \mathcal{P}$ define

$$H_{va,L} = \begin{cases} \{va\} \cup \{ub \mid ub \in L, ub \text{ is non-strictly forbidden for } \mathcal{L}(\mathcal{F})\} & \text{if } va \notin L \\ \emptyset & \text{if } va \in L. \end{cases}$$

Note that if $H_{va,L}$ is not empty then it is not contained in any maximal language in $\mathcal{M}(\mathcal{F})$. Define

$$\mathcal{H}_{va} = \cup_{L \in \mathcal{M}(\mathcal{F})} H_{va,L}.$$

Consider the non-empty subsets of \mathcal{H}_w that are not contained in any maximal language in $\mathcal{M}(\mathcal{F})$. Order these subsets with \subseteq and let \mathcal{Q}_w be the set of minimal subsets of \mathcal{H}_w that contain w and that are not contained in any maximal language in $\mathcal{M}(\mathcal{F})$. Include the \mathcal{Q}_w 's in the new extended forbidding set.

Lemma 2.4.1 *Let \mathcal{F} be given in minimal normal form. Let $\hat{\mathcal{F}} = \mathcal{G} \cup_{w \in \mathcal{P}} \mathcal{Q}_w$. Then, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\hat{\mathcal{F}})$.*

Proof. Let $L \in \mathcal{M}(\mathcal{F})$. Let $F \in \hat{\mathcal{F}}$. By construction of $\hat{\mathcal{F}}$, F is either in \mathcal{G} or in \mathcal{Q}_w for some $w \in \mathcal{P}$. If $F \in \mathcal{G}$, then $F = \{w\}$ for some w and w is strictly forbidden. Hence, $F \not\subseteq \text{sub}(L)$. If $F \in \mathcal{Q}_w$ for some $w \in \mathcal{P}$, then F is a minimal subset of \mathcal{H}_w that contains w and is not contained in any maximal language. Again, $F \not\subseteq L$ and since L is maximal $F \not\subseteq \text{sub}(L)$. So, $L \in \mathcal{L}(\hat{\mathcal{F}})$ which establishes $\mathcal{M}(\mathcal{F}) \subseteq \mathcal{L}(\hat{\mathcal{F}})$. By Lemma 2.3.6, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\hat{\mathcal{F}})$.

For the converse, note that by construction each forbiddler from \mathcal{F} is in $\hat{\mathcal{F}}$. If $F = \{w\}$ then $F \in \mathcal{G}$ since \mathcal{F} is in minimal normal form. If F has more than one word, then by Lemma 2.3.7, there is a maximal language $L_F \in \mathcal{M}(\mathcal{F})$ such that L_F contains all words from F but one. Let $w \in F$ be such that $w \notin L_F$. Since \mathcal{Q}_w contains all minimal sets that contain w and are not in any maximal language, it follows that $F \in \mathcal{Q}_w$. Hence, $\mathcal{L}(\hat{\mathcal{F}}) \subseteq \mathcal{L}(\mathcal{F})$. ■

The extended forbidding set $\hat{\mathcal{F}}$ obtained with the construction above is called *maximal set of forbidders*. The example below uses the forbidding set from Example 2.3.2 to illustrate the above construction.

Example 2.4.2 Let $\mathcal{F} = \{aa, bb\}, \{ab, ba\}$. Following the maximal set construction $\mathcal{C} = \{a^i ba, b^i ab, ab^i a, ba^i b \mid i \geq 1\} \cup \{a^i bb, b^i aa \mid i > 1\}$. The minimal elements are

$\mathcal{G} = \{\{ab^i a\}, \{ba^i b\} \mid i \geq 1\} \cup \{\{aabb\}, \{bbaa\}\}$. Note that ba is a word such that b is a word in L_1 but $ba \notin L_1$. Also, $ba \in L_2$. Thus, $\mathcal{P} = \{aa, bb, ab, ba\}$. The set H_{ba} can be obtained in the following way:

$$\begin{aligned} H_{ba,L_1} &= \{ba, ab, aa, aab, \dots, a^n b, \dots \mid n \geq 1\}, \\ H_{ba,L_2} &= H_{ba,L_3} = \emptyset, \\ H_{ba,L_4} &= \{ba, bb, ab, abb\}. \end{aligned}$$

The minimal subsets of $\mathcal{H}_{ba} = H_{ba,L_1} \cup H_{ba,L_4}$ are $Q_{ba} = \{\{ba, a^n b\} \mid n \geq 1\} \cup \{\{ba, aa, bb\}, \{ba, abb\}\}$. Using similar arguments

$$\begin{aligned} Q_{ab} &= \{\{ab, b^n a\} \mid n \geq 1\} \cup \{\{ab, baa\}, \{ab, aa, bb\}\}, \\ Q_{aa} &= \{\{aa, b^n a\} \mid n > 1\} \cup \{\{aa, bb\}, \{aa, ab, ba\}, \{aa, abb\}\}, \\ Q_{bb} &= \{\{bb, a^n b\} \mid n > 1\} \cup \{\{aa, bb\}, \{ab, ba, bb\}, \{bb, baa\}\}. \end{aligned}$$

In this example the extended forbidding set $\hat{\mathcal{F}}$ is a forbidding set, which is not in minimal normal form. The minimal normal form of $\hat{\mathcal{F}}$ is \mathcal{F} , and $\mathcal{F} \subseteq \hat{\mathcal{F}}$.

The characterization of extended f -families presented below follows from the above construction.

Theorem 2.4.3 *Let \mathcal{L} be a family of languages with the set of maximal languages $\mathcal{M}(\mathcal{L})$. The following are equivalent:*

- (i) *For all $L \in \mathcal{M}(\mathcal{L})$, L is factorial and if $L' \subset L$ then $L' \in \mathcal{L}$.*
- (ii) *\mathcal{L} is an extended f -family.*

Proof. If \mathcal{L} satisfies (i), then the maximal set of forbidders provides an extended forbidding set $\hat{\mathcal{F}}$ such that $\mathcal{L} = \mathcal{L}(\hat{\mathcal{F}})$. By Remark 2.3.5, the maximal languages of an extended f -family satisfy (i). ■

Lemmas 2.3.4 and 2.3.6 show that if \mathcal{L} is an f -family, then Theorem 2.4.3 (i) also holds. However, the next example shows that Theorem 2.4.3 (i) may define an

extended f -family that is not an f -family.

Example 2.4.4 Let $\hat{\mathcal{F}} = \{\{aba, ab^2a, ab^3a, \dots\}\}$ and $\mathcal{L} = \mathcal{L}(\hat{\mathcal{F}})$. Then $\mathcal{M}(\hat{\mathcal{F}}) = \mathcal{M}(\mathcal{L})$, hence satisfies the two properties of Theorem 2.4.3 (i), but \mathcal{L} is not an f -family.

Corollary 2.4.5 *Let \mathcal{L} be a family of languages with the set of maximal languages $\mathcal{M}(\mathcal{L})$. If $\mathcal{M}(\mathcal{L})$ is finite and if for all $L \in \mathcal{M}(\mathcal{L})$, L is factorial and $L' \subset L$ implies $L' \in \mathcal{L}$, then \mathcal{L} is an f -family.*

2.5 Generated Languages

The notion of generated languages appeared in [16]. In order to define step by step derivation of languages defined by an enforcing set, the authors in [8, 9, 40] define \mathcal{E} -extensions. For an enforcing set \mathcal{E} and languages K and L , L is an \mathcal{E} -extension of K (written as $K \vdash_{\mathcal{E}} L$), if for each $(X, Y) \in \mathcal{E}$, $X \subseteq K$ implies $L \cap Y \neq \emptyset$. As defined, it is not necessarily the case that $K \subseteq L$; however, in the process of derivation of a language, this premise is included. The process of computation of fe -systems is represented in [8, 9, 40] by a Γ -tree. A Γ -tree is a rooted, finitely branching tree where the labels of the nodes are finite languages from $\mathcal{L}(\mathcal{F})$ and the label of each child is an \mathcal{E} -extension of the label of the parent that contains it as a subset.

The authors in [8] take the smallest \mathcal{E} -extensions steps in order to generate a language that satisfies a given enforcing set \mathcal{E} . It follows that every language L that contains X from an enforcer $(X, Y) \in \mathcal{E}$ has to contain a minimal set of words defined by the enforcing set \mathcal{E} . In this section the classes of languages defined rather than derived by an fe -system are considered. A new definition, namely of generated and minimal generated languages, is introduced. These notions may be seen as “faster steps” through the Γ -tree defined in [8, 9, 40] and in that sense expand the notion of \mathcal{E} -extensions.

Definition 2.5.1 Let $X \in \mathcal{E}^{(1)}$. A language $g(X)$ generated by X is a language

that satisfies the following two conditions:

- (i) $X \subseteq g(X)$
- (ii) $g(X) \underline{sat} (X', Y')$ for every $(X', Y') \in \mathcal{E}$.

A generated language $g_m(X)$ is called *minimal*, if no proper subset of it is a generated language.

Let \mathcal{E} be an enforcing set and let $X \in \mathcal{E}^{(1)}$. Denote the family of generated languages of X with respect to \mathcal{E} by $G_X^\mathcal{E}$ or simply G_X when \mathcal{E} is understood. The family of minimal generated languages of X with respect to \mathcal{E} is denoted by $M_X^\mathcal{E}$ or simply M_X when \mathcal{E} is understood. The set $\mathcal{M}(\mathcal{E}) = \cup_{X \in \mathcal{E}^{(1)}} M_X$ is called the *minimal generated set* of \mathcal{E} .

Remark 2.5.2 It follows from the definition of minimal generated languages that if \mathcal{E} is an enforcing set and $X \in \mathcal{E}^{(1)}$, then for every language L such that $L \underline{sat} \mathcal{E}$, $X \subseteq L$ implies that L contains as a subset a minimal generated language $g_m(X) \in M_X$.

Note that a generated language always satisfies \mathcal{E} , whereas an \mathcal{E} -extension may not. The following example shows how \mathcal{E} -extensions and minimal generated languages differ.

Example 2.5.3 Let $\mathcal{E} = \{(\{a, aa\}, \{bb, ba\}), (\{ba\}, \{ab\})\}$ and consider the language $X = \{a, aa\}$ in $\mathcal{E}^{(1)}$. Then $\{a, aa, bb\}$ and $\{a, aa, ba, ab\}$ are minimal generated languages for X . The set $\{a, aa, bb, ba\}$ is not a generated language for X , but it is an \mathcal{E} -extension for the language X and it does not satisfy \mathcal{E} . Finally, $\{a, aa, bb, ba, ab\}$ is generated, but it is not minimal.

Lemma 11.14 in [40] shows a redundancy in the enforcing set: if $(X, Y), (X', Y')$ are two different enforcers in \mathcal{E} such that $X \subseteq X'$ and $Y \subseteq Y'$, then $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$. Although not as simple, the following definition extends the notion of redundancy.

Definition 2.5.4 Given \mathcal{E} , the enforcer (X', Y') is *redundant* for \mathcal{E} , if there exists another enforcer $(X, Y) \in \mathcal{E}$ with $X \subseteq X'$ and $Y' \cap g_m(X) \neq \emptyset$ for all $g_m(X) \in M_X^{\mathcal{E}'}$, where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$.

In particular, if $X \subseteq X'$ and $Y \subseteq Y'$, then every $g_m(X) \in M_X^{\mathcal{E}'}$ is such that $g_m(X) \cap Y \neq \emptyset$, hence $g_m(X) \cap Y' \neq \emptyset$, i.e., (X', Y') is redundant.

Example 2.5.5 Let $\mathcal{E} = \{(\{a\}, \{b\}), (\{b\}, \{c, d\}), (\{a, e\}, \{c, d, f\})\}$, and consider $X = \{a\}$. Observe that any language that satisfies the first two enforcers and contains a , has to have either $\{a, b, c\}$ or $\{a, b, d\}$ as subsets, which satisfies the third (redundant) enforcer in both cases.

The following lemma shows that redundant enforcers can be erased from the enforcing set.

Lemma 2.5.6 *If (X', Y') is redundant for \mathcal{E} , then $\mathcal{L}(\mathcal{E}) = \mathcal{L}(\mathcal{E}')$, where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$.*

Proof. It is clear that $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$. Let $L \in \mathcal{L}(\mathcal{E}')$. If $X' \not\subseteq L$, then $L \in \mathcal{L}(\mathcal{E})$. Assume $X' \subseteq L$. Since (X', Y') is redundant, there is an enforcer $(X, Y) \in \mathcal{E}$ such that $X \subseteq X'$ and $g_m(X) \cap Y' \neq \emptyset$ for all $g_m(X)$ in $M_X^{\mathcal{E}'}$. Since L contains at least one $g_m(X)$ from $M_X^{\mathcal{E}'}$, $L \text{ sat}(X', Y')$. ■

Remark 2.5.7 Example 2.5.5 shows a finite enforcing set where for each $X \in \mathcal{E}^{(1)}$, there is a finite number of finite minimal generated languages of X . It is obvious that finite enforcing sets have a finite minimal generated set of finite generated languages, i.e., if \mathcal{E} is finite, then $\mathcal{M}(\mathcal{E})$ is finite and every $g_m \in \mathcal{M}(\mathcal{E})$ is a finite language.

Remark 2.5.8 An infinite finitary enforcing set \mathcal{E} may have an infinite minimal generated set $\mathcal{M}(\mathcal{E})$ in which the minimal generated languages g_m are finite or infinite, or it may have a finite $\mathcal{M}(\mathcal{E})$, which must contain an infinite generated language g_m .

The examples that follow and Lemma 2.5.13 prove this fact. The next two examples show that an infinite finitary enforcing set may have an infinite minimal generated set of infinite minimal generated languages or an infinite minimal generated set of finite minimal generated languages.

Example 2.5.9 Let $\mathcal{E} = \{(\{a^i\}, \{a^{2i}, b^{2i}\}), (\{b^i\}, \{a^{2i+1}, b^{2i+1}\}) \mid i > 0\}$. Then $M_{\{a\}}$ is an infinite family of infinite languages. To see this, we construct a tree rooted at a with two children a^2 and b^2 . The enforcer $(\{a^2\}, \{a^4, b^4\})$ defines two children for a^2 to be a^4 and b^4 and the enforcer $(\{b^2\}, \{a^5, b^5\})$ defines two children for b^2 to be a^5 and b^5 . Continuing this way one can define the children for each new node and the corresponding enforcer. Note that the labels of the nodes in the resulting tree are all distinct, the tree is infinite and the union of the labels of an infinite path that starts at the root is a minimal generated language for the set $X = \{a\}$. Since there is an infinite number of such infinite paths, there is an infinite number of minimal generated languages and each minimal generated language is infinite.

Example 2.5.10 The infinite enforcing set $\mathcal{E} = \{(\{a^i\}, \{b^i\}) \mid i \geq 1\}$ contains an infinite minimal generated set of finite minimal generated languages, i.e., $g_m(\{a^i\}) = \{a^i, b^i\}$ for $i \geq 1$.

The following two examples show that an infinite finitary enforcing set may have a finite minimal generated set, which contains an infinite minimal generated language.

Example 2.5.11 Consider $\mathcal{E} = \{(\emptyset, \{\lambda\}), (\{\lambda\}, \{a\})\} \cup \{(\{a^i\}, \{a^{i+1}\}) \mid \text{for } i \geq 1\}$. There is only one minimal generated language in M_\emptyset which is a^* . The same is true for M_X for any $X \in \mathcal{E}^{(1)}$. Hence, $\mathcal{M}(\mathcal{E}) = \{a^*\}$.

Example 2.5.12 Let $Z = \{w_1, w_2, \dots\}$ be an infinite set of words. Consider the enforcing set $\mathcal{E} = \{(\{w_1, w_2\}, \{w_3\}), (\{w_2, w_3\}, \{w_4\}), (\{w_2, w_4\}, \{w_1\})\} \cup \{(\{w_n, w_{n+1}\}, \{w_{n+2}\}), (\{w_n, w_{n+2}\}, \{w_1\}), (\{w_1, w_n\}, \{w_2\}) \mid n \geq 3\}$. It is obvious that this enforcing set is infinite and finitary. Notice that $\mathcal{M}(\mathcal{E})$ is a singleton

and its only minimal generated language contains all words in Z . In other words, $M_X = \{Z\}$ for all $X \in \mathcal{E}^{(1)}$.

The following lemma shows that an infinite finitary enforcing set with finite $\mathcal{M}(\mathcal{E})$ must have an infinite generated language.

Lemma 2.5.13 *Let \mathcal{E} be infinite and finitary, such that $\mathcal{M}(\mathcal{E})$ is finite. Then there exists an infinite generated language.*

Proof. Since $\mathcal{M}(\mathcal{E})$ is finite, there are a finite number of families of minimal generated languages M_X . Denote these families by M_1, M_2, \dots, M_k , i.e., $\mathcal{M}(\mathcal{E}) = \bigcup_{i=1}^k M_i$. Since there are infinitely many distinct X 's (due to \mathcal{E} being infinite) and finitely many M_i 's, there must exist at least one M_j such that for infinitely many X 's in $\mathcal{E}^{(1)}$, we have $M_X = M_j$. Let $g_m(X) \in M_j$. Since $g_m(X)$ is a minimal generated language for infinitely many X 's, it follows that $g_m(X)$ contains all these X 's as subsets. Hence, $g_m(X)$ is infinite. (In fact, all generated sets in M_j are infinite.)

■

Lemma 2.5.13 is used to show that infinite enforcing sets define non-open families of languages (Proposition 3.4.7).

Chapter 3

Topological Properties of fe -Families of Languages

This chapter provides basic topological properties of the space $\mathcal{P}(A^*)$ using the metric defined in [8] and [40]. It includes a characterization of continuous morphisms and investigates the topological properties of language families as subspaces of $\mathcal{P}(A^*)$. We show that fe -systems define classes of languages different than Chomsky families of languages.

3.1 The Cantor Space $\mathcal{P}(A^*)$

In this section, the space $\mathcal{P}(A^*)$ with the metric defined in [10] and [40] is investigated. This metric comes naturally from the one defined for the ω -words in [11] and [39] and the one used in symbolic dynamics (see [26, 27]). Although the metric is natural, the study of the space of formal languages (the language space) as a topological (metric) space did not appear in literature until [16]. Other topologies on formal languages are considered in [25].

Denote the symmetric difference of L_1 and L_2 by $L_1 \Delta L_2$.

Definition 3.1.1 (Language Metric) The *distance* between any two languages L_1 and L_2 in $\mathcal{P}(A^*)$ is:

$$d(L_1, L_2) = \begin{cases} \frac{1}{2^j} & \text{for } j = \min\{|w| \mid w \in L_1 \Delta L_2\} \quad \text{if } L_1 \neq L_2 \\ 0 & \text{if } L_1 = L_2. \end{cases}$$

For example, let $L_1 = ab^*a$ and $L_2 = a(bb)^*a$. The shortest word in the symmetric difference of L_1 and L_2 is aba ; hence, $d(L_1, L_2) = 2^{-3}$.

It is easy to see that d defined above is a metric. The open ball centered at L with radius δ is the set of all languages that are at a distance less than δ from L . It is denoted by $B_d(L, \delta)$. Clearly, $K \in B_d(L, \delta)$ if and only if $K^{\leq m} = L^{\leq m}$ for any m such that $2^{-m} < \delta$. There is a close relationship between the language metric and the one defined for ω -words in [11], whose definition is recalled below.

Definition 3.1.2 (ω -word Metric) The ω -word distance between any two words ξ and η in A^ω is:

$$\rho(\xi, \eta) = \begin{cases} \frac{1}{2^j}, & \text{for } j = \underline{\min} \{i \mid \xi_i \neq \eta_i\} \text{ if } \xi \neq \eta \\ 0, & \text{if } \xi = \eta \end{cases}$$

As it is well known, A^ω equipped with the metric ρ is homeomorphic to the Cantor space (see for ex. [11, 26, 39]). Open balls in A^ω centered at ξ with radius δ are denoted by $B_\rho(\xi, \delta)$. The homeomorphism defined in Proposition 3.1.4 establishes the connection between $\mathcal{P}(A^*)$ and $\{0, 1\}^\omega$.

Definition 3.1.3 Let K be a language. A *cylinder set* centered at K with bound m is $C(K)_m = \{L \mid L^{\leq m} = K^{\leq m}\}$.

Note that K and L belong to the same cylinder set with bound m if and only if $L \cap A^{\leq m} = K \cap A^{\leq m}$. The collection of cylinder sets corresponds to the open balls for $\mathcal{P}(A^*)$ and hence is a basis for the topology defined by d . Given m , $\mathcal{P} = \{C(K)_m \mid K \in \mathcal{P}(A^*)\}$ forms a finite partition on $\mathcal{P}(A^*)$. For example, if $m = 1$ and $A = \{a, b\}$, then there are eight cylinder sets in the partition \mathcal{P} . Namely, $\mathcal{P} = \{\emptyset, \{\lambda\}, \{a\}, \{\lambda, a\}, \{b\}, \{\lambda, b\}, \{a, b\}, \{\lambda, a, b\}\}$. For example, the cylinder set $C(\emptyset)_1$ consists of all languages whose words have length greater or equal to 2. It is easy to see that every language from $\mathcal{P}(A^*)$ belongs to exactly one of these eight cylinder sets.

The above definition corresponds to the definition for cylinder sets in X^ω defined with $C_i(a_0 \cdots a_k) = \{\xi \mid \xi_i \xi_{i+1} \cdots \xi_k = a_0 a_1 \cdots a_k\}$ (see for ex. [26]).

Assume that the symbols in A are ordered and words in A^* are also ordered lexicographically, i.e., there exists an ordering map $\iota : \mathbb{N} \rightarrow A^*$ such that $w(i) = w_i$. Let $X = \{0, 1\}$.

Proposition 3.1.4 *Let $\phi : \mathcal{P}(A^*) \rightarrow X^\omega$ be a map such that $\phi(L) = \xi$, where $\xi(i) = 1$ if and only if $w_i \in L$. Then ϕ is a homeomorphism.*

Proof. Obviously, ϕ is a bijection. Let $B_d(L, \delta)$ with $\delta > \frac{1}{2^m}$ be an open ball in $\mathcal{P}(A^*)$. For each $w \in L^{\leq m}$ let $i(w)$ be the order of w in A^* , i.e., $i(w) = \iota^{-1}(w)$. Let $\phi(L) = \xi$ and let $j = \max\{i(w) \mid w \in L^{\leq m}\}$. Then $\phi(B_d(L, \delta)) = C_0(\xi_0 \cdots \xi_j)$. Hence both ϕ and ϕ^{-1} are continuous. ■

This proves the following theorem.

Theorem 3.1.5 *The space $\mathcal{P}(A^*)$ is homeomorphic to the Cantor space.*

Proposition 3.1.4 shows that ϕ maps cylinder sets into cylinder sets. As in X^ω , every cylinder set in $\mathcal{P}(A^*)$ is clopen (closed and open); hence, compact.

Present below is a direct proof of the theorem. A topological space is called perfect, if it has no isolated points. Note that no isolated point is a limit point (see [29]).

Theorem 3.1.6 *The space $\mathcal{P}(A^*)$ is*

- (i) *compact*
- (ii) *perfect*
- (iii) *totally disconnected.*

Therefore, it is homeomorphic to the Cantor space.

Proof. Compact. Let $\{L_n\}_{n \geq 0}$ be a sequence of languages in $\mathcal{P}(A^*)$. It is sufficient to show that $\{L_n\}_{n \geq 0}$ contains a convergent subsequence. If $\{L_n\}_{n \geq 0}$ is a finite family

of languages, then there is a language K and an infinite index set $I \subseteq \{0, 1, \dots\}$ such that $K = L_i$ for $i \in I$. In this case $\{L_i\}_{i \in I}$ is a convergent subsequence. Now assume that $\{L_n\}_{n \geq 0}$ is an infinite family of languages. Cantor diagonalization is used next. Consider $\mathcal{P}(A^0) = \{\emptyset, \{\lambda\}\}$. From the infinitely many languages in $\{L_n\}$, each of them either has λ or not, hence there is at least one infinite index set $I_0 \subseteq \{0, 1, \dots\}$ and $X_0 \in \mathcal{P}(A^0)$, for which $X_0 = L_i^{(0)}$ for all $i \in I_0$. All these languages $\{L_i\}_{i \in I_0}$ have the property that either λ is in all of them or λ is in neither of them. Proceed by induction and assume that for $m \geq 0$ there is an infinite index set I_m and a set $X_m \in \mathcal{P}(A^m)$ such that $L_j^{(m)} = X_m$ for all $j \in I_m$. Now consider the finite set $\mathcal{P}(A^{m+1})$. Since I_m is infinite, there is an infinite subset $I_{m+1} \subseteq I_m$ of indexes and at least one set $X_{m+1} \in \mathcal{P}(A^{m+1})$ with $L_j^{(m+1)} = X_{m+1}$ for all $j \in I_{m+1}$. For each m pick K_m from the set $\{L_j\}_{j \in I_m}$ and consider $\{K_m\}_{m \geq 0}$. Then $K = \cup_{m \geq 0} X_m$ is the limit of K_m , since by construction for all $n \geq 0$ it holds that $K_m^{\leq n} = K^{\leq n}$ for all $m \geq n$.

Perfect. It is sufficient to show that every language is a limit point of $\mathcal{P}(A^*)$. Let $L \in \mathcal{P}(A^*)$. If L is finite, then there is m such that $L \subseteq A^m$. Consider the sequence of languages $\{L_n\}_{n \geq 0}$ such that $L_n = L^{\leq n}$ for $n \leq m$ and for $n > m$, $L_n = L \cup \{a^n\}$ for some $a \in A$. Then for all $i > m$ $L_n^{\leq i} = L^{\leq i}$ for all $n > i$, i.e., $L_n \rightarrow L$. If L is infinite, then the sequence of finite languages $L_n = L^{\leq n}$ converges to L (Lemma 3.3.1).

Totally disconnected. In order to show that $\mathcal{P}(A^*)$ is totally disconnected, it suffices to show that the connected components are singletons, i.e., any two languages L_1 and L_2 belong to different connected components. If $L_1 \neq L_2$, then there is a minimal m such that $L_1 \triangle L_2 \cap A^m \neq \emptyset$. Let w be a word in the intersection $L_1 \triangle L_2 \cap A^m$. Consider the sets $\mathcal{B} = \{L \mid L \in \mathcal{P}(A^*) \text{ and } w \in L\}$ and $\mathcal{C} = \{L \mid L \in \mathcal{P}(A^*) \text{ and } w \notin L\}$. Then $d(\mathcal{B}, \mathcal{C}) \geq 2^{-m}$, both are open and their union is the whole space. Hence, they separate L_1 and L_2 . To see that they are open, note that $\mathcal{B} = \cup_{L \in \mathcal{B}} B_d(L, 2^{-(m+1)})$. Similarly for \mathcal{C} . ■

3.2 Continuous Functions

In this section homomorphisms that extend to continuous functions on $\mathcal{P}(A^*)$ are characterized. Note that concatenation of languages $L_1L_2 = \{uv \mid u \in L_1, v \in L_2\}$ makes $\mathcal{P}(A^*)$ a monoid with identity $\{\lambda\}$. A function $\bar{h} : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$ is an extension of a homomorphism $h : A^* \rightarrow B^*$ if $\bar{h}(L) = \{h(w) \mid w \in L\}$. If \bar{h} is an extension of h then \bar{h} is a monoid morphism, i.e., $\bar{h}(L_1L_2) = \bar{h}(L_1)\bar{h}(L_2)$. The homomorphism h is λ -free, if $h(a) \neq \lambda$ for all $a \in A$.

Theorem 3.2.1 *Let $\bar{h} : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$ be an extension of a morphism $h : A^* \rightarrow B^*$. Then \bar{h} is continuous if and only if h is λ -free.*

Proof. Consider a cylinder $C(K)_m$ centered at $K \in \mathcal{P}(B^*)$. Let $L = \{w \mid w \text{ is in a language in } \bar{h}^{-1}(K^{\leq m})\}$. The cylinder set $C(L)_m$ centered at L maps into $C(K)_m$. Let $L_1^{\leq m} = L^{\leq m}$ and $\bar{h}(L_1) = K_1$. Let $u \in K_1$ and $|u| \leq m$. Then there is a word $u' \in L_1$ with $h(u') = u$. Since $L_1^{\leq m} = L^{\leq m}$, we have $u' \in L$ and hence $u = h(u') \in K$, i.e., $K_1^{\leq m} \subseteq K^{\leq m}$. By the symmetry of the argument for L and L_1 we have $K_1^{\leq m} = K^{\leq m}$.

Conversely, assume that there is $a \in A$ with $h(a) = \lambda$. Then either $h(A) = \{\lambda\}$ or there is b such that $h(b) \neq \lambda$. Consider $L_0 \subseteq A^*bA^*bA^*$ where b is such that $h(b) \neq \lambda$. Let $m > |h(b)|$. For every n define $L_n = L_0 \cup \{a^k b \mid k \geq n\}$. Then $L_0^{\leq n} = L_n^{\leq n}$ for all n but $d(\bar{h}(L_0), \bar{h}(L_n)) = 2^{-|h(b)|} > 2^{-m}$. Hence, \bar{h} is not continuous. In the event that $h(A) = \{\lambda\}$, it is obvious that \bar{h} is not continuous since $\bar{h}(\mathcal{P}(A^*)) = \{\emptyset, \{\lambda\}\}$. ■

By Proposition 3.1.4, $\mathcal{P}(A^*)$ and $\{0, 1\}^\omega$ are homeomorphic. Theorem 2.1 in [39] classifies the continuous maps on X^ω . It states that a map $\varphi : X^\omega \rightarrow Y^\omega$ is continuous if and only if it is an extension of a totally unbounded (infinite languages map into infinite languages) and sequential (image of a prefix of a word is a prefix of the image of the same word) mapping $\varphi : X^* \rightarrow Y^*$. Let $X = Y = \{0, 1\}$ and consider two arbitrary alphabets A and B . Both $\mathcal{P}(A^*)$ as well as $\mathcal{P}(B^*)$ are homeomorphic to X^ω . A homomorphism $\bar{h} : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$ extends to a map \hat{h}

such that $\hat{h}F_A = F_B\bar{h}$ where F_A, F_B are the corresponding homeomorphisms defined in Proposition 3.1.4 from $\mathcal{P}(A^*)$ and $\mathcal{P}(B^*)$ to $\{0, 1\}^\omega$ respectively (see Figure 3.1). It is easy to see that any homomorphism $\bar{h} : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$ defines a sequential \hat{h} and the λ -free requirement for h corresponds to totally unbounded \hat{h} . It follows that Theorem 3.2.1 is a corollary of Theorem 2.1 in [39] and Proposition 3.1.4.

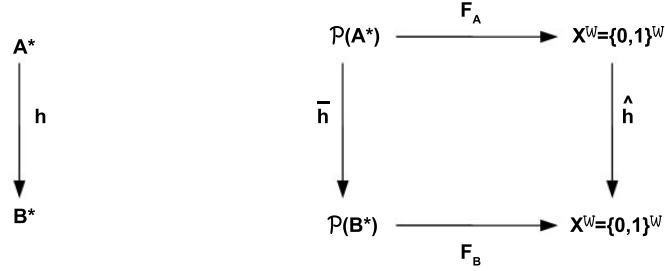


Figure 3.1: Continuous functions

It should be noted, however, that representing fe -families as sets of infinite sequences is difficult and unnatural. For example, describing an f -family $\mathcal{L}(\mathcal{F})$ with $\mathcal{F} = \{\{ab\}\}$ follows our intuition, i.e., every language in this family is a subset of b^*a^* .

An attempt to describe the same family using infinite sequences would be quite burdensome since one has to have at hand the order (index) of all words that contain ab as a subword. Then convert the language $K = \{w \mid ab \in \underline{sub}(w)\}$, to an infinite sequence ξ as in Proposition 3.1.4 and exclude from X^ω all sequences η that contain at least one 1 at a position where ξ also contains a 1.

Infinite sequences will not be discussed further in this work.

Proposition 3.2.2 *If $h : A^* \rightarrow B^*$ is an injective morphism, then h extends to a continuous \bar{h} .*

Proof. Suppose h does not extend to a continuous \bar{h} . Then $h(a) = \lambda$ for some $a \in A$ and $h(\lambda) = \lambda$. Contradiction, since h is one-to-one. ■

The proof of the following facts are straight forward and are omitted. These are examples of continuous functions that are not morphisms on languages.

Proposition 3.2.3 *The following functions $\mathcal{P}(A^*) \rightarrow \mathcal{P}(A^*)$ are continuous.*

- (i) S_L defined with $S_L(K) = KL$ for a fixed L .
- (ii) P_L defined with $P_L(K) = LK$ for a fixed L .
- (iii) U_L defined with $U_L(K) = L \cup K$ for a fixed L .
- (iv) I_L defined with $I_L(K) = L \cap K$ for a fixed L .
- (v) C defined with $C(K) = K^c$ (K^c is the complement of K).
- (vi) H defined with $H(K) = KK$.
- (vii) T defined with $T(K) = K^*$.

3.3 Chomsky Families as Subspaces of $\mathcal{P}(A^*)$

Let FIN, REG, CF, CS, RE denote the families of finite, regular, context free, context sensitive and recursively enumerable languages respectively. This section shows that these families do not correspond to “nice” topological spaces.

A sequence of languages $\{L_n\}_{n \geq 0}$ is *convergent* to a language L , if for each $m \in \mathbb{N}$ there is $M \in \mathbb{N}$, such that $L_i^{\leq m} = L^{\leq m}$ whenever $i > M$. This is denoted by $L_n \rightarrow L$.

The next lemma follows directly from the definitions. (See also [10, 40].)

Lemma 3.3.1 *For each language L , the sequence $\{L_m\}_{m \geq 0} = \{L^{\leq m}\}_{m \geq 0}$ of finite parts of L converges to L , i.e., $L^{\leq m} \rightarrow L$.*

The above lemma shows that every infinite language is a limit of a sequence of finite languages. Hence the following theorem.

Theorem 3.3.2 (i) *Every family of languages $\mathcal{L} \neq \mathcal{P}(A^*)$ that contains FIN is not closed.*

(ii) *None of the classes FIN, REG, CF, CS, RE is topologically closed.*

(iii) *The FIN family is dense in $\mathcal{P}(A^*)$.*

(iv) *Every open ball in $\mathcal{P}(A^*)$ contains a non r.e. language.*

Proof.

(i) Follows from the fact that if K is any language and $K \notin \mathcal{L}$, then $K^{\leq n} \rightarrow K$ and each $K^{\leq n}$ is finite. So, K belongs to the closure of \mathcal{L} , i.e., $\text{closure}(\mathcal{L}) \neq \mathcal{L}$.

(ii) Follows from (i).

(iii) Follows from the fact that every language is a limit of a sequence of finite languages (Lemma 3.3.1).

(iv) Let R be a non r.e. language and define $R^{-j} = R \setminus R^{\leq j}$. For every language $L \in \mathcal{P}(A^*)$ and for every $j > 0$ we have that $d(L, L^{\leq j} \cup R^{-j}) < 2^{-j}$.

■

The above theorem shows that the well known Chomsky families of languages do not have “nice” properties in this topology. In the study of formal languages these families contain languages that are classified by means much different than topological properties and are separated either by the types of automata that recognize them or by the types of grammars. In this sense, the two regular languages a^+ and a^* would be considered very close to each other. But in topological sense, they are at distance 1 from each other (the largest distance possible!). Other topologies on the space of formal languages have also shown to be not suitable for characterizing and classifying the Chomsky hierarchy [25].

3.4 Topological Properties of fe -Families

This section shows that the topology coming from the language metric is a natural framework for investigating fe -systems. The authors in [8, 40] show that fe -systems

define families of languages that are closed sets. This fact along with Theorem 3.3.2 lead to the following.

Theorem 3.4.1 *For all X , $X \in \{ FIN, REG, CF, CS, RE \}$, there is no fe-system Γ such that $\mathcal{L}(\Gamma) = X$.*

By [8, 40], $\mathcal{L}(\mathcal{F})$ is a closed set for every \mathcal{F} . As closed subsets of a compact metric space, $\mathcal{L}(\mathcal{F})$ are compact. The following proposition shows that they are not open.

Proposition 3.4.2 *Let \mathcal{F} be a nontrivial forbidding set. Then $\mathcal{L}(\mathcal{F})$ is not open.*

Proof. Since \mathcal{F} is not empty, there exists a forbidding F in \mathcal{F} such that F is finite, non-empty, and the words in F are distinct from the empty word λ .

Let $L \in \mathcal{L}(\mathcal{F})$ be given and choose $a \in A$. Then for each $s > 0$ consider $L_s = L \cup \{a^s w \mid w \in F\}$. Observe that $L_s \in B_d(L, \frac{1}{2^s})$ but $L_s \notin \mathcal{L}(\mathcal{F})$, since $F \subseteq \text{sub}(L_s)$. Hence, $\mathcal{L}(\mathcal{F})$ is not open. ■

Note that if \mathcal{F} is trivial, then by Remark 2.1.3, $\mathcal{L}(\mathcal{F}) = \mathcal{P}(A^*)$ and is open by definition. Also, the above proof shows a stronger result, which we state in the following corollary.

Corollary 3.4.3 *Let \mathcal{F} be a nontrivial forbidding set. Then every nonempty subset $V \subseteq \mathcal{L}(\mathcal{F})$ is not open.*

Proof. Proceed as in the previous proof, except that $L \in V$. Since $L_s \notin \mathcal{L}(\mathcal{F})$ for each s , then $L_s \notin V$. So, V is not open. ■

Note that if $V = \emptyset$, it is open by the definition.

Authors in [8] and [40] show that $\mathcal{L}(\mathcal{E})$ are closed sets in $\mathcal{P}(A^*)$. Hence these sets are also compact. The following discusses under what conditions they are open.

Proposition 3.4.4 *If \mathcal{E} is finite, then $\mathcal{L}(\mathcal{E})$ is open.*

Proof. If \mathcal{E} is empty, then $\mathcal{L}(\mathcal{E}) = \mathcal{P}(A^*)$, hence, it is open. Assume \mathcal{E} is not empty and let $\mathcal{E} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$. Let $X = X_1 \cup \dots \cup X_n$ and $Y = Y_1 \cup \dots \cup Y_n$ and let $m = \max\{|w| \mid w \in X \cup Y\}$. Let $K \in \mathcal{L}(\mathcal{E})$ and let L be any language such that $K^{\leq m} = L^{\leq m}$. Observe that $L \in \mathcal{L}(\mathcal{E})$. Let $(X_i, Y_i) \in \mathcal{E}$. If $X_i \not\subseteq L$, then $L \text{ sat } (X_i, Y_i)$. If $X_i \subseteq L$, then $X_i \subseteq K$ and since $Y_i \cap L = Y_i \cap K \neq \emptyset$ it holds that $L \text{ sat } (X_i, Y_i)$. Therefore $L \in \mathcal{L}(\mathcal{E})$. This shows that $B_d(K, \frac{1}{2^m}) \subseteq \mathcal{L}(\mathcal{E})$. Hence, $\mathcal{L}(\mathcal{E})$ is open. ■

Proposition 3.4.4 confirms that the boundary conditions of finite enforcing sets are not very restrictive, and by the observation in Theorem 3.3.2 they contain non r.e. languages. In fact, we have the following observation.

Remark 3.4.5 For each cylinder set $C = C(K)_m$ the enforcing set $\mathcal{E}_C = \{(\emptyset, \{w\}) \mid w \in K^{\leq m}\}$ is such that $C \subseteq \mathcal{L}(\mathcal{E}_C)$.

The above proposition shows that a *fe*-system with empty forbidders and finite enforcers is an open set that contains basis elements for the topology on $\mathcal{P}(A^*)$. The infinite enforcing sets have potential to provide families of languages that do not contain non r.e. languages. The following observations show that in the case of infinite enforcing sets, the defined family is always non-open.

Example 3.4.6 If \mathcal{E} is infinite, then $\mathcal{L}(\mathcal{E})$ is not necessarily open. To see this, consider the brute enforcing set $\mathcal{E} = \{(\emptyset, \{w\}) \mid w \in L \text{ and } L \text{ is infinite}\}$. Let $K \in \mathcal{L}(\mathcal{E})$. Since L is infinite, for each m there is $w_m \in L$ such that $|w_m| > m$. Then $L_m = K \setminus \{w_m\}$ is such that $L_m^{\leq m} = K^{\leq m}$, but $L_m \notin \mathcal{L}(\mathcal{E})$.

The enforcing set discussed in the above example is not finitary, but a similar argument can be made for its finitary equivalent enforcing set $\mathcal{E}' = \{(\emptyset, \{w_1\}), (\{w_1\}, \{w_2\}), \dots\}$ and the infinite language $L = \{w_1, w_2, \dots\}$. The following proposition shows that the above example is part of a general rule.

Proposition 3.4.7 *Let \mathcal{E} be infinite and finitary. Then $\mathcal{L}(\mathcal{E})$ is not open.*

Proof. Two cases are considered. In the case that $\mathcal{M}(\mathcal{E})$ is infinite we proceed as follows. Consider an infinite sequence of generated sets. Since $\mathcal{L}(\mathcal{E})$ is compact and every generated set is in $\mathcal{L}(\mathcal{E})$, there is a convergent subsequence, say $\{K_n\}_{n \geq 0}$ of generated sets. Denote its limit by K . Then, since $\mathcal{L}(\mathcal{E})$ is closed, K is in $\mathcal{L}(\mathcal{E})$. To show that $\mathcal{L}(\mathcal{E})$ is not open we observe that every open ball centered at K contains a language that is not in $\mathcal{L}(\mathcal{E})$. Let $m \geq 0$. There is an N such that for all $n \geq N$, $K_n^{\leq m} = K^{\leq m}$. Since there are infinitely many such generated sets K_n , there exists at least one (say $K_l, l \geq N$) which contains a word longer than m . Remove this word from K_l to obtain $\overline{K_l}$. Now $\overline{K_l} \notin \mathcal{L}(\mathcal{E})$ but $d(\overline{K_l}, K) < \frac{1}{2^m}$.

In the case that $\mathcal{M}(\mathcal{E})$ is finite it follows from Lemma 2.5.13 that there exists an infinite minimal generated set, which we denote by K . As in the case of infinite number of generated sets, the fact that every open ball centered at K contains a language that is not in $\mathcal{L}(\mathcal{E})$ is shown. Since K is infinite, for every $m \geq 0$ there is a word $w_m \in K$, such that $|w_m| > m$. Now for every $m \geq 0$ construct the language $L_m = K \setminus \{w_m\}$. Then for every $m \geq 0$, $L_m \in B_d(K, \frac{1}{2^m})$ but $L_m \notin \mathcal{L}(\mathcal{E})$ because L_m is a proper subset of a minimal generated set. ■

Propositions 3.4.4 and 3.4.7 establish a topological difference between finite and infinite enforcing sets. They show that infinite finitary enforcing sets cannot be equivalent to finite enforcing sets, because the first type of sets describes non-open families of languages and the latter - open. This fact is stated in the following corollary.

Corollary 3.4.8 *For every infinite finitary enforcing set there is no finite enforcing set equivalent to it and vice versa.*

The fact that $\mathcal{L}(\mathcal{F}, \mathcal{E})$ is always closed follows from the equality $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{E})$ and the fact that both $\mathcal{L}(\mathcal{F})$ and $\mathcal{L}(\mathcal{E})$ are closed. However, this set may not be open.

Theorem 3.4.9 *A fe-system $\Gamma = (\mathcal{F}, \mathcal{E})$ defines a nonempty open family of languages if and only if $\mathcal{F} = \emptyset$ and \mathcal{E} is finite.*

Proof. If $\mathcal{F} = \emptyset$ and \mathcal{E} is finite, then $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{E})$, which is by Proposition 3.4.4 is open. Conversely, let $\mathcal{L}(\mathcal{F}, \mathcal{E}) = V$ where $V \neq \emptyset$ and is open. Then, since $V \subset \mathcal{L}(\mathcal{F})$ from Proposition 3.4.2 and Corollary 3.4.3, it follows that $\mathcal{F} = \emptyset$. Then, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{E})$ which implies that \mathcal{E} is finite. ■

Note that in the case that $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \emptyset$ the fe -family is open, but as Examples 2.2.2 and 2.2.3 show \mathcal{F} may be nonempty and \mathcal{E} may be infinite.

Although every cylinder set is included in an e -family (Remark 3.4.5), there are cylinder sets (open balls) that define families of languages that cannot be defined by fe -systems.

Example 3.4.10 For the cylinder set $C = C(\{a, ba\})_2$ there is no fe -system Γ such that $\mathcal{L}(\Gamma) = C$. To see this note that C is open and nonempty, hence $\mathcal{F} = \emptyset$ and \mathcal{E} is finite. Then any nonempty combination of words $\{b, aa, ab, bb\}$ must be “excluded” from all languages by means of enforcing only, which is impossible.

The above example extends to the following fact.

Proposition 3.4.11 *Let P be a nonempty finite set of words and let \mathcal{L} be a family of languages such that $L \in \mathcal{L}$ if and only if $P \cap L = \emptyset$. Then for every fe -system Γ , $\mathcal{L}(\Gamma) \neq \mathcal{L}$.*

Proof. Let P and \mathcal{L} be as defined in the proposition. Suppose there exists a fe -system Γ such that $\mathcal{L}(\Gamma) = \mathcal{L}$. Then the language $K = A^* \setminus P$ belongs to $\mathcal{L}(\Gamma)$. Let the maximum length of a word in P be n . Then all words of length greater than n are in K , which implies that $\mathcal{F} = \emptyset$. But then the words from P cannot be excluded by enforcing only. Contradiction, hence no such Γ exists. ■

Proposition 3.4.11 can be proved by topological observation, as well. Let the maximal length of words in P be n . Let $L \in \mathcal{L}$ and consider the cylinder $C(L)_m$ for some $m \geq n$. Then, $C(L)_m \subseteq \mathcal{L}$, hence \mathcal{L} is open. Since $\mathcal{L} \neq \emptyset$, from Theorem

3.4.9 it follows that $\mathcal{F} = \emptyset$ and \mathcal{E} is finite. But then, the words from P cannot be excluded by enforcing only.

Corollary 3.4.12 *Let $C = C(K)_m$ be a cylinder set with $K \neq A^{\leq m}$. Then C is not a fe -family.*

Proof. Let $P = K^c \cap A^{\leq m}$. Then the corollary follows from Proposition 3.4.11. ■

Corollary 3.4.13 *A cylinder set $C = C(K)_m$ is a fe -family if and only if $K = A^{\leq m}$.*

Proof. By letting $\mathcal{F} = \emptyset$ and enforcing $A^{\leq m}$ as in Remark 3.4.5 we obtain a fe -family equal to the cylinder set. The converse follows from Corollary 3.4.12. ■

Chapter 4

Morphisms and fe -Families of Languages

Morphisms $h : A^* \rightarrow B^*$ that extend to $\bar{h} : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$, where A and B are two not necessarily distinct alphabets are considered. The relationship between morphic images of fe -families and fe -families of morphic images of fe -systems is investigated. Conditions under which morphic images of fe -families are fe -families themselves are studied. In the following, h is written instead of \bar{h} when it is clear which function is used from the context.

4.1 Morphic Maps and fe -Families

In general, a morphic image of a fe -family is not necessarily the same as the fe -family of the morphic image of the fe -system. In this section, we investigate under what morphisms these families coincide.

Proposition 4.1.1 *Let \mathcal{F} be a forbidding set over the alphabet A and $h : A^* \rightarrow B^*$ be a surjective λ -free morphism. Then $\mathcal{L}(h(\mathcal{F})) \subseteq h(\mathcal{L}(\mathcal{F}))$.*

Proof. Let $L \in \mathcal{L}(h(\mathcal{F}))$ and let $K = h^{-1}(L)$. Since h is onto, $h(K) = L$. Since h is a morphism, if $w \in \underline{sub}(K)$, then $h(w) \in \underline{sub}(h(K))$. Let $F \in \mathcal{F}$, then $h(F) \in h(\mathcal{F})$. If $F \subseteq \underline{sub}(K)$, then $h(F) \subseteq \underline{sub}(L)$, which contradicts the fact that $L \in \mathcal{L}(h(\mathcal{F}))$. So, $K \in \mathcal{L}(\mathcal{F})$ and hence $L \in h(\mathcal{L}(\mathcal{F}))$. ■

Here the λ -free requirement is essential, since if $h(a) = \lambda$ for some $a \in A$ then the image of the forbidding set $\{\{a\}\}$ is not a forbidding set. Note that an injective morphism is necessarily λ -free which, also, implies that every bijective morphism (isomorphism) is λ -free.

Proposition 4.1.2 *Let \mathcal{F} be given and $h : A^* \rightarrow B^*$ be an injective morphism. Then $h(\mathcal{L}(\mathcal{F})) \subseteq \mathcal{L}(h(\mathcal{F}))$.*

Proof. Let $L \in h(\mathcal{L}(\mathcal{F}))$. Then there is $K \in \mathcal{L}(\mathcal{F})$, such that $h(K) = L$. Let $F' \in h(\mathcal{F})$, then there is $F \in \mathcal{F}$, such that $h(F) = F'$. Since $K \in \mathcal{L}(\mathcal{F})$, $K \underline{con} F$. This means that there is $w \in F$, such that $w \notin \underline{sub}(K)$. Consider $h(w) = w'$. Note that $w' \in F'$. Next, it is shown that $L \underline{con} F'$ by showing that $w' \notin \underline{sub}(L)$. Suppose $w' \in \underline{sub}(L)$. Then there is $x' \in L$, such that $w' \in \underline{sub}(x')$. Also, there is $x \in K$, such that $h(x) = x'$. Since h is one-to-one, only w can map to w' , whence $w \in \underline{sub}(x)$. Contradiction, since $w \notin \underline{sub}(K)$. Therefore, $L \underline{con} F'$, which implies $L \in \mathcal{L}(h(\mathcal{F}))$. ■

The following example shows that that surjectivity is essential in Proposition 4.1.1, injectivity is essential in Proposition 4.1.2, and equality does not necessarily hold in both propositions.

Example 4.1.3

- (a) Let $A = \{a, b, c\}$ and $B = \{d, e\}$ with $h(a) = h(b) = d$ and $h(c) = e$. Let \mathcal{F} contain only one forbider $F = \{ac, bc\}$ and let $K = \{bc\}$. Then $h(K) = h(F)$ and it is not in $h(\mathcal{L}(\mathcal{F}))$, but $h(K) \notin \mathcal{L}(h(\mathcal{F}))$. This example shows that equality does not always hold in Proposition 4.1.1. It, also, shows that injectivity is essential in Proposition 4.1.2.
- (b) To observe that Proposition 4.1.1 does not hold if h is not surjective consider $A = \{a, b\}$ and $B = \{c, d, e\}$ with $h(a) = c$ and $h(b) = d$. Let $\mathcal{F} = \{\{aa\}\}$. Then $L = \{e\} \in \mathcal{L}(h(\mathcal{F}))$, but $L \notin h(\mathcal{L}(\mathcal{F}))$ since $h(\mathcal{L}(\mathcal{F})) \subseteq \{c, d\}^*$. This

example, also, shows that equality does not necessarily hold in Proposition 4.1.2.

Similar properties hold for enforcing families. In this case the requirement that h is a morphism is not necessary.

Proposition 4.1.4 *Let \mathcal{E} be given and $h : A^* \rightarrow B^*$ be a surjective map. Then $\mathcal{L}(h(\mathcal{E})) \subseteq h(\mathcal{L}(\mathcal{E}))$.*

Proof. Let $L \in \mathcal{L}(h(\mathcal{E}))$ and let $K = \{w \mid h(w) \in L\}$. Since h is onto, $h(K) = L$. Suppose $K \not\subseteq \mathcal{E}$. Then there is an enforcer $(X, Y) \in \mathcal{E}$ that is not satisfied by K . The image of this enforcer $(h(X), h(Y))$ is in $h(\mathcal{E})$ and as such is satisfied by L . Since $X \subseteq K$, we have that $h(X) \subseteq L$, and so $h(K) \cap h(Y)$ is not empty. If $w' \in h(K) \cap h(Y)$, then $w' \in h(Y)$ so, there is $w \in Y$, such that $h(w) = w'$. The fact that $w' \in h(K)$ implies that $w \in K$, since K contains all preimages of words in L . Consequently, $w \in K \cap Y$, which means that $K \cap Y \neq \emptyset$. This contradicts the assumption that K does not satisfy (X, Y) . Thus, $K \in \mathcal{L}(\mathcal{E})$, which implies that $L \in h(\mathcal{L}(\mathcal{E}))$. ■

Proposition 4.1.5 *Let \mathcal{E} be an enforcing set and h be an injective map $h : A^* \rightarrow B^*$. Then $h(\mathcal{L}(\mathcal{E})) \subseteq \mathcal{L}(h(\mathcal{E}))$.*

Proof. Let $L \in h(\mathcal{L}(\mathcal{E}))$. Then there is $K \in \mathcal{L}(\mathcal{E})$, such that $h(K) = L$. Let $(X', Y') \in h(\mathcal{E})$ with $h((X, Y)) = (X', Y')$. If $X' \not\subseteq L$, then L satisfies this enforcer trivially. If $X' \subseteq L$, then $X \subseteq K$ since h is injective. Since $K \subseteq \mathcal{E}$, it follows that $Y \cap K \neq \emptyset$. Let $w \in Y \cap K$. Then $w \in Y$ and $h(w) \in Y'$. Also, $w \in K$, which implies that $h(w) \in L$. This means that $L \subseteq h(\mathcal{E})$, i.e., $L \in \mathcal{L}(h(\mathcal{E}))$. ■

Consider again $A = \{a, b, c\}$ and $B = \{d, e\}$ with $h(a) = h(b) = d$ and $h(c) = e$. Let $\mathcal{E} = \{(\{ab\}, \{cc\})\}$. Since the language $K = \{aa\}$ maps into $L = \{dd\}$, we have that $L \in h(\mathcal{L}(\mathcal{E}))$. However, $L \notin \mathcal{L}(h(\mathcal{E}))$. This example shows that equality does

not always hold in Proposition 4.1.4. It, also, shows that injectivity is essential for Proposition 4.1.5. To see that surjectivity is essential for Proposition 4.1.4 consider $A = \{a, b\}$ and $B = \{c, d, e\}$. Once again, let $h(a) = c$ and $h(b) = d$. For $\mathcal{E} = \{(\{a\}, \{b\})\}$ and $L = \{e\}$ we have that $L \in \mathcal{L}(h(\mathcal{E}))$ but $L \notin h(\mathcal{L}(\mathcal{E}))$. This example, also, shows that equality does not always hold in Proposition 4.1.5.

The next corollary follows straight forward.

Corollary 4.1.6 *Let $h : A^* \rightarrow B^*$ be a morphism that extends to a morphism $\bar{h} : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$. Let \mathcal{F} be a forbidding set, \mathcal{E} an enforcing set and $\Gamma = (\mathcal{F}, \mathcal{E})$ an fe -system. The following holds:*

- (i) *If h is bijective then $h(\mathcal{L}(\mathcal{F})) = \mathcal{L}(h(\mathcal{F}))$ and $h(\mathcal{L}(\mathcal{E})) = \mathcal{L}(h(\mathcal{E}))$.*
- (ii) *If h is surjective and λ -free, then $\mathcal{L}(h(\Gamma)) \subseteq h(\mathcal{L}(\Gamma))$.*
- (iii) *If h is injective, then $h(\mathcal{L}(\Gamma)) \subseteq \mathcal{L}(h(\Gamma))$.*
- (iv) *If h is bijective, then $h(\mathcal{L}(\Gamma)) = \mathcal{L}(h(\Gamma))$.*

By the above corollary, a bijective homomorphism, i.e., an isomorphism always maps fe -families into fe -families.

4.2 Characterizing Morphic Images as fe -Families

An investigation of under what conditions fe -families are mapped into fe -families is presented in this section. Unlike what the results from the previous section may suggest, f -families and e -families behave differently under the same morphism. Morphisms that increase the length of a word fail to map f -families into f -families, but may not affect the e -families. Similarly, morphisms that are not surjective fail to map e -families into e -families, but may not affect the f -families.

Proposition 4.2.1 *Let $h : A^* \rightarrow B^*$ where $|A| > 1$ be a morphism, such that there is a symbol $a \in A$ with $|h(a)| > 1$. Then there is a forbidding set \mathcal{F} such that $h(\mathcal{L}(\mathcal{F}))$ is not an f -family.*

Proof. Let $a \in A$ with $|h(a)| > 1$. Let $\mathcal{F} = \{\{c\} \mid c \in A \text{ and } c \neq a\}$. Then $\mathcal{L}(\mathcal{F}) = \{L \mid L \subseteq a^*\}$. Consider $h(\mathcal{L}(\mathcal{F}))$. The language $\{b\} \notin h(\mathcal{L}(\mathcal{F}))$ for every $b \in B$. Suppose that there is a forbidding set \mathcal{F}' such that $h(\mathcal{L}(\mathcal{F})) = \mathcal{L}(\mathcal{F}')$. Then for every $b \in B$, $\{b\}$ must be a forbidden and since the only nonempty subword of b is b itself, $\{b\}$ must be in \mathcal{F}' . This implies that $\mathcal{L}(\mathcal{F}') = \{\emptyset, \{\lambda\}\}$, which contradicts the fact that there are non-trivial languages in $h(\mathcal{L}(\mathcal{F}))$ (for example $h(\{a\}) \in h(\mathcal{L}(\mathcal{F}))$). Hence the proposition follows. ■

Note that even though the image of an f -family under such morphisms might not be an f -family, it can still be an fe -family, as shown in the next example.

Example 4.2.2 Let $A = B = \{a, b\}$ and $h(a) = aa$. Let $\mathcal{F} = \{\{b\}\}$. Then $\mathcal{L}(\mathcal{F}) = \{L \mid L \subseteq a^*\}$. Consider $h(\mathcal{L}(\mathcal{F}))$. It is not an f -family as observed in the proof of Proposition 4.2.1, but it is an fe -family. Consider the fe -system $\Gamma = (\mathcal{F}', \mathcal{E}')$ where $\mathcal{E}' = \{(\{a^{2n+1}\}, \{b\}) \mid n \geq 1\}$ and $\mathcal{F}' = \{\{b\}\}$. Observe that $h(\mathcal{L}(\mathcal{F})) = \mathcal{L}(\Gamma)$.

The above proposition, also, follows from Lemma 2.3.4 since, when $|h(a)| > 1$ for some symbol a , we can find an \mathcal{F} such that the maximal languages in $\mathcal{M}(\mathcal{F})$ map into languages that are not factorial. If there exists a morphism $h : \mathcal{P}(A^*) \rightarrow \mathcal{P}(B^*)$ that maps every f -family into an f -family, then $h(A) \subseteq B$. The following example presents a morphism of this type mapping f -families into f -families.

Example 4.2.3

- (a) Consider $A = \{a, b\}$ and $B = \{c, d\}$ and a morphism h such that $h(a) = h(b) = c$. If $\mathcal{F} = \{\{aa, ab\}, \{ba\}, \{bb\}\}$ then $\mathcal{L}(\mathcal{F})$ consists of all languages that are subsets of $a^* \cup b$. Hence, $h(\mathcal{L}(\mathcal{F}))$ is the family of all languages that are subsets of c^* and $\mathcal{F}' = \{\{d\}\}$.
- (b) For h , A , and B as above, set $\mathcal{F} = \{\{ab, ba\}, \{aa\}, \{bb\}\}$. Then $\mathcal{L}(\mathcal{F})$ consists of languages that don't have words of length larger or equal to 3. Thus, we can set $\mathcal{F}' = \{\{d\}, \{ccc\}\}$.

Proposition 4.2.4 *Let $h : A \rightarrow B$ extend to a morphism $h : A^* \rightarrow B^*$. Then for every \mathcal{F} there exists an extended \mathcal{F}' such that $h(\mathcal{L}(\mathcal{F})) = \mathcal{L}(\mathcal{F}')$.*

Proof. Let \mathcal{F} be a forbidding set. Let L be a maximal language in $h(\mathcal{L}(\mathcal{F}))$. Then, there is a language $L' \in \mathcal{M}(\mathcal{F})$, such that $h(L') = L$. It is sufficient to observe that (i) from Theorem 2.4.3 holds for L . Let $K \subseteq L$. Then every word in K has a preimage in L' . So there is $K' \subseteq L'$ such that $h(K') = K$. Since $K' \in \mathcal{L}(\mathcal{F})$ it holds that $K \in h(\mathcal{L}(\mathcal{F}))$. Observe that, since h maps symbol to symbol, L is factorial. Consider $w \in L$ and $x \in \underline{\text{sub}}(w)$. Then there is $w' \in L'$ and $x' \in \underline{\text{sub}}(w')$ such that $h(w') = w$ and $h(x') = x$. Since L' is factorial, it follows that $x' \in L'$ which implies that $x \in L$. ■

The following result states that surjectivity is essential for mapping e -families into e -families.

Proposition 4.2.5 *Let $h : A^* \rightarrow B^*$ be a non surjective morphism. Then for every enforcing set \mathcal{E} , $h(\mathcal{L}(\mathcal{E}))$ is not an e -family.*

Proof. The proposition follows from the fact that there exists a word $w \in B^*$ such that $h^{-1}(w) = \emptyset$. Suppose there exists \mathcal{E}' such that $h(\mathcal{L}(\mathcal{E})) = \mathcal{L}(\mathcal{E}')$. Let $L \in h(\mathcal{L}(\mathcal{E}))$ and consider a language $K \in \mathcal{L}(\mathcal{E}')$ that contains $L \cup \{w\}$ as a subset. (Such a language always exists. In particular, B^* is one such language.) Then K is in $h(\mathcal{L}(\mathcal{E}))$, as well, which contradicts the fact that $h^{-1}(w) = \emptyset$. Hence, no such \mathcal{E}' exists. ■

Although the image of an e -family under a non surjective morphism is not an e -family, it could be an fe -family, as shown in the following example.

Example 4.2.6 Consider $A = \{a, b\}$, $B = \{a, b, c\}$ and $h : A^* \rightarrow B^*$ with $h(a) = a$ and $h(b) = b$. Then, given \mathcal{E} let $\Gamma = (\{\{c\}\}, \mathcal{E})$. We have that $h(\mathcal{L}(\mathcal{E})) = \mathcal{L}(\Gamma)$.

Thus, a morphism that maps e -families to e -families is necessarily surjective. The following examples illustrate such morphisms and point to the difficulty in constructing enforcers in the image family. This difficulty stems from the fact that the new enforcers do not necessarily have a preimage in the old enforcers, as seen in the following two examples. In both cases the enforcers defining the image family have completely different structure than the ones we started with.

Example 4.2.7

- (a) Let $A = \{a, b\}$ and $B = \{c\}$. Let $h : A^* \rightarrow B^*$ be a morphism such that $h(a) = h(b) = c$. Consider

$$\mathcal{E} = \{(\{aa\}, \{a^3\}), (\{aa\}, \{a^4\}), (\{ab\}, \{a^5\}), (\{ba\}, \{a^5\}), (\{bb\}, \{a^5\})\}.$$

The minimal generated sets for \mathcal{E} are: $g_m(\{aa\}) = \{a^2, a^3, a^4\}$, $g_m(\{ab\}) = \{ab, a^5\}$, $g_m(\{ba\}) = \{ba, a^5\}$, and $g_m(\{bb\}) = \{bb, a^5\}$. They all map into two sets $\{c^2, c^3, c^4\}$ and $\{c^2, c^5\}$. Consider a language $K \in h(\mathcal{L}(\mathcal{E}))$ that contains the word cc . Since $h^{-1}(cc) = \{aa, ab, ba, bb\}$, K must contain an image of a minimal generated set, i.e., K has as a subset either $\{c^2, c^3, c^4\}$ or $\{c^2, c^5\}$. Thus, if there is \mathcal{E}' such that $h(\mathcal{L}(\mathcal{E})) = \mathcal{L}(\mathcal{E}')$, then there must be an enforcer (X, Y) in \mathcal{E}' with $X = \{cc\}$. Note that $\{ba, a^5, b^3\}$ and $\{ba, a^5, b^4\}$ are in $\mathcal{L}(\mathcal{E})$ and they map into $\{c^2, c^3, c^5\}$ and $\{c^2, c^4, c^5\}$. So, there are the following enforcers $\mathcal{E}' = \{(\{c^2\}, \{c^3, c^4, c^5\}), (\{c^2, c^3\}, \{c^4, c^5\}), (\{c^2, c^4\}, \{c^3, c^5\})\}$. In this case $h(\mathcal{L}(\mathcal{E})) = \mathcal{L}(\mathcal{E}')$.

- (b) Consider $A = \{a, a', b, b', c, c'\}$ and $B = \{a, b, c\}$. Let $h : A^* \rightarrow B^*$ such that $h(a) = h(a') = a$, $h(b) = h(b') = b$, and $h(c) = h(c') = c$. Let

$$\mathcal{E} = \{(\{a, b\}, \{abc\}), (\{a', b'\}, \{abc\}), (\{a', c\}, \{abc\}), (\{a', c'\}, \{abc\}), (\{b', c\}, \{abc\}), (\{b', c'\}, \{abc\})\}.$$

Then $h(\mathcal{L}(\mathcal{E})) = \mathcal{L}(\mathcal{E}')$ where $\mathcal{E}' = \{(\{a, b, c\}, \{abc\})\}$.

Recall that an open map is a function that maps open sets into open sets.

Corollary 4.2.8 *Let $h : A^* \rightarrow B^*$ be a morphism that defines an open map on the space of languages. If for every fe -family $\Gamma = (\mathcal{F}, \mathcal{E})$ over alphabet A there is an fe -family $\Gamma' = (\mathcal{F}', \mathcal{E}')$ over B such that $h(\mathcal{L}(\Gamma)) = \mathcal{L}(\Gamma')$ then h is surjective.*

Proof. Assume that every fe -family maps with h onto an fe -family. Then such a family with empty enforcers, or with an empty forbidding set also maps into an fe -family. Let $\Gamma = (\mathcal{F}, \mathcal{E})$ be such that $\mathcal{F} = \emptyset$ and \mathcal{E} be finite. Then $\mathcal{L}(\Gamma)$ is open and $h(\mathcal{L}(\Gamma))$ is open, as well. This means that in $\Gamma' = (\mathcal{F}', \mathcal{E}')$ the forbidding set must be empty and the set of enforcers must be finite. By Proposition 4.2.5 h is surjective. ■

Unfortunately, the converse does not hold even when h maps symbols to symbols surjectively. Consider the following example.

Example 4.2.9 Let $A = \{a, b, \}$, $B = \{c\}$ and $a \mapsto c$, $b \mapsto c$. Let $\mathcal{F} = \{\{aa, bb\}, \{ab\}, \{ba\}\}$ and

$$\begin{aligned} \mathcal{E} &= \{(\{aa\}, \{bb\}), (\{bb\}, \{aa\}), (\{a\}, \{a^3\}), (\{b\}, \{b^3\})\} \\ &\cup \{(\{a^i\}, \{a^{i+1}\}), (\{b^i\}, \{b^{i+1}\}), (\{a^i\}, \{a\}), (\{b^i\}, \{b\}) \mid i \geq 3\}. \end{aligned}$$

Then $\mathcal{L}(\Gamma)$ besides the trivial, contains exactly two languages $\{a, a^3, a^4, \dots\}$ and $\{b, b^3, b^4, \dots\}$. The only non trivial set in $h(\mathcal{L}(\Gamma))$ is $K = \{c, c^3, c^4, \dots\}$. What can Γ' such that $\mathcal{L}(\Gamma') = \{K\}$ be? First note that any forbidding set in normal form must be at most a singleton $\{c^i\}$. But in that case, no power of c larger than i is allowed in a language of $\mathcal{L}(\Gamma')$. So $\mathcal{F}' = \emptyset$. But then, if $K \underline{sat} \mathcal{E}'$, we have that $c^* \underline{sat} \mathcal{E}'$ too. Hence, it is impossible to exclude c^2 using an enforcing set \mathcal{E}' only, i.e., there is no Γ' such that $\mathcal{L}(\Gamma') = h(\mathcal{L}(\Gamma))$.

Chapter 5

Forbidding and Enforcing of Graphs

In this chapter graph *fe*-systems are defined and investigated. They were inspired by language *fe*-systems and self-assembly of shapes. Forbidding and enforcing systems on graphs define new classes of graphs based on two sets of boundary conditions. Forbidding conditions state that certain combinations of graphs cannot be subgraphs of a graph. Enforcing conditions state that whenever certain graphs are subgraphs of a graph, then they are required to be embedded in pre-specified larger subgraphs of that graph. All graphs that obey the forbidding and enforcing constraints specified by a *fe*-system form the *fe*-family of graphs.

5.1 Definitions

The graphs $G = (V, E)$ in this chapter are simple, meaning no loops and no multi-edges are allowed. They are, also, undirected. A graph is *connected* if between every pair of vertices there is a path. In this chapter all graphs are connected, unless indicated otherwise.

An *isomorphism* between two simple graphs G and H is a vertex bijection $\varphi : V_G \rightarrow V_H$ such that for each $u, v \in V_G$, u and v are adjacent in G if and only if $\varphi(u)$ and $\varphi(v)$ are adjacent in H . Implicitly, there is also an edge bijection $E_G \rightarrow E_H$ such that $uv \rightarrow \varphi(u)\varphi(v)$. Two simple graphs G and H are called *isomorphic* if there is an isomorphism from G to H .

A *trivial graph* is a graph consisting of one vertex and no edges and a *null graph*

is a graph whose vertex- and edge-sets are empty. The trivial graph is denoted by Λ and the null graph by \emptyset . In the set of all connected graphs, the only graph with a vertex of degree 0 is the trivial graph. A connected graph which is not isomorphic to Λ or to \emptyset will be called *non-trivial*.

A *subgraph* of a graph G is a graph H whose vertex and edge sets are subsets of these of G . For that matter, any graph isomorphic to H is, also, considered a subgraph of G . This is denoted by $H \leq G$ or $H < G$, depending on whether H could be isomorphic to G or not. In this chapter, both $G \cong H$ and $G = H$ denote that G is isomorphic to H . So, \leq is the graph embedding, i.e., every graph isomorphic to a subgraph of graph G is considered to be a subgraph of G . The set $\underline{sub}(G)$ contains all subgraphs of G up to isomorphism, i.e., $\underline{sub}(G) = \{H \mid H \leq G\}$. Thus, a finite set of graphs F is a subset of $\underline{sub}(G)$ (denoted by $F \subseteq \underline{sub}(G)$), if for every graph from F , there is a subgraph of G isomorphic to it. Similarly, $F \not\subseteq \underline{sub}(G)$, if there is a graph in F for which no subgraph of G is isomorphic to it. Let F be a set of graphs. Define $\underline{sub}(F) = \{H \mid \text{there is } K \in F \text{ such that } H \leq K\}$. Or else, $\underline{sub}(F) = \cup_{K \in F} \underline{sub}(K)$.

A *non-connected* graph X is a subgraph of G denoted by $X \leq G$, if for every connected component C in X there is a $H \leq G$ such that C is isomorphic to H and there is an embedding $\phi : X \rightarrow G$, such that for every two components C_1 and C_2 of X , $V(\phi(C_1)) \cap V(\phi(C_2)) = \text{emptyset}$. (So, if X has two disjoint 3-cycles then G has two disjoint 3-cycles, as well.)

A graph is *finite* if its vertex (resp. edge set) is finite. Otherwise, it is infinite. The set of all finite connected graphs is denoted by \mathcal{U} .

Some commonly used notation is observed: P_n is a path on n vertices, C_n is a cycle of length n , K_n is the complete graph on n vertices, $K_{m,n}$ is a bipartite graph. In this regard, $P_0 = \emptyset$, $P_1 = \Lambda$, and P_2 is just an edge. D_4 denotes a 4-cycle with an extra edge connecting two non-adjacent vertices.

The following notation is used in [18]. $N_{i,j,k}$ is a graph that consists of a K_3 and three vertex-disjoint paths of lengths i , j , and k , with each path rooted at exactly one of the three vertices of K_3 and no two paths rooted at the same vertex. Also,

define $H_{i,j,k}$ to be the tree consisting of three paths with lengths i , j , and k , rooted at v and otherwise vertex-disjoint. For the purpose of $N_{i,j,k}$ and $H_{i,j,k}$ the length of the paths is determined without considering the common vertex, i.e., $K_{1,3} = H_{1,1,1}$ and a 3-cycle with an extra edge is denoted by $N_{1,0,0}$.

Given a finite set of graphs of non trivial graphs $F = \{H_1, \dots, H_n\}$ define the graph $S_{H_1 P_{i_1} H_2 \dots P_{i_{n-1}} H_n}$ such that H_1 is connected through a path P_{i_1} to H_2 and continuing this way H_{n-1} is connected through $P_{i_{n-1}}$ to H_n . Furthermore, $V(H_1) \cap V(P_{i_1}) = \{u'_1\}$, $V(P_{i_1}) \cap V(H_2) = \{u_2\}$, \dots , $V(P_{i_{n-1}}) \cap V(H_n) = \{u_n\}$, where all u 's are distinct and every graph from F and every path is otherwise vertex-disjoint from every other graph or path. (Figure 5.1 depicts the graph $S_{C_3 P_4 C_4}$.)

5.2 Connecting Graphs

This section defines and investigates *connecting graphs* of a finite set of graphs as graphs that contain all graphs from the set as subgraphs. In this sense, connecting graphs do in fact “connect” all graphs from such a set.

Definition 5.2.1 Given a finite set of graphs F , a graph G is a *connecting graph* of F (or G connects F), if $F \subseteq \underline{\text{sub}}(G)$. S is called a *minimal connecting graph* of F if S is a connecting graph and for every connecting graph H of F , $H \leq S$ implies $H = S$. Given F , the family of all connecting graphs of F is called the *connect* of F and is denoted by $C(F)$ and the family of minimal connecting graphs of F is called the *minimal connect* of F and is denoted by $C_{\min}(F)$.

Remark 5.2.2 For every finite set of graphs F there is a graph G such that $F \subseteq \underline{\text{sub}}(G)$. Moreover, for every such G there is a minimal graph S , such that $F \subseteq \underline{\text{sub}}(S)$ and $S \leq G$.

One such G can be obtained by ordering the graphs in $F = \{H_1, \dots, H_n\}$ and connecting H_i with H_{i+1} with an edge that has one vertex in H_i and the other vertex in H_{i+1} . Obviously, G is connected and $F \subseteq \underline{\text{sub}}(G)$. Then, S can be obtained by removing edges from G in such a way that this does not result in a graph that is



Figure 5.1: The graph $S_{C_3 P_n C_4}$ from Example 5.2.3

no longer connected nor that $F \not\subseteq \underline{sub}(G)$. If no more edges can be removed from G a minimal connecting graph S for F has been reached.

Another connecting graph G' of F can be obtained by taking the graph with the maximum number of vertices (say m) among the graphs in F and letting $G = K_m$. Then, by removing edges until the graph is no longer connected or no longer a connecting graph of F a minimal connecting graph $S \leq G'$ is obtained.

A finite set of graphs may have many minimal connecting graphs. Consider the following example.

Example 5.2.3 Let $F = \{C_3, C_4\}$. D_4 is a minimal connecting graph. Now consider the graph consisting of a 3-cycle and a 4-cycle, connected by a path of length n denoted with $S_{C_3 P_n C_4}$ (see Figure 5.1). One end of the path P_n is a vertex in C_3 and the other is a vertex in C_4 . Each of the cycles and the path is otherwise disjoint from the other two. It is clear that any value of n produces a minimal connecting graph. There are, in fact, infinitely many minimal connecting graphs for F .

Another way to define minimality is to require that S be *vertex-minimal*. In this case, D_4 is the unique minimal connecting graph for the F considered in Example 5.2.3.

Definition 5.2.4 Given a finite set of graphs F , S is called a *vertex-minimal connecting graph* of F if S is a connecting graph and for every connecting graph H in $C(F)$ it holds that $|V(S)| \leq |V(H)|$.

The following example shows that even if this definition of minimality is used, the minimal connecting graph is not necessarily unique.

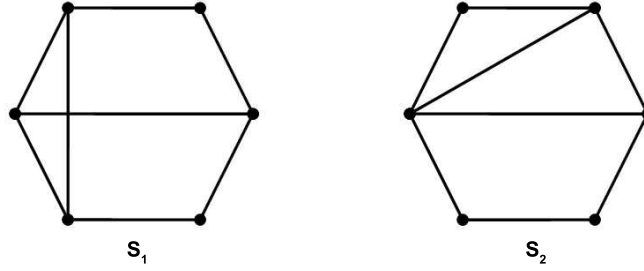


Figure 5.2: Minimal connecting graphs of F from Example 5.2.5

Example 5.2.5 Let $F = \{C_3, C_4, C_5, C_6\}$. Then both S_1 and S_2 depicted in Figure 5.2 are vertex-minimal connecting graphs of F .

For the rest of this chapter “minimal” connecting graphs are as in Definition 5.2.1.

It is obvious that if a graph is a connecting graph of a finite set of graphs, it is also a connecting graph of every subset of this set of graphs. This fact is stated formally in the next proposition.

Proposition 5.2.6 Let F_1 and F_2 be two finite sets of graphs such that $F_1 \subseteq F_2$. Then $C(F_2) \subseteq C(F_1)$.

Example 5.2.7 Consider the set of graphs $F = \{K_{1,3}, C_3, C_4\}$. Notice that $K_{1,3}$ is a subgraph of every connecting graph of $F' = \{C_3, C_4\}$ and so $C_{min}(F) = C_{min}(F')$.

The following proposition generalizes the above example.

Proposition 5.2.8 Let F be a finite set of graphs such that there is a $H \in F$ with $H \leq S$ for every $S \in C(F')$, where $F' = F \setminus \{H\}$. Then $C_{min}(F) = C_{min}(F')$.

Proof. Let $S \in C_{min}(F)$. By Proposition 5.2.6 $S \in C(F')$. Hence, there is a $T \in C_{min}(F')$ such that $T \leq S$. Since $T \in C(F)$, it follows that $T = S$, hence $C_{min}(F) \subseteq C_{min}(F')$. Conversely, let $S \in C_{min}(F')$. Since $H \leq S$, it follows that $S \in C(F)$. Then, there is $T \in C_{min}(F)$ such that $T \leq S$. By Proposition 5.2.6 $T \in C(F')$ and it follows that $T = S$. Hence, $C_{min}(F') \subseteq C_{min}(F)$. ■

As seen in Example 5.2.3 there are infinitely many connecting graphs for $F = \{C_3, C_4\}$, i.e., $C_{min}(F)$ is infinite. The following is an investigation of sets F for which $C_{min}(F)$ is infinite. The next remark is straight forward.

Remark 5.2.9 If $F = \{H\}$, then $C_{min}(F) = \{H\}$.

Definition 5.2.10 A set of graphs F is called *subgraph free* if for every pair of graphs $K, H \in F$, $K \not\leq H$ and $H \not\leq K$. A finite set of graphs F is called *connecting free* if for every graph $H \in F$ there exists $S \in C(F \setminus \{H\})$, such that $S \not\leq H$.

Example 5.2.11 The set F from Example 5.2.7 is subgraph free, but not connecting free.

Proposition 5.2.12 *If a finite set of graphs is connecting free, then it is subgraph free, but the converse does not hold.*

Proof. Let F be a connecting free finite set of graphs. Let $H, K \in F$. Since F is connecting free, it follows that there is a connecting graph $S \in C(F \setminus \{H\})$ such that $H \not\leq S$, which implies that $H \not\leq K$. Similarly, $K \not\leq H$. Hence, F is subgraph free. Example 5.2.11 shows that the converse does not hold. ■

Remark 5.2.13 It is clear that if F is a subgraph free set of graphs and $F' \subseteq F$, then F' is subgraph free.

Proposition 5.2.14 *Let F be a connecting free set of graphs and $F' \subseteq F$. Then F' is connecting free.*

Proof. Let F be connecting free and $F' \subseteq F$. Let $K \in F'$. $K \leq S$ for every $S \in C(F' \setminus \{K\})$ implies $K \leq S$ for every $S \in C(F \setminus \{K\})$. Hence, F' is connecting free. ■

It is easy to see that if F has only two graphs then the notions of connecting free and subgraph free coincide. Proposition 5.2.12 and Remark 5.2.9 prove the following remark.

Remark 5.2.15 Let the set of graphs F is such that $|F| = 2$. Then, F is connecting free if and only if F is subgraph free.

The fact below can easily be proved and it is used in the proposition that follows.

Remark 5.2.16 Let $F = \{P_n, H\}$ where H is a graph. Then F is not subgraph free for any $n \leq 3$.

Clearly, if F is as in the above remark and $n \leq 3$, then $C_{min}(F)$ is a singleton.

Proposition 5.2.17 Let $F = \{P_n, H\}$ where $n \geq 0$. Then, $C_{min}(F)$ is finite.

Proof. Let n and H be given. If F is not subgraph free, then $C_{min}(F)$ is a singleton. Assume F is subgraph free, i.e., $P_n \not\leq H$ and $H \not\leq P_n$. By Remark 5.2.16 $n \geq 4$. Also, H is not a path. The longest path in H can be “extended” to P_n . There are finitely many vertices and paths in H and finitely many ways of extending any path from any vertex to P_n . Hence, $C_{min}(F)$ is finite. ■

Proposition 5.2.18 Let $F = \{H_1, H_2, H_3\}$. Then, either there is a k such that for every $n \geq k$, $H_3 \leq S_{H_1P_nH_2}$ or there is a s such that for every $m \geq s$, $H_3 \not\leq S_{H_1P_mH_2}$.

Proof. Let F, H_1, H_2 , and H_3 are as in the conditions of the proposition and let $S_n = S_{H_1P_nH_2}$ where $n \geq 2$. If there is a k such that for every $n \geq k$ it holds that $H_3 \leq S_n$, the proposition holds. Otherwise, for every k there is a $n \geq k$ such that $H_3 \not\leq S_n$. Consider the smallest such n . Then, if for every $m \geq n$, $H_3 \not\leq S_m$, the proposition holds. Otherwise, there is a m such that $m > n$ and $H_3 \leq S_m$. If for every $l \geq m$ $H_3 \leq S_l$, the proposition holds. Otherwise, there is a $l > m$ such that $H_3 \not\leq S_l$, but in this case for every $p \geq l$ it holds that $H_3 \not\leq S_p$. To see this suppose there is a $p > l$ such that $H_3 \leq S_p$. Then, $H_3 \not\leq H_1, H_2$, since otherwise $H_3 \leq S_l$. Also, $H_3 \not\leq S_{H_1P_p}$ and $H_3 \not\leq S_{P_pH_2}$ for otherwise $H_3 \leq S_l$. (Here $S_{H_1P_p}$ is H_1 with a path P_p that shares a common vertex with H_1 and is otherwise disjoint from H_1 and $S_{P_pH_2}$ is defined analogously.) Hence, H_3 is “partly” in H_1 , “contains” the entire P_p

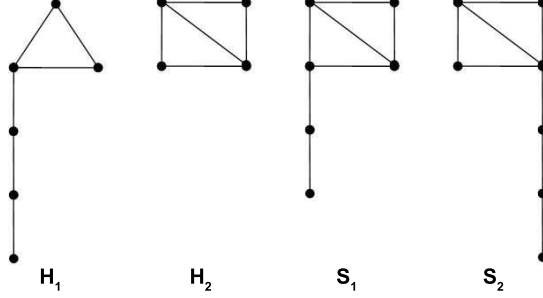


Figure 5.3: Graphs related to Example 5.2.20

and is “partly” in H_2 . But in this case $H_3 \not\leq S_m$, which contradicts the fact that $H_3 \leq S_m$. Hence, for every $p \geq l$ it holds that $H_3 \not\leq S_p$. ■

As an example that illustrates the above proof consider $F = \{C_3, C_4, T\}$, where T is a graph consisting of a P_3 with each end being of degree 3 such that the two neighbors of the end vertex not on the path are of degree 1. Then, $k = n = 2$, $m = 3$, and $l = 4$.

Definition 5.2.19 Let H and K be graphs with $H \not\leq K$ and $K \not\leq H$. The graph H is called a *tailed subgraph* of a graph K denoted by $H \leq_t K$, if there is a path P_n where $n \geq 2$ such that $H \leq S_{KP_n}$, where S_{KP_n} is the graph K with a path P_n rooted at one of the vertices of K and otherwise vertex-disjoint from K . A set of graphs F is called *tailed subgraph free* if for every two graphs $H, K \in F$ it holds that $H \not\leq_t K$ and $K \not\leq_t H$.

Example 5.2.20 Let $F = \{H_1, H_2\}$ where H_1 and H_2 are as in Figure 5.3. The graph H_1 is a tailed subgraph of H_2 . It is easy to see that S_1 and S_2 from the same figure are minimal connecting graphs of F .

The set F from Example 5.2.20 is subgraph free, but not tailed subgraph free. The following example illustrates the converse.

Example 5.2.21 Let $F = \{N_{1,0,0}, D_4\}$. Since $N_{1,0,0} \leq D_4$, F is not subgraph free, but F is tailed subgraph free.

It is easy to see that there are sets F that are both subgraph free and tailed subgraph free (see Example 5.2.3). Also, $F = \{H_1, H_2, H_3, H_4\}$ where $H_1 \leq H_2$ and $H_3 \leq_t H_4$ is neither subgraph free nor tailed subgraph free.

There are finite sets of graphs F not containing paths that have a finite $C_{min}(F)$.

Proposition 5.2.22 *Let $F = \{H_1, H_2, H_3\}$ be a subgraph free set of graphs. Then, for every $j = 1, 2, 3$ either H_j is a tailed subgraph of H_i for some $i \neq j$, $i = 1, 2, 3$ or there is a s such that for every $m \geq s$ $H_j \not\leq S_{H_i P_m H_l}$ where $i \neq l$, $j \neq i, l$, and $i, l = 1, 2, 3$.*

Proof. Let $j \in \{1, 2, 3\}$ be given and consider H_j . Let $S_{H_i P_m H_l}$ be such that $i \neq l$, $j \neq i, l$, and $i, l = 1, 2, 3$. If there is a s such that for every $m \geq s$ $H_j \not\leq S_{H_i P_m H_l}$, then the proposition holds. Otherwise, by Proposition 5.2.18 there is a k such that for every $n \geq k$ $H_j \leq S_{H_i P_n H_l}$. If $H_j \not\leq_t H_i, H_l$, then H_j must be “partly” in H_i , “partly” in H_l and must contain the entire P_n . Since $P_n \leq H_j$ for all $n \geq k$ it follows that H_j is infinite, which contradicts the fact that H_3 is finite. Therefore, H_j is a tailed subgraph of at least one of the graphs in $F \setminus \{H_j\}$. ■

Proposition 5.2.23 *Let $F = \{H_1, H_2\}$ be a subgraph free and tailed subgraph free set of graphs. Then, $C_{min}(F)$ is infinite.*

Proof. Consider $S_n = S_{H_1 P_n H_2}$. Since F is both subgraph and tailed subgraph free, the case where S_n is not going to be minimal is when one of the graphs is “partly” in H_1 , “partly” in H_2 and contains the entire S_n for some n . Since both H_1 and H_2 are finite graphs, there is a minimal k such that for every $n \geq k$ S_n is minimal. ■

Remark 5.2.24 Let $F = \{H_1, H_2\}$ be such that $H_1 \leq H_2$. Then, $C_{min}(F) = \{H_2\}$.

Proposition 5.2.25 *Let $F = \{H_1, H_2\}$ be such that $H_1 \leq_t H_2$. Then, $C_{min}(F)$ is finite.*

Proof. Let n_v be the length of the smallest path, which attached to H_2 at a vertex v will produce a minimal connecting graph S_v , i.e., $S_v = S_{H_2 P_{n_v}}$ where $V(H_2) \cap V(P_{n_v}) = \{v\}$. For every $v \in V(H_2)$ there is exactly one such S_v , hence there are finitely many S_v 's. Let K be the graph constructed from one copy of H_2 where at every vertex v from $V(H_2)$ a path of length one less than n_v is attached and that path is otherwise vertex-disjoint from the rest of the paths and H_2 . Take K and add edges to it (without adding new vertices) to complete it to K_m , i.e., consider K_m where $m = |V(K)|$. Let $G \in C(F)$. Then, there is a minimal connecting graph S of F with $S \leq G$ such that either $S = S_v$ for some S_v or $S \leq K_m$. Hence, $C_{min}(F)$ is finite. ■

Proposition 5.2.26 *Let F be a subgraph free and tailed subgraph free finite set of graphs such that $|F| \geq 2$. Then, $C_{min}(F)$ is infinite.*

Proof. It is clear that F does not contain any paths. Let $F = \{H_1, H_2, \dots, H_n\}$. If $n = 2$, by Proposition 5.2.23, $C_{min}(F)$ is infinite. Let $n > 2$. Consider the graph $S = S_{H_1 P_{i_1} H_2 \dots H_{n-1} P_{i_{n-1}} H_n}$ where $i_j \geq 2$ for every $j = 1, \dots, n-1$ and the values for i_j where $j = 1, \dots, n-1$ are determined as follows. Obviously, the graph S connects F for any i_j . Construct S in the following way. Let $S = S_{H_1 P_1 H_2}$. By Proposition 5.2.23, there is a smallest k such that S is minimal for every $n \geq k$. Let $l = k$. By Proposition 5.2.22, there is s such that for every $m \geq s$ $H_3 \not\leq S_{H_1 P_m H_2}$. Let t be the smallest such s and let $i_1 = \max\{k, t\}$. Let $S = S_{H_1 P_{i_1} H_2 P_1 H_3}$. If $H_1 \leq S_l = S_{H_2 P_1 H_3}$ for some $l \geq 2$, there is a smallest k such that for every $m \geq k$ $H_1 \not\leq S_l$. Let $l = k$. Otherwise, let $S = S_{H_1 P_{i_1} H_2 P_{i_2} H_3}$ where $i_2 = l$ and move to H_4 (if any). Check whether $H_4 \leq S_{H_1 P_{i_1} H_2}$ and whether $H_4 \leq S_{H_2 P_{i_2} H_3}$ and increase i_1 and i_2 so that this no longer holds. Then, “add” H_4 to S . Continue this way and at every step that H_i is added check whether H_1, \dots, H_{i-2} are subgraphs of $S_l = S_{H_{i-1} P_1 H_i}$ for some l , and in such case expand P_{i-1} until they are no longer subgraphs of S_l . Now that all values for i_j where $j = 1, \dots, n-1$ are determined, each time a connecting path in S is increased, this will result in a new minimal connecting graph of F .

Thus, $C_{min}(F)$ is infinite. ■

Theorem 5.2.27 *Let F be a subgraph free set of graphs such that $|F| \geq 2$, where for every $H, K \in F$ $H \not\leq_t K$ and vice versa, except if H or K is a path and there are at least two graphs that are not paths. Then, $C_{min}(F)$ is infinite.*

Proof. If F has no paths, the theorem holds by Proposition 5.2.26. If there is a path $P_k \in F$, (then $n \geq 3$) let $P_k = H_n$. Note that F can have at most one path. Proceed as in the proof of Proposition 5.2.26 until graph H_{n-1} is “added” to S . If H_n is already a subgraph of S , then the proof is completed. If H_n is not already a subgraph of S then $P_{i_{n-1}}H_n$ is a path P_l with such a length that $P_k \leq S$, but $P_{k+1} \not\leq S$. The theorem now follows. ■

Proposition 5.2.28 *Let F be a finite set of graphs. If $C_{min}(F)$ is infinite, then $|F| \geq 2$ and there exist $H, K \in F$ such that H and K are neither subgraphs of each other nor tailed subgraphs of each other.*

Proof. If $|F| = 1$, then $C_{min}(F)$ is finite. Suppose that $|F| \geq 2$ such that for every $H, K \in F$ either one of them is a subgraph of the other or one of them is a tailed subgraph of the other, i.e., $H \leq K$ or $K \leq H$ or $H \leq_t K$ or $K \leq_t H$. Let $F = \{H_1, \dots, H_n\}$. Then, either $H_1 \leq H_2$ or $H_2 \leq H_1$ or $H_1 \leq_t H_2$ or $H_2 \leq_t H_1$. In the first two cases take the larger graph and call it H' , in the last two cases extend one of the graphs by a path to obtain a minimal connecting graph of H_1 and H_2 . This can be done in finitely many ways. Let H' be one such graph. Proceed to H_3 . Then, for each of H_i , where $i = 1, 2$ either $H_i \leq H_3$ or $H_3 \leq H_i$ or $H_3 \leq_t H_i$ or $H_i \leq_t H_3$. In each case, the minimal connecting graphs of H_1 and H_2 can be extended in finitely many ways to obtain the minimal connecting graphs of H_1, H_2 , and H_3 . Continue this way until H_n . Consequently, $C_{min}(F)$ is finite. The proposition now follows. ■

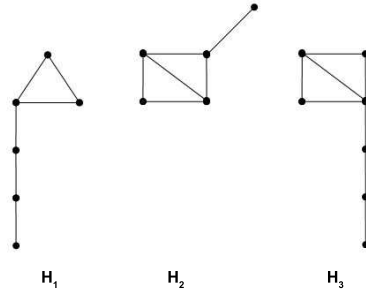


Figure 5.4: Graphs related to Example 5.2.29

Unlike the subgraph relation, the tailed subgraph relation is not an order. Consider the following example.

Example 5.2.29 Refer to Figure 5.4. $H_1 \not\leq_t H_1$ since $H_1 \leq H_1$ and this is true for any graph H . Also, $H_2 \leq_t H_3$ and $H_3 \leq_t H_2$ but the graphs H_2 and H_3 are not isomorphic. In fact, any two graphs that are tailed subgraphs of each other are not isomorphic. Finally, $H_1 \leq_t H_2$ and $H_2 \leq_t H_3$, but $H_1 \not\leq_t H_3$, since $H_1 \leq H_3$. Hence, \leq_t is neither reflexive, nor antisymmetric, nor transitive.

Definition 5.2.30 Define the graph $G_{\mathcal{U}}$ as a directed graph where the labels of the vertices are graphs and is constructed as follows. $G_{\mathcal{U}}$ starts at a single vertex labelled Λ , which vertex is said to be of level 0. For every vertex (graph) v from level n there is a vertex (graph) u in level $n - 1$, such that v can be obtained from u by adding an edge.

Figure 5.5 depicts the first four levels of the graph $G_{\mathcal{U}}$. Each level corresponds to the number of edges in its graph. Note that there is one graph at each of the levels 0, 1, and 2. There are 3 graphs at level 3, 5 graphs at level 4, 12 graphs at level 5, and so on. All paths that start at any vertex are infinite.

The facts in the next proposition follow directly from the definitions and are easy to prove.

Proposition 5.2.31 (i) *There is a one-to-one correspondence between the vertices of $G_{\mathcal{U}}$ and the graphs in \mathcal{U} .*

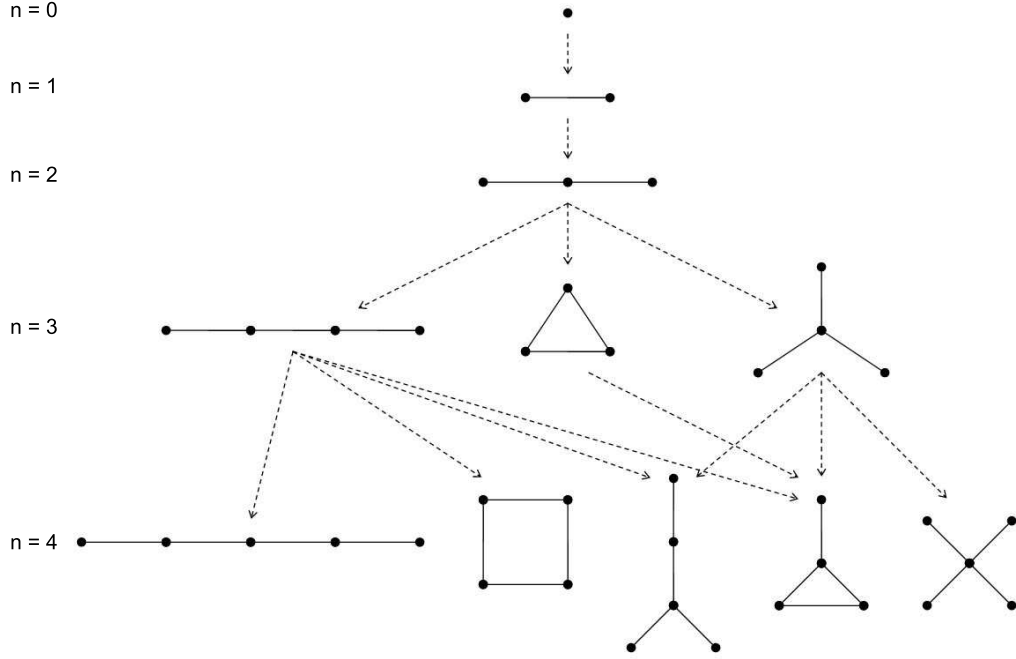


Figure 5.5: The graph $G_{\mathcal{U}}$ from Definition 5.2.30

- (ii) For every graph $G \in \mathcal{U}$ there is a not necessarily unique path in $G_{\mathcal{U}}$ that begins at Λ and ends at G and every vertex H in this path is such that $H \leq G$.
- (iii) For every $G \in \mathcal{U}$ and every path that begins at G , if H is a vertex on that path, then $G \leq H$.
- (iv) Let G be a vertex in $G_{\mathcal{U}}$ at level n . Then, for every H at level $n + 1$ if $(G, H) \notin E(G_{\mathcal{U}})$ it holds that $G \not\leq H$.
- (v) For every $G \in \mathcal{U}$ the set $\text{sub}(G) \setminus \{\emptyset\}$ equals the set of all graphs H , such that H lies on a path from Λ to G .
- (vi) If $S \in C(F)$ for some finite set of graphs F , then there is a path from each $H \in F$ to S . Furthermore, S is minimal if none of its predecessors are in $C(F)$.

For non-connected graphs the minimal connecting graphs are defined analogously.

Definition 5.2.32 Given a non-connected graph X , G is called a *connecting graph* of X if $X \leq G$. S is called a *minimal connecting graph* of X if S is a connecting graph and for every connecting graph H of X , $H \leq S$ implies $H = S$. Given a non-connected graph X denote the family of all connecting graphs of X with $C(X_g)$ and the family of minimal connecting graphs of X with $C_{min}(X_g)$.

For a non-connected graph X , the index g is added in $C(X_g)$ to distinguish between the connecting graphs of the components and the connecting graphs of the set. Thus, if X is viewed as a set of graphs, the connecting graphs would be $C(X)$.

As stated in the next section, non-connected graphs X appear in graph fe -systems only as a first component of an enforcer. In this case, not the minimality of connecting graphs of X per se is of interest, but rather a minimal connecting graph of X that contains a specific copy of X and is embedded in a specified connected graph G . This type of minimality is relative to G and to the copy of X in G . Hence, the following definition.

Definition 5.2.33 Let X be a non-connected graph and G be a graph such that $X \leq G$. Let \hat{X} be a copy of X in G . S is called a *minimal connecting graph of \hat{X} relative to G* if S is a connecting graph of \hat{X} in G and if H is also a connecting graph of \hat{X} in G with $H \leq S$ then $H = S$. Denote the minimal connecting graphs of \hat{X} relative to G with $C_{min}^G(\hat{X})$.

Example 5.2.34 If X consists of an edge and an isolated vertex, the minimal connecting graph of X is P_3 and it is unique, but for a specific copy \tilde{X} of X in some connected graph G , the minimal connecting graph containing that specific copy \tilde{X} (edge e and vertex v) in G may be P_5 (see Figure 5.6).

Definition 5.2.35 Let G be a graph. G^1 is an *extension by an edge* of G , if $E(G^1) = E(G) \cup \{e\}$ where $e \notin E(G)$ and has one vertex in G and the other is of degree 1.

Thus, $N_{1,0,0}$ is an extension by an edge of K_3 .

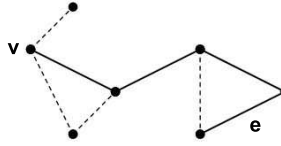


Figure 5.6: A minimal connecting graph of \tilde{X} in G from Example 5.2.34

Remark 5.2.36 For every connected graph G and every non-connected graph X such that $X \leq G$ and for every copy \hat{X} of X in G , there is a minimal connecting graph S relative to G , i.e., there is a $S \in C_{min}^G(\hat{X})$.

Let X be a finite non-connected graph. Then X has a finite number of components say H_1, \dots, H_n , where $n \geq 2$. Let the number of vertices of each component be k_i and let $m = \sum_{i=1}^n k_i$. Then, the graph K_m is a connecting graph of X . Consider a copy of \hat{X} of X in G . Consider the set of all graphs T that can be obtained by removing any number of edges from G , such that T is connected and contains the copy \hat{X} . Then, there is a minimal connecting graph S of \hat{X} relative to G , such that $S = T$ for some T . Note that S is not necessarily unique.

Now consider an infinite sequence of extensions by an edge of K_m, K_m^1, K_m^2, \dots such that $K_m^i \leq K_m^{i+1}$. Then, $X \leq K_m^i$ for every $i \geq 1$. Hence, if X is a finite non-connected graph, then $C^G(X)$ is infinite.

Instead of the extensions by an edge, one can also take the sequence $K_k, K_{k+1}, \dots, K_n, \dots$, which also shows that $C^G(X)$ is infinite.

Another way to construct a connecting graph for X is to order the connected components and connect them by paths. More specifically, let $\{H_1, \dots, H_n\}$ be the connected components of X in some order. Let $S = S_{H_1 P_1 H_2 \dots P_{n-1} H_n}$. Then, S is a connecting graph of X and varying the length of the “connecting” paths will produce infinitely many connecting graphs of X . In addition, there is an infinite sequence of extensions by an edge of S, S^1, S^2, \dots , such that $S^i \leq S^{i+1}$ and $X \leq S^i$ for every $i \geq 1$.

5.3 Graph f -Systems and Their Properties

Definition 5.3.1 A *forbidding set* \mathcal{F} is a (possibly infinite) family of finite nonempty sets of finite non-trivial connected graphs; each element of a forbidding set is called a *forbidder*.

A graph G is said to be *consistent* with a forbidder F , denoted by $G \underline{\text{con}} F$, if G is connected and $F \not\subseteq \underline{\text{sub}}(G)$. A graph G is consistent with a forbidding set \mathcal{F} , if $G \underline{\text{con}} F$ for all $F \in \mathcal{F}$. $G \underline{\text{ncon}} \mathcal{F}$ denotes that G is not consistent with \mathcal{F} .

For a forbidding set \mathcal{F} the family of \mathcal{F} -consistent graphs is $\mathcal{L}(\mathcal{F}) = \{G \mid G \underline{\text{con}} \mathcal{F}\}$.

The family $\mathcal{L}(\mathcal{F})$ is said to be *defined* by the forbidding set \mathcal{F} . A family \mathcal{L} is called a *f-family*, if there is a forbidding set \mathcal{F} such that $\mathcal{L} = \mathcal{L}(\mathcal{F})$.

Remark 5.3.2 Given a forbidder F and a graph G either $G \underline{\text{con}} F$ or G connects F , i.e., $\mathcal{L}(\mathcal{F})^c = C(F)$.

The following boundary observations state that if nothing is forbidden everything is allowed and that the trivial and null graphs are always in a f -family of graphs.

Remark 5.3.3 (i) $\mathcal{L}(\mathcal{F}) = \mathcal{U}$ if and only if \mathcal{F} is empty.

(ii) The null graph \emptyset and the trivial graph Λ are in $\mathcal{L}(\mathcal{F})$ for every \mathcal{F} .

Proof. If the forbidding set is empty, then $F \not\subseteq \underline{\text{sub}}(G)$ for all $G \in \mathcal{U}$ trivially. Conversely, suppose that $\mathcal{L}(\mathcal{F}) = \mathcal{U}$. If \mathcal{F} is not empty, then there is a forbidder F and any graph G for which $F \subseteq \underline{\text{sub}}(G)$ is not consistent with \mathcal{F} . The second observation is trivial. ■

Definition 5.3.4 Let \mathcal{F} be a forbidding set. If for each $F \in \mathcal{F}$, $|F| = 1$ then \mathcal{F} is called a *strict* forbidding set.

In general, forbidders may contain more than one element. The following example shows how a general forbidding set differs from a strict one.

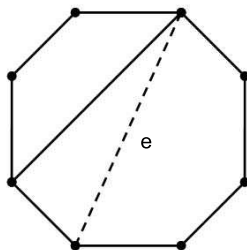


Figure 5.7: Graph related to Example 5.3.6

Example 5.3.5 Let $\mathcal{F} = \{C_3, C_4\}$. A graph $G \in \mathcal{L}(\mathcal{F})$ can have a 3-cycle or a 4-cycle as subgraphs, but not both. However, G does not need to have a cycle to be in $\mathcal{L}(\mathcal{F})$, i.e., all trees are in $\mathcal{L}(\mathcal{F})$, as well.

Example 5.3.6 Let $\mathcal{F} = \{C_3, C_4\}, \{C_5, C_6\}$. The graph depicted in figure 5.7 has a 4-cycle and a 6-cycle as subgraphs. If the edge e is not in the graph, then the graph is consistent with \mathcal{F} . If the graph contains the edge e , then it is not consistent with both forbidders. Of course, trees are consistent with \mathcal{F} .

Historically, the concept of *forbidden* graphs has been used to characterize Hamiltonian graphs and in Turan type problems (see for ex. [2, 7, 17]). For a comprehensive list of references refer to [2, 18]. In existing literature, *forbidden* graphs are a finite (or in some cases infinite) set of graphs $\{F_1, F_2, \dots\}$ where each of these F_i is a forbidden (induced) subgraph of G . In that respect, our *forbidding sets* definition differs from the *forbidden* graphs in that, it employs forbidders that are not necessarily singletons but a finite set of graphs and it considers subgraphs as opposed to induced subgraphs.

Definition 5.3.7 An *enforcing set* \mathcal{E} is a possibly infinite family of ordered pairs (X, Y) , such that X is a not necessarily connected graph, $Y = \{Y_1, \dots, Y_m\}$ where Y_i is a connected graph for $i = 1, \dots, m$, $X < Y_i$ for every $Y_i \in Y$, and $Y \neq \emptyset$. The elements of an enforcing set are called *enforcers*.

A graph G is said to *satisfy* an enforcer (X, Y) if G is connected and whenever $X \leq G$ there is $Y_i \in Y$ such that $X < Y_i \leq G$. Moreover, for every embedding

$\theta : X \rightarrow G$ there exists an embedding $\psi : X \rightarrow Y_i$ and there is an embedding $\phi : Y_i \rightarrow G$ such that $\theta = \phi\psi$. In this case we write $G \underline{sat}(X, Y)$. A graph G satisfies an enforcing set \mathcal{E} if G satisfies every enforcer in that set. For an enforcing set \mathcal{E} the family of all graphs that satisfy \mathcal{E} is denoted by $\mathcal{L}(\mathcal{E})$. A family of graphs \mathcal{L} is called an *e-family* if there exists an enforcing set \mathcal{E} such that $\mathcal{L} = \mathcal{L}(\mathcal{E})$.

To ease notation, enforcers are sometimes denoted with E , i.e., $E = (X, Y)$. In the case that $X \not\subseteq G$, G is said to satisfy the enforcer trivially. Enforcers in which $X = \emptyset$ or $X = \Lambda$ are called *brute*. In this case, a graph from Y has to be a subgraph of G in order for G to satisfy the enforcer.

An enforcer is called *trivial* if every graph satisfies it and nontrivial otherwise. In language enforcing sets enforcers (X, Y) where $X \cap Y \neq \emptyset$ are called trivial, because every language satisfies them. Thus, if a language enforcing set consists of trivial enforcers only, it defines the entire $\mathcal{P}(A^*)$. If a language enforcing set doesn't have trivial enforcers, then it defines the entire language set if and only if it is empty.

The following is an investigation of whether the graph enforcing sets can have trivial enforcers.

Proposition 5.3.8 *Trivial enforcers do not exist for graph enforcing sets.*

Proof. Let (X, Y) be an enforcer. If X is a connected graph, then $X \underline{nsat}(X, Y)$. Let X be non-connected. Let k be the maximum number of vertices in a graph $\hat{Y} \in Y$ and let X_1, \dots, X_n be the connected components of X . Consider the graph $S = S_{X_1 P_k X_2 P_k \dots P_k X_n}$. Then $S \underline{nsat}(X, Y)$. ■

Since there are no trivial enforcers in the graph case, we have the following remark.

Remark 5.3.9 (i) $\mathcal{L}(\mathcal{E}) = \mathcal{U}$ if and only if $\mathcal{E} = \emptyset$

(ii) $\emptyset \in \mathcal{L}(\mathcal{E})$ for every \mathcal{E} that does not have brute enforcers of the kind (\emptyset, Y) .

(iii) $\Lambda \in \mathcal{L}(\mathcal{E})$ for every \mathcal{E} that does not have brute enforcers.

Definition 5.3.10 An enforcer (X, Y) is called *strict* if $|Y| = 1$.

In some sense, strict enforcers “force” the graph from Y into the graph G for each occasion of X in G . Consider the following example consisting of strict enforcers only.

Example 5.3.11 Consider the enforcing set $\mathcal{E} = \{(C_3, \{N_{1,1,1}\}), (N_{1,1,1}, \{N_{2,2,2}\}), \dots, (N_{n,n,n}, \{N_{n+1,n+1,n+1}\}), \dots\}$. If a graph G has a 3-cycle, then each copy of a 3-cycle in G has to be embedded in a $N_{n,n,n}$, for any n , which will cause G to be infinite. Hence, $\mathcal{L}(\mathcal{E})$ defines the family of “triangle-free” graphs.

The two notions of forbidding and enforcing on graphs are combined in the following definition.

Definition 5.3.12 A *forbidding-enforcing system* is a construct $\Gamma = (\mathcal{F}, \mathcal{E})$ such that \mathcal{F} is a forbidding set and \mathcal{E} is an enforcing set. The family of graphs $\mathcal{L}(\Gamma)$ defined by this system consists of all graphs G that are consistent with \mathcal{F} and satisfy \mathcal{E} , i.e., $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{E})$.

A family of graphs \mathcal{L} is called a *forbidding-enforcing family* or *fe-family*, if there exists a *fe-system* $(\mathcal{F}, \mathcal{E})$, such that $\mathcal{L} = \mathcal{L}(\mathcal{F}, \mathcal{E})$.

Example 5.3.13 As shown later in this chapter, $\mathcal{F} = \{\{C_3\}, \{C_4\}, \dots\}$ defines the family of trees and $\mathcal{E} = \{(P_3, \{C_3\})\}$ defines the family of complete graphs. (See Proposition 5.5.3 and Proposition 5.5.5). Consider the forbidding-enforcing system $\Gamma = (\mathcal{F}, \mathcal{E})$. The obtained family of graphs that obey the system is $\mathcal{L}(\Gamma) = \{\emptyset, \Lambda, P_2, C_3\}$.

Two sets of forbidders (or two enforcing sets, or two forbidding-enforcing systems) are equivalent, if they define the same family of graphs. The equivalence relation is denoted by \sim . Also, two forbidders (enforcers) are equivalent (again denoted by \sim) if the singleton forbidding (enforcing) sets containing each of them are equivalent, e.g., $F \sim F'$ if and only if $\mathcal{F} \sim \mathcal{F}'$ where $\mathcal{F} = \{F\}$ and $\mathcal{F}' = \{F'\}$. Similarly, $E \sim E'$ if and only if $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E} = \{E\}$ and $\mathcal{E}' = \{E'\}$.

From the above definitions it follows that there is no forbidding set \mathcal{F} such that $\mathcal{L}(\mathcal{F})$ is empty, but there are enforcing sets \mathcal{E} such that $\mathcal{L}(\mathcal{E}) = \emptyset$ and there are fe -systems $(\mathcal{F}, \mathcal{E})$, such that $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \emptyset$ as shown later in this chapter.

Remark 5.3.14 $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{U}$ if and only if $\mathcal{F} = \emptyset$ and $\mathcal{E} = \emptyset$.

The next proposition states some of the immediate properties of graph fe -systems. They follow directly from the definitions above and match exactly the properties of language fe -systems as stated in [40].

Proposition 5.3.15 *Let \mathcal{F} and \mathcal{F}' be forbidding sets, \mathcal{E} and \mathcal{E}' be enforcing sets, and G and H be connected graphs.*

- (i) *If $H \leq G$ and $G \underline{con} \mathcal{F}$, then $H \underline{con} \mathcal{F}$.*
- (ii) *If $\mathcal{F}' \subseteq \mathcal{F}$, then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$.*
- (iii) *If $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$.*
- (iv) *If $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}', \mathcal{E}')$.*
- (v) $\mathcal{L}(\mathcal{F} \cup \mathcal{F}') = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{F}')$.
- (vi) $\mathcal{L}(\mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{E}) \cap \mathcal{L}(\mathcal{E}')$.
- (vii) $\mathcal{L}(\mathcal{F} \cup \mathcal{F}', \mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{F}, \mathcal{E}) \cap \mathcal{L}(\mathcal{F}', \mathcal{E}')$.

5.4 Forbidding through Enforcing

Example 5.4.1 Let $\mathcal{E} = \{(K_{1,3}, \{H_{2,2,2}\}), \dots, (H_{n,n,n}, \{H_{n+1,n+1,n+1}\}), \dots\}$. Then, $K_{1,3} \leq G$ if and only if $G \underline{nsat} \mathcal{E}$. Hence $\mathcal{L}(\mathcal{E}) = \mathcal{L}(\mathcal{F})$ for $\mathcal{F} = \{\{K_{1,3}\}\}$.

The above example shows that there are forbidders (forbidding sets) which could be replaced entirely by enforcing sets.

Proposition 5.4.2 *Let $\mathcal{F} = \{F\}$ and let $S \in C_{min}(F)$. Consider $\mathcal{E}_F^S = \{(S, \{S^1\}), \dots, (S^{n-1}, \{S^n\}), \dots \mid \text{where } S < S^1 \text{ and } S^{i-1} \leq S^i \text{ for every } i \geq 2\}$. Then, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E}_F)$, where $\mathcal{E}_F = \cup_{S \in C_{min}(F)} \mathcal{E}_F^S$.*

Proof. If $G \underline{sat} \mathcal{E}_F$, G cannot have any of the minimal connecting graphs for F as a subgraph. Hence, $F \not\subseteq \underline{sub}(G)$. Conversely, if $G \underline{con} \mathcal{F}$ it follows that $F \not\subseteq \underline{sub}(G)$, i.e., there is an $H \in F$ such that $H \not\subseteq G$. Therefore, $G \underline{sat} \mathcal{E}_F$. ■

The above proposition can be extended to a general forbidding set \mathcal{F} with more than one forbider by including enforcers like \mathcal{E}_F for every $F \in \mathcal{F}$ and considering their union $\mathcal{E} = \cup_{F \in \mathcal{F}} \mathcal{E}_F$. Then, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E})$. The next theorem states this conclusion formally.

Theorem 5.4.3 *For every forbidding set \mathcal{F} , there is an enforcing set \mathcal{E} such that $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E})$.*

Proof. Let \mathcal{F} be a forbidding set. If $\mathcal{F} = \emptyset$, then let $\mathcal{E} = \emptyset$. By Remarks 5.3.3 and 5.3.9 $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E}) = \mathcal{U}$. Let \mathcal{F} have at least one forbider. For every forbider $F \in \mathcal{F}$ construct the enforcing set \mathcal{E}_F as in Proposition 5.4.2. Consider $\mathcal{E} = \cup_{F \in \mathcal{F}} \mathcal{E}_F$. Let $G \underline{con} \mathcal{F}$ and let $(X, Y) \in \mathcal{E}$. Since X is a connecting graph of at least one $F \in \mathcal{F}$ it follows $X \not\subseteq G$. Therefore, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{E})$. If $G \underline{sat} \mathcal{E}$, then for every enforcer $(X, Y) \in \mathcal{E}$, $X \not\subseteq G$. More specifically, none of the minimal connecting graphs of \mathcal{F} are subgraphs of G . Hence, $G \underline{con} \mathcal{F}$. Thus, $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{F})$. Consequently, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E})$ ■

This does not, however, render forbidding sets obsolete. It would be much more practical and useful to represent a graph family by finite structures like the forbid-ers F , rather than infinite sets like \mathcal{E}_F . In addition, *fe*-systems may potentially be applied in DNA computing and self assembly of graphs, where the finiteness of constraints is of great importance.

5.5 Characterizations of Some Classes of Graphs by *fe*-Systems

Forbidding and enforcing sets can be used to define familiar classes of graphs. The characterizations that follow show that relatively simple *fe*-systems (in some in-

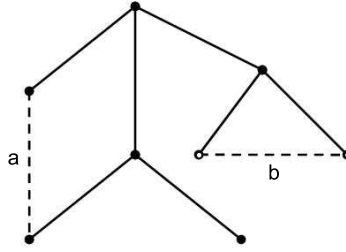


Figure 5.8: Tree related to Proposition 5.5.1

stances finite) are capable of defining large classes of graphs.

Proposition 5.5.1 (Trees.) *Let $\mathcal{F} = \{\{C_3\}, \{C_4\}, \dots, \{C_n\}, \dots\}$. Then $\mathcal{L}(\mathcal{F}) = \{G \mid G \text{ is a tree}\}$. In other words, G is a tree if and only if $G \in \mathcal{L}(\mathcal{F})$.*

Proof. Follows from the fact that $\mathcal{L}(\mathcal{F})$ contains every graph that does not have a cycle. ■

Figure 5.8 illustrates how the presence of edges a or b will make the graph non consistent with $\{C_4\}$ or $\{C_3\}$ respectively.

The Bipartite Graph Characterization Theorem states that a graph is bipartite if and only if the length of each of its cycles is even (see [18]). Hence the following f -family characterization of bipartite graphs.

Proposition 5.5.2 (Bipartite graphs.) *Let $\mathcal{F} = \{\{C_3\}, \{C_5\}, \dots, \{C_{2k+1}\}, \dots\}$. Then $\mathcal{L}(\mathcal{F})$ contains every graph that does not have an odd cycle, i.e., $\mathcal{L}(\mathcal{F}) = \{G \mid G \text{ is bipartite}\}$.*

In both propositions 5.5.1 and 5.5.2 the forbidding sets are infinite and the forbidders are singletons. The following proposition shows how a finite forbidding set can define an infinite family of graphs and provides an f -family characterization of paths and cycles.

Proposition 5.5.3 (Paths and cycles.) *Let $\mathcal{F} = \{\{K_{1,3}\}\}$. Then $\mathcal{L}(\mathcal{F}) = \{P_n \mid n \geq 0\} \cup \{C_n \mid n \geq 3\}$. In other words, G is a path or a cycle if and only if $G \in \mathcal{L}(\mathcal{F})$.*

Proof. Clearly, $\mathcal{L}(\mathcal{F})$ contains every connected graph that does not have a vertex with degree more than 2. ■

The next corollary provides an f -family characterization of paths.

Corollary 5.5.4 (Paths.) *Let $\mathcal{F} = \{\{K_{1,3}\}, \{C_3\}, \{C_4\}, \dots\}$. Then, $\mathcal{L}(\mathcal{F}) = \{G \mid G \text{ is a path}\}$.*

The above corollary follows from Propositions 5.5.3, 5.5.1, and 5.3.15.

In propositions 5.5.1, 5.5.2, and 5.5.3 each forbidders is a singleton. Thus, the graphs appearing in the forbidders are in some sense “strictly” forbidden as subgraphs. These propositions show that some classes of graphs can be classified using forbidders only. The following characterization shows that a singleton enforcing set defines the class of complete graphs.

Proposition 5.5.5 (Complete graphs.) *Let $\mathcal{E} = \{(P_3, \{C_3\})\}$. Then \mathcal{E} defines the class of complete graphs, i.e., $\mathcal{L}(\mathcal{E}) = \{G \mid G \text{ is a complete graph}\}$.*

Proof. If a graph is complete, any three vertices form a 3-cycle. On the other hand, suppose $G \in \mathcal{L}(\mathcal{E})$ and G has two vertices u and v that are not adjacent. Since G is connected, there is a path P_n with $n \geq 3$ from u to v . Let the order of the vertices in the path be $u_1 = u, u_2, \dots, u_{n-1}, u_n = v$. Since $u_1u_2u_3$ is a P_3 , then the edge $\{u_1, u_3\}$ must be in the graph. Similarly, $u_1u_3u_4$ implies $\{u_1, u_4\}$ is in the graph. Continuing this way, $u_1u_{n-1}u_n$ implies that $\{u_1, u_n\} \in E(G)$. ■

The following definition of trimmed extension sets is used in the characterization of k -regular graphs.

Definition 5.5.6 \hat{H} is called an *extension* of $K_{1,k}$ if it can be constructed from $K_{1,k}$ as follows. Label the vertex of degree k with v and all other vertices with v_1, \dots, v_k . Let $V(\hat{H}) = V(K_{1,k}) \cup \{v'_1, \dots, v'_k\}$ and $E(\hat{H}) = E(K_{1,k}) \cup \{\{v_1, v'_1\}, \dots, \{v_k, v'_k\} \mid \text{where } v'_i \neq v \text{ for every } i = 1, \dots, k\}$.

An extension \hat{H}^T is called *trimmed*, if it is obtained from some \hat{H} as follows. If \hat{H} is such that for every $i = 1, \dots, k$ and every $j = 1, \dots, k$ it holds that $v_i \neq v'_j$ where $v_i, v'_j \in V(\hat{H})$, then \hat{H} is trimmed, i.e., $\hat{H}^T = \hat{H}$. Otherwise, there is a v_i such that $v_i = v'_j$ for some $j \neq i$. In this case, remove the edge $\{v_i, v'_i\}$ from $E(\hat{H})$ and the resulting \hat{H}^T is trimmed.

The set of graphs $H^k = \{\hat{H}^T \mid \hat{H}^T \text{ is a trimmed extension of } K_{1,k}\}$ is called the *trimmed extension set* of $K_{1,k}$.

Note that in the definition of extension, the vertices v_i are all distinct, but the vertices v'_i are not necessarily distinct, i.e., it may be the case that $v'_s = v'_t$ for some $s \neq t$. Also, it is possible that some $v'_i = v_j$ for some $i \neq j$. However, $v'_i \neq v_i$ for every $i = 1, \dots, k$.

It should be noted that the number of vertices of an extension of $K_{1,k}$ is at most $2k + 1$, so H^k is a finite set. Also, notice that if $\hat{H}^T \in H^k$ (following the above labelling) $\deg(v_i) > 1$ for every $i = 1, \dots, k$ and all vertices of \hat{H}^T are of degree at most k .

The reason $\deg(v_i) > 1$ for every $i = 1, \dots, k$ is that either the edges $\{v, v_i\}$ and $\{v_i, v'_i\}$ are in $E(\hat{H}^T)$, or $\{v, v_i\}$ and $\{v_i, v_j\}$ are in $E(\hat{H}^T)$ for some $j \neq i$. To see that all vertices in \hat{H}^T are of degree at most k consider the following. Suppose there is a vertex u with $\deg(u) > k$. Then, $v \neq u$. Also, $v_i \neq u$ for every $i = 1, \dots, k$, since all v_i 's are distinct and there can be at most $k - 1$ edges incident with v_i and v'_j for all $j \neq i$ (in this case the edge $\{v_i, v'_i\}$ is removed so they are $k - 1$) plus the edge $\{v, v_i\}$ the degree of v_i becomes at most k . Then, $u = u'_j$ for some j . If $u'_j = u_i$ for some i the case is explained above. Then, at most $u'_j = u'_s$ for all $j, s = 1, \dots, k$ in which case $\deg(u) \leq k$ again. Hence, $\deg(u) \leq k$ for every $u \in V(\hat{H}^T)$.

Theorem 5.5.7 (k -regular graphs.) *Let $k \geq 3$. Let $\mathcal{F} = \{\{K_{1,k+1}\}\}$. Consider the enforcing set $\mathcal{E} = \{E_1, \dots, E_{k+2}\}$ where $E_1 = (\emptyset, \{P_2\})$, $E_2 = (\Lambda, \{P_2\})$, $E_3 = (P_2, \{K_{1,k}\})$, $E_4 = (P_3, \{K_{1,k}\})$, $E_{i+2} = (K_{1,i}, K_{1,k})$ for $i = 3, \dots, k-1$, and $E_{k+2} = (K_{1,k}, H^k)$ where H^k is the trimmed extension set of $K_{1,k}$. Then, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \{G \mid G \text{ is } k\text{-regular}\}$.*

Proof. Let G be a k -regular graph. Obviously, $G \underline{\text{con}} \mathcal{F}$ and $G \underline{\text{sat}} \{E_1, \dots, E_{k+1}\}$. Let $\hat{K}_{1,k}$ with central vertex v and vertices v_1, \dots, v_k be a copy of $K_{1,k}$ in G . Since G is k -regular, for every $i = 1, \dots, k$ there exists an edge $\{v_i, v'_i\} \in E(G)$ such that $v'_i \neq v$ and $v'_i \neq v_i$. For every $i = 1, \dots, k$ consider one such edge $\{v_i, v'_i\}$ and consider the graph \hat{H} that consists of $\hat{K}_{1,k}$ along with these edges. It is clear that all vertices in \hat{H} are of degree at most k . If for every $i = 1, \dots, k$ and for every $j = 1, \dots, k$ it holds that $v_i \neq v'_j$ then \hat{H} is trimmed, so $\hat{H} \in H^k$ and $G \underline{\text{sat}} E_{k+2}$. Otherwise, there exist i and j , such that $v_i = v'_j$. Consider all such v_i and remove from \hat{H} all edges $\{v_i, v'_i\}$ for these v_i 's. Then, \hat{H} becomes trimmed, i.e., $\hat{H} \in H^k$. Hence, $\hat{K}_{1,k}$ is embedded in \hat{H}^T for some $\hat{H}^T \in H^k$. Therefore, $G \underline{\text{sat}} E_{k+2}$.

Conversely, let $G \in \mathcal{L}(\mathcal{F}, \mathcal{E})$. Since $G \underline{\text{con}} \mathcal{F}$, all vertices in G are of degree k or less. Since $G \underline{\text{sat}} \{E_1, E_2\}$, all vertices in G are of degree greater than 0 and G has at least one edge. Let $u \in V(G)$. Since $\deg(u) \geq 1$ there is an edge $e = \{u, u'\} \in E(G)$. Since $G \underline{\text{sat}} E_3$ the edge e has at least one vertex of degree k . If $\deg(u) = k$, then G is 3-regular. If not, consider the copy $\hat{K}_{1,k}$ of $K_{1,k}$ in G in which $e = \{v, v_1\}$ where v is the central vertex of $\hat{K}_{1,k}$ and $u = v_1$ (resp. $u' = v$). Since $G \underline{\text{sat}} E_{k+2}$, it follows that $\deg(v_1) > 1$. Let $\deg(v_1) = s$, where $1 < s \leq k$. If $s = 2$, then E_4 ensures that $\deg(v_1) = k$. When $3 \leq s \leq k$, the fact that $G \underline{\text{sat}} \{E_5, \dots, E_{k+1}\}$, implies that $\deg(v_1) = k$. Thus, $\deg(u) = k$, i.e., G is 3-regular. ■

Corollary 5.5.8 (3-regular graphs.) *Let $\mathcal{F} = \{\{K_{1,4}\}\}$ and $\mathcal{E} = \{(\emptyset, \{P_2\}), (\Lambda, \{P_2\}), (P_2, K_{1,3}), (P_3, K_{1,3}), (K_{1,3}, \{H_1, H_2, H_3, H_4, H_5\})\}$, where H_1, H_2, H_3, H_4 and H_5 are as indicated in Figure 5.9. Then, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \{G \mid G \text{ is 3-regular}\}$.*

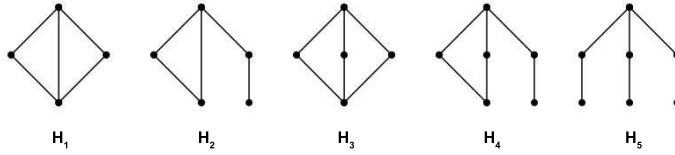


Figure 5.9: Graphs related to \mathcal{E} in Corollary 5.5.8

The proof follows directly from the above theorem. A direct proof which does not use Theorem 5.5.7 is presented below.

Proof. Label the enforcers in the order that they appear in the statement of the corollary as follows: $\mathcal{E} = \{E_1, E_2, E_3, E_4, E_5\}$. Assume, G is a 3-regular graph. Then $G \underline{con} \mathcal{F}$. Also, G has at least 4 vertices and 6 edges, so $G \underline{sat} E_1, E_2$ and since it is 3-regular, $G \underline{sat} E_3, E_4$. Let $\hat{K}_{1,3}$ be a copy of $K_{1,3}$ in G . Let v be its central vertex and v_1, v_2, v_3 be the three distinct vertices adjacent to v . Since G is 3-regular, there exist edges $\{v_1, v'_1\}$, $\{v_2, v'_2\}$, and $\{v_3, v'_3\}$, with $v'_1 \neq v$, $v'_2 \neq v$, and $v'_3 \neq v$. Then, either there is a i for which there is a $j \neq i$ such that $v_i = v'_j$ or not. If there is such i , then $\hat{K}_{1,3}$ is enclosed in either H_1 or H_2 in G (see Figure 5.9). Otherwise, none of the v_i 's equals a v'_j , hence $\hat{K}_{1,3}$ is enclosed in either H_3 , or H_4 , or H_5 in G . Hence, $G \underline{sat} E_5$ and thus $G \in \mathcal{L}(\mathcal{F}, \mathcal{E})$.

Conversely, assume that $G \in \mathcal{L}(\mathcal{F}, \mathcal{E})$. Since $G \underline{con} \mathcal{F}$, G does not have a vertex of degree 4 or higher. Since $G \underline{sat} \{E_1, E_2\}$, G has at least one edge e . Since $G \underline{sat} E_3$ the edge e has at least one vertex of degree 3. E_5 requires that the other vertex of e is either of degree 2 or of degree 3. If it is of degree 2, then E_3 requires that it is of degree 3. Thus, G is 3-regular. ■

Chapter 6

Normal Forms for Graph fe -Systems

In general, forbidding (enforcing) sets may have superfluous forbidders (enforcers) that can be removed without changing the fe -family. This chapter contains an investigation of ways to remove redundant forbidders and enforcers and presents some normal forms for graph fe -systems.

6.1 Normal Forms for Forbidding Sets

Forbidding sets could be redundant and could be reduced by removing some parts of them. In this section, ways to reduce single forbidders, as well as, removal of entire forbidders from a forbidding set without changing the f -family are investigated. The subgraph free and subgraph incomparable normal forms match the subword free and subword incomparable normal forms from the language fe -systems from [40]. However, the difference of the structures involved requires a new way of proving some of these normal forms. In addition, this section includes the connecting free and connecting incomparable normal forms which are new and do not have an analog in the language fe -systems. They come to interest only with graph fe -systems.

Example 6.1.1 Consider the forbidding set $F = \{K_{1,3}, C_4, D_4\}$. A graph G is consistent with F if either $K_{1,3} \not\subseteq G$ or $C_4 \not\subseteq G$ or $D_4 \not\subseteq G$. In either case D_4 is not allowed as a subgraph of G . Hence, $G \underline{con} F$ implies $G \underline{con} F'$ where $F' = \{D_4\}$. Conversely, if $D_4 \not\subseteq G$ then obviously, $G \underline{con} F$. So, $G \underline{con} F$ if and only if $G \underline{con} F'$. Therefore, F can be replaced with F' , which in essence reduces F to F' .

The forbidding set F in the above example is redundant because $K_{1,3} \leq D_4$ and $C_4 \leq D_4$. In other words, the forbidding set F is not *subgraph free*.

Definition 6.1.2 A forbidding set F is called *subgraph free* if the set F is subgraph free (i.e., for every $K, H \in F$ neither $K \leq H$ nor $H \leq K$). A forbidding set \mathcal{F} is called *subgraph free* if all of its forbidding sets are subgraph free.

In relation to the $G_{\mathcal{U}}$ graph from the previous chapter, a forbidding set is subgraph free if for every two graphs H, K from this forbidding set, there is no path from H to K or vice versa in the $G_{\mathcal{U}}$ graph.

When reducing a forbidding set by discarding graphs which are subgraphs of other graphs in that forbidding set, the newly obtained forbidding set is *maximal* in some sense. Every graph in a maximal forbidding set does not have subgraphs in that forbidding set. A formal definition follows.

Definition 6.1.3 Given a forbidding set F define $F_{max} = \{H \in F \mid K \in F \text{ with } H \leq K \text{ if and only if } H = K\}$.

It is clear that F_{max} is subgraph free. In addition, for every F there is an algorithm for finding F_{max} . Given a forbidding set F , the forbidding set F_{max} is unique.

Using the $G_{\mathcal{U}}$ graph, F_{max} can be found by considering all vertices from F and removing all vertices K from which there is a path from K to some vertex H in F .

Lemma 6.1.4 Let \mathcal{F} be a forbidding set and F be a forbidding set in \mathcal{F} such that there exist graphs H and K in F with $H \leq K$. Then $\mathcal{F} \sim (\mathcal{F} \setminus \{F\}) \cup \{F'\}$, where $F' = F \setminus \{H\}$.

Proof. Let \mathcal{F}, F, H and K satisfy the conditions of the lemma. Let $\mathcal{F}' = (\mathcal{F} \setminus \{F\}) \cup \{F'\}$. It is obvious that $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F})$. Let $G \in \mathcal{L}(\mathcal{F})$. It follows that there exists $L \in F$ such that $L \leq G$. If $L \neq H$, then $G \in \mathcal{L}(F')$. If $L = H$, then since $H \leq K$, it follows that $K \leq G$. Hence, $G \in \mathcal{L}(F)$ and $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. ■

Subgraph free is a normal form, as stated in the next lemma.

Lemma 6.1.5 *For every forbidding set there exists an equivalent subgraph free forbidding set.*

Proof. Let \mathcal{F} be given. Consider $\mathcal{F}_{max} = \{F_{max} \mid F \in \mathcal{F}\}$. Clearly, for all $F \in \mathcal{F}$ and for all graphs G we have that $F \subseteq \underline{sub}(G)$ if and only if $F_{max} \subseteq \underline{sub}(G)$. Hence, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{F}_{max})$. ■

In general, $\underline{sub}(F_1) = \underline{sub}(F_2)$ does not necessarily imply that $F_1 = F_2$. For example, consider $\underline{sub}(\{K_{1,3}, D_4\}) = \underline{sub}(\{C_4, D_4\})$. In terms of $G_{\mathcal{U}}$, if F_1 is a set of vertices in $G_{\mathcal{U}}$ and some additional vertices that are on some paths from Λ to vertices in F are added to F_1 to obtain F_2 , then $\underline{sub}(F_1) = \underline{sub}(F_2)$, but $F_1 \neq F_2$ necessarily. However, the following lemma shows that this cannot happen if the finite sets of graphs are subgraph free.

Lemma 6.1.6 *If F_1 and F_2 are subgraph free, then $\underline{sub}(F_1) = \underline{sub}(F_2)$ if and only if $F_1 = F_2$.*

Proof. Assume that F_1 and F_2 are given and are subgraph free. Let $\underline{sub}(F_1) = \underline{sub}(F_2)$ and suppose $F_1 \neq F_2$. Then there is a graph H that is in one of the sets, say $H \in F_1$, but not in the other, i.e., $H \notin F_2$. Since $H \in \underline{sub}(F_1)$ implies that $H \in \underline{sub}(F_2)$, there is $K \in F_2$ such that $H \preceq K$. Since $K \in \underline{sub}(F_1)$, there is $L \in F_1$ such that $K \leq L$. Then, $H \preceq L$ implies that F_1 is not subgraph free, which contradicts the conditions of the lemma. Therefore, $F_1 = F_2$. ■

The above lemma can, also, be observed through the $G_{\mathcal{U}}$ graph. The only vertices that will not change $\underline{sub}(F)$ when added to it are the vertices on the paths from Λ to the vertices of F . If F is remain subgraph free, such vertices cannot be added to F . Thus, any vertices added to F will change $\underline{sub}(F)$.

Some subgraph free forbidders can be reduced even further as shown in the following example.

Example 6.1.7 Let $F = \{K_{1,3}, C_3, C_4\}$ and $F' = \{C_3, C_4\}$ as in Example 5.2.7. Then, $F \subseteq \underline{sub}(G)$ if and only if $F' \subseteq \underline{sub}(G)$. Hence, F can be reduced to F' .

The above forbidding F could be reduced because every connecting graph of F' contains $K_{1,3}$ as a subgraph. Consider another example.

Example 6.1.8 Let $F = \{K_{1,4}, K_4, C_5\}$ and $F' = \{K_4, C_5\}$. Notice that every connecting graph of F' has $K_{1,4}$ as a subgraph, hence F can be reduced to F' .

In the graph G_U , this can be observed by noticing that for every S such that there is a path from K_4 to S and a path from C_5 to S , there exists a path from $K_{1,4}$ to S .

The next lemma generalizes this reduction.

Lemma 6.1.9 Let \mathcal{F} be given and let $F \in \mathcal{F}$ be such that there is an $H \in F$ with $H \leq S$ for every $S \in C(F')$, where $F' = F \setminus \{H\}$. Then $\mathcal{F} \sim (\mathcal{F} \setminus \{F\}) \cup \{F'\}$.

Proof. Let $\mathcal{F}' = (\mathcal{F} \setminus \{F\}) \cup \{F'\}$. It is obvious that $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F})$. Let $G \underline{con} \mathcal{F}$. If $K \not\leq G$ for some $K \in F$ with $K \neq H$, then $G \underline{con} F'$. Otherwise, $H \not\leq G$, which implies that $G \notin C(F')$. Hence, $G \underline{con} F'$. Therefore, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$ and $\mathcal{F} \sim \mathcal{F}'$. ■

The above lemma, also, follows from Proposition 5.2.8. Let $G \underline{con} F$. If $G \underline{ncon} F'$ then $G \in C(F')$ and there is a $S \in C_{min}(F')$ such that $S \leq G$, but from Proposition 5.2.8 it follows that $S \in C_{min}(F)$ which implies that $F \subseteq \underline{sub}(G)$. This contradicts the fact that $G \underline{con} F$. Therefore, $G \underline{con} F'$.

The reductions from Lemma 6.1.9 can be observed through the G_U graph. Consider the graphs from F as vertices in G_U . If for some $H \in F$ it holds that for all S for which there is a path from all $K \in F \setminus \{H\}$ to S , there is, also, a path from H to S , then H can be removed from F without changing the forbidding family.

A forbidding for which reductions of the type in Lemma 6.1.9 are no longer possible is called *connecting free*.

Definition 6.1.10 A forbiddler F is called *connecting free* if the set F is connecting free (i.e., if for every graph $H \in F$ there is a connecting graph $S \in C(F \setminus \{H\})$, such that $H \not\preceq S$). A forbidding set is called *connecting free* if all of its forbidders are connecting free.

In terms of the graph G_U , a connecting graph S of a forbiddler F is a vertex S in $V(G_U)$, such that there is a path from each $H \in F$ to S . A forbiddler F is connecting free, if for every $H \in F$ there is a vertex S , such that for every $K \in F \setminus \{S\}$, there is a path from K to S , but there is no path from H to S .

Proposition 6.1.11 *If a forbidding set is connecting free, then it is subgraph free, but the converse does not hold.*

Proof. Let \mathcal{F} be a connecting free forbidding set. Let $F \in \mathcal{F}$ and $H, K \in F$. Since F is connecting free, it follows that there is a connecting graph $S \in C(F \setminus \{H\})$ such that $H \not\preceq S$, which implies that $H \not\preceq K$. Similarly, $K \not\preceq S$. Hence, \mathcal{F} is subgraph free. Example 6.1.7 shows that the converse does not hold. ■

The alternate proof below employs the G_U graph.

Proof. Assume that \mathcal{F} is connecting free. Let $F \in \mathcal{F}$ and $H, K \in F$. If there is a path from the vertex K to the vertex H in the graph G_U , then there will be a path from H to every connecting graph $S \in C(F \setminus \{K\})$. Consequently, there will be a path from K to S for every such S , which contradicts the fact that F is connecting free. Therefore, F is subgraph free. Example 6.1.7 shows that the converse does not hold. ■

The above proposition, also, follows from Proposition 5.2.12. If \mathcal{F} is connecting free, then every $F \in \mathcal{F}$ is connecting free and by Proposition 5.2.12 F is subgraph free. Hence, \mathcal{F} is subgraph free.

Proposition 6.1.12 *Let F and F' be two forbidders. Then, $C_{min}(F) = C_{min}(F')$ if and only if $C(F) = C(F')$.*

Proof. Obviously, if $C(F) = C(F')$, then $C_{min}(F) = C_{min}(F')$. Let $C_{min}(F) = C_{min}(F')$ and $G \in C(F)$. Then, there is an $S \in C_{min}(F)$ such that $S \leq G$. Hence, $S \in C_{min}(F')$. Since $S \underline{ncon} F'$ it follows that $G \underline{ncon} F'$, i.e., $G \in C(F')$. Similarly, $C(F') \subseteq C(F)$. Hence, $C(F) = C(F')$. ■

Proposition 6.1.13 *Let F and F' be two forbidders such that $C(F) = C(F')$. Then, $F \sim F'$.*

Proof. Follows from the fact that $G \underline{ncon} F$ if and only if $G \underline{ncon} F'$. ■

The following lemma shows that given a forbider there exists an equivalent forbider that is connecting free.

Lemma 6.1.14 *Let F be a forbider. Then there exists a connecting free F_{free} such that $F_{free} \subseteq F$ and $F_{free} \sim F$.*

Proof. Let F be a forbider. If F is connecting free, the lemma follows. Otherwise, there is a $H \in F$ such that $H \leq S$ for every $S \in C(F)$. Consider $F_1 = F \setminus \{H\}$. If F_1 is connecting free, let $F_1 = F_{free}$. Otherwise, there is a $H_1 \in F_1$ such that $H_1 \leq S$ for every $S \in C(F_1)$. Continue this way and consider the sequence $F = F_0, F_1, \dots$. Since F is finite, eventually an F_k is reached that is connecting free. Note that $F_0 \supsetneq F_1 \supsetneq \dots \supsetneq F_k$ and $F_{i+1} = F_i \setminus \{H_i\}$. By Proposition 5.2.8 $C_{min}(F_i) = C_{min}(F_{i+1})$ for every i so $C_{min}(F) = C_{min}(F_{free})$. Then, by Proposition 6.1.12 and Proposition 6.1.13 it follows that $F \sim F_{free}$. ■

The corollary below follows from Lemma 6.1.14.

Corollary 6.1.15 *Given a forbidding set \mathcal{F} and a forbider $F \in \mathcal{F}$. If F' and F'' are two connecting free forbidders obtained from F as in Lemma 6.1.14, then $F' \sim F''$.*

The following theorem states that connecting free is a normal form.

Theorem 6.1.16 *For every forbidding set there exists an equivalent connecting free forbidding set.*

Proof. Let \mathcal{F} be given. From Lemma 6.1.14, for every $F \in \mathcal{F}$ there is F_{free} such that F_{free} is connecting free and $G \underline{con} F$ if and only if $G \underline{con} F_{free}$. Define $\mathcal{F}' = \{F_{free} \mid F \in \mathcal{F}\}$. It is clear that \mathcal{F}' is connecting free and that $\mathcal{F} \sim \mathcal{F}'$. ■

Another way to reduce redundancy is to remove entire forbidders that are superfluous. Consider the following example.

Example 6.1.17 Let $\mathcal{F} = \{\{C_3\}, \{N_{1,1,1}\}, \{N_{2,2,2}\}, \dots\}$. If a graph does not have a 3-cycle as a subgraph, then it does not have any of $N_{i,i,i}$ for $i \geq 1$ as a subgraph. Thus $\mathcal{F} \sim \mathcal{F}'$ where $\mathcal{F}' = \{\{C_3\}\}$.

The key in the preceding example is that the set of subgraphs of every forbinder contains C_3 and more precisely the set of $\underline{sub}(\{C_3\})$.

The next lemma is a generalization of example 6.1.17.

Lemma 6.1.18 *Let \mathcal{F} be a forbidding set, and let $F_1, F_2 \in \mathcal{F}$ with $F_1 \neq F_2$. If $\underline{sub}(F_1) \subseteq \underline{sub}(F_2)$, then $\mathcal{F} \sim \mathcal{F} \setminus \{F_2\}$.*

Proof. It is clear that $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F} \setminus \{F_2\})$. Suppose $G \underline{con}(\mathcal{F} \setminus \{F_2\})$. Then $F_1 \not\subseteq \underline{sub}(G)$, which implies that $\underline{sub}(F_1) \not\subseteq \underline{sub}(G)$. Since $\underline{sub}(F_1) \subseteq \underline{sub}(F_2)$ it holds that $\underline{sub}(F_2) \not\subseteq \underline{sub}(G)$. Consequently, $F_2 \not\subseteq \underline{sub}(G)$. Hence, $G \in \mathcal{L}(\mathcal{F})$ and $\mathcal{L}(\mathcal{F} \setminus \{F_2\}) \subseteq \mathcal{L}(\mathcal{F})$. ■

The following lemma is a generalization of lemma 6.1.18 which allows the removal of a (possibly infinite) set of forbidders, rather than just one forbinder.

Lemma 6.1.19 *Let \mathcal{F}' and \mathcal{F} be forbidding sets with $\mathcal{F}' \subseteq \mathcal{F}$, such that for each $F \in \mathcal{F}$ there is a $F' \in \mathcal{F}'$ with $\underline{sub}(F') \subseteq \underline{sub}(F)$. Then $\mathcal{F}' \sim \mathcal{F}$.*

Proof. Obviously, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Let $G \in \mathcal{L}(\mathcal{F}')$. Let $F \in \mathcal{F}$. If $F \in \mathcal{F}'$ then $G \underline{\text{con}} F$. Otherwise, there is a $F' \in \mathcal{F}'$ such that $F' \neq F$ and $\underline{\text{sub}}(F') \subseteq \underline{\text{sub}}(F)$. Since $F' \not\subseteq \underline{\text{sub}}(G)$ it follows that $\underline{\text{sub}}(F') \not\subseteq \underline{\text{sub}}(G)$. Consequently, $\underline{\text{sub}}(F) \not\subseteq \underline{\text{sub}}(G)$ and $F \not\subseteq \underline{\text{sub}}(G)$. Hence, $G \underline{\text{con}} F$ and $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F})$. ■

Definition 6.1.20 A forbidding set \mathcal{F} is called *subgraph incomparable* if for any two forbidders $F_1, F_2 \in \mathcal{F}$ it holds that $\underline{\text{sub}}(F_1) \not\subseteq \underline{\text{sub}}(F_2)$ and $\underline{\text{sub}}(F_2) \not\subseteq \underline{\text{sub}}(F_1)$.

Lemma 6.1.21 *For every forbidding set there is an equivalent subgraph incomparable forbidding set.*

Proof. Let \mathcal{F} be a forbidding set. Because of lemma 6.1.5 assume that \mathcal{F} is subgraph free. Define $\mathcal{F}' = \{F \in \mathcal{F} \mid \text{there is no } \hat{F} \in \mathcal{F} \text{ such that } \hat{F} \neq F \text{ and } \underline{\text{sub}}(\hat{F}) \subseteq \underline{\text{sub}}(F)\}$. The subgraph free condition is necessary to ensure that forbidders that are different ($F_1 \neq F_2$), but have the same subgraphs i.e., $\underline{\text{sub}}(F_1) = \underline{\text{sub}}(F_2)$ are represented by their maximal forbidding F_{\max} . It is clear that the forbidding set \mathcal{F}' is subgraph incomparable.

Notice that $\mathcal{F}' \subseteq \mathcal{F}$ by definition. The rest of the proof shows that for every $F \in \mathcal{F}$ there exists a $F' \in \mathcal{F}'$ with $\underline{\text{sub}}(F') \subseteq \underline{\text{sub}}(F)$. Let $F \in \mathcal{F}$. If $F \in \mathcal{F}'$, the claim is true. If not, it has been excluded from \mathcal{F}' by definition, i.e., there is some $F_1 \in \mathcal{F}$ such that $\underline{\text{sub}}(F_1) \subseteq \underline{\text{sub}}(F)$ and $F_1 \neq F$. Again, either $F_1 \in \mathcal{F}'$ or there is $F_2 \in \mathcal{F}$ with $\underline{\text{sub}}(F_2) \subseteq \underline{\text{sub}}(F_1)$ and $F_2 \neq F_1$. Continue this way and consider the sequence of forbidders F_0, F_1, F_2, \dots , where $F_0 = F$, $F_{i+1} \neq F_i$, and $\underline{\text{sub}}(F_{i+1}) \subseteq \underline{\text{sub}}(F_i)$, for each $i \geq 0$. Since \mathcal{F} is subgraph free, $F_{i+1} \neq F_i$ implies that $\underline{\text{sub}}(F_{i+1}) \neq \underline{\text{sub}}(F_i)$ (see Proposition 6.1.6), therefore, $\underline{\text{sub}}(F_{i+1}) \subset \underline{\text{sub}}(F_i)$. Since F is finite, i.e., it has a finite number of finite connected graphs, then $\underline{\text{sub}}(F)$ is finite, as well, which means that there are finitely many strict subsets of it. Therefore, in a finite number of steps, a forbidding F_k is reached, such that $F_k \in \mathcal{F}'$ and $\underline{\text{sub}}(F_k) \subseteq \underline{\text{sub}}(F)$. Thus, the conditions of 6.1.19 are satisfied and it follows that $\mathcal{F}' \sim \mathcal{F}$. ■

The next example discusses a forbidding set that is connecting free (respectively subgraph free) and subgraph incomparable, but it is not necessarily minimal, i.e., the forbidding set contains redundant forbidders.

Example 6.1.22 Let $\mathcal{F} = \{\{K_{1,3}\}, \{C_3, C_4\}\}$ and consider $\mathcal{F}' = \{\{K_{1,3}\}\}$. To see that $\mathcal{F} \sim \mathcal{F}'$ note that every graph that contains both C_3 and C_4 as subgraphs contains $K_{1,3}$, as well. Hence, $G \underline{\text{con}} \{K_{1,3}\}$ implies $G \underline{\text{con}} \{C_3, C_4\}$.

The preceding example shows that the notion of subgraph incomparable needs to be generalized to include not only the set of subgraphs of a forbinder, but also, the subgraphs of connecting graphs of that forbinder.

Lemma 6.1.23 *Let \mathcal{F} be a forbidding set and $F_1, F_2 \in \mathcal{F}$ such that $C(F_2) \subseteq C(F_1)$. Then $\mathcal{F} \sim (\mathcal{F} \setminus \{F_2\})$.*

Proof. Let \mathcal{F}, F_1 , and F_2 be as in the conditions of the lemma. It is obvious that $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F} \setminus \{F_2\})$. Let $G \in \mathcal{L}(\mathcal{F} \setminus \{F_2\})$. Since $G \underline{\text{con}} F_1$, it follows that $G \notin C(F_2)$, i.e., $G \underline{\text{con}} F_2$. ■

The above lemma is generalized below to allow removal of possibly infinitely many forbidders.

Lemma 6.1.24 *Let \mathcal{F} and \mathcal{F}' be forbidding sets with $\mathcal{F}' \subseteq \mathcal{F}$ such that for each $F \in \mathcal{F}$ there is a $F' \in \mathcal{F}'$ such that $C(F) \subseteq C(F')$. Then $\mathcal{F}' \sim \mathcal{F}$.*

Proof. Obviously, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Let $G \in \mathcal{L}(\mathcal{F}')$ and $F \in \mathcal{F}$. Since $G \notin C(F')$ it follows that $G \notin C(F)$. Hence, $G \underline{\text{con}} F$ and the lemma follows. ■

Recall that for a graph G and a forbinder F , either $G \underline{\text{con}} F$ or $G \in C(F)$. Two forbidders F_1 and F_2 are equivalent if and only if $G \underline{\text{con}} F_1$ implies $G \underline{\text{con}} F_2$ and vice versa. Hence, the following remark.

Remark 6.1.25 Let \mathcal{F} be a forbidding set and $F_1, F_2 \in \mathcal{F}$. Then, F_1 and F_2 are equivalent if and only if $C(F_1) = C(F_2)$.

So, if F_1 and F_2 are not equivalent, either there exists a connecting graph $S \in C(F_1)$ such that $F_2 \not\subseteq \underline{\text{sub}}(S)$ or there exists a connecting graph $T \in C(F_2)$ such that $F_1 \not\subseteq \underline{\text{sub}}(T)$.

Definition 6.1.26 Two forbidders F_1 and F_2 are *connecting incomparable* if there exists a connecting graph $S \in C(F_1)$ such that $S \notin C(F_2)$ and there exists a connecting graph $T \in C(F_2)$ such that $T \notin C(F_1)$. A forbidding set \mathcal{F} is *connecting incomparable* if every pair of forbidders in it is connecting incomparable.

In terms of the $G_{\mathcal{U}}$ graph, F_1 and F_2 are *connecting incomparable* if there is a vertex S in $G_{\mathcal{U}}$ with a path from every $H \in F_1$ to S , such that there is a $K \in F_2$ from which there is no path to S and vice versa.

Proposition 6.1.27 *If a forbidding set is connecting incomparable, then it is subgraph incomparable, but the converse does not hold.*

Proof. Let \mathcal{F} be connecting incomparable. Let $F_1, F_2 \in \mathcal{F}$ with $F_1 \neq F_2$. Since \mathcal{F} is connecting incomparable, there exists $T \in C(F_2)$, such that $T \notin C(F_1)$. This implies that $F_1 \not\subseteq \underline{\text{sub}}(F_2)$. Hence, $\underline{\text{sub}}(F_1) \not\subseteq \underline{\text{sub}}(F_2)$. Similarly, $\underline{\text{sub}}(F_2) \not\subseteq \underline{\text{sub}}(F_1)$. Thus, \mathcal{F} is subelement incomparable. Example 6.1.22 shows that the converse does not hold. ■

Corollary 6.1.28 *Let \mathcal{F} be a forbidding set and $F, F' \in \mathcal{F}$ with $F \neq F'$. If F and F' are connecting incomparable, then they are not equivalent.*

The following theorem establishes that connecting incomparable is a normal form.

Theorem 6.1.29 *For every forbidding set there exists an equivalent connecting incomparable forbidding set.*

Proof. Let \mathcal{F} be a forbidding set. Because of Theorem 6.1.16 \mathcal{F} is assumed to be connecting (hence subgraph) free. Define $\hat{\mathcal{F}} = \{\hat{F} \in \mathcal{F} \mid \text{there is no } F \in \mathcal{F} \text{ such that } F \not\sim \hat{F} \text{ and } C(\hat{F}) \subseteq C(F)\}$. The set $\hat{\mathcal{F}}$ may contain equivalent forbidders. Divide (\hat{F}) into equivalent classes and let \mathcal{F}' contain exactly one forbiddler from each equivalent class. It is clear that \mathcal{F}' is connecting incomparable.

Notice that $\mathcal{F}' \subseteq \mathcal{F}$ by definition. The rest of the proof shows that for every $F \in \mathcal{F}$ there exists a $F' \in \mathcal{F}'$ with $C(F) \subseteq C(F')$. Let $F \in \mathcal{F}$. If $F \in \mathcal{F}'$, the claim is true. If not, it has been excluded from \mathcal{F}' by definition. Either there is some $\tilde{F} \in \mathcal{F}'$ such that $F \sim \tilde{F}$, in which case the claim is true or F has been excluded from (\hat{F}) in which case there is a F_1 such that $C(F) \subseteq C(F_1)$ and $F \not\sim F_1$. If $F_1 \in \mathcal{F}'$ the claim is true. If not, either $F_1 \sim F_2$ for some $F_2 \in \mathcal{F}'$ in which case the claim is true or there is $F_2 \not\sim F_1$ with $C(F_1) \subseteq C(F_2)$. Continue this way and consider the sequence of forbidders F_0, F_1, F_2, \dots where $F_0 = F$, $F_{i+1} \not\sim F_i$, and for each $i \geq 0$ it holds that $C(F_i) \subseteq C(F_{i+1})$. Since $F_i \not\sim F_{i+1}$ from Remark 6.1.25 it follows that $C(F_i) \neq C(F_{i+1})$, therefore, $C(F_i) \subset C(F_{i+1})$. Let $S \in C(F)$. If the sequence F_0, F_1, F_2, \dots is infinite, it follows that S is a connecting graph of infinitely many forbidders, which contradicts the fact that S is finite. Therefore, a $F_k \in \mathcal{F}'$ is reached such that $C(F) \subseteq C(F_k)$. Thus, the conditions of Lemma 6.1.24 are satisfied and it follows that $\mathcal{F}_{min} \sim \mathcal{F}$. ■

Because Theorem 6.1.21 shows that connecting free and connecting incomparable is a normal form consider the definition below.

Definition 6.1.30 The forbidding set \mathcal{F} is in *reduced normal form* if it is both connecting free and connecting incomparable.

Proposition 6.1.31 *Let \mathcal{F} be a forbidding set. For every $F \in \mathcal{F}$ choose one connecting graph $S_F \in C(F)$ and consider $\mathcal{F}' = \{\{S_F\} \mid F \in \mathcal{F}\}$. Then, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$.*

Proof. Let $G \underline{\text{con}} \mathcal{F}$ and let $\{S_F\} \in \mathcal{F}'$. Then, S_F is a connecting graph for some $F \in \mathcal{F}$. Since $F \not\subseteq \underline{\text{sub}}(G)$, it holds that $\{S_F\} \not\subseteq \underline{\text{sub}}(G)$. Hence, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. ■

The converse, however, is not necessarily true even if \mathcal{F} is in reduced normal form. Consider $\mathcal{F} = \{\{C_3, C_4\}\}$ and $D_4 \in C(\{C_3, C_4\})$. Let $\mathcal{F}' = \{\{D_4\}\}$. Then, the graph $G = S_{C_3P_4C_4}$ from Example 5.2.3 and Figure 5.1 is such that $G \underline{\text{con}} \mathcal{F}'$ but $G \underline{\text{ncon}} \mathcal{F}$. The following proposition considers a special case.

Proposition 6.1.32 *Let \mathcal{F} be a forbidding set. For every $F \in \mathcal{F}$ construct the forbidding set $\mathcal{F}_F = \{\{S\} \mid S \in C_{\min}(F)\}$ and consider $\mathcal{F}' = \cup_{F \in \mathcal{F}} \mathcal{F}_F$. Then, $\mathcal{F} \sim \mathcal{F}'$.*

Proof. Let $G \underline{\text{con}} \mathcal{F}$ then for every $\{S\} \in \mathcal{F}'$ there is a $F \in \mathcal{F}$ such that $F \subseteq \underline{\text{sub}}(S)$. Since $F \not\subseteq \underline{\text{sub}}(G)$, it follows that $\{S\} \not\subseteq \underline{\text{sub}}(G)$. Hence, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Conversely, let $G \underline{\text{con}} \mathcal{F}'$ and let $F \in \mathcal{F}$. Suppose $F \subseteq \underline{\text{sub}}(G)$. Then, $G \in C(F)$ and there is a $S \in C_{\min}(F)$ such that $S \leq G$. This contradicts the fact that $G \underline{\text{con}} \mathcal{F}'$. Hence, $F \not\subseteq \underline{\text{sub}}(G)$ and $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F})$. Consequently, $\mathcal{F} \sim \mathcal{F}'$. ■

Proposition 6.1.31 now follows from the above proposition. Proposition 6.1.32 proves that every forbidding set is equivalent to a forbidding set consisting of singleton forbidders only. Hence, the following theorem.

Theorem 6.1.33 *Every forbidding set is equivalent to a strict forbidding set.*

As Example 5.2.3 states, even a simple forbider $F = \{C_3, C_4\}$ has an infinite number of minimal connecting graphs and dealing with such a strict forbidding set will be cumbersome.

Remark 6.1.34 Any strict forbidding set is connecting (subgraph) free. Also, connecting incomparable is equivalent to subgraph incomparable for such a set.

Proposition 6.1.35 *If \mathcal{F} consists of singleton forbidders only and it is a connecting (subgraph) incomparable forbidding set, then, it is minimal, i.e., $\mathcal{L}(\mathcal{F}) \subset \mathcal{L}(\mathcal{F} \setminus \{F\})$ for any $F \in \mathcal{F}$.*

Proof. Let $F = \{K\} \in \mathcal{F}$. It is clear that $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F} \setminus \{F\})$. Obviously, $K \notin \mathcal{L}(\mathcal{F})$. If $H \neq F$, then $H \not\subseteq \underline{\text{sub}}(K)$ since \mathcal{F} is subgraph incomparable. Hence, $\mathcal{L}(\mathcal{F}) \neq \mathcal{L}(\mathcal{F} \setminus \{F\})$. ■

Corollary 6.1.36 *For every forbidding set there is a an equivalent minimal strict forbidding set.*

Proof. Let \mathcal{F} be given and construct \mathcal{F}' as in Theorem 6.1.33. Then, from \mathcal{F}' construct a connecting (subgraph) incomparable forbidding set \mathcal{F}'' as in Theorem 6.1.29. Then, by Proposition 6.1.35, \mathcal{F}'' is minimal, i.e., $\mathcal{L}(\mathcal{F}'') \subset \mathcal{L}(\mathcal{F}'' \setminus \{F\})$ for any $F \in \mathcal{F}''$. ■

Lemma 6.1.37 *Let \mathcal{F} be a strict forbidding set and \mathcal{F}_1 and \mathcal{F}_2 be two forbidding sets obtained as \mathcal{F}'' in Corollary 6.1.36. Then $\mathcal{F}_1 = \mathcal{F}_2$.*

Proof. Let $F \in \mathcal{F}_1$. Since $\mathcal{F}_1 \sim \mathcal{F}_2$, it follows that there is a $H \in \mathcal{F}_2$ such that $H \subseteq \underline{\text{sub}}(F)$, hence, $\underline{\text{sub}}(H) \subseteq \underline{\text{sub}}(F)$. Similarly, there is a $K \in \mathcal{F}_1$ such that $K \subseteq \underline{\text{sub}}(H)$ which implies that $\underline{\text{sub}}(K) \subseteq \underline{\text{sub}}(H)$. Since both K and F are in \mathcal{F}_1 and $\underline{\text{sub}}(K) \subseteq \underline{\text{sub}}(F)$ by Lemma 6.1.29 it follows that $\underline{\text{sub}}(K) = \underline{\text{sub}}(F)$. Since both F and K are singletons, it follows that $K = F$ which implies that $F \in \mathcal{F}_2$. Thus, $\mathcal{F}_1 \subseteq \mathcal{F}_2$. Similarly, $\mathcal{F}_2 \subseteq \mathcal{F}_1$. Consequently, $\mathcal{F}_1 = \mathcal{F}_2$. ■

Theorem 6.1.38 *For every forbidding set there is a equivalent unique minimal strict forbidding set in reduced normal form.*

Proof. Let \mathcal{F} be given and let \mathcal{F}' be the strict forbidding set constructed as in Theorem 6.1.33 consisting of singleton forbidders of all minimal connecting graphs

of the forbidders in \mathcal{F} . Construct a connecting (subgraph) incomparable set $\hat{\mathcal{F}}$ from \mathcal{F}' . Then, $\hat{\mathcal{F}}$ is in reduced normal form. From Corollary 6.1.36 it is minimal and Lemma 6.1.37 establishes that it is unique. ■

6.2 Normal Forms for Enforcing Sets

In this section, redundancy within enforcing sets is investigated. The first remark illustrates an obvious redundancy.

Remark 6.2.1 Let \mathcal{E} be an enforcing set and $(X, Y) \in \mathcal{E}$. Then, $\hat{Y} \underline{sat}(X, Y)$ for every $\hat{Y} \in Y$.

Note that if an enforcing set \mathcal{E} contains a brute enforcer (\emptyset, Y) , then if $G \in \mathcal{L}(\mathcal{E})$, it follows that G has at least one vertex, hence $G \underline{sat}(\emptyset, \{\Lambda\})$. The following remark states that formally.

Remark 6.2.2 If \mathcal{E} contains a brute enforcer (\emptyset, Y) with $Y \neq \{\Lambda\}$ and $(\emptyset, \{\Lambda\})$, then $(\emptyset, \{\Lambda\})$ is redundant.

Example 6.2.3 Let \mathcal{E} be an enforcing set, which contains both $E_1 = (P_3, C_3)$ and $E_2 = (P_3, \{C_3, C_4\})$ as enforcers. It is easy to see that if a graph G satisfies E_1 then it, also, satisfies E_2 . Thus, the enforcer E_2 is redundant and can be discarded.

The next proposition generalizes this type of redundancy.

Proposition 6.2.4 Let (X, Y') and (X, Y'') are enforcers in an enforcing set \mathcal{E} with $Y' \subseteq Y''$. Then $\mathcal{E} \sim \mathcal{E}'$, where $\mathcal{E}' = \mathcal{E} \setminus \{(X, Y'')\}$.

Proof. Obviously, $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$. Let G be in $\mathcal{L}(\mathcal{E}')$. If $X \not\leq G$ then $G \underline{sat}(X, Y'')$. Suppose $X \leq G$. Then there is $\hat{Y} \in Y'$ such that $X \leq \hat{Y} \leq G$. Since $\hat{Y} \in Y''$ it follows that $G \underline{sat}(X, Y'')$ and $G \in \mathcal{L}(\mathcal{E}')$. ■

The above result is extended to removing a subset of redundant enforcers of this type instead of just one such enforcer.

Proposition 6.2.5 *Let \mathcal{E} be an enforcing set and (X, Y) an enforcer in \mathcal{E} . Then, $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E}' = \mathcal{E} \setminus \mathcal{E}''$ with $\mathcal{E}'' = \{(X'', Y'') \mid X = X'' \text{ and } Y \subseteq Y''\}$.*

Proof. Obviously, $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$. Let $G \in \mathcal{L}(\mathcal{E}')$ and $(X, Y'') \in \mathcal{E}''$. If $X \not\leq G$, then $G \underline{\text{sat}}(X, Y'')$ trivially. Suppose, $X \leq G$, then since $G \underline{\text{sat}}(X, Y)$, there is a $\hat{Y} \in Y$ such that $X \leq \hat{Y} \leq G$. Hence, $G \underline{\text{sat}}(X, Y'')$ and $G \underline{\text{sat}} \mathcal{E}''$. ■

A graph in Y can be a subgraph of another graph in Y , which in some cases leads to the type of redundancy examined next.

Proposition 6.2.6 *Let (X, Y) be an enforcer in an enforcing set \mathcal{E} , such that if Y_i and Y_j are in Y with $Y_i \leq Y_j$ and whenever $X \leq Y_j$, it holds that $X \leq Y_i \leq Y_j$. Then $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E}' = (\mathcal{E} \setminus \{(X, Y)\}) \cup \{(X, Y')\}$ with $Y' = Y \setminus \{Y_j\}$.*

Proof. Let $G \underline{\text{sat}} \mathcal{E}$. It is sufficient to show that $G \underline{\text{sat}}(X, Y')$. If $X \not\leq G$, then $G \underline{\text{sat}}(X, Y')$ trivially. Suppose $X \leq G$. Since $G \underline{\text{sat}}(X, Y)$, there is $Y_k \in Y$ such that $X \leq Y_k \leq G$. If $Y_k \neq Y_j$, then $G \underline{\text{sat}}(X, Y')$. Otherwise, $X \leq Y_j \leq G$. This implies that $X \leq Y_i \leq G$. Consequently, $G \underline{\text{sat}}(X, Y')$. Hence, $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$. The converse is trivial, since $Y' \subseteq Y$. ■

Example 6.2.7 Let $\mathcal{E} = \{(P_3, \{C_4, K_5\})\}$. Then, by the above proposition, $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E}' = \{(P_3, C_4)\}$.

The following example shows that even in a very restrictive case, it is still very difficult to define redundancy for enforcers with different first components.

Example 6.2.8 Let \mathcal{E} be given. Let $(X, Y), (X', Y') \in \mathcal{E}$ such that $X \leq X'$, $X' \leq \hat{Y}$ for every $\hat{Y} \in Y$, and for every $\hat{Y}' \in Y'$ there is a $\hat{Y} \in Y$ such that $\hat{Y}' \leq \hat{Y}$. Then $\{(X', Y')\}$ is not necessarily redundant. Consider $\mathcal{E} = \{(X, Y)\}, (\{X', Y'\})$ where $X = C_3$, $Y = \{\hat{Y}\}$, and \hat{Y} and X' are as in Figure 6.1. Then the graph G from the same figure satisfies the first enforcer, but does not satisfy \mathcal{E} .

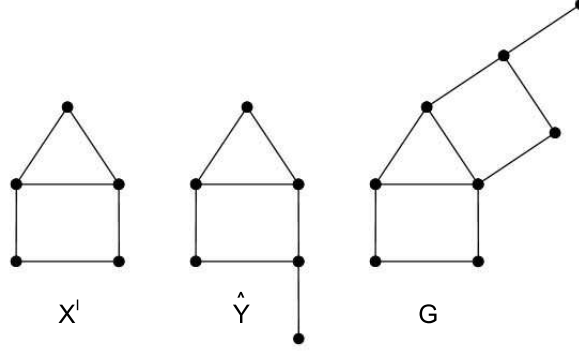


Figure 6.1: Graphs related to \mathcal{E} in Example 6.2.8

Let \mathcal{E} be an enforcing set. Define $\mathcal{E}^{(1)} = \{X \mid (X, Y) \in \mathcal{E}\}$ and $\mathcal{E}^{(2)} = \{Y \mid (X, Y) \in \mathcal{E}\}$.

Definition 6.2.9 Let $X \in \mathcal{E}^{(1)}$. A graph $g(X)$ is *generated* by X if the following two conditions hold:

- (i) $X \leq g(X)$
- (ii) $g(X) \underline{\text{sat}} (X', Y')$ for every $(X', Y') \in \mathcal{E}$.

A generated graph $g_m(X)$ is called *minimal*, if no proper subgraph of it is a generated graph.

Let \mathcal{E} be an enforcing set and let $X \in \mathcal{E}^{(1)}$. Denote the family of generated graphs of X with respect to \mathcal{E} with $G_X^\mathcal{E}$ or simply G_X when \mathcal{E} is understood. The family of minimal generated graphs of X with respect to \mathcal{E} is denoted by $M_X^\mathcal{E}$ or simply M_X when \mathcal{E} is understood. Let $\mathcal{M}(\mathcal{E}) = \cup_{X \in \mathcal{E}^{(1)}} M_X$.

The next proposition follows directly from the definition of generated graphs.

Proposition 6.2.10 *Let \mathcal{E} be an enforcing set and $X \in \mathcal{E}^{(1)}$, then*

- (i) *For every graph G such that $G \underline{\text{sat}} \mathcal{E}$, $X \leq G$ implies $G \in G_X$.*
- (ii) *$g(X) \in \mathcal{L}(\mathcal{E})$ if and only if $g(X)$ is finite.*

(iii) If there exists an infinite $g_m(X)$, $G \text{ sat } \mathcal{E}$ implies $X \not\leq G$.

Example 6.2.11 (a) Consider $\mathcal{E} = \{(P_3, \{C_3\})\}$. The minimal generated graph of P_3 is (isomorphic to) C_3 . It is finite and it satisfies \mathcal{E} . K_4 is a generated graph, but not minimal since $C_3 \leq K_4$.

(b) Let $\mathcal{E}' = \{(P_3, \{P_4\}), (P_4, \{P_5\}), \dots, (P_n, \{P_{n+1}\}), \dots\}$. The minimal generated graph $g_m(P_3)$ is infinite and no graph in $\mathcal{L}(\mathcal{E})$ contains P_3 as a subgraph. Thus, $\mathcal{L}(\mathcal{E}) = \{\emptyset, \Lambda, P_2\}$.

Example 6.2.12 Let $\mathcal{E} = \{(P_3, \{C_3\})\}$ and $\mathcal{E}' = \{(P_3, \{C_3\}), (P_4, \{C_4\}), \dots, (P_n, \{C_n\}), \dots\}$. It is clear that $\mathcal{L}(\mathcal{E}') \subseteq \mathcal{L}(\mathcal{E})$. Let $G \in \mathcal{L}(\mathcal{E})$. Let $P_n \leq G$ with $P_n = v_1, v_2, \dots, v_n$. Since the path $v_1 v_2 v_3$ demands that $\{v_1, v_3\} \in E(G)$ and $v_1 v_3 v_4$ demands that $\{v_1, v_4\} \in E(G)$, it follows that $\{v_1, v_{n-1}\}$ and $\{v_1, v_n\} \in E(G)$, which proves that $\mathcal{E} \sim \mathcal{E}'$.

The above example shows that an infinite enforcing set is equivalent to a finite enforcing set. This raises the question whether there are infinite enforcing sets that are not equivalent to any finite enforcing set and the following two examples and proposition provide an affirmative answer.

Example 6.2.13 There is no finite enforcing set \mathcal{E} such that $\mathcal{L}(\mathcal{E}) = \{\emptyset, \Lambda, P_2\}$. Suppose there is an \mathcal{E} for which $\mathcal{L}(\mathcal{E}) = \{\emptyset, \Lambda, P_2\}$. Then P_3 is not in $\mathcal{L}(\mathcal{E})$ (as well as any graph which has P_3 as a subgraph). P_3 needs to be “excluded” from $\mathcal{L}(\mathcal{E})$. Hence, an enforcer (P_3, Y) is needed and all generated graphs $g(P_3)$ need to be infinite, but this would mean that \mathcal{E} is infinite.

Example 6.2.14 Consider $\tilde{\mathcal{E}} = \{(P_3, \{C_3\}), (P_3, \{C_4\}), \dots, (P_3, \{C_n\}), \dots\}$. Then there is no finite enforcing set \mathcal{E} , such that $\mathcal{E} \sim \tilde{\mathcal{E}}$. First, observe that $\mathcal{L}(\tilde{\mathcal{E}}) = \{\emptyset, \Lambda, P_2\}$. To show this consider $G \in \mathcal{L}(\tilde{\mathcal{E}})$. If $P_3 \leq G$ it follows that for every n $P_3 \leq C_n \leq G$ which contradicts the fact that G is finite. Hence, $P_3 \not\leq G$. Therefore, G can only be \emptyset, Λ , or P_2 . Then, by the above example the claim follows.

The above two examples prove the following proposition.

Proposition 6.2.15 *There are infinite enforcing sets which do not have a finite equivalent.*

Definition 6.2.16 An enforcing set \mathcal{E} is called *finitary* if for all $X \in \mathcal{E}^{(1)}$ there is a finite number of enforcers (X, Y_i) in \mathcal{E} .

Example 6.2.17 The infinite and not finitary enforcing set $\hat{\mathcal{E}} = \{(P_3, \{C_3, C_4\}), (P_3, \{C_3, C_5\}), \dots, (P_3, \{C_3, C_n\}), \dots\}$ defines the class of complete graphs. If $P_3 \not\leq G$ then G is complete. Assume that $P_3 \leq G$. If $P_3 \leq C_3 \not\leq G$ then for any $n \geq 4$, $P_3 \leq C_n \leq G$, which contradicts the fact that G is finite. It follows that $P_3 \leq C_3 \leq G$, which means that $\hat{\mathcal{E}} \sim \mathcal{E}$ for $\mathcal{E} = \{(P_3, C_3)\}$.

This fact shows that there is a non-finitary enforcing set that is equivalent to a finite enforcing set.

From the above examples it follows that $\hat{\mathcal{E}} \sim \mathcal{E}'$, where $\hat{\mathcal{E}}$ is as in Example 6.2.17 and \mathcal{E}' is as in Example 6.2.12.

Definition 6.2.18 Given \mathcal{E} , the enforcer (X', Y') is *redundant* for \mathcal{E} , if there exists an enforcer $(X, Y) \in \mathcal{E}$ with $X \leq X'$ and for every $g(X) \in G_X^{\mathcal{E}'}$ where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$ whenever $X' \leq g(X)$ there exists $\hat{Y} \in Y'$, such that $X' \leq \hat{Y} \leq g(X)$ or $g(X)$ is infinite.

Example 6.2.19 Consider $\mathcal{E} = \{(C_3, N_{1,1,1}), (N_{1,0,0}, N_{1,1,1})\}$. Then, the second enforcer is redundant.

The following proposition shows that redundant enforcers can be erased from the enforcing set.

Proposition 6.2.20 *If (X', Y') is redundant for \mathcal{E} , then $\mathcal{E} \sim \mathcal{E}'$, where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$.*

Proof. It is clear that $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$. Let $G \in \mathcal{L}(\mathcal{E}')$. If $X' \not\leq G$, then $G \in \mathcal{L}(\mathcal{E})$. Assume $X' \leq G$. Then, $X \leq G$ which implies that $G = g(X)$ for some $g(X) \in G_X^{\mathcal{E}'}$.

Since (X', Y') is redundant, every copy of X' in G is “enclosed” in some \hat{Y} from Y' in G . Hence, $G \underline{sat} \mathcal{E}$. ■

The following lemma shows that an infinite finitary enforcing set with finite $\mathcal{M}(\mathcal{E})$ must have an infinite generated graph.

Lemma 6.2.21 *Let \mathcal{E} be infinite and finitary, such that $\mathcal{M}(\mathcal{E})$ is finite. Then there exists an infinite generated set.*

Proof. Since $\mathcal{M}(\mathcal{E})$ is finite, there is a finite number of families of minimal generated graphs M_X . Denote these families by M_1, M_2, \dots, M_k , i.e., $\mathcal{M}(\mathcal{E}) = \bigcup_{i=1}^k M_i$. Since there are infinitely many distinct X 's (due to \mathcal{E} being infinite) and finitely many M_i 's, there must exist at least one M_j such that for infinitely many X 's in $\mathcal{E}^{(1)}$, it holds that $M_X = M_j$. Let $g_m(X) \in M_j$. Since $g_m(X)$ is a generated graph for infinitely many X 's, it follows that $g_m(X)$ contains all these X 's as subgraphs. Hence, $g_m(X)$ is infinite. (In fact all generated graphs in M_j are infinite.) ■

Chapter 7

Forbidding and Enforcing on Partially Ordered Sets

This chapter presents a new way of defining families of subposets or single subposets based on forbidding and enforcing constraints. The first section is a direct generalization of the language fe -systems to posets. In the remaining sections the single subposet fe -families, which were inspired by graph fe -systems are defined. In this case, the forbidding sets generalize the graph forbidding sets, but the enforcing sets are different than these discussed for graphs in that enforcing is “weak” rather than the “strong” version presented in Chapter 5.

In this chapter a partially ordered set is usually denoted by P and the power set of P with $\mathcal{P}(P)$. The order in P is denoted by \leq . A set of elements from P is called a *subposet*. Thus, subposets are elements in $\mathcal{P}(P)$. A partially ordered set does not necessarily contain a smallest element, but in case it does, the smallest element is denoted by λ . In this case, $\lambda \leq p$ for every $p \in P$. A subposet $L \subseteq P$ is a *chain* if for any $p, q \in L$ it holds that either $p \leq q$ or $q \leq p$. A subposet $K \subseteq P$ is an *antichain* if for any $p, q \in K$ with $p \neq q$ neither $p < q$ nor $q < p$ holds. Basic discussion of posets and order can be found in [6, 38].

Definition 7.0.22 Given $p \in P$ define the *shadow* of p as $\underline{sub}(p) = \{q \mid q \leq p\}$. For $L \in \mathcal{P}(P)$ define $\underline{sub}(L) = \cup_{p \in L} \underline{sub}(p)$.

In literature, $\underline{sub}(p)$ is, also, called a *principal ideal*, a *down-set*, or a *lower set* (see for example [38]).

The proofs of some propositions in this chapter depend on the finiteness of $\underline{sub}(p)$. Hence, the following definition.

Definition 7.0.23 An element $p \in P$ is called *finite*, if $\underline{sub}(p)$ is finite. Otherwise, p is infinite.

Some examples of posets include:

- (i) The subsets $\mathcal{P}(S)$ of a set S ordered by inclusion and denoted by $(\mathcal{P}(S), \subseteq)$.
- (ii) The words A^* over a given alphabet A with subword order denoted with (A^*, \underline{sub}) where $u \leq v$ if and only if u is a subword of v ($u \in \underline{sub}(v)$).
- (iii) Graphs with subgraph order (\mathcal{G}, \leq) where \mathcal{G} is the set of simple connected graphs and $H \leq G$ if and only if H is a subgraph of G .
- (iv) Natural numbers with divisibility $(N, |)$ where $p \leq q$ if and only if p divides q ($p \mid q$).

In all of the above examples $\lambda \in P$ and all elements of P are finite. The poset of integers (Z, \leq) does not have a smallest element λ and contains infinite elements.

7.1 *fe*-Families as Sets of Subposets

In this section, the forbidding-enforcing systems defined on formal languages are generalized to partially ordered sets. The partially ordered set (A^*, \underline{sub}) , where A is an alphabet and \underline{sub} is the subword relation between words is used as an example of a general poset. In this generalization, languages are associated with subposets $L \in \mathcal{P}(P)$.

Definition 7.1.1 A *forbidding set* \mathcal{F} is a (possibly infinite) family of finite nonempty sets from $\mathcal{P}(P)$, not containing λ if $\lambda \in P$; each element F of the forbidding set \mathcal{F} is called a *forbidder*.

A subposet $L \subseteq P$, is *consistent* with a forbiddier F , denoted by $L \underline{con} F$ if $F \not\subseteq \underline{sub}(L)$. A subposet L is consistent with a forbidding set \mathcal{F} ($L \underline{con} \mathcal{F}$) if $L \underline{con} F$ for all $F \in \mathcal{F}$.

For a forbidding set \mathcal{F} the family of \mathcal{F} -consistent subposets is $\mathcal{L}(\mathcal{F}) = \{L \mid L \underline{con} \mathcal{F}\}$, i.e., $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{P}(P)$.

The family $\mathcal{L}(\mathcal{F})$ is said to be *defined* by the forbidding set \mathcal{F} . A family of subposets \mathcal{L} is called an *f-family*, if there is a forbidding set \mathcal{F} such that $\mathcal{L} = \mathcal{L}(\mathcal{F})$. The maximal subposets are defined with respect to inclusion and $\mathcal{M}(\mathcal{F})$ denotes the set of maximal subposets in $\mathcal{L}(\mathcal{F})$.

The corresponding boundary observations from Chapter 2 hold for posets, as well.

Remark 7.1.2 (i) $\mathcal{L}(\mathcal{F}) = \mathcal{P}(P)$ if and only if \mathcal{F} is empty.

(ii) $\emptyset \in \mathcal{L}(\mathcal{F})$ for every \mathcal{F} .

(iii) If $\lambda \in P$, then $\{\lambda\} \in \mathcal{L}(\mathcal{F})$ for every \mathcal{F} .

In addition to (A^*, \underline{sub}) , consider the following examples.

Example 7.1.3 Consider (\mathcal{G}, \leq) . Let $\mathcal{F} = \{\{C\} \mid C \text{ is a cycle}\}$. Then $\mathcal{L}(\mathcal{F}) = \{G \mid G \text{ is a set of trees}\}$.

Example 7.1.4 Consider (N, \mid) . Let $\mathcal{F} = \{\{2, 3\}, \{5, 7\}\}$ and $L_1 = \{n \mid 2 \nmid n \text{ and } 5 \nmid n\}$, $L_2 = \{n \mid 2 \nmid n \text{ and } 7 \nmid n\}$, $L_3 = \{n \mid 3 \nmid n \text{ and } 5 \nmid n\}$, and $L_4 = \{n \mid 3 \nmid n \text{ and } 7 \nmid n\}$. Then $\mathcal{L}(\mathcal{F}) = \{L \mid L \subseteq L_i \text{ for } i = 1, \dots, 4\}$. In fact, $\mathcal{M}(\mathcal{F}) = \{L_1, L_2, L_3, L_4\}$. Thus, $L = \{1, 4, 4^2, \dots\}$ is consistent with \mathcal{F} .

Again, \mathcal{F} is called a *strict* forbidding set if it contains singleton forbidders only.

Example 7.1.5 Once again, consider (N, \mid) . Let $\mathcal{L}(\mathcal{F}) = \{\{2\}\}$. Then, $\mathcal{L}(\mathcal{F})$ is the power set of odd numbers.

Definition 7.1.6 An *enforcing set* \mathcal{E} is a possibly infinite family of ordered pairs (X, Y) , where X and Y are finite subsets of P such that $Y \neq \emptyset$. The elements (X, Y) of an enforcing set \mathcal{E} are called *enforcers*.

A subset L is said to *satisfy* an enforcer (X, Y) if $X \subseteq L$ implies $Y \cap L \neq \emptyset$. For an enforcing set \mathcal{E} the family of all subsets L that satisfy \mathcal{E} is denoted by $\mathcal{L}(\mathcal{E})$. A family of subsets \mathcal{L} is called an *e-family* if there exists an enforcing set \mathcal{E} such that $\mathcal{L} = \mathcal{L}(\mathcal{E})$.

In the case that $X \not\subseteq L$, L is said to satisfy the enforcer trivially. Enforcers in which $X = \emptyset$ are called *brute*. In this case, an element from Y has to be in L in order for L to satisfy the enforcer. An enforcer (X, Y) for which $X \cap Y \neq \emptyset$ is called *trivial*, since it is always satisfied. The notation and definition for generated subsets is extended accordingly.

Remark 7.1.7 (i) $\mathcal{L}(\mathcal{E}) = \mathcal{P}(P)$ if and only if $\mathcal{E} = \emptyset$ or \mathcal{E} has trivial enforcers only.

(ii) $\emptyset \in \mathcal{L}(\mathcal{E})$ for every \mathcal{E} that does not have brute enforcers.

(iii) $P \in \mathcal{L}(\mathcal{E})$ for every \mathcal{E} .

In what follows, unless otherwise stated, all enforcers are non-trivial.

Definition 7.1.8 An enforcer (X, Y) is called *strict* if $|Y| = 1$.

In some sense, strict enforcers “force” Y into the subset L . Consider the following example of a strict enforcing set.

Example 7.1.9 Consider (\mathcal{G}, \leq) and let $\mathcal{E} = \{(\{P_2\}, \{C_3\})\}$. Then, $\mathcal{L}(\mathcal{E})$ consists of all graphs that contain a 3-cycle as a subgraph, Λ , and \emptyset .

Respectively, the two notions of forbidding and enforcing are combined into the following definition.

Definition 7.1.10 A *forbidding-enforcing system* is a construct $\Gamma = (P, \mathcal{F}, \mathcal{E})$ (or shortly $(\mathcal{F}, \mathcal{E})$ when P is understood), such that \mathcal{F} is a forbidding set and \mathcal{E} is an enforcing set over the poset P . The family of subposets $\mathcal{L}(\Gamma)$ defined by this system consists of all subposets L that are consistent with \mathcal{F} and satisfy \mathcal{E} , i.e., $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{E})$.

A family of subposets \mathcal{L} is called a *forbidding-enforcing family* or *fe-family*, if there exists a *fe-system* $(\mathcal{F}, \mathcal{E})$, such that $\mathcal{L} = \mathcal{L}(\mathcal{F}, \mathcal{E})$.

Example 7.1.11 Consider $(N, |)$. Let $\mathcal{E} = \{(\emptyset, \{1, 2\})\}$. Then, $\mathcal{L}(\mathcal{E})$ contains sets of numbers L that contain either 1 or 2.

Two sets of forbidders (or two enforcing sets, or two forbidding-enforcing systems) are equivalent if they define the same family of subposets. The equivalence relation is denoted by \sim .

From the above definitions it follows that there is no enforcing set \mathcal{E} such that $\mathcal{L}(\mathcal{E}) = \emptyset$. Neither there is a forbidding set \mathcal{F} such that $\mathcal{L}(\mathcal{F}) = \emptyset$.

Remark 7.1.12 $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{P}(P)$ if and only if $\mathcal{F} = \emptyset$ and $\mathcal{E} = \emptyset$ or \mathcal{E} consists of trivial enforcers only.

Example 7.1.13 Consider $(N, |)$ with $(\mathcal{F}, \mathcal{E})$ as in Examples 7.1.5 and 7.1.11. Then $\mathcal{L}(\mathcal{F}, \mathcal{E})$ contains all sets of odd numbers that contain 1.

The immediate properties of *fe-systems* on languages can be extended to families of subposets and can easily be verified. A poset P is assumed.

Proposition 7.1.14 Let \mathcal{F} and \mathcal{F}' be forbidding sets, \mathcal{E} and \mathcal{E}' be enforcing sets, and L and K be subposets of P .

- (i) If $L \underline{\text{con}} \mathcal{F}$, then $\underline{\text{sub}}(L) \underline{\text{con}} \mathcal{F}$.
- (ii) If $K \subseteq L$ and $L \underline{\text{con}} \mathcal{F}$, then $K \underline{\text{con}} \mathcal{F}$.
- (iii) If $\mathcal{F}' \subseteq \mathcal{F}$, then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$.

- (iv) If $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$.
- (v) If $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}', \mathcal{E}')$.
- (vi) $\mathcal{L}(\mathcal{F} \cup \mathcal{F}') = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{F}')$.
- (vii) $\mathcal{L}(\mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{E}) \cap \mathcal{L}(\mathcal{E}')$.
- (viii) $\mathcal{L}(\mathcal{F} \cup \mathcal{F}', \mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{F}, \mathcal{E}) \cap \mathcal{L}(\mathcal{F}', \mathcal{E}')$.

The first property above implies that if $L \underline{con} \mathcal{F}$ then $L \cap K \underline{con} \mathcal{F}$ for every subposet K (see also [40]). Hence, $\mathcal{L}(\mathcal{F})$ is closed under intersection. Namely, if $L \underline{con} \mathcal{F}$ and $K \underline{con} \mathcal{F}$ then $L \cap K \underline{con} \mathcal{F}$. However, $\mathcal{L}(\mathcal{F})$ is not closed under union. Take for example $(N, |)$ with $\mathcal{F} = \{\{2, 3\}\}$. Then $L = \{2\}$ and $K = \{3\}$ are consistent with $\mathcal{L}(\mathcal{F})$, but $L \cup K$ is not. The e -families are neither closed under intersection nor under union (see [40]).

The normal forms for fe -systems on formal languages from [8, 9, 40] follow directly for subposets with certain restrictions.

Example 7.1.15 Consider $(N, |)$ and a forbidding $\{2, 3, 6\}$. A subposet L is consistent with this forbidding if either $2 \notin \underline{sub}(L)$ or $3 \notin \underline{sub}(L)$ or $6 \notin \underline{sub}(L)$. In all cases, $6 \notin \underline{sub}(L)$. Thus, $L \underline{con} \{2, 3, 6\}$ implies $L \underline{con} \{6\}$. Conversely, if $L \underline{con} \{6\}$ it follows that $6 \notin \underline{sub}(L)$. Hence, $\{2, 3, 6\} \notin \underline{sub}(L)$. Thus, $L \underline{con} \{2, 3, 6\}$ if and only if $L \underline{con} \{6\}$.

The above example shows that the forbidding $\{2, 3, 6\}$ is redundant. It is not *subelement free* because $2 \in \underline{sub}(6)$.

Definition 7.1.16 A forbidding set \mathcal{F} is called *subelement free* if all of its forbidings are antichains.

This definition generalizes the *subword free* condition for forbidding sets on languages and Lemma 11.2 in [40] holds, i.e., for every forbidding set there is an equivalent subelement free forbidding set.

In general, two distinct forbidders F_1 and F_2 may have the same set of subelements like $\{2, 6\}$ and $\{3, 6\}$ in relation to $(N, |)$, but this cannot happen if the forbidders are antichains (see Lemma 2.1 in [40]), i.e., if K and L are antichains, then $\underline{sub}(K) = \underline{sub}(L)$ implies that $K = L$.

Not only elements within a forbinder may be redundant, but also forbidders themselves can be redundant. The following example illustrates such redundancy.

Example 7.1.17 Consider again $(N, |)$. Let $\mathcal{F} = \{\{3\}, \{3^2\}, \dots, \{3^n\}, \dots\}$. If a number is not divisible by 3, then it is not divisible by 3^i for $i \geq 2$. Hence, the above forbidding set is equivalent to $\mathcal{F}' = \{\{3\}\}$.

The reason for the above redundancy is that $\underline{sub}(\{3\}) \subseteq \underline{sub}(F)$ for every forbinder $F \in \mathcal{F}$. The following lemma is from [40] and it holds for posets, as well.

Lemma 7.1.18 *Let \mathcal{F} be a forbidding set and let $F_1, F_2 \in \mathcal{F}$ with $F_1 \neq F_2$. If $\underline{sub}(F_1) \subseteq \underline{sub}(F_2)$, then $\mathcal{F} \sim \mathcal{F}'$ where $\mathcal{F}' = \mathcal{F} \setminus \{F_2\}$.*

Lemma 11.4 from [40], also, holds for posets. It allows the removal of possibly infinitely many such forbidders.

The following definition generalizes the notion of subword incomparable forbidders from [40].

Definition 7.1.19 Two different forbidders F_1 and F_2 are said to be *subelement incomparable* if $\underline{sub}(F_1) \not\subseteq \underline{sub}(F_2)$ and $\underline{sub}(F_2) \not\subseteq \underline{sub}(F_1)$. A forbidding set is called *subelement incomparable* if each pair of distinct forbidders is subelement incomparable.

Remark 7.1.20 Note that, if P is a poset and $F \in \mathcal{P}(P)$ is a finite subposet. Then, $\underline{sub}(F)$ is not necessarily finite. It is finite only if all elements $p \in F$ are finite.

Some results for *fe*-systems on languages in [40] were proved using the fact that $\underline{sub}(L)$ is finite for a finite language L . Because of the above remark, these results

cannot be extended to posets directly. Some examples where $\underline{sub}(F)$ is finite are, (A^*, \underline{sub}) , $(N, |)$, (\mathcal{G}, \leq) , and $(\mathcal{P}(S), \subseteq)$.

Again, a forbidding set is in minimal normal form, if it is both subelement free and subelement incomparable. In the event that $\underline{sub}(F)$ is finite for every finite subposet $F \subseteq P$, Theorem 11.10 from [40] holds, i.e., for each forbidding set there is a unique equivalent forbidding set in minimal normal form.

Definition 7.1.21 Let \mathcal{F} be a forbidding set. The set $\underline{max} \mathcal{F} = \{H \mid H \text{ is finite and } H \underline{ncon} \mathcal{F}\}$ is called the maximal forbidding set of \mathcal{F} .

From [40], it holds that $\mathcal{F} \sim \underline{max} \mathcal{F}$.

Some properties of language enforcing sets hold for posets, as well, but their proofs need to be adjusted. Consider the following two observations from [40].

Lemma 7.1.22 *If \mathcal{E} is finite, then $\mathcal{L}(\mathcal{E})$ contains a finite subposet.*

Proof. Let $\mathcal{E} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$. Let $X = \cup_{i=1}^n X_i$ and $Y = \cup_{i=1}^n Y_i$. Then, $L = X \cup Y$ is such that $L \underline{sat} \mathcal{E}$ and L is finite. In fact, $K = Y$ is another such subposet. ■

Lemma 7.1.23 *If P contains an infinite chain $p_1 \leq p_2 \leq \dots \leq p_n \leq \dots$ then, there is an \mathcal{E} such that no finite enforcing set is equivalent to \mathcal{E} .*

Proof. Consider the enforcing set $\mathcal{E} = \{(\emptyset, \{p_1\}), (\{p_1\}, \{p_2\}), \dots\}$. Then every $L \in \mathcal{L}(\mathcal{E})$ contains the infinite chain $p_1, p_2, \dots, p_n, \dots$, i.e., is infinite and by Lemma 7.1.22 a finite \mathcal{E}' such that $\mathcal{E}' \sim \mathcal{E}$ does not exist. ■

Some observations for redundancy of enforcing sets for languages can be extended to posets directly (see [40]). Thus, trivial enforcers (for which $X \cap Y \neq \emptyset$) provide an obvious redundancy. Also, if the enforcing set contains two enforcers (X, Y) and (X', Y') such that $X \subseteq X'$ and $Y \subseteq Y'$, then (X', Y') is redundant.

For the remainder of this section it is assumed that an enforcing set \mathcal{E} consists of non-trivial enforcers only, i.e., if $(X, Y) \in \mathcal{E}$ then $X \cap Y = \emptyset$ and for each pair of enforcers (X, Y) and (X', Y') either $X \not\subseteq X'$ or $Y \not\subseteq Y'$.

The procedure of evolving through \mathcal{E} -extensions was proposed for enforcing of languages in [8, 9, 40]. It is generalized to posets.

Definition 7.1.24 For an enforcing set \mathcal{E} and subposets L and L' , L' is called an \mathcal{E} -extension of L , written $L \vdash_{\mathcal{E}} L'$, if $X \subseteq L$ implies $L' \cap Y \neq \emptyset$, for every $(X, Y) \in \mathcal{E}$.

Let \mathcal{E} be an enforcing set. Define $\mathcal{E}^{(1)} = \{X \mid (X, Y) \in \mathcal{E}\}$. The definition of generated sets for languages can, also, be extended to posets.

Definition 7.1.25 Let $X \in \mathcal{E}^{(1)}$. A subposet $g(X)$ generated by X is a set such that the following two conditions hold:

- (i) $X \subseteq g(X)$
- (ii) $g(X) \underline{sat} (X', Y')$ for every $(X', Y') \in \mathcal{E}$.

A generated set $g_m(X)$ is called *minimal*, if no proper subset of it is a generated set.

The notation from Chapter 2 is used here, as well. Let \mathcal{E} be an enforcing set and let $X \in \mathcal{E}^{(1)}$. Denote the family of generated sets of X with respect to \mathcal{E} with $G_X^{\mathcal{E}}$ or simply G_X when \mathcal{E} is understood. The family of minimal generated sets of X with respect to \mathcal{E} is denoted by $M_X^{\mathcal{E}}$ or simply M_X when \mathcal{E} is understood. Let $\mathcal{M}(\mathcal{E}) = \cup_{X \in \mathcal{E}^{(1)}} M_X$.

Remark 7.1.26 It follows from the definition of minimal generating sets that if \mathcal{E} is an enforcing set and $X \in \mathcal{E}^{(1)}$, then for every subposet L such that $L \underline{sat} \mathcal{E}$, $X \subseteq L$ implies that L contains as a subset a set $g_m(X) \in M_X$.

All results from Section 2.5 follow.

The following definition generalizes the notion of redundancy.

Definition 7.1.27 Given \mathcal{E} , (X', Y') is called *redundant* for \mathcal{E} , if there exists $(X, Y) \in \mathcal{E}$, with $X \subseteq X'$ and $Y' \cap g_m(X) \neq \emptyset$ for all $g_m(X) \in M_X^{\mathcal{E}'}$, where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$.

In particular, if $X \subseteq X'$ and $Y \subseteq Y'$, then (X', Y') is redundant. Redundant enforcers can be removed from the enforcing set without changing the e -family (see Section 2.5).

As in the case of languages, $\mathcal{M}(\mathcal{E})$ can be finite or infinite with finite or infinite generated sets. If \mathcal{E} is finite, then construct $L = X \cup Y$ as in the proof of Lemma 7.1.22 and observe that every $g_m \in \mathcal{M}(\mathcal{E})$ is such that $g_m \subseteq L$. Since L is finite, $\mathcal{M}(\mathcal{E})$ consists of finitely many finite minimal generated sets.

In the case that \mathcal{E} is infinite and not equivalent to any finite set and $\mathcal{M}(\mathcal{E})$ is finite, then Lemma 2.5.13 holds and there is an infinite minimal generated set in $\mathcal{M}(\mathcal{E})$.

7.2 fe -Systems Defining a Single Subset

Inspired by fe -systems on graphs and other structures, fe -systems on posets are defined, so that the corresponding fe -family is a single subposet. Thus, the fe -family is a class of graphs, a language, a set of numbers rather than a set of sets of such structures.

Definition 7.2.1 A *forbidding set* \mathcal{F} is a (possibly infinite) family of finite nonempty subsets of P , not containing λ , if $\lambda \in P$; each element F of the forbidding set \mathcal{F} is called a *forbidder*.

An element w of P , is said to be *consistent* with a forbidder F , denoted by $w \underline{\text{con}} F$, if $F \not\subseteq \underline{\text{sub}}(w)$. An element w is consistent with a forbidding set \mathcal{F} , if $w \underline{\text{con}} F$ for all $F \in \mathcal{F}$. This is denoted by $w \underline{\text{con}} \mathcal{F}$.

For a forbidding set \mathcal{F} the family of \mathcal{F} -consistent elements is $\mathcal{L}(\mathcal{F}) = \{w \mid w \underline{\text{con}} \mathcal{F}\}$. $\mathcal{L}(\mathcal{F}) \in \mathcal{P}(P)$, i.e., it is a subposet of P .

The family $\mathcal{L}(\mathcal{F})$ is said to be *defined* by the forbidding set \mathcal{F} . A subposet L is called an f -family, if there is a forbidding set \mathcal{F} such that $L = \mathcal{L}(\mathcal{F})$.

The following boundary observations state that if nothing is forbidden everything is allowed and that the smallest element(if any) is always in an f -family.

Remark 7.2.2 (i) $\mathcal{L}(\mathcal{F}) = P$ if and only if \mathcal{F} is empty.

(ii) If P contains λ , then $\lambda \in \mathcal{L}(\mathcal{F})$ for every \mathcal{F} .

Example 7.2.3 Consider (A^*, \underline{sub}) and let $\mathcal{F} = \{\{aa, bb\}, \{ab, ba\}\}$. Then, $\mathcal{L}(\mathcal{F}) = \{a^* \cup b^* \cup a^*b \cup ab^* \cup b^*a \cup ba^*\}$ and in fact, it is a regular language.

Example 7.2.4 Consider (\mathcal{G}, \leq) and let $\mathcal{F} = \{\{C_3, C_4\}\}$. Then $\mathcal{L}(\mathcal{F}) = \{G \mid \text{either } C_3 \not\leq G \text{ or } C_4 \not\leq G\}$.

Definition 7.2.5 A forbidding set \mathcal{F} is called *strict* if it contains singleton forbid-ers only.

Example 7.2.6 Revisit (\mathcal{G}, \leq) and the strict forbidding set $\mathcal{F} = \{\{C_3\}, \{C_4\}, \dots\}$. Then, $L(\mathcal{F})$ is the set of all trees.

Definition 7.2.7 Weak enforcing. An *enforcing set* \mathcal{E} is a possibly infinite family of ordered pairs (X, Y) , where X and Y are finite subposets of P , such that $Y \neq \emptyset$. The elements (X, Y) of an enforcing set \mathcal{E} are called *enforcers*.

An element w is said to *weakly satisfy* an enforcer (X, Y) if $X \subseteq \underline{sub}(w)$ implies $Y \cap \underline{sub}(w) \neq \emptyset$. An element w *weakly satisfies* an enforcing set \mathcal{E} , if w weakly satisfies every enforcer in that set. For an enforcing set \mathcal{E} the family of all elements w that weakly satisfy \mathcal{E} is denoted by $\mathcal{L}(\mathcal{E})$. Obviously, $\mathcal{L}(\mathcal{E}) \in \mathcal{P}(P)$. A subposet L is called an *e-family* if there exists an enforcing set \mathcal{E} such that $L = \mathcal{L}(\mathcal{E})$.

In the case that $X \not\subseteq \underline{sub}(w)$, w is said to satisfy the enforcer trivially. If $X \subseteq \underline{sub}(w)$ for some enforcer (X, Y) , then (X, Y) is *applicable* to w (see [8, 9, 40]). Enforcers in which $X = \emptyset$ are called *brute*. In this case, an element from Y has to be in $\underline{sub}(w)$ in order for w to satisfy the enforcer. If P contains λ , then enforcers with $X = \{\lambda\}$ are also “brute” since $\lambda \in \underline{sub}(w)$ for every $w \in P$.

The “weak” enforcing in the case of (A^*, \underline{sub}) ensures that if a word has all words from a pre-specified set X as subwords than it also has at least one subword from another set Y .

Example 7.2.8 Consider (A^*, \underline{sub}) . Let $X = \{aa, bb\}$ and $Y = \{ab\}$. Then, $baa, aabba \underline{sat}(X, Y)$, but $bbaa \underline{nsat}(X, Y)$.

Definition 7.2.9 Strong enforcing. An *enforcing set* \mathcal{E} is a possibly infinite family of ordered pairs (X, Y) , where X is an element in P and Y is a finite subposet of P , such that $Y \neq \emptyset$ and $X < y$ for every $y \in Y$. The elements (X, Y) of an enforcing set \mathcal{E} are called *enforcers*.

An element $w \in P$ is said to *strongly satisfy* an enforcer (X, Y) if whenever $X \leq w$ there is a $y \in Y$ such that $X \leq y \leq w$. Moreover, for every embedding $\theta : X \rightarrow w$ there exists an embedding $\psi : X \rightarrow y$ for some $y \in Y$ and there is an embedding $\phi : y \rightarrow w$ such that $\theta = \phi\psi$. This is denoted by $w \underline{sat}(X, Y)$. An element w *strongly satisfies* an enforcing set \mathcal{E} , if w strongly satisfies every enforcer in that set. For an enforcing set \mathcal{E} the family of all elements w that strongly satisfy \mathcal{E} is denoted $\mathcal{L}(\mathcal{E})$. Obviously, $\mathcal{L}(\mathcal{E}) \in \mathcal{P}(P)$. A subposet L is called an *e-family*, if there exists an enforcing set \mathcal{E} such that $L = \mathcal{L}(\mathcal{E})$.

The following example illustrates how weak enforcing differs from strong enforcing defined above and in Chapter 5.

Example 7.2.10 Consider (\mathcal{G}, \leq) . Then, $\mathcal{E} = \{(\{P_3\}, \{C_3\})\}$ ensures that every connected graph with at least 3 vertices contains a 3-cycle, which is different than the strong enforcing definition where $\mathcal{E} = \{(P_3, \{C_3\})\}$ defines the class of complete graphs (see Proposition 5.5.5).

Note that for weak enforcing, if $X \cap Y \neq \emptyset$, then (X, Y) is satisfied by any $w \in P$. This is a direct generalization from [40]. Such enforcers are called trivial. For strong enforcing, $X \underline{nsat}(X, Y)$ for any $X \neq \lambda$ and Y such that $\lambda \notin Y$ (see also Proposition 5.3.8). In what follows, all enforcers are non-trivial.

When it is clear whether the enforcing set is weak or strong, just “enforcing” is used, instead of “weak enforcing” or “strong enforcing”.

The following remark holds for both weak and strong enforcing.

Remark 7.2.11 (i) $\mathcal{L}(\mathcal{E}) = P$ if and only if $\mathcal{E} = \emptyset$.

(ii) If $\lambda \in P$, then $\lambda \in \mathcal{L}(\mathcal{E})$ for every \mathcal{E} that does not have brute enforcers.

Definition 7.2.12 An enforcer (X, Y) is called *strict* if $|Y| = 1$.

In some sense, strict enforcers “force” Y into the subelements of w whenever $X \subseteq \underline{sub}(w)$ or $X \leq w$. Consider the following example consisting of strict enforcers only.

Example 7.2.13 Consider (A^*, \underline{sub}) and the weak enforcing set $\mathcal{E} = \{(\{\lambda\}, \{a\}), (\{a\}, \{ab\}), (\{ab\}, \{abb\}), \dots\}$. Then no finite word satisfies \mathcal{E} . The infinite word ab^ω satisfies \mathcal{E} , as well as wab^ω for any $w \in A^*$. Hence, $\mathcal{L}(\mathcal{E}) = \emptyset$.

The *fe*-systems definition can be extended to posets accordingly.

Definition 7.2.14 A *forbidding-enforcing system* is a construct $\Gamma = (P, \mathcal{F}, \mathcal{E})$ (or shortly $(\mathcal{F}, \mathcal{E})$ when P is understood), such that \mathcal{F} is a forbidding set and \mathcal{E} is an enforcing set. The family of elements (subposet) $\mathcal{L}(\Gamma)$ defined by this system consists of all elements w that are consistent with \mathcal{F} and satisfy \mathcal{E} , i.e., $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{E})$.

A subposet L is called a *forbidding-enforcing family* or *fe-family*, if there exists a *fe*-system $(\mathcal{F}, \mathcal{E})$, such that $L = \mathcal{L}(\mathcal{F}, \mathcal{E})$.

Two sets of forbidders (or two enforcing sets, or two forbidding-enforcing systems) are equivalent if they define the same family of elements (the same subposet). The equivalence relation is denoted by \sim . Two forbidders F and F' (similarly two enforcers E and E') are equivalent denoted by $F \sim F'$ ($E \sim E'$), if all elements that are consistent with F (satisfy E) are also consistent with F' (also satisfy E') and vice versa.

In the language case there is no enforcing set \mathcal{E} such that $\mathcal{L}(\mathcal{E}) = \emptyset$ and there is no forbidding set \mathcal{F} such that $\mathcal{L}(\mathcal{F}) = \emptyset$. The following remark shows that this does not necessarily hold for the single poset definitions.

Remark 7.2.15 If $\lambda \notin P$, then $\mathcal{F} = \{\{p\} \mid p \in P\}$ is such that $\mathcal{L}(\mathcal{F}) = \emptyset$. If $\lambda \in P$, there is no \mathcal{F} such that $\mathcal{L}(\mathcal{F}) = \emptyset$.

The following remark holds for both weak and strong enforcing.

Remark 7.2.16 If for every $p \in P$ there is an infinite chain $p \leq p_1 \leq p_2 \leq \dots$ and all elements in p are finite, then there is an \mathcal{E} such that $\mathcal{L}(\mathcal{E}) = \emptyset$. For example, take $\mathcal{E}_p = \{(\{p\}, \{p_1\}), (\{p_1\}, \{p_2\}), \dots\}$ and let $\mathcal{E} = \cup_{p \in P} \mathcal{E}_p$ for weak enforcing and similarly (change $\{p_i\}$ with p_i in the first components), for strong enforcing.

The next two examples illustrate such posets and hold for both weak and strong enforcing.

Example 7.2.17 Consider (A^*, sub) and let $a \in A$. Then, $\mathcal{E} = \{(\{\lambda\}, \{w\}), (\{w\}, \{wa\}), (\{wa\}, \{wa^2\}), \dots \mid w \in A^*\}$ is such that $\mathcal{L}(\mathcal{E}) = \emptyset$.

Example 7.2.18 Consider (\mathcal{G}, \leq) . Then for every graph G there is an extension by an edge graph G^1 . Consider a sequence $G \leq G^1 \leq G^2 \leq \dots$, such that G^i is an extension by an edge of G^{i-1} . Let $\mathcal{E} = \{(\emptyset, \{G\}), (\{G\}, \{G^1\}), (\{G^1\}, \{G^2\}), \dots\}$. Then, $\mathcal{L}(\mathcal{E}) = \emptyset$. In particular, the sequence of graphs can be P_2, P_3, \dots

Remark 7.2.19 For any poset P there exist *fe*-systems defining empty *fe*-families. For example, if $\lambda \notin P$, let $\mathcal{F} = \{\{p\} \mid p \in P\}$. Then, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \emptyset$ for any \mathcal{E} . In the case that $\lambda \in P$, consider $\mathcal{F} = \{\{p\} \mid p \in P\}$ and $\mathcal{E} = \{(\{\lambda\}, \{p\})\}$ where $p \in P$. Then, $\mathcal{L}(\mathcal{F}, \mathcal{E}) = \emptyset$ for weak enforcing. It holds for strong enforcing, as well, but $\{\lambda\}$ is replaced by λ .

Remark 7.2.20 $\mathcal{L}(\mathcal{F}, \mathcal{E}) = P$ if and only if $\mathcal{F} = \emptyset$ and $\mathcal{E} = \emptyset$.

The following proposition contains immediate properties of forbidding and enforcing sets. These properties follow directly from the definitions, Proposition 5.3.15, and the corresponding properties for language *fe*-systems from [8, 9, 40].

Proposition 7.2.21 *Let \mathcal{F} and \mathcal{F}' be forbidding sets, \mathcal{E} and \mathcal{E}' be enforcing sets, and u and v be elements in P .*

- (i) *If $u \leq v$ and $v \underline{con} \mathcal{F}$, then $u \underline{con} \mathcal{F}$.*
- (ii) *If $\mathcal{F}' \subseteq \mathcal{F}$, then $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$.*
- (iii) *If $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{E}) \subseteq \mathcal{L}(\mathcal{E}')$.*
- (iv) *If $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{E}' \subseteq \mathcal{E}$, then $\mathcal{L}(\mathcal{F}, \mathcal{E}) \subseteq \mathcal{L}(\mathcal{F}', \mathcal{E}')$.*
- (v) $\mathcal{L}(\mathcal{F} \cup \mathcal{F}') = \mathcal{L}(\mathcal{F}) \cap \mathcal{L}(\mathcal{F}')$.
- (vi) $\mathcal{L}(\mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{E}) \cap \mathcal{L}(\mathcal{E}')$.
- (vii) $\mathcal{L}(\mathcal{F} \cup \mathcal{F}', \mathcal{E} \cup \mathcal{E}') = \mathcal{L}(\mathcal{F}, \mathcal{E}) \cap \mathcal{L}(\mathcal{F}', \mathcal{E}')$.

If intersection is defined for the elements of the poset P in such a way that for any two elements u and v , $\underline{sub}(u \cap v) \subseteq \underline{sub}(u) \cap \underline{sub}(v)$, then the first property above implies that if $w \underline{con} \mathcal{F}$ then $w \cap u \underline{con} \mathcal{F}$ for any u . Such *f*-families are closed under intersection. Consider for example (A^*, \underline{sub}) where intersection $u \cap v = w$ is defined such that $w \in Pref(u) \cap Pref(v)$ and $wa \notin Pref(u) \cap Pref(v)$ for any $a \in A$.

However, if union is defined in a similar way, i.e., $\underline{sub}(u \cup v) \supseteq \underline{sub}(u) \cup \underline{sub}(v)$, then $\mathcal{L}(\mathcal{F})$ is not closed under union. Take for example $\mathcal{F} = \{\{u, v\}\}$, such that $u \not\leq v$ and $v \not\leq u$. Then $u \underline{con} \mathcal{F}$ and $v \underline{con} \mathcal{F}$, but $u \cup v \underline{ncon} \mathcal{F}$ since $\{u, v\} \subseteq \underline{sub}(u \cup v)$. Concatenation of words is an example of such “union”.

7.3 Upper Bounds

For some posets P it holds that for every finite set F with elements from P , there is a $p \in P$ such that $F \subseteq \underline{sub}(p)$. Examples of such posets are (A^*, \underline{sub}) where p may

be a concatenation of words in F ; (\mathcal{G}, \leq) where all graphs of F may be consecutively connected by an edge; $(N, |)$ where p is the LCM of the numbers in F .

In this section, the discussion of connecting graphs from Chapter 5 is extended to posets. The notion of “connecting graphs” is generalized to “upper bounds”.

Definition 7.3.1 Let P be a poset and F be a finite subset of P . An element $s \in P$ is an *upper bound* of F (or $s \in P$ *connects* F) if $F \subseteq \underline{sub}(s)$. An upper bound s of F is called *minimal* if for every upper bound h of F it holds that $h \leq s$ implies $h = s$. The set of upper bounds of F is called the *connect* of F and is denoted by $C(F)$ and the set of minimal upper bounds is called the *minimal connect* of F and is denoted by $C_{min}(F)$.

So, for a forbidden F and an element $w \in P$, either $w \underline{con} F$ or w “connects” (is an upper bound of) F . For every $w \in C(F)$ there is a $s \in C_{min}(F)$ such that $s \leq w$.

It is obvious that if an element is an upper bound of a finite set of elements, it is also an upper bound of every subset of this set of elements. This fact is a direct extension of Proposition 5.2.6.

Proposition 7.3.2 Let F and F' be two finite subsets of P such that $F' \subseteq F$. Then $C(F) \subseteq C(F')$.

For a given finite set F , there may be infinitely many minimal upper bounds of F . Example 5.2.5 discusses one case for graphs. Another example is discussed next.

Example 7.3.3 Consider (A^*, \underline{sub}) and the set $F = \{aa, bb\}$. Then, $C_{min}(F) = aa(ba)^*bb \cup bb(ab)^*aa$. Hence, there are infinitely many minimal upper bounds of F .

In the case of natural numbers with divisibility, for every finite set of numbers, the minimal connecting element is unique and it is equal to the least common multiple (LCM) of the numbers in the set.

Remark 7.3.4 Consider $P = (N, |)$. For every finite $F \subseteq P$, $C_{min}(F) = \{l(F)\}$ where $l(F)$ is the LCM of the numbers in F .

In the following examples $F' \subseteq F$, but $C_{min}(F) \not\subseteq C_{min}(F')$.

Example 7.3.5 Consider (A^*, \underline{sub}) with $F' = \{aa, bb\}$ and $F = \{aa, bb, ab\}$. Then $bbaab \in C_{min}(F)$, but $bbaab \notin C_{min}(F')$.

Consider a special case, which is a direct generalization of Proposition 5.2.8.

Proposition 7.3.6 Let F be a finite subposet of P such that there is a $q \in F$ with $q \leq s$ for every $s \in C(F')$, where $F' = F \setminus \{q\}$. Then $C_{min}(F) = C_{min}(F')$.

Definition 7.3.7 Let X be a finite subposet of P . Define X^1 to be an *extension* of X , if $X^1 \in C(X)$ and $\underline{sub}(X) \subsetneq \underline{sub}(X^1)$.

The above definition says that if $X = \{p\}$, then p cannot be an extension of itself, even though it is an upper bound of X , but $q \in P$ such that $q \neq p$ and $p \leq q$ is an extension of p . So, every extension is an upper bound, but the converse does not hold.

Definition 7.3.8 Let P be a poset. If every finite set in P has an upper bound, then P is called *weakly extendable* and if every finite set in P has an extension, P is called *extendable*.

Proposition 7.3.9 If a poset P is extendable, then it is weakly extendable, but the converse does not hold.

Proof. Let P be extendable. By Definitions 7.3.7 and 7.3.8, if X is a finite subposet of P , then there is an extension X^1 of X that is an upper bound of X . Hence, P is weakly extendable. Consider a finite set S and let $P = (\mathcal{P}(S), \subseteq)$. Then, S is an upper bound for every finite subset $X \subset \mathcal{P}(S)$, but S does not have an extension. ■

The poset $(A^{\leq n}, \underline{sub})$ is neither extendable, nor weakly extendable. The posets (A^*, \underline{sub}) , (\mathcal{G}, \leq) , and $(N, |)$ are extendable posets that contain finite elements only.

The forbidding through enforcing observations from Chapter 5 can be generalized for extendable P consisting of finite elements only.

Proposition 7.3.10 *Let $\mathcal{F} = \{F\}$ and $\mathcal{E}_F = \{(\{s\}, \{s^1\}), \dots, (\{s^{n-1}\}, \{s^n\}), \dots \mid$ for every $s \in C_{\min}(F)$ and s^i is an extension of s^{i-1} for every $i > 1\}$. Then, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E}_F)$.*

Similar to Chapter 5, the above proposition can be extended to a general forbidding set \mathcal{F} with more than one forbidders by including enforcers like \mathcal{E}_F for every $F \in \mathcal{F}$ and considering their union $\mathcal{E} = \cup_{F \in \mathcal{F}} \mathcal{E}_F$. Then, $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E})$.

Theorem 7.3.11 *Let P be extendable, containing finite elements only. Then, for every forbidding set \mathcal{F} , there is an enforcing set \mathcal{E} such that $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{E})$.*

The proofs to Proposition 7.3.10 and Theorem 7.3.11 are a direct generalization of the proof of Proposition 5.4.2 and Theorem 5.4.3; thus, omitted.

7.4 Normal Forms for Forbidding Sets

This section discusses redundancy of forbidding sets and how it can be avoided without changing the forbidding family (subposet). It is a generalization of the normal forms for graph f -families from Chapter 6 and some normal forms for language f -families from [8, 9, 40].

Example 7.4.1 Consider a forbidders $\{p_1, p_2, q\}$ where $p_i \leq q$ for $i = 1, 2$. An element w is consistent with this forbidders if either $p_1 \not\leq w$, or $p_2 \not\leq w$, or $q \not\leq w$. In all three cases, $q \not\leq w$. Thus, $w \underline{\text{con}} \{p_1, p_2, q\}$ implies $w \underline{\text{con}} \{q\}$. Conversely, if $w \underline{\text{con}} \{q\}$ it follows that $q \not\leq w$ which implies that $w \underline{\text{con}} \{p_1, p_2, q\}$. Hence, $w \underline{\text{con}} \{p_1, p_2, q\}$ if and only if $w \underline{\text{con}} \{q\}$.

The above example shows that the forbidders $\{p_1, p_2, q\}$ is redundant. It is not “subelement free”, because $p_1 \not\leq q$ and $p_2 \not\leq q$.

Definition 7.4.2 A forbidding set \mathcal{F} is called *subelement free* if all of its forbidders are antichains. Define $F_{\max} = \{p \in F \mid \text{for every } q \in F p \leq q \text{ implies } p = q\}$. Define $\mathcal{F}_{\max} = \{F_{\max} \mid F \in \mathcal{F}\}$.

Clearly, F_{max} is an antichain

Example 7.4.3 Consider $(N, |)$. The forbidding $F = \{2, 3, 6\}$ is not an antichain, but $F_{max} = \{6\}$ is. Let $n \in N$. Then, $n \underline{con} F$ if and only if n is not divisible by 6.

“Subelement free” is a normal form. The proof of the following lemma is a direct generalization of the proof of Lemma 6.1.5.

Lemma 7.4.4 *For every forbidding set there exists an equivalent subelement free forbidding set.*

Generalizing from graph fe -systems, two distinct forbidbers F_1 and F_2 may have the same set of subelements. For example, consider $(N, |)$ and let $F_1 = \{2, 6\}$ and $F_2 = \{3, 6\}$. This cannot happen, however, if the forbidbers are subelement free, as stated in the following lemma.

Lemma 7.4.5 *Let F and H be subelement free subposets. Then, $\underline{sub}(F) = \underline{sub}(H)$ implies that $F = H$.*

The “connecting free” definition for graphs is extended to posets.

Definition 7.4.6 A forbidding F is called *connecting free* if for every $p \in F$ there is an upper bound $s \in C(F \setminus \{p\})$, such that $p \not\leq s$. A forbidding set \mathcal{F} is called *connecting free* if all forbidbers $F \in \mathcal{F}$ are connecting free.

The following example shows that a forbidding set that is an antichain is not necessarily connecting free.

Example 7.4.7 Consider $(N, |)$. Let $F = \{6, 10, 15\}$. It is an antichain since neither of its elements divides another element. F is not connecting free, however, since all of the upper bounds of $F' = \{6, 10\}$ are divisible by 30; therefore, by 15. F can be reduced to F' , since if $w \underline{con} F'$, then $w \underline{con} F$. Conversely, if $w \underline{con} F$ either $6 \nmid w$, or $10 \nmid w$, or $15 \nmid w$. In the first two cases $w \underline{con} F'$. Suppose $15 \nmid w$. Then, either $3 \nmid w$ or $5 \nmid w$. In the first case $6 \nmid w$ and in the second $10 \nmid w$. Therefore, $w \underline{con} F'$.

The following facts about connecting free forbidders are a direct generalization of these in Chapter 6.

Lemma 7.4.8 *Let \mathcal{F} be given and let $F \in \mathcal{F}$ be such that $F = \{p_1, \dots, p_n, q\}$ and $q \leq s$ for every $s \in C(F')$, where $F' = F \setminus \{q\}$. Then, $\mathcal{F} \sim (\mathcal{F} \setminus \{F\}) \cup \{F'\}$.*

Proposition 7.4.9 *Connecting free implies subelement free, but the converse does not hold.*

Lemma 7.4.10 *Let F be a forbidded. Then there exists a connecting free F_{free} such that $F_{free} \subseteq F$ and $F_{free} \sim F$.*

Example 7.4.11 Consider $(N, |)$ and let $F = \{6, 10, 15\}$. Then $F_{free} = \{10, 15\}$, or $F_{free} = \{6, 15\}$, or $F_{free} = \{6, 10\}$. All three of these forbidders are equivalent.

Corollary 7.4.12 *Given a forbidding set \mathcal{F} and a forbidded $F \in \mathcal{F}$. If F' and F'' are two connecting free forbidders obtained from F as in Lemma 7.4.10, then $F' \sim F''$.*

The following theorem states that connecting free is a normal form.

Theorem 7.4.13 *For every forbidding set there exists an equivalent connecting free forbidding set.*

Note that the set \mathcal{F}' from the proof of the preceding theorem is not unique (Example 7.4.11).

Not only elements within a forbidded can be redundant, but also forbidders themselves can be redundant. Next, the subelement incomparable normal form, which is a generalization of the normal forms for graph f -systems from Chapter 6 and language f -systems from [40] is discussed.

Example 7.4.14 For $(N, |)$ consider $\mathcal{F} = \{\{3\}, \{3^2\}, \dots, \{3^n\}, \dots\}$. If a number is not divisible by 3, it is certainly not divisible by any power of 3. Hence, the above forbidding set is equivalent to $\mathcal{F}' = \{\{3\}\}$.

The reason for the above redundancy is that $3 \in \underline{sub}(F)$ for every forbidders $F \in \mathcal{F}$. The following lemma provides a generalization.

Lemma 7.4.15 *Let \mathcal{F} be a forbidding set and let $F_1, F_2 \in \mathcal{F}$ with $F_1 \neq F_2$. If $\underline{sub}(F_1) \subseteq \underline{sub}(F_2)$, then $\mathcal{F} \sim \mathcal{F} \setminus \{F_2\}$.*

The following lemma ensures that possibly infinitely many such forbidders can be removed.

Lemma 7.4.16 *Let \mathcal{F} and \mathcal{F}' be forbidding sets with $\mathcal{F}' \subseteq \mathcal{F}$, such that for each $F \in \mathcal{F}$ there is a $F' \in \mathcal{F}'$ with $\underline{sub}(F') \subseteq \underline{sub}(F)$. Then $\mathcal{F} \sim \mathcal{F}'$.*

Definition 7.4.17 Two forbidders F_1, F_2 are *subelement incomparable* if $\underline{sub}(F_1) \not\subseteq \underline{sub}(F_2)$ and $\underline{sub}(F_2) \not\subseteq \underline{sub}(F_1)$. A forbidding set is called *subelement incomparable* if each pair of distinct forbidders is subelement incomparable.

The following lemma states that *subelement incomparable* is a normal form.

Lemma 7.4.18 *Let P be a poset containing finite elements only. Then, for each forbidding set over P there is an equivalent subelement incomparable forbidding set.*

In language *f*-systems the corresponding subword incomparable normal form along with subword free is minimal and unique (see [?, 9, 40]). In the case of single subposet *f*-families, subelement incomparable is not minimal as illustrated in the following example.

Example 7.4.19 Consider $(N, |)$. Let $\mathcal{F} = \{\{15\}, \{6, 10\}\}$. Both forbidders are obviously connecting free and neither of the three numbers divide another, so the forbidders are subelement incomparable, as well. However, the forbidding set \mathcal{F} can be reduced further to the set $\mathcal{F}' = \{\{15\}\}$. To see that $\mathcal{F} \sim \mathcal{F}'$ consider the fact that every number that is divisible by both 6 and 10 is also divisible by 15. Hence, $\{6, 10\} \subseteq \underline{sub}(w)$ for some element $w \in N$ implies $\{15\} \subseteq \underline{sub}(w)$.

The preceding example shows that the notion of subelement incomparable can be generalized to include not only the set of subelements of a forbidders, but also, the connecting elements of that forbidders.

Lemma 7.4.20 *Let \mathcal{F} be a forbidding set and $F_1, F_2 \in \mathcal{F}$ with $F_1 \neq F_2$ such that $C(F_2) \subseteq C(F_1)$. Then $\mathcal{F} \sim (\mathcal{F} \setminus \{F_2\})$.*

Example 7.4.21 Consider $(N, |)$ where $F_1 = \{2, 15\}$ and $F_2 = \{6, 10\}$. Then, since the LCM is 30 for both F_1 and F_2 , they are equivalent. Note that F_1 and F_2 are connecting (subelement) free and subelement incomparable.

Lemma 7.4.22 *Let \mathcal{F} and \mathcal{F}' be forbidding sets with $\mathcal{F}' \subseteq \mathcal{F}$ such that for each $F \in \mathcal{F}$ there is an $F' \in \mathcal{F}'$ such that $C(F) \subseteq C(F')$. Then, $\mathcal{F}' \sim \mathcal{F}$.*

Proposition 7.4.23 *Let \mathcal{F} be a forbidding set and $F_1, F_2 \in \mathcal{F}$. Then, F_1 and F_2 are equivalent if and only if $C(F_1) = C(F_2)$.*

So, if F_1 and F_2 are not equivalent either there exists an upper bound $s \in C(F_1)$ such that $F_2 \not\subseteq \underline{\text{sub}}(s)$ or there exists an upper bound $h \in C(F_2)$ such that $F_1 \not\subseteq \underline{\text{sub}}(h)$.

Definition 7.4.24 Two forbidders F_1 and F_2 are *connecting incomparable* if there exists an upper bound $s \in C(F_1)$ such that $F_2 \not\subseteq \underline{\text{sub}}(s)$ and there exists an upper bound $h \in C(F_2)$ such that $F_1 \not\subseteq \underline{\text{sub}}(h)$. A forbidding set \mathcal{F} is connecting incomparable if every pair of forbidders in it is connecting incomparable.

Proposition 7.4.25 *Connecting incomparable implies subelement incomparable, but the converse does not hold.*

Corollary 7.4.26 *Let \mathcal{F} be a forbidding set and $F, F' \in \mathcal{F}$ with $F \neq F'$. If F and F' are connecting incomparable, then they are not equivalent.*

The following lemma establishes that connecting incomparable is a normal form.

Lemma 7.4.27 *Let P contain finite elements only. Then for every forbidding set there exists an equivalent connecting incomparable forbidding set.*

The connecting incomparable normal form is not unique.

In general, if two forbidders F and H share a common minimal upper bound s it does not necessarily follow that $F \sim H$. Also, even for a minimal upper bound s of F it does not necessarily follow that $F \sim \{s\}$. For example, consider $\{aa, bb\}$ over (A^*, \underline{sub}) . Then, $s = aabb$ is a minimal upper bound and $bbaa \underline{con} \{s\}$, but $bbaa \underline{ncon} \{aa, bb\}$.

However, for $(N, |)$ it holds that for every F , $F \sim \{s\}$ for the minimal upper bound (LCM) s of F .

Proposition 7.4.28 *Let $P = (N, |)$. Then, for every forbidding set $\mathcal{F} = \{F_1, F_2, \dots\}$ there exists an equivalent forbidding set $\mathcal{F}' = \{\{s_1\}, \{s_2\}, \dots\}$, where s_i is the LCM of the numbers in F_i .*

Proof. Let $w \underline{con} \mathcal{F}$ and let $\{s\} \in \mathcal{F}'$. Then, there is $F \in \mathcal{F}$ such that s is the LCM of the numbers in F . Since $F \not\subseteq \underline{sub}(w)$, $s \notin \underline{sub}(w)$. Hence, $w \underline{con} \{s\}$ and $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Conversely, if $w \underline{con} \{s\}$ it follows that $s \nmid w$. This implies that some divisor of s say k is such that $k \nmid w$. Consider the smallest such number k . Then, either $k \in F$ or $k \mid l$ for some $l \in F$. Hence, $w \underline{con} F$. ■

Because of the above proposition, all forbidders over $(N, |)$ can be replaced with singletons, i.e., one can obtain strict forbidding sets only. Because of subelement incomparable normal form, if $\{s\}$ and $\{k \times s\}$ are two forbidders in \mathcal{F}' , then $\{k \times s\}$ is redundant. So, $\{k \times s\}$ can be removed from \mathcal{F}' . Furthermore, since all forbidders are singletons, parentheses can be omitted, i.e., $\mathcal{F}' = \{h_1, \dots, h_n, \dots\}$, where for every i and every j it holds that $h_i \nmid h_j$. So, a number $w \in \mathcal{L}(\mathcal{F})$ if $h_i \nmid w$ for every $i \geq 1$.

A strict forbidding set is always connecting (subelement) free and subelement incomparable coincides with connecting incomparable. So, the forbidding set described above for $(N, |)$ is indeed minimal and unique.

The next proposition is a generalization of Proposition 6.1.31.

Proposition 7.4.29 *Let P be weakly extendable. Let \mathcal{F} be a forbidding set. For every $F \in \mathcal{F}$ choose one minimal upper bound $s_F \in C_{\min}(F)$ and consider $\mathcal{F}' = \{\{s_F\} \mid F \in \mathcal{F}\}$. Then, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$.*

Proof. Let $w \underline{\text{con}} \mathcal{F}$ and let $\{s_F\} \in \mathcal{F}'$. Then, s_F is a minimal upper bound for some $F \in \mathcal{F}$. Since $F \not\subseteq \underline{\text{sub}}(w)$, it holds that $\{s_F\} \not\subseteq \underline{\text{sub}}(w)$. Hence, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. ■

To see that equality does not necessarily hold consider once again $F = \{aa, bb\}$ over $(A^*, \underline{\text{sub}})$ and let $\{s_F\} = \{aabb\} \in \mathcal{F}'$. The minimal upper bound $bbaa \underline{\text{con}} \{s\}$, but $bbaa \underline{\text{ncon}} \{aa, bb\}$. However, if all minimal upper bounds of all forbidders from \mathcal{F} are included as singleton forbidders of \mathcal{F}' , then $\mathcal{F} \sim \mathcal{F}'$.

Proposition 7.4.30 *Let P be weakly extendable and \mathcal{F} be a forbidding set. For every $F \in \mathcal{F}$ construct the forbidding set $\mathcal{F}_F = \{\{s\} \mid s \in C_{\min}(F)\}$ and consider $\mathcal{F}' = \cup_{F \in \mathcal{F}} \mathcal{F}_F$. Then, $\mathcal{F} \sim \mathcal{F}'$.*

Proof. Let $w \underline{\text{con}} \mathcal{F}$ and let $\{s\} \in \mathcal{F}'$. Then, s is a minimal upper bound for some $F \in \mathcal{F}$. Since $F \not\subseteq \underline{\text{sub}}(w)$, it holds that $\{s\} \not\subseteq \underline{\text{sub}}(w)$. Hence, $\mathcal{L}(\mathcal{F}) \subseteq \mathcal{L}(\mathcal{F}')$. Conversely, let $w \underline{\text{con}} \mathcal{F}'$ and let $F \in \mathcal{F}$. Suppose $F \subseteq \underline{\text{sub}}(w)$. Then, $w \in C(F)$ and there is $s \in C_{\min}(F)$ such that $s \leq w$. This contradicts the fact that $w \underline{\text{con}} \mathcal{F}'$. Hence, $F \not\subseteq \underline{\text{sub}}(w)$. Therefore, $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F})$. Consequently, $\mathcal{F} \sim \mathcal{F}'$. ■

Proposition 7.4.30 proves that every forbidding set is equivalent to a strict forbidding set. This fact is stated formally below.

Theorem 7.4.31 *Every forbidding set is equivalent to a strict forbidding set.*

It should be noted that some posets may have infinitely many minimal upper bounds even for some two element forbidders. Chapter 5 considers one such poset,

namely (\mathcal{G}, \leq) . In such cases, converting a non-strict forbidding set to a strict one will be very cumbersome and impractical.

However, for posets in which every finite subposet has a unique minimal upper bound a unique minimal normal form can be achieved. The poset $(N, |)$ is one such example.

Proposition 7.4.32 *Let \mathcal{F} be a forbidding set over $(N, |)$. Let $\hat{\mathcal{F}} = \{\{s_F\} \mid F \in \mathcal{F}\}$. Then, let $\mathcal{F}' = \{\{s\} \in \hat{\mathcal{F}} \mid \text{there is no } \{h\} \in \hat{\mathcal{F}} \text{ with } h \leq s\}$. Then, \mathcal{F}' is connecting free, connecting incomparable, minimal, and unique and $\mathcal{F} \sim \mathcal{F}'$.*

The proof is omitted, since the proposition follows from a more general statement presented below.

Theorem 7.4.33 *Let P be a weakly extendable poset, such that every finite subposet of P has a unique minimal upper bound. Then, for every forbidding set \mathcal{F} over P , there exists a unique minimal forbidding set \mathcal{F}' such that $\mathcal{F} \sim \mathcal{F}'$.*

Proof. Let $\hat{\mathcal{F}} = \{\{s_F\} \mid F \in \mathcal{F}\}$. Let $\mathcal{F}' = \{\{s\} \in \hat{\mathcal{F}} \mid \text{there is no } \{h\} \in \hat{\mathcal{F}} \text{ with } h \leq s\}$. Obviously, \mathcal{F}' is connecting free and connecting incomparable. From Proposition 7.4.30 it follows that $\mathcal{F} \sim \mathcal{F}'$. Consider $\mathcal{F}'' = \mathcal{F}' \setminus \{s\}$. Then, $\mathcal{L}(\mathcal{F}') \subseteq \mathcal{L}(\mathcal{F}'')$. Clearly, $s \text{ ncon } \mathcal{F}'$. Let $\{h\} \in \mathcal{F}''$. If $h \in \underline{\text{sub}}(s)$, $\underline{\text{sub}}(\{h\}) \subseteq \underline{\text{sub}}(\{s\})$ which contradicts the definition of \mathcal{F}' . Therefore, $\{h\} \not\subseteq \underline{\text{sub}}(s)$ and $\mathcal{L}(\mathcal{F}') \subset \mathcal{L}(\mathcal{F}'')$, which establishes the minimality of \mathcal{F}' .

Let \mathcal{F}_1 and \mathcal{F}_2 are two such minimal forbidding sets with $\mathcal{F}_1 \sim \mathcal{F}_2$ and let $F \in \mathcal{F}_1$. Then, $F = \{s\}$ for some s and $s \notin \mathcal{L}(\mathcal{F}_2)$. It follows that there is $F' = \{h\} \in \mathcal{F}_2$ such that $F' \subseteq \underline{\text{sub}}(s)$, i.e., $h \in \underline{\text{sub}}(s)$. Since $h \notin \mathcal{L}(\mathcal{F}_1)$, it follows that there is $F'' = \{k\} \in \mathcal{F}_1$, such that $k \in \underline{\text{sub}}(h)$. This implies that $k \in \underline{\text{sub}}(s)$ and $\underline{\text{sub}}(\{k\}) \subseteq \underline{\text{sub}}(\{s\})$. Since both $\{k\}$ and $\{s\}$ are in \mathcal{F}_1 and \mathcal{F}_1 is subelement incomparable, it follows that $k = s$. Hence, $s = h$ and $\{s\} \in \mathcal{F}_2$. Thus, $\mathcal{F}_1 \subseteq \mathcal{F}_2$. Similarly, $\mathcal{F}_2 \subseteq \mathcal{F}_1$, which proves that $\mathcal{F}_1 = \mathcal{F}_2$. ■

7.5 Normal Forms for Enforcing Sets

This section discusses redundancy within enforcing sets. Note that the definition for weak enforcing is not a direct generalization of graph e -systems discussed in Chapter 5. There is no structure in the poset in general, so specific posets will have specific structure. Thus, in the graph (word) case there can be many copies of a single graph (word) in another graph (word), but in a general poset there is only one “occurrence” of a set of elements X in the subelements of an element.

The following discussion is for weak enforcing sets only. Normal forms for enforcing in graph fe -systems can be generalized directly for strong enforcing sets.

Remark 7.5.1 Let \mathcal{E} be an enforcing set. If $(X, Y), (X, Y') \in \mathcal{E}$ such that $Y \subseteq Y'$, then $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E}' = \mathcal{E} \setminus \{(X, Y')\}$.

Lemma 11.14 from [40] holds.

Lemma 7.5.2 Let \mathcal{E} be an enforcing set. If $(X, Y), (X', Y') \in \mathcal{E}$ such that $X \subseteq X'$ and $Y \subseteq Y'$, then $\mathcal{E} \sim \mathcal{E}'$ where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$.

As in the language and graph fe -systems, generated sets provide a more general definition of redundancy. Let \mathcal{E} be an enforcing set. Define $\mathcal{E}^{(1)} = \{X \mid (X, Y) \in \mathcal{E}\}$.

Definition 7.5.3 Let $X \in \mathcal{E}^{(1)}$. An element $g(X)$ is *generated* by X if the following two conditions hold:

- (i) $X \subseteq \underline{sub}(g(X))$
- (ii) $g(X) \underline{sat}(X', Y')$ for every $(X', Y') \in \mathcal{E}$.

A generated element $g_m(X)$ is called *minimal*, if for every generated element $p(X)$ $p \leq g_m(X)$, implies that $p = g_m(X)$.

Let \mathcal{E} be an enforcing set and let $X \in \mathcal{E}^{(1)}$. Denote the subposet of generated elements of X with respect to \mathcal{E} with $G_X^{\mathcal{E}}$ or simply G_X when \mathcal{E} is understood. The

subposet of minimal generated elements of X with respect to \mathcal{E} is denoted by $M_X^\mathcal{E}$ or simply M_X when \mathcal{E} is understood. Let $\mathcal{M}(\mathcal{E}) = \cup_{X \in \mathcal{E}^{(1)}} M_X$.

The above definition of generated elements allows the elements $g_m(X)$ to be such that $\underline{sub}(g_m(X))$ is infinite. Note that if $w \underline{sat} \mathcal{E}$ and $X \in \mathcal{E}^{(1)}$ is such that $X \subseteq \underline{sub}(w)$, then $w \in G_X$.

Example 7.5.4 Let $\mathcal{E} = \{(\{2\}, \{3, 4\}), (\{3\}, \{5, 7\})\}$. Some of the elements in $\mathcal{L}(\mathcal{E})$ are 1, 4, 5, 7, and 30, while 2, 3, and 6 do not satisfy \mathcal{E} . The minimal generated sets for $\{2\}$ are $g_m(2) = \{2, 4\}$, $g_m(2) = \{2, 3, 5\}$, and $g_m(2) = \{2, 3, 7\}$. The set $\{2, 4, 7\}$ is generated, but not minimal. The set $\{2, 3\}$ is not a generated set and does not satisfy \mathcal{E} .

Example 7.5.5 Consider $(\mathcal{F}, \mathcal{E})$ with $\mathcal{F} = \{\{5\}\}$ and \mathcal{E} as in Example 7.5.4. Then $90 \in \mathcal{L}(\mathcal{E})$ but $90 \notin \mathcal{L}(\mathcal{F}, \mathcal{E})$.

Redundancy for enforcing sets can be defined as in Chapter 2.

Definition 7.5.6 Given \mathcal{E} , the enforcer (X', Y') is *redundant* for \mathcal{E} , if there exists an enforcer $(X, Y) \in \mathcal{E}$, with $X \subseteq X'$ and $g_m(X) \cap Y'$ for every $g_m(X) \in M_X^{\mathcal{E}'}$ where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$

In particular, if $X \subseteq X'$ and $Y \subseteq Y'$ with $X' \subseteq \underline{sub}(y)$ for every $y \in Y$, then (X', Y') is redundant.

Example 7.5.7 Consider $(N, |)$ with $\mathcal{E} = \{(\{2\}, \{3\}), (\{3\}, \{5, 7\}), (\{2, 13\}, \{5, 7, 11\})\}$. Observe that any even number that satisfied the first two enforcers is divisible by 5 or 7, which satisfies the third enforcer. In other words, the minimal generated sets of $\{2\}$ in \mathcal{E}' are $\{2, 3, 5\}$ and $\{2, 3, 7\}$ and they both intersect $\{5, 7, 11\}$.

The following lemma from Chapter 2 holds in this case, as well. It shows that redundant enforcers can be erased from the enforcing set.

Lemma 7.5.8 *If (X', Y') is redundant for \mathcal{E} , then $\mathcal{L}(\mathcal{E}) = \mathcal{L}(\mathcal{E}')$, where $\mathcal{E}' = \mathcal{E} \setminus \{(X', Y')\}$.*

An enforcing set \mathcal{E} is said to be *finitary* if for all $X \in \mathcal{E}^{(1)}$ there are finite number of enforcers (X, Y_i) in \mathcal{E} . It is shown in [8] that every infinite enforcing set is equivalent to a finitary enforcing set in the case of language *fe*-systems. The following example shows that if P contains an infinite chain, there exists an infinite finitary enforcing set for which $\mathcal{M}(\mathcal{E})$ is finite.

As in the case with family of subsets (and families of languages), $\mathcal{M}(\mathcal{E})$ can be finite or infinite with “finite” or “infinite” elements.

Example 7.5.9 Let $C = \{w_1, w_2, \dots\}$ be an infinite chain in P , i.e., $w_i \leq w_{i+1}$ for every $i \geq 1$. Consider the enforcing set $\mathcal{E} = \{(\{w_1, w_2\}, \{w_3\}), (\{w_2, w_3\}, \{w_4\}), (\{w_2, w_4\}, \{w_1\})\} \cup \{(\{w_n, w_{n+1}\}, \{w_{n+2}\}), (\{w_n, w_{n+2}\}, \{w_1\}), (\{w_1, w_n\}, \{w_2\}) \mid n \geq 3\}$. It is obvious that this enforcing set is infinite and finitary. Notice that $\mathcal{M}(\mathcal{E})$ is a singleton and its only generated set contains all elements in Z . In other words, $M_X = Z$ for all $X \in \mathcal{E}^{(1)}$.

The following lemma shows that an infinite finitary enforcing set with finite $\mathcal{M}(\mathcal{E})$ must have an infinite generated element.

Proposition 7.5.10 *Let P contain infinite elements. Let \mathcal{E} be infinite and finitary, such that $\mathcal{M}(\mathcal{E})$ is finite. Then there exists an infinite generated element.*

Proof. Since $\mathcal{M}(\mathcal{E})$ is finite, there is a finite number of families of minimal generated elements M_X . Denote these (families) sets by M_1, M_2, \dots, M_k , i.e., $\mathcal{M}(\mathcal{E}) = \bigcup_{i=1}^k M_i$. Since there are infinitely many distinct X 's (due to \mathcal{E} being infinite) and finitely many M_i 's, there must exist at least one M_j such that for infinitely many X 's in $\mathcal{E}^{(1)}$, we have $M_X = M_j$. Let $g_m(X) \in M_j$. Since $g_m(X)$ is a generated element for infinitely many X 's, it follows that $g_m(X)$ contains all these X 's in $\text{sub}(g_m(X))$. Hence, $g_m(X)$ (and each generated element in M_j) is infinite. ■

Corollary 7.5.11 *Let P contain finite elements only. If \mathcal{E} is infinite and finitary, then $\mathcal{M}(\mathcal{E})$ is infinite.*

Proof. Suppose that $\mathcal{M}(\mathcal{E})$ is finite and proceed as in the above proof. This implies that an infinite generated element exists, which contradicts the fact that P contains finite elements only. Hence, $\mathcal{M}(\mathcal{E})$ is infinite. ■

Chapter 8

Computing with *fe*-Systems

Forbidding and enforcing systems were inspired by molecular reactions. Naturally, their computing capabilities are being studied in general and through specific models. In [8] the structure of computations in a *fe*-system is represented by a tree, called Γ -tree. The authors in [8], also, show how *fe*-systems can model the 3-SAT problem and the structure of DNA molecules and operations on them. Another computational aspect of *fe*-system is studied in [5] where two new variants of membrane systems are defined using *fe*-systems. In [13], *fe*-systems are used to model 3D DNA self-assembled graph structures.

The *fe*-systems definitions presented in the previous chapters were used to *define*, rather than derive the family of structures that obeys the *fe*-system. In computing, the computation process begins with a set of initial conditions. Therefore, additional definitions are introduced to obtain a family of structures *derived* by an *fe*-system.

An *fe*-system *generated by object* K is a pair $\gamma = (K, \Gamma)$ where K is an object (i.e., language, graph, subposet, etc.) and Γ is an *fe*-system. One can consider K as a set or structure defining the initial conditions. The class of objects defined by γ is $\mathcal{L}(\gamma) = \{L \mid K \leq L \text{ and } L \in \mathcal{L}(\Gamma)\}$. Suppose K and L are two objects which generate γ_K and γ_L respectively. Then, $K \leq L$ implies $\mathcal{L}(\gamma_L) \subseteq \mathcal{L}(\gamma_K)$.

8.1 Modeling Molecular Bonding and Splicing Systems

In DNA computing molecules and atoms are abstracted with symbols and larger molecules or macromolecules with strings (words) over a specified alphabet. In this case the suitable fe -systems variant to model molecules is the one defined on languages as described in Chapter 2.

Bonding relationships (both covalent and weak hydrogen or ionic bonds) in molecules can be described through fe -systems of languages. As an illustration, this section considers the structure of the double stranded DNA and the actions of endonucleases and ligases.

A DNA strand can be conveniently represented as a string over a suitable alphabet. The basic alphabet is $A = \{a, c, t, g\}$ where the symbols represent the four different kinds of bases of nucleotides in the DNA molecule: “adenine”, “cytosine”, “thymine”, and “guanine”. Watson-Crick complementarity dictates that a pairs with t and g with c . For a symbol $x \in A$, its Watson-Crick complement is denoted with \bar{x} . This notation is extended to words over A with \bar{w} being the complement of w , w^R being the reverse of w , and \bar{w}^R the reverse complement of w . For example, if $w = aatcga$ then $\bar{w} = ttacct$ and $\bar{w}^R = tcgatt$. Thus, \bar{w}^R is the complement of w^R . Each nucleotide consists of a sugar, a phosphate group and a nitrogenous base. A strand of DNA is obtained when a covalent bond is established between the phosphate group of one nucleotide and the hydroxyl group (from the sugar) of another nucleotide. This leaves a phosphate on one end, standardly denoted as 5' and a hydroxyl group on the other end, standardly denoted as 3'. Driven by Watson-Crick complementarity two strands of DNA can form a double stranded molecule by forming hydrogen bonds between complementary nucleotides. A base pair of nucleotides can be represented with a pair of symbols $\begin{pmatrix} x \\ \bar{x} \end{pmatrix}$, where $x \in A$ and \bar{x} is the Watson-Crick complement of x . Similarly, a piece of double stranded molecule is represented as a concatenation of such symbols, say $\begin{pmatrix} 5' aatcga 3' \\ 3' ttacct 5' \end{pmatrix}$. It is customary to write the upper strand in the 5' – 3' direction, (which orients the lower strand in the 3' – 5' direction) and thus omit the 5' and 3' from the notation.

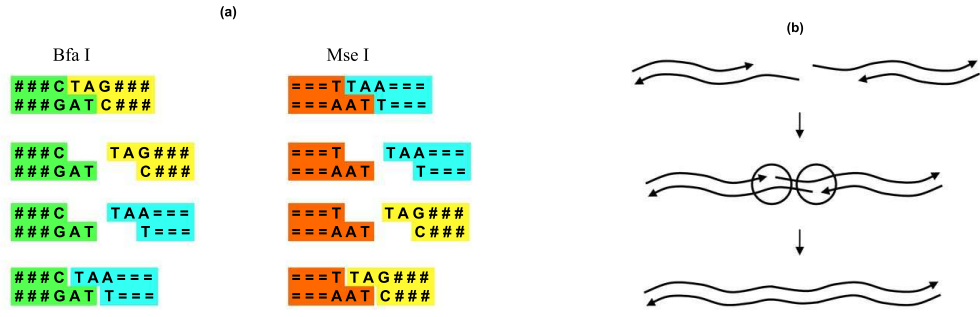


Figure 8.1: (a) Cutting with restriction enzymes (b) DNA recombination

For example, the above molecule can be written simply as $\begin{pmatrix} aatcga \\ ttagct \end{pmatrix}$. The situation when a covalent bond between two nucleotides in a double-stranded DNA molecule is missing is called a “nick”.

Endonucleases recognize short sequences of DNA and perform “cuts”, which are either blunt or leave “sticky” overhangs. Figure 8.1 left shows examples of the recognition sites of the enzymes *BfaI* and *MseI*. Even though, both enzymes have different recognition sites they leave the same (single stranded) 5' overhang *ta*. Two strings with complementary overhangs can “stick” together and form a completely new string. Thus, a molecule with 5' overhang *ta* on top attaches to a lower strand overhang *at*, by forming a hydrogen bond between *t* and *a* and *a* and *t* respectively. A ligase glues the nicks at the positions where the molecules have joined establishing a covalent bond between *t* and the nucleotide next to it on the top strand. Similarly, the lower strand 5'-end of *a* forms a covalent bond with the 3'-end of the adjoining nucleotide. The above described operation of “cutting” and “pasting” with enzymes is known in literature as *splicing*. Figure 8.1 right shows the two steps of DNA recombination.

Consider the alphabet $\Sigma = \{ \begin{pmatrix} x \\ \bar{x} \end{pmatrix}, \begin{pmatrix} x \end{pmatrix}, \begin{pmatrix} \cdot \\ x \end{pmatrix}, \begin{pmatrix} *x \\ \bar{x} \end{pmatrix}, \begin{pmatrix} x \\ \bar{x}* \end{pmatrix}, \begin{pmatrix} \cdot \\ *x \end{pmatrix}, \begin{pmatrix} \cdot \\ x* \end{pmatrix} \mid x \in A \}$. Here $*$ is used to indicate an existing nick or to identify the 5' end. The \cdot indicates a missing nucleotide, so $\cdot \cdot$ indicate two missing nucleotides. This alphabet is similar to the one used in [8]. To ease the notation the inner parentheses are ignored when concatenating symbols.

A language of valid single or double stranded DNA belongs to the family $\mathcal{L}(\Gamma)$



Figure 8.2: Missing nucleotides: (a) diagonally (b) in one strand.

where Γ is the *fe*-system $\Gamma = (\mathcal{F}, \mathcal{E})$ with $\mathcal{F} = \left\{ \left\{ \binom{x \cdot}{\cdot y} \right\}, \left\{ \binom{\cdot x}{y \cdot} \right\}, \left\{ \binom{s \cdot \bar{z}}{\cdot z} \right\}, \left\{ \binom{\bar{z} \cdot s}{z \cdot} \right\} \mid x \in A \cup *A, y \in A \cup A*, s, z \in A \right\}$ and enforcers $\mathcal{E} = \left\{ \left(\{z\}, \{z, \bar{z}^R\} \right) \mid z = \binom{w}{\bar{w}}, \bar{z}^R = \binom{\bar{w}^R}{w^R} \text{ for some } w = x \text{ or } w = *x \text{ where } x \text{ is a word in the alphabet } A \right\}$. Each one of the four types of forbidders describes the situation when the covalent bond between the phosphate and the hydroxyl group is missing, hence the words do not represent a valid chain of nucleotides, or a valid double stranded DNA. The first two types of forbidders $\left\{ \binom{x \cdot}{\cdot y} \right\}$ (depicted in Figure 8.2 (a)) and $\left\{ \binom{\cdot x}{y \cdot} \right\}$ state that a double stranded molecule cannot be formed when two nucleotides positioned diagonally are missing. The third $\left\{ \binom{s \cdot \bar{z}}{\cdot z} \right\}$ and fourth $\left\{ \binom{\bar{z} \cdot s}{z \cdot} \right\}$ (depicted in Figure 8.2 (b)) types of forbidders state that a covalent bond cannot be formed in one strand when the other strand has missing nucleotides. An additional requirement for a valid DNA molecule is that if some double string z denotes a molecule α , then its reversed complement \bar{z}^R is in fact the same molecule and it should be enforced. Hence $\mathcal{L}(\Gamma)$ denotes the class of all possible single or double stranded DNA.

The process of ligating, or closing the “nicks” is defined with $\mathcal{E}' = \left\{ (X = \{w \binom{x \cdot y}{\bar{x} \bar{y}} z\}, X \cup \{w \binom{xy}{\bar{x}\bar{y}} z\}), (X' = \{u \binom{xy}{\bar{x}\bar{y}} v\}, X' \cup \{u \binom{xy}{\bar{x}\bar{y}} v\}) \mid x, y \in A \text{ and } u, v, w, \text{ and } z \text{ are words over } \Sigma \right\}$.

Cutting with *BfaI* can be modeled with $\mathcal{E}_{BfaI} = \left\{ (X = \{w \binom{ctag}{gatc} z\}, Y = X \cup \{w \binom{c \cdot \cdot}{gat*}\}) \mid \text{for each pair of words } w \text{ and } z \right\}$. Then, recombination that occurs when two pieces with overhang *ta* one from *BfaI* and another one from *MseI* anneal can be modeled in two steps. First, annealing (self-assembly by the hydrogen bonds) can be expressed by $\mathcal{E}_{ta} = \left\{ (X = \{w \binom{c \cdot \cdot}{gat*}, \binom{*ta}{\cdot \cdot t} z\}, X \cup \{w \binom{c*ta}{gat*} z\}) \mid \text{for each pair of words } w \text{ and } z \right\}$. Then, the ligation is obtained by the enforcers \mathcal{E}' .

Now consider a molecular mix with an initial set of DNA strands K , a set

of endonucleases E , and a ligase. Let \mathcal{E}_e be a set of enforcers for each $e \in E$ equivalent to the ones described for *BfaI* together with the necessary enforcers for annealing overhangs. Let $\Gamma = (\mathcal{F}, \mathcal{E} \cup \mathcal{E}' \cup_{e \in E} \mathcal{E}_e)$. Then, $\mathcal{L}(\Gamma)$ defines all possible sets of molecules that can occur. The *fe*-system generated by K is $\gamma = (K, \Gamma)$ and describes the set of all DNA strands that can be obtained by recombination of the original set of molecules with the given enzymes. This is an equivalent definition for splicing systems initially defined as string rewriting type of rules in [19].

8.2 Information Processing by *fe*-Systems

It is possible to describe solutions to computational problems through *fe*-systems. Section 8.1 shows how *fe*-systems can describe the languages that are generated by the well-known computational model of splicing systems (see for example [36, 33]). This sections shows how a solution to a well-known NP complete problem, the k -colorability problem can be described by *fe*-systems.

Let S be a set and $\mathcal{P}(S)$ its powerset. Consider the poset $(\mathcal{P}(S), \subseteq)$, where \subseteq is the subset relation. The k -colorability problem asks whether given a graph and a finite set of k colors it is possible to assign one color to each vertex in such a way that adjacent vertices have distinct colors. Such assignment of colors is called a k -coloring.

Let $G = (V, E)$ be a graph with n vertices, i.e., $V = \{v_1, v_2, \dots, v_n\}$ and C be a set of k colors i.e., $C = \{c_1, c_2, \dots, c_k\}$. A k -coloring will be viewed as a subset of $V \times C$.

Consider the poset $P = (\mathcal{P}(V \times C), \subseteq)$ and the following *fe*-system. Every vertex should be assigned exactly one color which can be done by a combination of brute enforcing and forbidding. The brute enforcing $\mathcal{E} = \{(\emptyset, \{(v, c_1), (v, c_2), \dots, (v, c_k)\}) \mid v \in V\}$ ensures that every vertex is assigned *at least one* color. The forbidding set $\mathcal{F} = \{\{(v, c), (v, c')\} \mid v \in V \text{ and } c, c' \in C \text{ with } c \neq c'\}$ allows only these vertices that are assigned at most one color.

Also, no two adjacent vertices should be colored the same. This is obtained with

the following forbidders: $\mathcal{F}' = \{\{(u, c), (v, c)\} \mid \{u, v\} \in E \text{ and } c \in C\}$.

Let $\Gamma = (\mathcal{F} \cup \mathcal{F}', \mathcal{E})$ be an *fe*-system. Then G is k -colorable if and only if $\mathcal{L}(\Gamma) \neq \emptyset$. Further more, any set $K \in \mathcal{L}(\Gamma)$ contains exactly one k -coloring of G . Note that up to permutations of the colors, all solutions to the k -colorability problem can be obtained by fixing one vertex and one color. So, set $\bar{v} \in V$ and $\bar{c} \in C$ and let $K = \{(\bar{v}, \bar{c})\}$. Then G is k -colorable if and only if $\mathcal{L}(K, \Gamma) \neq \emptyset$. Further more, any set $K \in \mathcal{L}(K, \Gamma)$ contains exactly one k -coloring of G .

Conclusions

This work introduces a new way of classifying structures through specifying boundary constraints of forbidden and enforced structures. Each one of the different types of fe -systems presented discusses a different poset and restrictions. In the case of formal languages, the poset of all words with subword order is used and then families of subposets are investigated. The language fe -systems were first defined in [8] and studied in [9, 10, 40]. This work and [16] show that forbidding-enforcing systems (fe -systems) provide a completely new way to define classes of languages. None of the Chomsky families can be defined in this way. This paper investigates topological and morphic properties of language fe -systems. Although it characterizes extended f -families (Theorem 2.4.3), a characterization of f -families may be of interest. Also, any characterization of e -families remains to be investigated. The author believes that the introduction of minimal generated sets is the first step towards this goal.

Morphisms are natural maps to consider between languages. Although this work characterizes the morphisms that map f -families to extended f -families and provides results about morphic images of f -families and e -families, the question of what morphisms map fe -families into fe -families remains open.

Normal forms provide a foundation for studying any type of fe -systems. In language fe -systems, [8, 9, 40] present normal forms for forbidding sets and for enforcing sets. This paper provides new normal forms for language fe -systems. Even if a forbidding set is in minimal normal form and an enforcing set is in finitary normal, when combined in a fe -system, the system as a whole may be reduced further. In this respect, any “interaction” between the forbidding set and the enforcing set of an fe -system should be investigated further.

The poset of graphs with the subgraph order where fe -systems define a subposet (class of graphs) provides a new way of classifying graphs. The definitions and properties of fe -graphs led to a new graph-theoretical notion - connecting graphs of a set

of graphs. This work presents several properties of connecting graphs. An interesting future direction is investigating further the properties of connecting graphs as a separate topic. However, in a separate direction, the results for connecting graphs related to graph fe -systems should be explored. In this paper and in [15] several characterizations of familiar classes of graphs are presented (e.g., trees, paths, complete graphs, and k -regular graphs are characterized using graph fe -systems). More such characterizations can be pursued (e.g., planar graphs).

This paper states some normal forms for forbidding sets and enforcing sets for graph fe -systems. Connecting free and connecting incomparable are shown to be minimal in some cases. Continuing the investigation of minimal and unique normal forms for forbidding sets is the natural next step. Enforcing sets are inherently difficult to study. New normal forms for enforcing sets would be of interest. Generating sets may provide a natural first step in this direction. It is worth mentioning that many other graph fe -systems models can be defined and investigated. If in the forbidding definition the word *subgraph* is replaced with *induced subgraph* then a general version of the historical concept of forbidding sets is obtained. A comprehensive list of papers considering such type of forbidding where the forbidding sets are strict is available at [18]. Non-strict forbidding sets for induced subgraphs have not yet been discussed in general. Since there are a lot of conditions for hamiltonicity described through strict “induced” forbidding sets, a more general definition of allowing not only singleton enforcers can provide direct generalizations. Other versions of enforcing can, also, be pursued. It will be interesting to know whether relaxing the embedding condition for enforcers (“weak” enforcing) leads to new characterizations. Also, restricting the first component of an enforcer to connected graphs may lead to new graph fe -systems properties. Another variant of enforcing sets that can be defined is a model where the first component of an enforcer is a set of connected graphs, not just one graph.

A generalization of the weak version of enforcing to posets is presented in Chapter 7. It can, also, be defined on some specific posets like A^* with subword order. Considering a topology on A^* (see [34]) may enhance the understanding of fe -

systems on (A^*, \underline{sub}) .

Although the general types of fe -systems should be studied in the context of posets and even categories, investigating each variant of fe -systems in a specific poset (for example languages, words, graphs, groups, and matrices) may provide for new properties of fe -families due to the inherent structure of the poset. Investigating fe -systems in the poset of geometric structures made of small building blocks may provide a completely new approach into studying crystals and self-assembly processes.

The computing potential of fe -system is yet to be investigated in detail. Both theoretical and practical results should be pursued.

References

- [1] L. Adleman, *Molecular computation of solutions to combinatorial problems*, Science, **226** (1994), 1021-1024.
- [2] B. Bollobas, *Graph theory*, Springer-Verlag, New York (1979).
- [3] M. Cavaliere, D. Genova, *P Systems with Symport/Antiport of Rules*, Dept. of Computer Sciences and Artificial Intelligence, Univ. of Sevilla Tech. Rep 01/2004, Gh. Paun, A. Riscos-Nunez, A. Romero-Jimenez, F. Sancho Caparrini (Eds.), Second Brainstorming Week on Membrane Computing, Sevilla, Spain, Feb 2-7, 2004, 102–116.
- [4] M. Cavaliere, D. Genova, *P Systems with Symport/Antiport of Rules*, Journal of Universal Computer Science, **10** No. 5 (2004) 540–558.
- [5] M. Cavaliere, N. Jonoska, *Forbidding and enforcing in membrane computing*, Natural Computing **2** (2003) 215-228.
- [6] B. A. Davey, H. A. Priestley *Introduction to lattices and order*, Cambridge University Press, (1990).
- [7] D. Duffus, R. J. Gould, M. S. Jacobson, *Forbidden subgraphs and the hamiltonian theme*, The Theory and applications of graphs, ed. G. Chartrand, J. Alavi, D. Goldsmith, L. Lesniak, and D. Lick (1981) 297-316.
- [8] A. Ehrenfeucht, G. Rozenberg, *Forbidding-enforcing systems*, Theoretical Computer Science Vol. **292** (2003) 611-638.
- [9] A. Ehrenfeucht, H.J. Hoogeboom, G. Rozenberg, N. van Vugt, *Forbidding and enforcing* (E. Winfree, D.K. Gifford, eds.) AMS DIMACS series Vol. **54** (2000) 195-206.

- [10] A. Ehrenfeucht, H.J. Hoogeboom, G. Rozenberg, N. van Vugt, *Sequences of languages in forbidding-enforcing families*, Soft Computing Vol. **5** No. 2 (2001) 121-125.
- [11] J. Engelfriet, H.J. Hoogeboom, *X-automata on ω -words*, Theoretical Computer Science. **110** No. 1 (1993), 1–51.
- [12] R. J. Faudree, R. J. Gould, *Characterizing forbidden pairs for hamiltonian properties*, Discrete math. 173 (1997) 45-60.
- [13] G. Franco, N. Jonoska, *Forbidding-Enforcing Conditions in DNA Self-Assembly of Graphs*, in Nanotechnology: science and computing (J. Chen, et. al. eds.) Springer-Verlag (2006) 105 – 118.
- [14] R. Freund, M. Oswald, *P Systems with Antiport Rules for Evolution Rules*, Dept. of Computer Sciences and Artificial Intelligence, Univ. of Sevilla Tech. Rep 01/2004, Gh. Paun, A. Riscos-Nunez, A. Romero-Jimenez, F. Sancho Capparrini (Eds.), Second Brainstorming Week on Membrane Computing, Sevilla, Spain, Feb 2-7, 2004.
- [15] D. Genova, N. Jonoska, *Defining Structures through Forbidding and Enforcing Constraints*, Physica B: Physics of Condensed Matter, Elsevier **394** No. 2 (2007) 306-310
- [16] D. Genova, N. Jonoska, *Topological Properties of Forbidding-enforcing Systems*, Journal of Automata, Languages, and Combinatorics, (accepted).
- [17] R. J. Gould, M. S. Jacobson, *Forbidden subgraphs and hamiltonian properties of graphs*, Discrete Math. 42 (1982), 189-196.
- [18] J. L. Gross, J. Yellen, *Graph Theory and Its Applications*, CRC Press, (1999).
- [19] T. Head, *Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors*, Bull. Math. Biol. **49** No.6 (1987), 737-759.
- [20] T. Head, Gh. Paun, D. Pixton, *Language theory and molecular genetics*, Handbook of formal languages, (G. Rozenberg, A. Salomaa editors), Springer Verlag, **2** (1997) 295-358.

- [21] S. Hussini, L. Kari, S. Konstantinidis, *Coding properties of DNA languages*, *Theoretical Computer Science* **290** (2003) 1557–1579.
- [22] N. Jonoska, P. Sa-Ardien, N. C. Seeman, *Computation by self-assembly of DNA graphs*, *Genetic Programming and Evolvable Machines* **4** (2003) 123–137.
- [23] N. Jonoska, K. Mahalingam, J. Chen, *Involution codes: with application to DNA coded languages*, *Natural Computing*, **4** no. 2 (2005) pp. 141-162.
- [24] N. Jonoska, G. L. McColm, *Flexible versus Rigid Tile Assembly* in: C. S. Calude, et al, Proc. 5th Internat'l Conf. Unconv. Comp., Springer LNCS 4135 (2006) 139–151.
- [25] D. Kephart, *Topology, Morphisms and Randomness in the space of Formal Languages*, PhD thesis, University of South Florida, Tampa FL, USA, 2005.
- [26] B. Kitchens, *Symbolic Dynamics*, Springer-Verlag, New York, (1998).
- [27] D. Lind, B. Marcus, *An Introduction to Symbolic Dynamics and Coding*, Cambridge University Press, Cambridge, (1995).
- [28] K. Mahalingam, *Involution codes: with applications to DNA strand design*, PhD thesis, University of South Florida, Tampa FL, USA, 2004.
- [29] J. Munkres, *Topology*, Prentice Hall, Upper Saddle River, (2000).
- [30] Gh. Păun, *Membrane Computing. An Introduction*, Springer, Berlin, (2002).
- [31] Gh. Păun, *Membrane Computing: Some Non-Standard Ideas*, Aspects of Molecular Computing, Essays Dedicated to Tom Head on the Occasion of His 70th Birthday, (N. Jonoska, G Rozenberg, eds.), LNCS 2950, Springer-Verlag, Berlin (2004), 322–377.
- [32] Gh. Păun, G. Rozenberg, *A guide to membrane computing*. Theoret. Comput. Sci. Vol. **287** No. 1 (2002) 73–100.
- [33] Gh. Paun, G. Rozenberg, A. Salomaa, *DNA Computing, new computing paradigms*, Springer Verlag (1998).
- [34] J. E. Pin, *Varieties of formal languages*, Plenum Press (1986).
- [35] J. H. Reif, S. Sabu, P. Yin, *Complexity of Graph Self-Assembly in Accretive Sys-*

- tems and Self-Destructive Systems*, Proceedings of 11th International Meeting on DNA Computing, London, Ontario, Canada, June 6-9 (2005) 101–112.
- [36] G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*, Springer-Verlag, Berlin (1997).
- [37] N.C. Seeman, *De Novo design of sequences for nucleic acid structural engineering*, Journal of Biomolecular Structure and Dynamics, **8** no. 3 (1990) 573-581.
- [38] B. Schroeder, *Ordered sets: an introduction*, Birkhouser Boston, (2003).
- [39] L. Staiger, *ω -languages*, Handbook of Formal Languages, G. Rosenberg, A. Salomaa (eds.), Springer Verlag, Berlin, **3** (1997), 339–387.
- [40] N. van Vugt, *Models of Molecular Computing*, PhD thesis, Leiden University, May (2002).
- [41] E. Winfree, X. Yang, N.C. Seeman, *Universal Computation via Self-assembly of DNA: Some Theory and Experiments*. in DNA computers II, eds L. Landweber and E. Baum, AMS DIMACS series Vol. **44** (1998) 191–198.
- [42] Y. Zhang, N. C. Seeman, *The construction of a DNA truncated octahedron*, J. Am. Chem. Soc. **160** (1994) 1661–1669.

About the Author

Daniela Genova obtained Bachelor's of Arts in Business Administration/Accounting in 1995, Master's of Accounting/Taxation in 1997, and completed requirements for M.A. degree in Mathematics in 1999, at the University of South Florida, Tampa, FL. Her research interests are in the area of formal language theory and unconventional models of computation like DNA Computing and Membrane Computing. She presented her research in several local, state, and international conferences and co-authored three journal publications: in Membrane Computing with M. Cavaliere; in Formal Languages with N.Jonoska; and in Set and Graph Theory with N. Jonoska.