Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS

Tomáš Toufar

Algorithmic metatheorems for Matroids

Computer Science Institute of Charles University

Supervisor of the bachelor thesis: RNDr. Ondřej Pangrác, Ph.D.
Study programme: Computer science
Specialization: General computer science

Prague 2014

Acknowledgement

I would like to express thanks to my former advisor Daniel Král', who introduced me to the area of algorithmic metatheorems and supervised the early phase of writing of my thesis.

My sincere thanks goes to my advisor Ondřej Pangrác for his patience and helpful suggestions and also for excellent lectures on matroid theory at Charles University.

I would also like to thank to my friend Lukáš Mach, who managed to successfully present the results of the thesis at the IPEC workshop.

Last but not least, I would like to express my gratitude to my parents for their everlasting support. Without their encouragement, I would never be able to complete my thesis.

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague, December 5, 2014 Tomáš Toufar

Název práce: Algoritmické metavýsledky

Autor: Tomáš Toufar

Ústav: Informatický ústav Univerzity Karlovy

Vedoucí bakalářské práce: RNDr. Ondřej Pangrác, Ph.D.

Abstrakt: V práci definujeme nový šířkový parametr pro matroidy nazvaný *amalgamační šířka*. Tento šířkový parametr vychází z operace *amalgamace* matroidů. Parametr má úzký vztah k větvící šířce (branch width) na matroidech reprezentovatelných nad pevně zvoleným konečným tělesem – reprezentovatelné matroidy s omezenou větvící šířkou mají omezenou i amalgamační šířku. Přitom jsme stále schopni rozhodovat vlastnosti v monadické logice druhého řádu v lineárním čase pro matroidy s omezenou amalgamační šířkou a to i tehdy, když matroid není reprezentovatelný (pokud ovšem máme dekompozici danou). Navíc dokážeme spočíst koeficienty Tutteho polynomu matroidu v polynomiálním čase na třídách matroidů s omezenou amalgamační šířkou.

Klíčová slova: matroidy, algoritmy, šířkové parametry, monadická logika druhého řádu, Tutteho polynom

Title: Algoritmic metatheorems for matroids

Author: Tomáš Toufar

Department: Computer Science Institute of Charles University

Supervisor: RNDr. Ondřej Pangrác, Ph.D.

Abstract: In the thesis we define a new width parameter for matroids called *amalgam width* that is based on the operation of *matroid amalgamation*. The parameter is related to branch width for matroids representable over fixed finite field in the sense that class of representable matroids of bounded branch width has bounded amalgam width. The decomposition allows us to decide monadic second-order properties in linear time on matroids of bounded amalgam width, even for matroids that are not representable (provided we are given the decomposition). We also prove that the coefficients of the Tutte polynomial can be computed in polynomial time for matroids of bounded amalgam width.

Keywords: matroids, algorithms, width parameters, monadic second order logic, Tutte polynomial

Contents

Introduction

In the 1970s, some mathematicians noticed that a lot of very hard combinatorial problems can be solved efficiently by a variant of dynamic programming on graphs of bounded dimension [BB73]. A decade later, a parameter called *tree width* introduced by Robertson and Seymour [RS84, RS86] proved to be more useful for solving NP-complete problems on graphs. Such results have given rise to a whole field called *parameterized complexity* [DF99].

One of the most prominent results of parameterized complexity is the celebrated *Courcelle's theorem* [Cou90], stating that every property definable in *monadic second-order logic* can be decided in linear time on graphs given by a tree decomposition of bounded width. Together with the efficient construction of tree decompositions by Bodlaender [Bod96], we obtain that every property definable in MSO logic is decidable in linear time on graphs of bounded tree-width.

Because this result actually covers almost all theorems stating "Problem P is solvable in linear time on classes of graphs of bounded tree-width", it is often referred to as a *metatheorem*, or more specifically an *algorithmic metatheorem*.

It was natural to ask if similar results can be achieved for matroids. Even though tree-width can be defined for matroids [HW06, HW09], *branch-width* proved to be more natural width parameter for matroids. Algorithmic metatheorems usually have two parts: computing decompositions of small width; and solving hard problems on such decompositions. The solution to the first part has been found by Oum and Seymour [OS06] and FPT algorithm was later designed by Hliněný and Oum [HO08].

However, tractability of hard problems is a different matter. Algorithms deciding monadic second-order formulas [Hli06a] or evaluating Tutte polynomial [Hli06b] usually require the input matroid to be represented over a finite field. This is supported by the fact that it cannot be decided in subexponential time whether a matroid given by independence oracle is binary [Sey81] (i.e. representable over GF(2)). This result holds even for matroids of bounded branch-width. Some results about matroids represented over rationals [Hli06c] as well as several structural results [GGW02, GGW06, GGW07] suggest that matroids representable over finite fields are closely related to graphs, while general matroids can be much more complex.

Two width parameters were proposed to allow decision of MSO formulas on non-representable matroids – decomposition width by Kráľ [Kra12], and another decomposition by Strozecki [Str11] based on 2-sums. However, the latter allows the matroid to be split only along 2-separations, thus prohibiting non-trivial decompositions for 3-connected matroid. The first one does not suffer from this problem and allows splitting along more complex separation, but this operation does not correspond to any natural operation on matroids.

We propose a parameter called *amalgam-width* that is based on a natural "gluing" operation allowing joins along complex separations, yet still allows us to prove desired algorithmic results.

In Chapter 1, we introduce the notion of a matroid. Note that this is only a brief introduction, for the thorough treatment of matroid theory, we refer the reader to the monograph from Oxley [Oxl06]. We continue with the exposition of matroid amalgams in Chapter 2. The central concept — amalgam decomposition — is introduced in Chapter 3. The two key results of thesis, computing the Tutte polynomial and deciding monadic second-order formulas for matroids given by amalgam decomposition, are covered in Chapters 4 and 5, respectively. Finally, in Chapter 6, we show how can we construct an amalgam decomposition given an $\mathbb{F}$-representation of a matroid.

Some of the results of the thesis were presented by Lukáš Mach at IPEC symposium in 2013. The submitted paper was published at arXiv [MT13]. Note that the paper is much more brief and some of the definitions differ sightly from the final version of the thesis.

1. Matroid definitions

In this chapter, we introduce matroids which are objects of our main interest in this thesis. However, this chapter does not cover all the properties of matroids. The purpose of this chapter is to get the reader familiar with the terminology and the notation we are using throughout this thesis. For thorough introduction to matroids, we refer the reader to the monograph by Oxley [Oxl06].

1.1 Basic definitions and axiomatics

Matroids can be defined in several ways, we now present the most common definition of a matroid.

Definition 1.1.1. A *matroid* M is an ordered tuple $(E, \mathcal{I})$ where E is a finite set and $\mathcal{I}$ is subset of 2^E satisfying the following conditions:

(I1) $\emptyset \in \mathcal{I}$,

(I2) if $I' \subseteq I$ and $I \in \mathcal{I}$ then $I' \in \mathcal{I}$, and

(I3) if $I_1, I_2 \in \mathcal{I}$ and $|I_1| < |I_2|$, then there exists $x \in I_2 \setminus I_1$ such that $I_1 \cup \{x\} \in \mathcal{I}$.

The set E is called the *ground set*, the members of E are called *elements* of M and the sets in $\mathcal{I}$ are called *independent sets*. The subsets of E not present in $\mathcal{I}$ are referred to as *dependent sets*. We also write $E(M)$ to denote the ground set of M and $\mathcal{I}(M)$ for the family of independent sets. If the matroid is clear from the context and no confusion can arise, we simply write E and $\mathcal{I}$.

Definition 1.1.2. If $M = (E, \mathcal{I})$ is a matroid and X is a subset of E, we denote by $M|X$ the *restriction of* M *to* X, which is defined to be the matroid $(X, \mathcal{I} \cap 2^X)$. The same matroid can also be referred to as the *deletion of* $E \setminus X$ *from* M denoted by $M \setminus (E \setminus X)$.

It is straightforward to check that the operation in Definition 1.1.2 produces a matroid.

Definition 1.1.3. A *circuit* of a matroid $M = (E, \mathcal{I})$ is an inclusion-wise minimal dependent set of M.

The family $\mathcal{C}$ of all circuits of any matroid M satisfies the following conditions:

(C1) $\emptyset \notin \mathcal{C}$,

(C2) if $C, C' \in \mathcal{C}$ and $C' \subseteq C$ then $C = C'$, and

(C3) if $C_1, C_2 \in \mathcal{C}$, $C_1 \neq C_2$ and $e \in C_1, C_2$, then there exists $C_3 \in \mathcal{C}$ such that $C_3 \subseteq (C_1 \cup C_2) \setminus \{e\}$.

Moreover, every family $\mathcal{C}$ satisfying (C1) - (C3) defines a matroid that has family of circuits $\mathcal{C}$.

The family of circuits of a matroid M is denoted by $\mathcal{C}(M)$. The circuit of size 1 is called a *loop*. If two elements e and f form a circuit, we say that e and f are *parallel*. The set containing e and all elements parallel to e is a *class of parallel elements*. If a matroid M contains no loop and no two elements are parallel, we say that M is *simple*. The *simplification of* M is the matroid obtained from M by deleting all loops and removing all parallel edges except one from each class of parallel elements. The simplification of M is denoted by $\widetilde{M}$.

Definition 1.1.4. An inclusion-wise maximal independent set of matroid M is called a *base* of M.

The family $\mathcal{B}$ of all bases of M satisfies the following conditions:

(B1) $\mathcal{B} \neq \emptyset$, and

(B2) If $B_1, B_2 \in \mathcal{B}$ and $e \in B_1 \setminus B_2$ then there exists element $f \in B_2 \setminus B_1$ such that $(B_1 \setminus \{e\}) \cup \{f\} \in \mathcal{B}$.

Again, every family $\mathcal{B}$ of finite sets satisfies (B1) and (B2) defines matroids having $\mathcal{B}$ as its family of bases.

We denote the family of all bases of a matroid M by $\mathcal{B}(M)$.

It is easy to see that all inclusion-wise maximal independent sets have the same cardinality.

Definition 1.1.5. The function from 2^E to nonnegative integers which assigns each set X the size of the maximal independent subset of X is called the *rank function* of M.

The rank function has the following properties:

(R1) If X is subset of E, then $0 \leq r(X) \leq |X|$,

(R2) If $X \subseteq Y \subseteq E$, then $r(X) \leq r(Y)$, and

(R3) For X and Y are subsets of E, then

$$r(X \cap Y) + r(X \cup Y) \leq r(X) + r(Y).$$

The condition (R3) is referred to as the *submodularity of the rank function.*

Definition 1.1.6. Let M be a matroid and let cl be function from 2^E to 2^E defined as follows:

$$\mathrm{cl}_M(X) = \{x : r(X) = r(X \cup \{x\})\}.$$

This function is called the *closure operator of* M. We call the set $\mathrm{cl}_M(X)$ the *closure of* X *in* M. We often write only $\mathrm{cl}(X)$ when the matroid is clear from the context.

Moreover, taking a closure of any set does not increase its rank, that is for any set X holds $r(X) = r(\mathrm{cl}(X))$.

Subsets of X satisfying $\mathrm{cl}(X) = X$ are called *closed sets* or *flats* of M. Flats of rank $r(M) - 1$ are called *hyperplanes* of M.

Proposition 1.1.7. *The closure operator of a matroid $M = (E, \mathcal{I})$ has the following properties:*

(CL1) If $X \subseteq E$, then $X \subseteq \mathrm{cl}(X)$.

(CL2) If $X \subseteq Y \subseteq E$, then $\mathrm{cl}(X) \subseteq \mathrm{cl}(Y)$.

(CL3) If $X \subseteq E$, then $\mathrm{cl}(\mathrm{cl}(X)) = \mathrm{cl}(X)$.

(CL4) If $X \subseteq E$, $x \in E$, and $y \in \mathrm{cl}(X \cup \{x\}) \setminus \mathrm{cl}(X)$, then $x \in \mathrm{cl}(X \cup \{y\})$

Moreover if F_1, F_2 are flats, then so is $F_1 \cap F_2$.

We also present two fundamental classes of matroids. One of them arises from graphs, the second one is derived from matrices. Both of them are frequently encountered throughout matroid theory.

Definition 1.1.8. Let $G = (V, E)$ be an undirected graph. The *cycle matroid of G* is the matroid M with the ground set E, where a set $I \subseteq E$ is independent if and only if it does not contain a cycle in G.

If a matroid M is isomorphic to the cycle matroid of a graph, we than M is said to be *graphic*.

Definition 1.1.9. Let A be a matrix over a field $\mathbb{F}$. The *vector matroid of A* is the matroid with ground set corresponding to columns of A, where set I is independent in M if and only if its corresponding set of columns is linearly independent.

If there exists a matrix A over a field $\mathbb{F}$ such that M is isomorphic to the vector matroid of A, we say that M is $\mathbb{F}$*-representable.*

1.2 Duality

We now introduce the notion of duality, one of the key concepts of matroid theory. Duality is important for the definition of matroid minors, which generalizes the concept of graph minors.

Definition 1.2.1. Let M be a matroid with and $\mathcal{B}$ be the family of bases of M. The *dual matroid of M*, denoted by M^* is the matroid on the ground set E with family of bases $\mathcal{B}^* = \{E \setminus B : B \in \mathcal{B}\}$.

Let us omit the proof that the set $\mathcal{B}^*$ satisfies axioms (B1) and (B2).

Definition 1.2.2. Let M be a matroid and T a subset of its ground set. The matroid obtained by *contracting* a set T is denoted by $M \,/\, T$ and is defined as $M \,/\, T = (M^* \setminus T)^*$

Proposition 1.2.3. *For all $X \subseteq E \setminus T$,*

$$\mathrm{cl}_{M/T}(X) = \mathrm{cl}_M(X \cup T) \setminus T.$$

Definition 1.2.4. Let M be a matroid. We say that M' is a *minor of M* if M can be obtained from M by a sequence of deletions and contractions.

In the next lemma we establish that the operations of deleting and contracting commute.

Lemma 1.2.5. *Let M be a matroid and T_1 and T_2 two disjoint subsets of its ground set. It holds:*

- $(M \,/\, T_1) \,/\, T_2 = M \,/\, (T_1 \cup T_2)$
- $(M \setminus T_1) \,/\, T_2 = (M \,/\, T_2) \setminus T_1$

By the last lemma the order in which the elements are deleted and contracted does not matter, which gives us somewhat simpler description of matroid minors.

Corollary 1.2.6. *A matroid M' is a minor of M if and only if there exist sets D and T such that $M' = M \setminus D \,/\, T$.*

2. Matroid amalgams

If two matroids M_1 and M_2 have a common restriction N, it is natural to ask if these two matroids can be joined together through N. In this chapter, we present some of the existing conditions guaranteeing that matroids can be joined in such a way. We will focus on the operation of a generalized parallel connection which is the core tool in our decomposition.

2.1 Basic definitions

Definition 2.1.1. Let M_1 and M_2 be two matroids on ground sets E_1 and E_2, respectively. Denote $E = E_1 \cup E_2$ and $T = E_1 \cap E_2$ and suppose that $M_1|T = M_2|T = N$. If M is a matroid with the ground set E such that $M|E_1 = M_1$ and $M|E_2 = M_2$, we say that M is an *amalgam* of M_1 and M_2.

Let us remark that an amalgam is not uniquely determined by M_1 and M_2 nor it is guaranteed to exist. The equality of matroids on intersections of their ground sets is not sufficient for the existence of their amalgam.

If the sets E_1 and E_2 are disjoint, an amalgam of M_1 and M_2 always exists and one such amalgam is the direct sum of M_1 and M_2.

In what follows the common restriction of M_1 and M_2 will be denoted by N and the rank function on this common restriction will be denoted r. Note that for every amalgam M of M_1 and M_2, the restriction of its rank function on N must be r. So we can use r for the rank function of the amalgam M without confusion.

There are two special kinds of amalgams: free amalgams and proper amalgams. Free amalgams are, as the name suggest, "freer" than any other amalgam. The definition of a proper amalgam is more complex; however it is easier to formulate sufficient conditions for its existence.

Definition 2.1.2. Let M_0 be an amalgam of M_1 and M_2. We say that M_0 is the *free amalgam* of M_1 and M_2 if for every amalgam M of M_1 and M_2 every set independent in M is also independent in M_0.

It is obvious that if the free amalgam exists, it is uniquely determined.

The definition of a proper amalgam requires two auxiliary functions which we now introduce. For a subset X of E, let

$$\eta(X) = r_1(X \cap E_1) + r_2(X \cap E_2) - r(X \cap T)$$
$$\zeta(X) = \min\{\eta(Y) : Y \supseteq X\}$$

Definition 2.1.3. If ζ is submodular on 2^E, we call the matroid on E with ζ as its rank function the *proper amalgam* of M_1 and M_2.

Proposition 2.1.4 ([Oxl06, Proposition 12.4.2, p. 413])**.** *Every proper amalgam is a free amalgam.*

Proof. Let M be an arbitrary amalgam of M_1 and M_2. The restriction of r_M to 2^{E_1} is r_1 by the definition of amalgam; similarly the restriction to 2^{E_2} is r_2. By submodularity of the rank function of M, we have for all subsets X of E and $Y \supseteq X$ the following inequality:

$$r_M(X) \leq r_M(Y) \leq r_1(Y \cap E_1) + r_2(Y \cap E_2) - r(Y \cap T) = \eta(Y).$$

Since the inequality holds for all $Y \supseteq X$ and $\zeta(X) = \min\{\eta(Y) : Y \supseteq X\}$, we have for all $X \subseteq E$:

$$r_M(X) \leq \zeta(X),$$

which implies that the proper amalgam is also a free amalgam. ☐

The next proposition provides a necessary and sufficient condition for an amalgam to be the proper amalgam of two given matroids.

Proposition 2.1.5 ([Oxl06, Proposition 12.4.3, p. 414])**.** *Let M_1 and M_2 be matroids and M their amalgam. Then M is the proper amalgam of M_1 and M_2 if and only if for every flat F of M it holds that*

$$r(F) = r(F \cap E_1) + r(F \cap E_2) - r(F \cap T).$$

However, note that this proposition is useful only when the matroid M is given, it says nothing about the existence of the proper amalgam of M_1 and M_2.

2.1.1 Modular flats

One of the easiest approaches for formulating sufficient conditions for the existence of proper amalgams is using modular flats. We are using several results about modular flats, however, most of them are stated without a proof. More results about modular flats as well as the omitted proofs are in [Oxl06].

Definition 2.1.6. Let X and Y be flats of a matroid M. The pair (X, Y) is a *modular pair of flats* if it satisfies

$$r(X) + r(Y) = r(X \cap Y) + r(X \cup Y).$$

If F is a flat of M such that (F, Y) is a modular pair of flats for every flat Y, then F is called a *modular flat* of M.

It is easy to see that E, $\mathrm{cl}(\emptyset)$ (the set of all loops), and all flats with rank one (classes of parallel elements) are modular flats.

Definition 2.1.7. Let M be a matroid and T subset of $E(M)$. We say that T is a *semiflat* if every element of $\mathrm{cl}(T) \setminus T$ is either a loop or parallel to some element of T. Furthermore, we say that T is a *modular semiflat* of M if T is a semiflat and $\mathrm{cl}(T)$ is modular flat of M.

Proposition 2.1.8 ([Oxl06, Proposition 6.9.5, p. 232])**.** *Let M be a matroid and F a modular flat of M. If D and F are disjoint, then F is a modular flat of $M \setminus D$.*

Proof. Let Y be a flat of $M \setminus D$ and denote $\mathrm{cl}_M(Y)$ by Y'. We aim to prove three equalities:

(i) $r(Y') = r(Y)$,

(ii) $r(F \cap Y') = r(F \cap Y)$,

(iii) $r(F \cup Y') = r(F \cup Y)$.

The equality (i) is trivial. Moreover,

$$Y' \setminus D = \mathrm{cl}_M(Y) \setminus D = \mathrm{cl}_{M \setminus D}(Y) = Y,$$

because Y is disjoint from D and Y is a flat of $M \setminus D$. From this equality and the fact that F is disjoint from D, we can conclude that $Y' \cap F = Y \cap F$ and therefore $r(F \cap Y') = r(F \cap Y)$. For the equality (iii), observe that $F \cup Y' \subseteq \mathrm{cl}_M(F \cup Y)$ so $r(F \cup Y') \leq r(F \cup Y)$. On the other hand, $F \cup Y \subseteq F \cup Y'$, so the equality $r(F \cup Y') = r(F \cup Y)$ holds.

Since F is a modular flat of M, (F, Y') is a modular pair of flats in M and we have the equality $r(F) + r(Y') = r(F \cap Y') + r(F \cup Y')$. By using (i), (ii) and (iii), we see that (F, Y) is a modular pair of flats in $M \setminus D$ and F is a modular flat of $M \setminus D$ as required. ☐

Proposition 2.1.9 ([Oxl06, Proposition 6.9.7, pp. 232–233])**.** *If X is a modular flat in M and Y is a modular flat in the matroid $M|X$, then Y is a modular flat in M.*

2.1.2 Generalized parallel connection

In this section, we focus on a particular kind of an amalgam — generalized parallel connection. We present conditions for its existence as well as some properties that are crucial for our algorithms.

Theorem 2.1.10 ([Oxl06, Corollary 12.4.12, p. 417])**.** *If T is a modular semiflat of M_1, then the proper amalgam of M_1 and any matroid M_2 such that $M_1|T = M_2|T$ exists.*

Definition 2.1.11. If the condition stated in Theorem 2.1.10 holds, we call the proper amalgam of M_1 and M_2 the *generalized parallel connection* of the matroids M_1 and M_2 and the amalgam is denoted by $M_1 \oplus_N M_2$.

To characterize closed sets of the generalized parallel connection, we need another definition.

Definition 2.1.12. Let M_1 and M_2 be two matroids on ground sets E_1 and E_2 and E the union of their ground sets. Let $\mathcal{L}(M_1, M_2)$ be the family of subsets X of E such that $X \cap E_1$ is a flat of M_1 and $X \cap E_2$ is a flat of M_2. The meet operation is defined as $X \wedge Y = X \cap Y$.

Note that since the set $\mathcal{L}(M_1, M_2)$ is finite and it has the greatest element, the definition of the meet operation uniquely determines the join operation such that $\mathcal{L}(M_1, M_2)$ with those two operations is a lattice. In the general case, the definition of join operation is more complex. The next proposition describes the structure of $\mathcal{L}(M_1, M_2)$ when generalized parallel connection of M_1 and M_2 exists.

Proposition 2.1.13 ([Oxl06, Proposition 12.4.13, pp. 418-419])**.** *The flats of $M_1 \oplus_N M_2$ are exactly the members of $\mathcal{L}(M_1, M_2)$.*

Lemma 2.1.14 ([Oxl06, Proposition 12.4.14 (iii), p. 419])**.** *If the generalized parallel connection of matroids M_1 and M_2 exists, E_2 is a modular semiflat of $M_1 \oplus_N M_2$.*

Since a generalized parallel connection of M_1 and M_2 is always the proper amalgam of M_1 and M_2, Proposition 2.1.5 gives us a simple formula for the calculating rank of closed sets using the rank functions of M_1 and M_2. The next proposition says how to determine both closure and rank of a set in generalized parallel connection.

Proposition 2.1.15. *Let M_1 and M_2 be two matroids and E the union of their ground sets. Further, let $M = M_1 \oplus_N M_2$ and $X \subseteq E$ and denote $X_i = \mathrm{cl}_i(X \cap E_i) \cup X$. It holds that*

$$\mathrm{cl}_M(X) = \mathrm{cl}_1(X_2 \cap E_1) \cup \mathrm{cl}_2(X_1 \cap E_2), \text{ and}$$

$$r_M(X) = r_{M_1}(X_2 \cap E_1) + r_{M_2}(X_1 \cap E_2) - r(T \cap [X_1 \cup X_2]).$$

Generalized parallel connections of matroids commute in the following sense: if we have a matroid and we paste another matroids to it as a generalized parallel connection, then the order of pasting does not matter.

Theorem 2.1.16. *Let K, M_1 and M_2 be matroids such that $M_1|T_1 = K|T_1$ and $M_2|T_2 = K|T_2$. If T_1 is modular in M_1 and T_2 is modular in M_2, then*

$$M_2 \oplus_{N_2} (M_1 \oplus_{N_1} K) = M_1 \oplus_{N_1} (M_2 \oplus_{N_2} K).$$

Proof. Since T_1 is modular in M_1 and T_2 is modular in M_2, all generalized parallel connections in the equality are well-defined.

By Proposition 2.1.13, flats of $M_1 \oplus_{N_1} K$ are exactly the sets of $\mathcal{L}(M_1, K)$. By using the same proposition again, we obtain that the flats of $M_2 \oplus_{N_2} (M_1 \oplus_{N_1} K)$ are sets such that their intersection with $E(M_2)$ is closed in M_2 and their intersection with $E(M_1 \oplus_{N_1} K)$ is closed in $M_1 \oplus_{N_1} K$. In other words, their intersection with $E(M_2)$ must be closed in M_2, intersection with $E(M_1)$ must be closed in M_1, and intersection with $E(K)$ must be closed in K.

Along the same lines we show that flats of $M_1 \oplus_{N_1} (M_2 \oplus_{N_2} K)$ are the same. We eventually conclude that matroids $M_2 \oplus_{N_2} (M_1 \oplus_{N_1} K)$ and $M_1 \oplus_{N_1} (M_2 \oplus_{N_2} K)$ are equal. ◻

Lemma 2.1.17 ([Oxl06, Proposition 12.4.14, p. 419])**.** *Let M_1 and M_2 be two matroids such that their generalized parallel connection exists.*

- *If $e \in E_1 \setminus \mathrm{cl}_1(T)$, then $(M_1 \oplus_N M_2) / e = (M_1 / e) \oplus_N M_2$.*
- *If $e \in E_2 \setminus \mathrm{cl}_2(T)$, then $(M_1 \oplus_N M_2) / e = M_1 \oplus_N (M_2 / e)$.*
- *If $e \in E_1 \setminus T$, then $(M_1 \oplus_N M_2) \setminus e = (M_1 \setminus e) \oplus_N M_2$.*
- *If $e \in E_2 \setminus T$, then $(M_1 \oplus_N M_2) \setminus e = M_1 \oplus_N (M_2 \setminus e)$.*

- *If $e \in T$, then $(M_1 \oplus_N M_2) \,/\, e = (M_1 \,/\, e) \oplus_{N/e} (M_2 \,/\, e)$.*

Some well-known simple matroid operations can be viewed as a special kind of the generalized parallel connection as stated in the next propositions.

Definition 2.1.18. Let M_1 and M_2 be two matroids, E the union of their ground sets and $E_1 \cap E_2 = \{p\}$. If p is neither a loop nor a coloop in M_1 or M_2, the matroid on a ground set E with the set of circuits $\mathcal{C}_S$ is called the *series connection* of M_1 and M_2, where $\mathcal{C}_S$ is defined as follows:

$$\mathcal{C}_S = \mathcal{C}(M_1 \setminus \{p\}) \cap \mathcal{C}(M_2 \setminus \{p\}) \cap \{C_1 \cap C_2 : p_i \in C_i \in \mathcal{C}(M_i) \text{ for } i = 1, 2\}$$

We denote the series connection of M_1 and M_2 by $S(M_1, M_2)$.

Proposition 2.1.19. *A series connection can be replaced by a sequence of generalized parallel connections and edge deletions. In particular, it holds that*

$$S(M_1, M_2) = M_1 \oplus_{N_1} (M' \oplus_{N_2} M_2) \setminus \{e_1, e_2\},$$

where $N_1 = M_1|\{e_1\}$, $N_2 = M_2|\{e_2\}$ and M' is matroid on ground set $E(M') = \{e_1, e_2, e\}$ such that $M' \cong U_{2,3}$.

Proposition 2.1.20. *The 2-sum and 3-sum of matroids can be replaced by generalized parallel connection and deletion of edges.*

3. Amalgam Decompositions

In this chapter, we present a decomposition of matroids based on matroid amalgams. We call the corresponding width parameter the amalgam width. This notion will allow us to compute some intractable problems for matroids with bounded amalgam width. More precisely, we will be able to decide MSO formulas in polynomial time on matroids with bounded width parameter.

3.1 Definition

Definition 3.1.1. An *amalgam decomposition of width k* of a matroid M is a rooted binary tree $\mathcal{T}$ with root r such that

- every internal node v of $\mathcal{T}$ is assigned a matroid K_v with ground set E_v, $|E_v| \leq k$, and four distinguished subsets J_v^1, J_v^2, J_v and D_v of E_v such that J_v^1 is disjoint from J_v^2, J_v is a modular semiflat of K_v, and $J_v \cap D_v = \emptyset$,
- the set J_r is empty
- the leaves of $\mathcal{T}$ one-to-one correspond to the elements of M, and
- M is the matroid M_r obtained by assigning each node v of $\mathcal{T}$ a matroid M_v by the following rules
  - M_v is $M|\{e\}$ with $J_v := \{e\}$ if v is a leaf corresponding to an element e, and
  - M_v is the matroid $M_{v_2} \oplus_{J_v^2} (M_{v_1} \oplus_{J_v^1} K_v) \setminus D_v$ where v_1 and v_2 are the two children of v, otherwise.

This definition deserves some explanation. Every internal node of $\mathcal{T}$ is assigned an auxiliary matroid K_v that is used only for gluing matroids together. The matroid M_v is an intermediate matroid used to eventually create the given matroid M at the root by an operation, which attach the intermediate matroids assigned to its children through the marked sets of K_v (by Theorem 2.1.16, the order in that we join the children to K_v does not matter). The parameter k bounds the size of auxiliary matroids.

It is not obvious that all generalized parallel connections in our definition are well-defined. The soundness of our definition is established in the next theorem.

Theorem 3.1.2. *If J_v is a modular semiflat of K_v, then J_v is a modular semiflat of M_v, where $M_v = M_{v_2} \oplus_{J_v^2} (M_{v_1} \oplus_{J_v^1} K_v) \setminus D_v$.*

Proof. Denote the matroid $M_{v_1} \oplus_{J_v^1} K_v$ by M_1 and the matroid $M_{v_2} \oplus_{J_v^2} (M_{v_1} \oplus_{J_v^1} K_v)$ by M_2.

By Lemma 2.1.14, $\mathrm{cl}_{M_1}(E(K_v))$ is a modular flat of M_1. Since every element of $\mathrm{cl}_{M_1}(E(K_v)) \setminus K_v$ must be a loop or parallel to some element of K_v, the set $\mathrm{cl}_{M_1}(J_v)$ is a modular flat of $\mathrm{cl}_{M_1}(K_v)$. Now by using Proposition 2.1.9, we obtain that $\mathrm{cl}_{M_1}(J_v)$ is a modular flat of M_1. Because every element of $\mathrm{cl}_{M_1}(J_v) \setminus J_v$ is either a loop or parallel to some element of J_v, the set J_v is a modular semiflat of M_1.

By an analogous argument, we get that J_v is a modular semiflat of M_2. Since $M_v = M_2 \setminus D_v$ and D_v is disjoint from J_v, by Proposition 2.1.8 we have that J_v is a modular semiflat of M_v as required. □

We show that every matroid admits an amalgam decomposition.

Proposition 3.1.3. *Let M be a matroid. There exists an amalgam decomposition $\mathcal{T}$ of M of width at most $|E(M)|$.*

Proof. Let $\mathcal{T}$ be an arbitrary rooted binary tree such that its leaves are in one-to-one correspondence to the elements of M. Denote by E_v the set of elements corresponding to the leaves below v.

For an internal node v with children v_1 and v_2, we set

- $K_v := M|E_v$,
- $J_v := E_v$,
- $J_v^1 := E_{v_1}$,
- $J_v^2 := E_{v_2}$, and
- $D_v := \emptyset$.

Since J_v is equal to $E(K_v)$, it must be a modular flat of K_v. It's easy to see that the described decomposition is indeed an amalgam decomposition of M of width $|E(M)|$. □

The last proposition leads to following definition.

Definition 3.1.4. Let M be a matroid. The *amalgam width* of M is the smallest number k such that M has an amalgam decomposition of width k.

We are now ready to explore algorithmic properties of amalgam decompositions. First, the decompositions capture the structure of a matroid, so it can be used as an input structure for matroid algorithms. Since the decomposition size is linear, it is a compact structure that can be used to encode a matroid M as input.

Our main result is that for bounded k, we can efficiently compute the Tutte polynomial of M and decide MSO formulas for matroids M given by an amalgam decomposition of width at most k.

3.2 Basic properties

We now prove two crucial properties of the amalgam decompositions – existence of a linear time algorithm for computing rank of a given set and linear size of the decomposition itself.

Before showing the rank computation, we need following lemma.

Lemma 3.2.1. *Let v be a node of an amalgam decomposition $\mathcal{T}$. Denote by M'_v the matroid obtained in the node v without deleting elements in D_v. Then a set F is a flat of M'_v if and only if $F \cap E(M_{v_1})$ is a flat of M_{v_1}, $F \cap E(M_{v_2})$ is a flat of M_{v_2}, and $F \cap E(K_v)$ is a flat of K_v. Moreover, the rank of a flat F can be computed as*

$$r(F) = r(F \cap E(M_{v_1})) + r(F \cap E(M_{v_2})) + r(F \cap E(K_v)) - r(F \cap J_v^1) - r(F \cap J_v^2).$$

Proof. Denote by S the set $E(M_{v_1}) \cup E(K_v)$. By Proposition 2.1.13, the set F is a flat of M'_v if and only if $F \cap E(M_{v_2})$ is a flat of M_{v_2} and $F \cap S$ is a flat of $M_{v_1} \oplus_{J_v^1} K_v$. Using Proposition 2.1.13 again, we obtain that $F \cap S$ is a flat of $M_{v_1} \oplus_{J_v^1} K_v$ if and only if $F \cap E(M_{v_1})$ is a flat of M_{v_1} and $F \cap E(K_v)$ is a flat of K_v. Together, we get the first part of the lemma.

The proof of rank formula is similar. Since F is a flat, by Proposition 2.1.5 it holds

$$r(F) = r(F \cap E(M_{v_2})) + r(F \cap S) - r(F \cap [E(M_{v_2}) \cap S]),$$

which can be simplified to

$$r(F) = r(F \cap E(M_{v_2})) + r(F \cap S) - r(F \cap J_v^2).$$

Using Proposition 2.1.5 on $r(F \cap S)$ in $M_{v_1} \oplus K_v$, we obtain

$$r(F) = r(F \cap E(M_{v_1})) + r(F \cap E(M_{v_2})) + r(F \cap E(K_v)) - r(F \cap J_v^1) - r(F \cap J_v^2).$$

□

Theorem 3.2.2. *For every fixed k, there exists a linear-time algorithm that given an amalgam decomposition with width at most k of a matroid M and set X of elements of M computes the rank of X in M.*

Proof. Let $\mathcal{T}$ be the given amalgam decomposition of M and let M' be a matroid obtained from the decomposition $\mathcal{T}$ if we consider that every set D_v is empty (i.e. we delete no intermediate elements). Clearly, $r_{M'}(X) = r_M(X)$, so it suffices to compute rank of X in M'.

We begin by computing $\mathrm{cl}_{M'}(X)$ and then compute its rank using Lemma 3.2.1.

The closure is computed recursively. For leafs, the computation is trivial. So let v be an internal node of $\mathcal{T}$. Denote by M'_v the matroid obtained from the decomposition $\mathcal{T}$ at the node v considering that every set D_v is empty. Furthermore, denote by X_v the set $X \cap E(M_v)$, by cl_1 the closure operator of M'_{v_1}, and by cl_2 the closure operator of M'_{v_2}.

From Lemma 3.2.1, we see that a set Y is closed in M'_v if and only if $Y \cap E(K_v)$ is closed in K_v, $Y \cap E(M'_{v_1})$ is closed in M'_{v_1}, and $Y \cap E(M'_{v_2})$ is closed in M'_{v_2}. We can obtain $\mathrm{cl}_{M'_v}(X_v)$ by repeatedly calling cl_1, cl_2, and cl_{K_v} until the intersections of the resulting set with $E(M'_{v_1})$, $E(M'_{v_2})$, and $E(K_v)$ are all closed. We need to call the closure only in the case when the sequence of last three calls adds an element into K_v. Since there are at most k elements in K_v, the number of recursive calls for the node v is bounded by $3k$. Moreover, as there are only $|E| - 1$ internal nodes in $\mathcal{T}$ and we are performing only constant amount of work in each node, the time complexity of computing closure is linear with respect to $|E|$.

The final computation of rank of a closed set is clearly linear, so the whole computation runs in linear time. □

Algorithm 1 Closure of X in M'_v for internal node v

repeat
  $X' \leftarrow X$
  $X \leftarrow \mathrm{cl}_1(X \cap E(M'_{v_1}))$
  $X \leftarrow \mathrm{cl}_2(X \cap E(M'_{v_2}))$
  $X \leftarrow \mathrm{cl}_{K_v}(X \cap E(K_v))$
until $X' \cap K_v = X \cap K_v$
return X

Algorithm 2 Computing rank of a flat X in M'_v

return $r(X\cap E(M'_{v_1}))+r(X\cap E(M'_{v_2}))+r(X\cap E(K_v))-r(X\cap J^1_v)-r(X\cap J^2_v)$

Proposition 3.2.3. *For every fixed k, the size of amalgam decomposition with width at most k of a matroid M is linear in $E(M)$.*

Proof. Let us denote the number of elements of M by n. The decomposition $\mathcal{T}$ has exactly n leaves and the tree is binary, so it has $n-1$ internal nodes. The matroid K_v in every internal node v has at most k elements, so its size does not depend on n. The sizes of sets J^1_v, J^2_v, J_v, and D_v are also bounded by k, so the size of every internal node depends on k only. The size of each leaf of $\mathcal{T}$ is clearly constant. We eventually conclude that the size of the decomposition is linear in n. ☐

4. Computing the Tutte Polynomial

The Tutte polynomial is a polynomial in two variables that can be used to express many interesting quantities for either graphs or matroids. The definition of Tutte polynomial for matroids naturally extends the definition for graphs.

By evaluating the Tutte polynomial at specific points, we can express for example number of all bases, all spanning sets, or all independent sets. In the graph case, we can obtain number of acyclic orientations, chromatic polynomial, flow polynomial, or all-terminal reliability of network [Wel90]. Not surprisingly, evaluating the Tutte polynomial is #P-hard at almost every point. This was proved by Jaeger, Vertigan, and Welsh [JVW90]. In this chapter, we show that for matroids given by an amalgam decomposition of bounded width we can compute the Tutte polynomial in polynomial time.

Definition 4.0.4. Let M be a matroid with a ground set E. The *Tutte polynomial of M* is a bivariate polynomial T_M defined as

$$T_M(x, y) = \sum_{F \subseteq E} (x-1)^{r(E)-r(F)} (y-1)^{|F|-r(F)}.$$

The main result of this chapter is that the Tutte polynomial can be efficiently computed using an amalgam decomposition.

Theorem 4.0.5. *For every fixed k, there exists a polynomial-time algorithm that given an amalgam decomposition with width at most k of a matroid M computes coefficients of the Tutte polynomial of M.*

The key idea is to count the number of sets with given size and rank. This way, we get the coefficients in 4.0.4, which can be easily transformed to the coefficients of every monomial in Tutte polynomial. However, in order to be able to compute the number of such sets in polynomial time, we need to introduce a few auxiliary notions.

4.1 The type of a set

Definition 4.1.1. Let v be a node of amalgam decomposition $\mathcal{T}$ of M and X be a subset of $E(M)$. The map f_v^X from $2^{J_v} \to 2^{J_v}$ satisfying

$$f_v^X(Y) = \mathrm{cl}_{M_v}\big((X \cap E(M_v)) \cup Y\big) \cap J_v$$

is called *the type of X in v.*

The purpose of type is to describe the local behavior of closure in matroid M_v restricted to the set J_v along which we are pasting matroids together. Informally speaking, if two subsets of $E(M_v)$ have the same type, they exhibit the same behavior outside of M_v. We will later use that fact in the computation of the Tutte polynomial as well as deciding MSO properties.

It is easy to see that the type of a set X in a node v depends only on the intersection of X with $E(M_v)$. If two sets have the same intersection with $E(M_v)$, they must necessarily have the same type.

Note that number of distinct types does not depend on n – more precisely, it is bounded by $(2^k)^{(2^k)}$.

Before proceeding, let us state few basic properties of types.

Proposition 4.1.2. *Let v be a node of an amalgam decomposition, X a subset of $E(M_v)$, and f be the type of a set X in v. If Y is a subset of J_v disjoint from X, then*

$$f(Y) = (\mathrm{cl}_{M_v/X}(Y) \cup X) \cap J_v.$$

Proof. Using Proposition 1.2.3, we have

$$(\mathrm{cl}_{M_v/X}(Y) \cup X) \cap J_v = ((\mathrm{cl}_{M_v}(X \cup Y) \setminus X) \cup X) \cap J_v = \mathrm{cl}_{M_v}(X \cup Y) \cap J_v = f(Y).$$

□

Corollary 4.1.3. *Let v be a node of an amalgam decomposition and f be the type of a set X in v. Then f satisfies the following properties:*

(i) *If $Y \subseteq J_v$, then $Y \subseteq f(Y)$.*

(ii) *If $Y \subseteq Y' \subseteq J_v$, then $f(Y) \subseteq f(Y')$.*

(iii) *If $Y \subseteq J_v$, then $f(f(Y)) = f(Y)$.*

(iv) *If $Y, Y' \subseteq J_v$, $f(Y) = Y$, and $f(Y') = Y'$, then $f(Y \cap Y') = Y \cap Y'$.*

Proof. From the definition of the type, it is clear that $f(Y) = f(Y \setminus X)$. Moreover, the type of X is the same as type of $X \cap E(M_v)$, therefore we can use Proposition 4.1.2 in the general case. The rest is a straightforward application of Proposition 1.1.7. □

An important property of types is that the type of a set X in a node v is fully determined by the types of X in children of v. To prove this fact we use following lemma about structure of closed sets.

Lemma 4.1.4. *Let v be a node of an amalgam decomposition of a matroid M, v_1, v_2 the children of v, and X a subset of $E(M)$. Denote the type of X in v_1 and v_2 by f_1 and f_2, respectively. Then the type of X in v is fully determined by the types f_1 and f_2. Moreover, for a subset Y of J_v, denote by F_Y the set $\mathrm{cl}_{M'_v}\big((X \cap E(M_v)) \cup Y\big)$ and by Z_Y the smallest subset of $E(K_v)$ satisfying*

(i) $f_1(Z_Y \cap J_v^1) = Z_Y \cap J_v^1$,

(ii) $f_2(Z_Y \cap J_v^2) = Z_Y \cap J_v^2$,

(iii) $Z_Y = \mathrm{cl}_{K_v}(Z_Y)$,

(iv) $Z_Y \supseteq Y$.

Then it holds that:

(a) $F_Y \cap E(K_v) = Z_Y$,

(b) $F_Y \cap E(M_{v_1}) = \mathrm{cl}_{M_{v_1}}\big((X \cap E(M_{v_1})) \cup (Z_Y \cap J_v^1)\big)$,

(c) $F_Y \cap E(M_{v_2}) = \mathrm{cl}_{M_{v_2}}\big((X \cap E(M_{v_2})) \cup (Z_Y \cap J_v^2)\big)$.

Proof. First, we need to establish that the set Z_Y is well-defined for every $Y \subseteq J_v$. To prove that, we show that the family of subsets satisfying (i)-(iv) is nonempty and closed under intersections.

Clearly, $E(K_v)$ satisfies (iii) and (iv). By Corollary 4.1.3 (i), the conditions (i) and (ii) also holds for $E(K_v)$. Therefore, the family is nonempty.

If Z_1 and Z_2 both satisfy (i)-(iv), then $Z_1 \cap Z_2$ satisfies (iii) and (iv). By Corollary 4.1.3 (iv), $Z_1 \cap Z_2$ also satisfies (i) and (ii), so the family is closed under intersection.

From Lemma 3.2.1, we see that $F_Y \cap E(K_v)$ must be a flat of K_v, therefore $F_Y \cap E(K_v)$ must satisfy condition (iii). From the same lemma, we have that $F_Y \cap E(M_{v_1})$ is a flat of M_{v_1}. Together with the definition of type, we obtain that $F_Y \cap E(K_v)$ satisfies condition (i). Same argument shows that $F_Y \cap E(K_v)$ satisfies (ii). Because F_Y is a superset of Y and Y is a subset of $E(K_v)$, we see that $F_Y \cap E(K_v)$ must be a superset of Y, thus condition (iv) holds for $F_Y \cap E(K_v)$. Since Z_Y is the smallest set satisfying (i)-(iv), we conclude that $F_Y \cap E(K_v) \supseteq Z_Y$.

Denote by F_Y^1 and F_Y^2 the sets $\mathrm{cl}_{M_{v_1}}\big((X \cap E(M_{v_1}) \cup (Z_Y \cap J_v^1))\big)$ and $\mathrm{cl}_{M_{v_2}}\big((X \cap E(M_{v_2}) \cup (Z_Y \cap J_v^2))\big)$, respectively. Because $F_Y \cap E(M_{v_1})$ is a flat and it must contain both $X \cap E(M_{v_1})$ and $Z_Y \cap J_v^1$, we see that $F_Y \cap E(M_{v_1}) \supseteq F_Y^1$. The same argument shows that $F_Y \cap E(M_{v_2})$ is a superset of F_Y^2. Together, we have shown that $F_Y \supseteq F_Y^1 \cup F_Y^2 \cup Z_Y$.

Denote by F'_Y the set $F_Y^1 \cup F_Y^2 \cup Z_Y$. To show that $F_Y = F'_Y$, it suffices to prove that F'_Y is a flat of M'_v. Note that because Z_Y satisfies conditions (i) and (ii), it must hold that $Z_Y \cap J_v^1 = F_Y^1 \cap J_v^1$ and $Z_Y \cap J_v^2 = F_Y^2 \cap J_v^2$. From this, we conclude that $F'_Y \cap E(K_v) = Z_Y$, $F'_Y \cap E(M_{v_1}) = F_Y^1$, and $F'_Y \cap E(M_{v_2}) = F_Y^2$. Because Z_Y, F_Y^1, and F_Y^2 are flats of K_v, M_{v_1}, and M_{v_2}, respectively, by Lemma 3.2.1 F'_Y must be a flat of M'_v.

Therefore, $F_Y = F'_Y$. From previous facts about the set F'_Y, (a)-(c) follows immediately.

To see that the type of X in v is determined by the types f_1 and f_2, it is enough to see that $f_v^X(Y)$ is by definition $F_Y \cap J_v$, which is by (a) equal to $Z_Y \cap J_v$. □

The last fact leads to the following definition.

Definition 4.1.5. Let v be a node of an amalgam decomposition of a matroid M, let v_1 and v_2 be the children of v, and X a subset of $E(M)$. Denote by f_1 the type of X in v_1 and by f_2 the type of X in v_2. Then the mapping $f\colon J_v \to J_v$ satisfying $f(Y) = Z_Y \cap J_v$ for every $Y \subseteq J_v$, where Z_Y is the smallest subset satisfying

(i) $f_1(Z_Y \cap J_v^1) = Z_Y \cap J_v^1$,

(ii) $f_2(Z_Y \cap J_v^2) = Z_Y \cap J_v^2$,

(iii) $Z_Y = \mathrm{cl}_{K_v}(Z_Y)$,

(iv) $Z_Y \supseteq Y$.

is called the *join type of* f_1 *and* f_2 and is denoted by $f_1 +_v f_2$.

As mentioned before, the type of set in a node v depends only on intersection of this set with $E(M_v)$. Therefore, if v is a node with children v_1 an v_2, X_1 is a subset of $E(M_{v_1})$, and X_2 is a subset of $E(M_{v_2})$, we can also write $f_{v_1}^{X_1} +_v f_{v_2}^{X_2}$ for $f_v^{X_1 \cup X_2}$.

Note that Lemma 4.1.4 also suggests a way how to compute the join type of f_1 and f_2. Given $Y \subseteq J_v$, we can find Z_Y by simply trying all subsets of $E(K_v)$ and picking the smallest one satisfying conditions (i)-(iv). The running time of such algorithm depends on k only.

We have seen that types of subsets of $E(M_1)$ and $E(M_2)$ determine the type of their union. As a next step, we show that types and ranks of such subsets determine the rank of their union.

Suppose that X_1 is a subset of $E(M_{v_1})$ with rank r_1 and type f_1 and X_2 is subset of $E(M_{v_2})$ with rank r_2 and type f_2. We aim to compute the rank of $X_1 \cup X_2$ using only the types r_1, r_2, types f_1, f_2, and the matroid K_v.

We will try to somehow mimic the computation of rank we have used in decompositions – compute the closure of a given set and then compute its rank using Proposition 2.1.5. However, this time we cannot compute the closure of $X_1 \cup X_2$, because we don't know which elements are in X_1 or X_2. Fortunately, we can still compute the intersection of closure of $X_1 \cup X_2$ with the set $E(K_v)$.

If we denote by F the set $\mathrm{cl}_{M'_v}(X_1 \cup X_2)$, by Lemma 4.1.4 we have that $F \cap E(K_v)$ is the set $Z_\emptyset$, that is smallest subset of $E(K_v)$ such that

(i) $f_1(Z_\emptyset \cap J_v^1) = Z_\emptyset \cap J_v^1$,

(ii) $f_2(Z_\emptyset \cap J_v^2) = Z_\emptyset \cap J_v^2$,

(iii) $Z_\emptyset = \mathrm{cl}_{K_v}(Z_\emptyset)$.

By using Lemma 3.2.1, we obtain

$$r(F) = r(F \cap E(M_{v_1})) + r(F \cap E(M_{v_2})) + r(F \cap E(K_v)) - r(F \cap J_v^1) - r(F \cap J_v^2),$$

which can be rewritten as

$$r(F) = r(F \cap E(M_{v_1})) + r(F \cap E(M_{v_2})) + r(Z_\emptyset) - r(Z_\emptyset \cap J_v^1) - r(Z_\emptyset \cap J_v^2).$$

Since we can find the set $Z_\emptyset$, we can compute the last three terms. It remains to show how to compute $r(F \cap E(M_{v_1}))$ and $r(F \cap E(M_{v_2}))$. By Lemma 4.1.4, we have that $F \cap E(M_{v_1}) = \mathrm{cl}_1(X \cup (Z_\emptyset \cap J_v^1))$. We already know the rank of X, so we need to find how much the addition of a certain subset of J_v^1 (in our case the set $Z_\emptyset \cap J_v^1$) to X increases the rank. This procedure is described in Algorithm 3.

Lemma 4.1.6. *Algorithm 3 eventually terminates, returning the* $r(X \cup Y) - r(X)$ *(i.e. the increase in the rank of* X *after adding the set* Y*). Moreover, the running time of the algorithm depends on* k *only.*

Algorithm 3 Compute the rank increase

INPUT: type f_v^X, a subset Y of J_v
OUTPUT: $r(X \cup Y) - r(X)$

$Z \leftarrow f_v^X(\emptyset)$
$l \leftarrow 0$
while $Y \setminus Z \neq \emptyset$ **do**
  pick arbitrary element e of $Y \setminus Z$
  $Z \leftarrow f_v^X(Z \cup \{e\})$
  $l \leftarrow l + 1$
end while
return l

Proof. Denote by Z_0 the set $f_v^X(\emptyset)$, by Z_i the set Z after i-th pass of the loop, From Corollary 4.1.3 (i), we see that the sequence Z_i is strictly increasing, so the loop terminates after at most $|J_v|$ steps.

We claim that if B is a base of X, then $B \cup \{e_1, e_2, \ldots, e_i\}$ is a base of $X \cup Z_i$. We prove this statement by induction on i. First of all, B is a base of $X \cup Z_0$, because every element of Z_0 lies in the closure of X. Now suppose that $i \geq 1$ and $B \cup \{e_1, \ldots, e_{i-1}\}$ is a base of $X \cup Z_{i-1}$. Because e_i is not an element of Z_{i-1}, it does not lie in the closure of $X \cup Z_{i-1}$ and therefore $B \cup \{e_1, \ldots, e_i\}$ is independent. Since Z_i is by definition $\mathrm{cl}_{M_v}(X \cup Z_{i-1} \cup \{e_i\}) \cap J_v$ and $B \cup \{e_1, \ldots, e_{i-1}\}$ spans $X \cup Z_{i-1}$, the set $B \cup \{e_1, \ldots, e_i\}$ must span Z_i. Therefore, $B \cup \{e_1, \ldots, e_i\}$ is a base of $X \cup Z_i$.

The algorithm terminates when $Z_l \supseteq Y$. At this point, the set $B \cup \{e_1, \ldots, e_l\}$ spans $X \cup Z_l$, so it must also span $X \cup Y$. Moreover, it is an independent subset of $X \cup Y$, so it must be a base of $X \cup Y$. From this fact, it can be easily seen that l is equal to $r(X \cup Y) - r(X)$.

As mentioned before, the loop terminates after at most $|J_v|$ iterations, so the number of iterations is bounded by k. Because running time of every statement depends on k only, the running time of the whole algorithm must depend on k only.

□

Given Algorithm 3 we are now able to compute the rank of union of sets X_1 and X_2 provided we know only their ranks and types.

The soundness of algorithm follows immediately from discussion preceding Lemma 4.1.6. Note that apart from arithmetic operations with ranks, the running time of Algorithm 4 can be bounded by some function of k.

We now have all tools required to prove the main theorem of this section.

Proof of Theorem 4.0.5. We now show how to compute the number of subsets of $E(M_v)$ with given rank, size, and type. We denote the number by $\mathrm{count}_v(r, s, f)$.

The algorithms runs from the bottom of the tree – first it computes the count for the leaves and then it picks arbitrary node such that both its children has its count already computed. The computation for the leaf is trivial, as there are only two possible cases, loop and non-loop element. When we get to the root r, there is only one type (since J_r is the empty set) and the number $\mathrm{count}_v(r, s, f_0)$ is the desired number of sets of given rank and size.

Algorithm 4 Rank of union of X_1 and X_2

INPUT: rank r_1 of set $X_1 \subseteq E(M_{v_1})$, type f_1 of X_1 in v_1,
rank r_2 of set $X_2 \subseteq E(M_{v_2})$, type f_2 of X_2 in v_2,
matroid K_v

OUTPUT: rank of $X_1 \cup X_2$

$\text{rankOfUnion}_v(r_1, f_1, r_2, f_2)$:

find smallest set $Z_\emptyset \subseteq E(K_v)$ such that
- $f_1(Z_\emptyset \cap J_v^1) = Z_\emptyset \cap J_v^1$
- $f_2(Z_\emptyset \cap J_v^2) = Z_\emptyset \cap J_v^2$
- $Z_\emptyset$ is a flat of K_v

$Y_1 \leftarrow Z_\emptyset \cap J_v^1$
$Y_2 \leftarrow Z_\emptyset \cap J_v^2$
$r_1' \leftarrow r_1 + \text{rankIncrease}(f_1, Y_1)$
$r_2' \leftarrow r_2 + \text{rankIncrease}(f_2, Y_2)$
return $r_1' + r_2' + r(Z_\emptyset) - r(Z_\emptyset \cap J_v^1) - r(Z_\emptyset \cap J_v^2)$

Algorithm 5 Computing all counts for an inner node v

INPUT: node v of $\mathcal{T}$ with children v_1, v_2
OUTPUT: $\text{count}_v(r, s, f)$ for all ranks r, sizes s, and types f
Initialize $\text{count}_v(r, s, f)$ to 0 for all r, s, f
for s_1 in $\{1, 2, \ldots, |E(M_{v_1})|\}$ **do**
  for s_1 in $\{1, 2, \ldots, |E(M_{v_2})|\}$ **do**
    for r_1 in $\{1, 2, \ldots, r(M_{v_1})\}$ **do**
      for r_2 in $\{1, 2, \ldots, r(M_{v_2})\}$ **do**
        for f_1 type in v_1 **do**
          for f_2 type in v_2 **do**
            $f \leftarrow f_1 +_{K_v} f_2$
            $s \leftarrow s_1 + s_2$
            $r \leftarrow \text{rankOfUnion}_v(r_1, f_1, r_2, f_2)$
            $\text{count}_v(r, s, f) \leftarrow$
              $\text{count}_v(r, s, f) + \text{count}_{v_1}(r_1, s_1, f_1) \cdot \text{count}_{v_2}(r_2, s_2, f_2)$
          end for
        end for
      end for
    end for
  end for
end for

Let us turn our attention to the analysis of the time complexity of Algorithm 5. Let r denote the rank of the matroid M. Each of the two outer loops make n iterations. The loops iterating over the rank make at most r iteration each. The number of iterations of loops iterating over the types can be bounded by a function of k, so as long as k is fixed and we are interested only in asymptotic behavior depending on size of input matroid n, we do not need to determine the number of iterations. The time complexity of the computation of the join type also depends on k only. The total time complexity of Algorithm 5 is $\mathcal{O}(n^2r^2)$.

For the computation of the Tutte polynomial itself, we need to call Algorithm 5 for every node. This is $n-1$ calls, so the total time complexity of computing the Tutte polynomial is $\mathcal{O}(n^3r^2)$. $\square$

5. Deciding MSO properties

Logic formulas are probably the most prevalent way of encoding decision problems. Our particular interest is the *monadic second-order logic*. It is a direct analogue of monadic second-order logic for graphs that was used in formulation of Courcelle's theorem.

Roughly speaking, monadic second-order (MSO) logic allows us to quantify over elements of matroid or over subsets of matroid. The choice of MSO logic over first-order logic is natural, because the structure of matroid is described by predicate taking subsets of arbitrary size as parameter. Although there is work dealing with deciding first-order properties [GKO11], we do not wish to restrict ourselves as the expressive power of first order logic is somewhat limited.

Even in the case of graphs, one of properties expressible in MSO logic is Hamiltonicity, which implies that in general, deciding MSO properties is hard.

Note that for every fixed matroid N, there exists MSO formula that is satisfied by M if and only if M contains N as a minor. If Rota's Conjecture [Rot71] is true[1], then it can be decided by MSO formula whether a matroid is representable over GF(q) for every prime power q. For more information about properties expressible in MSO logic, see paper by Hliněný [Hli03].

5.1 Definition of matroid MSO logic

First, we present basic definition concerning monadic second-order logic. For more detailed treatment of monadic second-order logic, see for example Chapter 7 in [Lib04].

Definition 5.1.1. We assume two countably infinite set of variables: *element variables* and *set variables*. Element variables are denoted by lower-case letters, set variables are denoted by upper-case letters. *Matroid monadic second order formulas* are defined inductively as follows:

- If x and y are element variables, then $x = y$ is a formula.
- If x is an element variable and X is a set variable, then $x \in X$ and $x \in \mathrm{cl}(X)$ are formulas.
- If ϕ is a formula, then $\neg\phi$ is a formula.
- If ϕ and ψ are formulas, then $\phi \wedge \psi$ is a formula.
- If ϕ is a formula and x is an element variable, then $\exists x\phi$ is a formula.
- If ϕ is a formula and X is an element variable, then $\exists X\phi$ is a formula.

Parentheses are used to denote the precedence of symbols.

Definition 5.1.2. We define *free variables* of a formula inductively:

[1] In August 2013, Geelen, Gerrards, and Whittle have reported that that they have proved Rota's Conjecture. However, according to their words, the publishing of the proof will take several years. As of time of finishing this thesis, the proof is still unpublished.

- The formula $x = y$ has free variables x and y.
- Formulas $x \in X$ and $x \in \mathrm{cl}(X)$ have free variables x and X.
- The formula $\neg\phi$ has the same free variables as ϕ.
- The free variables of the formula $\phi \wedge \psi$ are the free variables of ϕ and ψ.
- The free variables of formula $\exists x\phi$ are free variables of ϕ except x.
- The free variables of formula $\exists X\phi$ are free variables of ϕ except X.

The set of free variables of the formula ϕ is denoted by $\mathrm{Var}(\phi)$. If we want to emphasize that formula ϕ has free elements variables $\vec{x} = (x_1, x_2, \ldots, x_n)$ and free set variables $\vec{X} = (X_1, X_2, \ldots, X_m)$, we write $\phi(\vec{x}, \vec{X})$. The formula without any free variables is a *sentence*.

Definition 5.1.3. A *variable assignment* of formula ϕ for a matroid M is a mapping that maps the element variables of $\mathrm{Var}(\phi)$ to the elements of $E(M)$ and the set variables of $\mathrm{Var}(\phi)$ to the subsets of $E(M)$. If the formula ϕ has free element variables $x_1, \ldots x_n$ and free set variables $X_1, \ldots, X_m$, we sometimes identify the assignment with the tuple $(a_1, \ldots, a_n, A_1, \ldots A_m)$, where each a_i is an element of M and each A_i is a subset of M.

Definition 5.1.4. Given a formula $\phi(\vec{x}, \vec{X})$, a matroid M, and a variable assignment $(\vec{a}, \vec{A})$, we define that the formula $\phi(\vec{x}, \vec{X})$ is *true in M under assignment* $(\vec{a}, \vec{A})$ (denoted by $M \models \phi(\vec{a}, \vec{A})$) inductively:

- If $\phi \equiv x_1 = x_2$, then $M \models \phi(a_1, a_2)$ if and only if $a_1 = a_2$.
- If $\phi \equiv x_1 \in X_1$, then $M \models \phi(a_1, A_1)$ if and only if $a_1 \in A_1$.
- If $\phi \equiv x_1 \in \mathrm{cl}(X_1)$, then $M \models \phi(a_1, A_1)$ if and only if $a_1 \in \mathrm{cl}_M(A_1)$.
- $M \models \neg\phi(\vec{a}, \vec{A})$ holds if and only if $M \models \phi(\vec{a}, \vec{A})$ does not hold.
- $M \models (\phi \wedge \psi)(\vec{a}, \vec{A})$ holds if and only if both $M \models \phi(\vec{a}, \vec{A})$ and $M \models \psi(\vec{a}, \vec{A})$.
- $M \models (\exists y\phi(y, \vec{x}, \vec{X}))(\vec{a}, \vec{A})$ if and only if there is an element $b \in E(M)$ such that $M \models \phi(b, \vec{a}, \vec{A})$.
- $M \models (\exists Y\phi(\vec{x}, Y, \vec{X}))(\vec{a}, \vec{A})$ if and only if there is a subset $B \subseteq E(M)$ such that $M \models \phi(\vec{a}, B, \vec{A})$.

In case ϕ is a sentence, we simply say ϕ *is true in M* and write $M \models \phi$. We may also say that ϕ *holds in M* or that M *satisfies* ϕ.

We also use various shorthands whenever we feel it improves clarity:

- $x \neq y$ for $\neg(x = y)$,
- $\phi \vee \psi$ for $\neg(\neg\phi \wedge \neg\psi)$,
- $\phi \to \psi$ for $\neg\phi \vee \psi$,

- $\phi \leftrightarrow \psi$ for $(\phi \to \psi) \wedge (\psi \to \phi)$,
- $\forall x \psi$ for $\neg(\exists x \neg \psi)$
- $\forall X \psi$ for $\neg(\exists X \neg \psi)$
- $X = Y$ for $\forall x (x \in X \leftrightarrow x \in Y)$
- $X \subseteq Y$ for $\forall x (x \in X \to x \in Y)$
- $X \subsetneq Y$ for $X \subseteq Y \wedge X \neq Y$

Even though we avoided function symbols, we can introduce basic set operations as shorthands too. For example, the formula $\phi(e, X, Y) \equiv e \in \mathrm{cl}(X \cup Y)$ can be written without using the function symbol $\cup$ as

$$\phi'(e, X, Y) \equiv \exists Z \big((d \in Z \leftrightarrow (d \in X \vee d \in Y)) \wedge e \in \mathrm{cl}(Z) \big).$$

Other set operations such as $\cap, \setminus$ can be defined in similar way and we will use them to improve readability.

It is more common to define the matroid monadic second-order logic with the predicate $\mathrm{indep}(Y)$ that is true if and only if the set X is independent. Note that this slight change does not affect the expressive power of matroid second-order logic, since predicate $x \in \mathrm{cl}(X)$ can be expressed using predicate $\mathrm{indep}(Y)$ and vice versa:

$$\begin{aligned} \mathrm{indep}(I) &\equiv \forall e (e \in I \to \mathrm{cl}(X) \neq \mathrm{cl}(X \setminus \{e\})), \\ e \in \mathrm{cl}(X) &\equiv \exists C (\neg \,\mathrm{indep}(C) \wedge (\forall C' \subsetneq C : \mathrm{indep}(C')) \wedge C \subseteq X \cup \{e\}). \end{aligned}$$

Therefore, every result about expressibility or inexpressibility for matroid monadic second-order logic with the indep predicate apply to our version of matroid second-order logic as well. Unless we are concerned about quantifier rank of formulas, it really does not matter which predicate encoding the matroid are we using. In our case, the predicate related to closure is convenient because of prominent role of flats in generalized parallel connection (Proposition 2.1.5 and Proposition 2.1.13).

5.2 Tree automata

One common approach for designing the linear-time algorithm for deciding MSO formulas is the explicit construction of *tree automaton.* In this section, we present basic definitions related to tree automata. For more detailed treatment of finite automata see for example chapter 10 in [FG06].

Definition 5.2.1. A *nondeterministic finite tree automaton* is a 4-tuple (S, Σ, F, Δ), where

- S is a finite set of states,
- Σ is a finite alphabet,

- $F \subseteq S$ is the set of accepting states,
- $\Delta \subseteq (S \cup \{\bot\}) \times (S \cup \{\bot\}) \times \Sigma \times S$ is a transition relation.

We say that tree automaton (S, Σ, F, Δ) is *deterministic* if Δ is a function from $(S \cup \{\bot\}) \times (S \cup \{\bot\}) \times \Sigma$ to S.

Definition 5.2.2. A Σ-*tree* is a tuple (T, λ), where T is a rooted binary tree and λ is a mapping from set of its vertices into Σ.

Definition 5.2.3. A *run* of a tree automaton $\mathcal{A} = (S, \Sigma, F, \Delta)$ on a Σ-tree (T, λ) is a mapping $\rho : T \to S$ satisfying:

- $(\bot, \bot, \lambda(t), \rho(t)) \in \Delta$ if t is a leaf,
- $(\bot, \rho(t_r), \lambda(t), \rho(t)) \in \Delta$ if t has only right child t_r,
- $(\rho(t_l), \bot, \lambda(t), \rho(t)) \in \Delta$ if t has only left child t_l,
- $(\rho(t_l), \rho(t_r), \lambda(t), \rho(t)) \in \Delta$ if t has left child t_l and right child t_r.

A run ρ is *accepting* if $\rho(r) \in F$, where r is the root of tree T. We say that the automaton $\mathcal{A}$ *accepts* a sigma tree (T, λ) if there exists an accepting run of $\mathcal{A}$ on (T, λ).

It is clear that when $\mathcal{A}$ is deterministic, there is exactly one run for every Σ-tree (T, λ) and there is algorithm computing that run in linear time with respect to size of T.

It is an important fact that for every nondeterministic finite tree automaton with k states there exists a deterministic finite tree automaton with at most 2^k states that accepts the same set of Σ-trees. The proof is a straightforward modification of analogous proof for finite automatons (see for example Section 2.3 in [HMU01]), and we omit the proof here.

5.3 Main theorem

Theorem 5.3.1. *Let ψ be a fixed matroid MSO sentence and k a fixed integer. There exists a linear time algorithm that given an amalgam decomposition $\mathcal{T}$ of width at most k of an n-element matroid M decides if ψ is satisfied by M.*

We want to prove this result by induction on formulas. However, for the purpose of induction, we need to prove slightly more general theorem.

Theorem 5.3.2. *Let ψ be a fixed matroid MSO formula and k a fixed integer. There exists a linear-time algorithm that given*

- *an amalgam decomposition $\mathcal{T}$ of width at most k of a matroid M, and*
- *a variable assignment Q of ψ for a matroid M,*

decides if $M \models \psi(Q)$.

It is clear that Theorem 5.3.1 immediately follows from this generalization. Because we will be often working with restriction of variable assignment Q, the following definition is very useful.

Definition 5.3.3. Let F be a subset of $E(M)$. The *local view of* Q *at* F is a mapping from $\mathrm{Var}(\psi)$ denoted by Q_F such that

- $Q_F(x) = Q(x)$ if x is an element variable and $x \in F$
- $Q_F(x) = \boxtimes$ if x is an element variable and $x \notin F$
- $Q_F(X) = F \cap Q(X)$ if X is a set variable

Now we have all tools required to prove the main theorem.

Proof of 5.3.2. We prove the statement by the induction on the size of the formula ψ. Note that apart from the cases using quantifiers, all described automata are deterministic. We begin the induction on atomic formulas $e_1 = e_2$, $e \in X$ and $e \in \mathrm{cl}(X)$

First consider the case when ψ is formula $e \in X$. Observe that the validity of formula $e \in X$ can be determined from local view of Q at the leaf of $\mathcal{T}$ corresponding to the element e. The automaton only needs to propagate that information to the root.

The alphabet Σ will encode the local view of Q at $E(K_v)$. Note that this can be done such that the size of alphabet depends on k only. The automaton will have states UNDECIDED, ACCEPT, and REJECT, where ACCEPT is the only accepting state. The transition function Δ is defined as follows:

- If the processed node is a leaf (i.e. the state of both children is $\perp$)
  - If $Q_v(e) = \boxtimes$, go to UNDECIDED state,
  - If $Q_v(e) \neq \boxtimes$ and $Q_v(e) \in Q_v(X)$, go to ACCEPT state,
  - If $Q_v(e) \neq \boxtimes$ and $Q_v(e) \notin Q_v(X)$, go to REJECT state,
- If the processed node is an inner node
  - If one of the children is in ACCEPT state, go to ACCEPT state,
  - If one of the children is in REJECT state, go to REJECT state,
  - If both children are in UNDECIDED state, go to UNDECIDED state

We can handle the formula $e_1 = e_2$ in a similar fashion. Again, the alphabet Σ will encode the local view of Q at $E(K_v)$.

- If the processed node is a leaf
  - If $Q_v(e_1) = Q_v(e_2) = \boxtimes$, go to UNDECIDED state,
  - If exactly one of $Q_v(e_1)$, $Q_v(e_2)$ equals to $\boxtimes$, go to REJECT state
  - If both $Q_v(e_1)$, $Q_v(e_2)$ are not equal to $\boxtimes$, go to ACCEPT state
- If the processed node is an inner node
  - If one of the children is in ACCEPT state, go to ACCEPT state,

- If at least one of the children is in REJECT state, go to REJECT state,
- If both children are in UNDECIDED state, go to UNDECIDED state

The last base case of the induction is the atomic formula $e \in \mathrm{cl}(X)$. In this case, we need the alphabet to encode all possible non-isomorphic choices of the matroid K_v, the sets J_v^1, J_v^2, J_v and D_v together with all possible local views of Q at $E(K_v)$.

Recall the notion of type introduced in Definition 4.1.1. The state at node v has two parts – first is the type of X in M_v, the second is list of all subsets Y of J_v such that $e \in \mathrm{cl}_{M_v}((X \cap E(M_v)) \cup Y)$ in case e lies in $E(M_v)$ or a symbol $\boxtimes$ denoting that e does not lie in $E(M_v)$. We denote such list by L_v.

The computation of state for leaves is trivial, let us turn our attention to non-leaf case. The automaton will compute the state at node v with children v_1, v_2 by the following procedure:

- Compute join type of f_1 and f_2
- If both L_{v_1} and L_{v_2} are equal to $\boxtimes$, set L_v to $\boxtimes$
- If one of L_{v_i} is not equal to $\boxtimes$ (this cannot happen both for $i = 1$ and $i = 2$ at once), then:
  - For each $Y \subseteq J_v$ compute the set Z_Y described in Definition 4.1.5
  - If $Z_Y \cap J_v^i$ is in L_{v_i}, add Y to L_v

Finally, the set of accepting states is

$$F = \{(f, L) : \emptyset \in L\}.$$

From the properties of types, it should be clear that for every node v such that $e \in E(M_v)$, $e \in \mathrm{cl}_{M_v}(X \cup Y)$ if and only if $Y \in L_v$. Therefore, described automaton decides the formula $e \in \mathrm{cl}(X)$.

Automaton for the formula $\neg\psi$ will be the same as the automaton for the formula ψ except that we will change the set of accepting states to its complement.

The formula $\psi = \psi_1 \wedge \psi_2$ is decided by Cartesian product of automata for formulas ψ_1 and ψ_2. The Cartesian product is defined as follows:

$$\begin{aligned} \Sigma &= \Sigma_1 \times \Sigma_2 \\ S &= S_1 \times S_2 \\ F &= F_1 \times F_2 \\ \Delta((s_1, s_2), (s_1', s_2'), (\sigma_1, \sigma_2)) &= (\Delta_1(s_1, s_1', \sigma_1), \Delta_2(s_2, s_2', \sigma_2)) \end{aligned}$$

Running the product automaton is the same as running both automatons for ψ_1 and ψ_2 and accepting if and only if both automatons accept.

To decide a formula $\psi = (\exists X)(\psi' X)$, we use a nondeterministic automaton. Informally, the automaton "guesses" the value of X in the leaves and then the computation is carried out by automaton for the formula ψ'. As mentioned before, such nondeterministic automaton can be transformed into deterministic automaton at the cost of exponential increase in number of states. This is not a problem, since we only need that number of states is bounded by some function of $|\psi|$ and k.

Denote by $\mathcal{A}' = (S', \Sigma', F', \Delta')$ the automaton for the formula ψ'. The alphabet Σ is the same as Σ', except it does not carry information about values of local view for variable X, since X is no longer free in ψ. Instead, the local information about the choice for X will be held in state.

The set of states S consists of tuples (s', X_v), where $s' \in S'$ and X_v encodes the subset $X \cap E(K_v)$. The set of accepting states F is defined as $\{(s, X_v) : s \in F'\}$. The formal definition of Δ is somewhat unwieldy, so let us present the relation only informally. The existence of the automaton carrying out required computation should be clear from the description. In every leaf, the automaton "guesses" whether the corresponding element is present in the variable X. This corresponds to saying that both $(\bot, \bot, s, \emptyset)$ and $(\bot, \bot, s, \{v\})$ are in Δ. The automaton then continues with computation according to function Δ', the only difference is that the information about $X \cap E(K_v)$ is not present in the labeling of node v by alphabet Σ', but it is contained in the state of a v. Moreover, the Δ needs to compute the part of state encoding the intersection of X with $E(K_v)$, but this is trivial, since Δ has access to intersection of X with $E(K_{v_1})$ and $E(K_{v_2})$, where v_1 and v_2 are children of v.

Strictly speaking, this approach is not correct, because the set of states must be the same for all nodes (so we cannot use state representing, say $\{v\}$). This can be circumvented by numbering elements of $E(K_v)$ for every v by numbers 1 through k. The subset can be represented simply as a subset of $\{1, \ldots, k\}$. To compute the numbering of set $X \cap E(K_v)$ from numbering of sets $X \cap E(K_{v_1})$ and $X \cap E(K_{v_2})$, we need to know the relation between numbering of elements in node and numbering in its children (i.e. element numbered by 3 in v_1 has number 5 in its parent and so on). Note that this can be done in space bounded by a function of k, so the automaton is valid and our little "cheat" merely spares us from using overly complicated notation.

The case of formula $\psi = (\exists x)(\psi')$ is similar to previous case. However, in an element variable case it might happen that an automaton produces invalid guess. In the set variable case, we needed to choose some subset of $E(M)$ (we simply picked a subset of leaves), but in the element variable case, we need to pick exactly one leaf. Of course, because leaves have no information about computation in other leaves, we might mark more than one leaf as x or mark no leaf at all. This does not pose a problem, because we can propagate the guess information upwards and discard such invalid guesses as we encounter them. Otherwise the computation is analogous to the previous case.

The induction on formulas suggests how to construct a tree automaton $\mathcal{A}$ for formula ψ. Given an amalgam decomposition $\mathcal{T}$ of a matroid M, we can label nodes of $\mathcal{T}$ (according to the induction), then run the automaton $\mathcal{A}$ on $\mathcal{T}$. From the description of the alphabet Σ, it is clear that we can make such labeling in linear time with respect to $|E(M)|$. Since the computation time of automaton in one node is bounded by some function of k, $|\psi|$, the running time of the automaton is also linear. The described algorithm therefore has running time $f(k, |\psi|)|E(M)|$ as desired.

□

6. Constructing amalgam decompositions

Our construction is related to the well-known branch decomposition of matroids. First, let us present necessary definitions.

Definition 6.0.4. A *branch decomposition* of a matroid M is a tree $\mathcal{B}$ such that leaves of $\mathcal{B}$ one-to-one correspond to elements of M and every inner vertex has degree exactly three.

If e is an edge of $\mathcal{B}$ then e splits the tree into two subtrees $\mathcal{B}_1$ and $\mathcal{B}_2$. These trees forms a partition (E_1, E_2) of the ground set of M. The *width of an edge* e is defined as $r(E_1) + r(E_2) - r(E) + 1$. The *width of the branch decomposition* $\mathcal{B}$ is the maximum width of an edge $e \in \mathcal{B}$. Finally, the *branch width of a matroid* M is the minimum width of a branch decomposition of a matroid M.

Theorem 6.0.5. *Let $\mathbb{F}$ be a finite field and M be an $\mathbb{F}$-representable matroid of branch width at most k. Then there is a linear time algorithm that given an $\mathbb{F}$-representation of M and branch decomposition $\mathcal{B}$ of M outputs an amalgam decomposition $\mathcal{T}$ of M such that $\mathcal{T}$ has width at most $2|\mathbb{F}|^{3k-3}$.*

Proof. Choose an arbitrary edge e of $\mathcal{B}$ and subdivide it. Denote the new vertex v_r and root the tree at this vertex. Call the resulting rooted binary tree $\mathcal{T}$. We now describe how to set the matroid K_v and the sets J_v, J_{v_1}, J_{v_2}, and D_v in each node v of $\mathcal{T}$ to obtain an amalgam decomposition of M.

Since we are given an $\mathbb{F}$-representation of M, we can treat elements of M as vectors in the vector space $\mathbb{F}^d$ for some $d \in \mathbb{N}$. For a set $X \subseteq E$, we will also write $\langle X \rangle$ for the linear span of a set X in $\mathbb{F}^d$. Note that the rank of a set X is equal to $\dim\langle X \rangle$.

If v is a leaf then there is by the definition of amalgam decomposition only one option how to set the the matroid and the designated sets. Let v be an internal node of $\mathcal{T}$ with children v_1 and v_2. Denote by E_i the set of elements corresponding to leaves under v_i and by E' the set $E(M) \setminus (E_1 \cup E_2)$.

Denote by F_1 the set $\langle E_1 \rangle \cap \langle E_2 \cup E' \rangle$, by F_2 the set $\langle E_2 \rangle \cap \langle E_1 \cup E' \rangle$, and by F' the set $\langle E' \rangle \cap \langle E_1 \cup E_2 \rangle$. The set $E(K_v)$ will be the set $\langle F_1 \cup F_2 \cup F' \rangle$. Because the branch width of $\mathcal{B}$ is at most k, the dimension of each of the sets F_1, F_2, F' is at most $k-1$. Therefore, the dimension of $E(K_v)$ is at most $3k-3$, so $E(K_v)$ can have at most $|\mathbb{F}|^{3k-3}$ elements. The independent sets of K_v are exactly the linearly independent sets.

We define the set J_v as the simplification of the set F' – that is we delete all loops and from every class of parallel elements we choose exactly one element. Similarly, we define J_v^i as the simplification of the set F_i.

Because the definition requires that J_v^1 and J_v^2 are disjoint, we "split" every element of $E(K_v)$ that lies in both J_v^1 and J_v^2 into two parallel elements. This process can increase the size of $E(K_v)$ to at most double of its previous size, therefore we have $|E(K_v)| \leq 2|\mathbb{F}|^{3k-3}$. Finally, the set D_v is defined as $E(K_v) \setminus (J_v \cup E(M))$.

We need to check that this is a valid amalgam decomposition. Recall that this means verifying that

(i) J_v^1 is disjoint from J_v^2,

(ii) J_v is disjoint from D_v,

(iii) J_v is a modular semiflat of K_v,

(iv) J_{v_r} is the empty set (v_r is the root of $\mathcal{T}$),

Statements (i) and (ii) follow immediately. To prove (iii), first note that F' is a flat in K_v since it is a subspace. Because the J_v was formed by removing parallel edges and loops from F', it must be a semiflat. To see that it is modular, observe that in K_v flats correspond to subspaces of $\mathbb{F}^d$, so for flats X, Y the following holds:

$$r(X \cup Y) = \dim\langle X \cup Y\rangle = \dim X + \dim Y - \dim X \cap Y = r(X) + r(Y) - r(X \cap Y).$$

Finally to prove (iv), note that in the root $E_1 \cup E_2$ is the whole ground set of M, so E' is empty. Therefore F' is the zero dimensional vector space and J_{v_r} is empty.

Note that the time required to construct matroid K_v and the sets J_v^1, J_v^2, J_v, and D_v in each node depends only on $\mathbb{F}$ and k. Therefore, the running time of the algorithm is linear in size of the matroid. □

Together with fixed-parameter tractable algorithm of Hliněný and Oum [HO08], we obtain the following.

Corollary 6.0.6. *Let M be an $\mathbb{F}$-representable matroid of branch width at most k. Then there is a fixed-parameter tractable algorithm that given an $\mathbb{F}$-representation of M outputs an amalgam decomposition of width at most $2|\mathbb{F}|^{3k-3}$.*

Conclusion

In our work, we presented a width parameter and a decomposition that has three key properties that are expected from a well-behaved decomposition – computation of Tutte polynomial, deciding MSO properties, and constructing the decomposition. For the construction of the decomposition, we needed an $\mathbb{F}$-representation of matroid, however, as we already argued in the introduction, requiring an $\mathbb{F}$-representation for either deciding formulas or constructing decompositions seems to be unavoidable. The advantage over approach using branch width [Hli06a] is that we are able to decide MSO properties in linear time even for matroids that are not representable.

Our work improves the matroid decomposition used by Strozecki [Str11], as our way of joining matroids can replace both parallel and series connection. We also claim that the operation used in joining is natural, since generalized parallel connections are closely related to *pushouts* in a suitable category of matroids (see [Bry75] for details). This could bring a better insight into the structure of the decomposed matroid than the decomposition of Král' [Kra12].

However, because generalized parallel connections are not guaranteed to exist, we are required to deal with sufficient conditions to ensure that all operations involved are well-defined.

A keen reader surely noticed that some of our bounds (bound to number of all types to name one example) are far from being tight. In our results, we frequently need an arbitrary bound while caring little for how good the bound actually is. Therefore, we have deliberately chosen to use bounds that are simple but can be proved more easily and clearly. Similar remark also hold for the time complexity our algorithms. In lot of cases, we probably could have used a clever trick to reduce the multiplicative constant slightly, but we opted for a more simple description to make the presentation less obscured.

References

[BB73] U. Bertelé, F. Brioschi. *On Non-serial Dynamic Programming.* Journal of Combinatorial Theory, Series A, Volume 14 (2) (1973), 137-148.

[Bod96] H. Bodlaender. *A Linear-Time Algorithm for finding Tree-Decompositions of Small Treewidth.* SIAM J. Comput. 25(6):1305-1317 (1996).

[Bry75] Thomas H. Brylawski. *Modular constructions for combinatorial geometries.* Trans. Amer. Math. Soc. **203**, 1-44.

[Cou90] B. Courcelle. *The monadic second-order logic of graphs. I. Recognizable Sets of Finite Graphs.* Inf. Comput. 85(1):12–75, 199.

[DF99] R. Downey, M. Fellows. *Parameterized complexity.* Springer New York, 1999. ISBN 978-0-387-94883-6

[FG06] J. Flum, M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag Berlin Heidelberg, 2006. ISBN 3-540-29952-1

[GGW02] J. Geelen, A. M. H. Gerards, G. Whittle. *Branch-Width and Well-Quasi-Ordering in Matroids and Graphs.* J. Comb. Theory, Ser. B 84(2): 270-290 (2002)

[GGW06] J. Geelen, A. M. H. Gerards, G. Whittle. *On Rota's conjecture and excluded minors containing large projective geometries.* J. Comb. Theory, Ser. B 96(3): 405-425 (2006)

[GGW07] J. Geelen, A. M. H. Gerards, G. Whittle. *Excluding a planar graph from* GF(q)*-representable matroids.* J. Comb. Theory, Ser. B 97(6): 971-998 (2007)

[GKO11] T. Gavenčiak, D. Král', S. Oum. *Deciding first order logic properties of matroids.* arXiv:1108.5457v1 (2011)

[Hli03] P. Hliněný. *On Matroid Properties Definable in the MSO Logic.* Mathematical Foundation of Computer Science 2003, Lecture Notes in Computer Science 2747, Springer-Verlag (2003), 470-479.

[Hli06a] P. Hliněný. *Branch-width, parse trees, and monadic second-order logic for matroids.* Journal of Combinatorial Theory, Series B, Volume 96(3) (2006), 325-351.

[Hli06b] P. Hliněný. *The Tutte Polynomial for Matroids of Bounded Branch-Width.* Combinatorics, Probability & Computing 15(3): 397-409 (2006)

[Hli06c] P. Hliněný. *On Matroid Representability and Minor Problems.* MFCS 2006: 505-516

[HO08] P. Hliněný, S. Oum. *Finding Branch-Decompositions and Rank-Decompositions.* SIAM J. Comput. 38(3): 1012-1032 (2008)

[HW06] P. Hliněný, G. Whittle. *Matroid tree-width.* Eur. J. Comb. 27(7): 1117-1128 (2006)

[HW09] P. Hliněný, G. Whittle. *Addendum to matroid tree-width.* Eur. J. Comb. 30(4): 1036-1044 (2009)

[HMU01] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation – international edition (2. ed).* Addison-Wesley 2001, ISBN 978-0-201-44124-6

[JVW90] F. Jaeger, D. L. Vertigan, D. J. A. Welsh. *On the computational complexity of Jones and Tutte polynomials.* Math. Proc. Cambridge Philos. Soc. 108 (1990), 35-53.

[Kra12] D. Král'. *Decomposition width of matroids.* Discrete Applied Mathematics 160(6): 913-923, (2012).

[Lib04] Leonid Libkin. *Elements of Finite Model Theory.* Springer-Verlag, 2004. ISBN 3-540-21202-7

[MT13] Lukáš Mach, Tomáš Toufar. *Amalgam width of matroids.* arXiv:1304.0299v2 (2013)

[OS06] S. Oum, P. Seymour. *Approximating clique-width and branch-width.* J. Comb. Theory, Ser. B 96(4): 514-528 (2006)

[Oxl06] James Oxley. *Matroid Theory.* Oxford University Press, 2006. ISBN 0-19-920250-8

[RS84] N. Robertson, P. Seymour. *Graph minors. III. Planar tree-width.* J. Comb. Theory, Ser. B, Vol. 36 (1) (1984), 49-64.

[RS86] N. Robertson, P. Seymour. *Graph minors. II. Algorithmic Aspects of Tree-width.* J. Algorithms 7 (3) (1986), 309-322.

[Rot71] G. C. Rota. *Combinatorial Theory, Old and New.* Proc. of International Congress of Mathematics (Nice, 1970), Gauthier-Villars, Paris (1971), 229-233.

[Sey81] P. Seymour. *Recognizing graphic matroids.* Combinatorica 1(1): 75-78 (1981)

[Str11] Y. Strozecki. *Monadic second-order model-checking on decomposable matroids.* Discrete Applied Mathematics 159(10): 1022-1039, (2011).

[Wel90] D. J. A. Welsh. *The Tutte polynomial.* Random Structures Algorithm 15 (1990), 210-228.