Computing character tables and Cartan matrices of finite monoids with fixed point counting

Balthazar Charles

Computing character tables and Cartan matrices of finite monoids with fixed point counting.

Balthazar Charles[a]

[a]*LISN, Université Paris-Saclay, 6 Rue Noetzlin, Gif-sur-Yvette, 91190, France*

Abstract

In this paper we present an algorithm for efficiently counting fixed points in a finite monoid M under a conjugacy-like action. We then prove a formula for the character table of M in terms of fixed points, which allows for the effective computation of both the character table of M other a field of null characteristic, as well as its Cartan matrix, using a formula from [Thiéry '12], again in terms of fixed points. We discuss the implementation details of the resulting algorithms and provide benchmarks of their performances.

Introduction

The last two decades have seen the development of a new dynamic around the study of monoid representation theory. This is due to applications to certain types of discrete Markov chains and especially Markov chains used to randomly generate combinatorial objects first uncovered in the seminal article of Brown [**?**]. This has lead to an exploration of the combinatorial properties of monoid representations, for instance in [**?**], [**?**] or [**?**].

In this last article [**?**], Thiéry gives a formula for the Cartan matrix of a finite monoid of M in terms of the number of fixed points and the character table of M. More precisely, the formula involves computing the cardinality of the set $\{s \in M \mid hsk = s\}$ for any $h, k \in M$. In this paper, we set out to use this formula to effectively compute the Cartan matrix of the algebra of M over a perfect field $\mathbf{k}$ of null characteristic.

Two difficulties have to be overcome in the pursuit of this goal. Firstly, the cardinality of many interesting families of monoids tends the increase very quickly. For instance, the cardinality of the *full transformation monoid* T_n of all functions from $[\![1, n]\!]$ to $[\![1, n]\!]$ is n^n, making the naive computation of $|\{s \in M \mid hsk = s\}|$

impractical even for small n. To remedy this we provide an algorithm to efficiently compute this statistic. Secondly, to use the formula one has to compute the character table of the monoid. The Clifford-Munn-Ponizovskii Theorem (such as presented in [**?**]) gives an explicit description of the simple $\mathbf{k}M$-modules. This technically makes the computation of the character table possible, provided that we know how to compute the simple modules associated to certain groups. However, this approach is rather computationally inefficient as it necessitates iterating over explicitly constructed irreducible group representations and dealing with high dimensional tensor products and quotients. In contrast, the method we propose allows for a "bundling" of the group representations using the easier to compute $\mathcal{L}$-class modules. Moreover, although themselves high dimensional, these modules are combinatorial and the action of the monoid can be efficiently computed. We also note that although some results on character tables are known in the case of interesting families of monoids (full transformation monoids, inverse monoids), no algorithms are available to compute the character table of an arbitrary finite monoid. Thus, for the general case, we prove a formula for the character table that allows for computation exploiting the Green structure of the monoid for increased efficiency.

In the first section of this paper, we present the results necessary for our fixed-points counting algorithm, with a particular emphasis on the notions of Green classes and Schützenberger groups. In the second section, we prove a formula for the character table of a finite monoid, after recalling the necessary module and character theoretic results. In the third section, we give and discuss the algorithms and equation systems used for computing the Cartan matrix with two focuses: the algorithms for fixed point counting and the equation system to compute the character table. Finally, in the last section, we discuss the performance of these algorithms in terms of execution time and size of tractable problems.

I. Combinatorics of fixed point counting.

In this section we first recall essential and elementary results on the Green structure of finite monoids and on Schützenberger groups. The informed reader may skip this first paragraph, with the exception of the notations (4) that are used throughout this paper. We then use these results to devise a fixed point counting method.

In the totality of this paper, we assume that all monoids are finite. We will often use the following special case of finite monoid to illustrate the various results presented hereafter.

Definition 1 (The full transformation monoid)**.** Consider the set T_n of all transformations of the set $\{1, \ldots, n\}$, equipped with the multiplication given by map composition: $\forall f, g \in T_n, fg = f \circ g$ (with the argument on the right : $f \circ g(x) = f(g(x))$). This is a monoid, aptly named the *full transformation monoid*. A submonoid of T_n is called a transformation monoid and n is called its *rank*.

1. Green structure and Schützenberger groups

Although finite monoids have been considered to be much wilder objects than groups, it turns out that, with the right optics, they are actually highly structured by their internal multiplication. Consider the divisibility relation: x divides y if $x = yz$ for some z. If x, y, z are taken in a group G, the relation is trivial. If however, we take them in a general monoid M, left or right translation by an arbitrary element need not be surjective, making the question of $x \in M$ being a left (or right) multiple of $y \in M$ non-trivial. These questions of "divisibility" in a general monoid are studied under the name of Green structure, of which we give a brief overview necessary for our purpose in the subsection below. In the following subsection, we also present the related notion of Schützenberger groups.

1.1. Green Structure

Definition 2 (Green's relations)**.** Let M be a finite monoid and a, b two of its elements. The *Green relations* are:

- The $\mathcal{L}$ preorder is defined by $a \leq_{\mathcal{L}} b \Leftrightarrow b = ua$ for some $u \in M$. The associated equivalence relation is : $a \in \mathcal{L}(b) \Leftrightarrow a \leq_{\mathcal{L}} b$ and $b \leq_{\mathcal{L}} a \Leftrightarrow Ma = Mb$.
- The $\mathcal{R}$ preorder is defined by $a \leq_{\mathcal{R}} b \Leftrightarrow b = av$ for some $v \in M$. The associated equivalence relation is : $a \in \mathcal{R}(b) \Leftrightarrow a \leq_{\mathcal{R}} b$ and $b \leq_{\mathcal{R}} a \Leftrightarrow aM = bM$.
- The $\mathcal{J}$ preorder is defined by $a \leq_{\mathcal{J}} b \Leftrightarrow b = uav$ for some $u, v \in M$. The associated equivalence relation is: $a \in \mathcal{J}(b) \Leftrightarrow a \leq_{\mathcal{J}} b$ and $b \leq_{\mathcal{J}} a \Leftrightarrow MaM = MbM$.
- The $\mathcal{H}$ equivalence relation is defined by $a \in \mathcal{H}(b) \Leftrightarrow a \in \mathcal{L}(b)$ and $a \in \mathcal{R}(b)$.

It can be proven (see for instance [**?** , Theorem 1.9]) that in finite monoids, the relation $\mathcal{J}$ is the smallest equivalence relation containing $\mathcal{R}$ and $\mathcal{L}$. This is not

true in general, and this smallest relation is usually denoted by $\mathcal{D}$ in the literature. Since we are only interested in finite monoids, we shall only use the terms $\mathcal{J}$-relation, $\mathcal{J}$-class, etc. From the definition, it is clear that the relation $\mathcal{L}$ and $\mathcal{R}$ are finer than $\mathcal{J}$ and that $\mathcal{H}$ is finer that both $\mathcal{L}$ and $\mathcal{R}$. Because of this, "the $\mathcal{L}$-class of some $\mathcal{H}$-class H" or "the $\mathcal{J}$-class of some $\mathcal{R}$-class R", etc... are well-defined and we shall denote them by $\mathcal{L}(H), \mathcal{J}(R)$, etc.

Example 3 (Green relations in T_n)**.** Let a, b be two elements of M.

- We have $a\mathcal{L}b$ if and only if they have the same *kernel* $\ker a = \{a^{-1}\{i\} \mid i \in [\![1, n]\!]\}$.
- We have $a\mathcal{R}b$ if and only if they have the same image, $\operatorname{Im} a = \{a(i) \mid i \in [\![1, n]\!]\}$.
- Since T_n is finite, $\mathcal{J}$ is generated by $\mathcal{L}$ and $\mathcal{R}$ so $a\mathcal{J}b$ if and only if $\operatorname{Im}(a)$ and $\operatorname{Im}(b)$ (or equivalently $\ker(a)$ and $\ker(b)$) have the same cardinality.
- Since $\mathcal{H}$ is the intersection of $\mathcal{L}$ and $\mathcal{R}$, $a\mathcal{H}b$ if and only a and b have the same image and the same kernel.

These conditions are necessary conditions in any transformation monoid. To get that they are sufficient, we use the fact that $\mathfrak{S}_n \subset T_n$ and that we can rearrange both image and kernel as we please.

The following notation will prove useful, as the study of Green relations is, in part, the study of the maps given by left and right translations in the monoid.

Notation 4. Let h, k be elements of M and S be a subset of M. We denote by:

- $h\times_S$ the application from S to hS defined by $s \mapsto hs$,
- $\times_S k$ the application from S to Sk defined by $s \mapsto sk$,
- ${}_M\operatorname{Stab}(S) = \{m \in M \mid mS = S\}$,
- $\operatorname{Stab}_M(S) = \{m \in M \mid Sm = S\}$,
- $\operatorname{Fix}_S(h, k) = \{s \in S \mid hsk = s\}$.

Using these notations, let us recall Green's Lemma, which is one of the central elements of the theory of Green relations, as it shows that the structure of the relations is actually heavily constrained, making their study practical.

Lemma 5 (Green's Lemma)**.** *Let a, a' be two elements in the same $\mathcal{L}$-class and let λ, λ' such that $\lambda a = a'$ and $\lambda' a' = a$. Then $\lambda \times_{\mathcal{R}(a)}$ and $\lambda' \times_{\mathcal{R}(a')}$ are reciprocal bijections. Moreover, for any $\mathcal{L}$-class L $\lambda \times_{\mathcal{R}(a) \cap L}$ and $\lambda' \times_{\mathcal{R}(a') \cap L}$ are reciprocal bijections.*

Similarly, if a, b are two elements in the same $\mathcal{R}$-class and ρ, ρ' are such that $\rho a = b$ and $\rho' b = a$, then $\times_{\mathcal{L}(a)} \rho$ and $\times_{\mathcal{L}(b)} \rho'$ are reciprocal bijections. Moreover, for any $\mathcal{R}$-class R $\times_{\mathcal{L}(a) \cap R} \rho$ and $\times_{\mathcal{L}(b) \cap R} \rho'$ are reciprocal bijections.

An important consequence of Green's Lemma is that $\mathcal{J}$-classes can be neatly organized as *eggbox pictures*[1]: the $\mathcal{J}$-class can be represented as a rectangular array with the $\mathcal{L}$-classes as columns, the $\mathcal{R}$-classes as rows and the $\mathcal{H}$-classes, the eggs, in the cases. This level organization is actually what allows for efficient computer representation of monoids and most of their algorithmic exploration.

1.2. Schützenberger groups

The Green structure offers a second way of facilitating computer exploration of monoids through groups that arise as stabilizers of some Green classes. These are called the *Schützenberger groups* and – this a running theme of monoid theory – allow for a number of monoid theoretic questions to be formulated in terms of groups for which we dispose of an array of efficient algorithms.

Definition 6 (Schützenberger groups)**.** Let H be a $\mathcal{H}$-class. The set $\{h \times_H \mid h \in {}_M\operatorname{Stab}(H)\}$ equipped with map composition is a subgroup of $\mathfrak{S}(H)$ called the *left Schützenberger group* and denoted by $\Gamma(H)$.

Similarly, $(\{\times_H k \mid k \in \operatorname{Stab}_M(H)\}, \circ)$ is a subgroup of $\mathfrak{S}(H)$ called the *right Schützenberger group* and denoted by $\Gamma'(H)$.

Example 7. Consider $H = \mathcal{H}([1\ 3\ 1])$ (the elements of T_n are given in function notation in all examples). We have :

$$ {}_M\operatorname{Stab}(H) = \{[1\ 2\ 3], [3\ 2\ 1], [1\ 3\ 3], [3\ 1\ 1], [1\ 1\ 3], [3\ 3\ 1]\} $$

and subsequently, $\Gamma(H) = \{[1\ 2\ 3]\times_H, [3\ 3\ 1]\times_H\}$. Notice that, as elements of $\Gamma(H)$, $[3\ 3\ 1]\times_H = [3\ 2\ 1]\times_H$ and that the only important thing is the permutations induced by the elements of ${}_M\operatorname{Stab}(H)$ on $\operatorname{Im} H$. Thus, in the case of transformation monoids, the left Schützenberger group of a $\mathcal{H}$-class H can be represented as a subgroup of $\mathfrak{S}(\operatorname{Im} H)$. In the same way, the right

[1] Terminology introduced in [**?**]

Schützenberger groups can be represented as subgroups of $\mathfrak{S}(\ker H)$. This fact is used to represent the Schützenberger groups in Section III.

Our precedent remark on exploiting Schützenberger groups to get efficient algorithms for computational monoid theoretic questions is seconded by the fact that Schützenberger groups do not contain any "superfluous information" in the following sense.

Proposition 8. *Let H be a $\mathcal{H}$-class. The natural actions of $\Gamma(H)$ and $\Gamma'(H)$ on H are free and transitive.*

We reproduce below a proof for Proposition 8 from [**?**] for the purpose of showcasing the main argument. The argument itself is widely known, and we will use it multiple times in the remainder of this paper.

Proof. Two elements $h, h' \in H$ are in the same $\mathcal{L}$-class so there is some $u \in M$ such that $uh = h'$. By Green's Lemma, this means that $u \in {}_M \operatorname{Stab}(H)$, so $\Gamma(H)$ acts transitively on H. Suppose that $uh = h$ for some $u \in M$. Since h, h' are also in the same $\mathcal{R}$ class, there is some v such that $h' = hv$, so $uh' = uhv = hv = h'$: an element of $\Gamma(H)$ either fixes all points in H or fixes none. The only element of $\Gamma(H)$ that fixes all points (and, consequently, the only one that fixes any point) is the identity and thus the action is free. The same arguments apply for $\Gamma'(H)$. □

A special case that is interesting to note, and that will be important later, is the case where H is the $\mathcal{H}$-class of an idempotent :

Definition 9. An element $e \in M$ is *idempotent* if $e^2 = e$. Given an idempotent e, the set $G_e = \{x \in M \mid \exists y \in M, xy = yx = e\}$ is called the *maximal subgroup at e*. One can check that G_e is indeed a group and that $G_e = \mathcal{H}(e)$.

In that case, $\Gamma(H)$ and $\Gamma'(H)$ can be defined as before, and are canonically isomorphic to G_e, simply because $G_e \subset {}_M \operatorname{Stab}(H)$ naturally induce a map making it a subgroup of $\Gamma(H)$ and that since G_e acts freely and transitively on H, this map must be injective and surjective (and similarly for $\Gamma'(H)$).

Example 10. Consider, in Example 3, $e = [1\ 2\ 2]$ and $H = \mathcal{H}(e)$. e is an idempotent, and, indeed, $H = G_e$ is group : setting $t = [2\ 1\ 1]$, we have $e^2 = e, t^2 = e$ and $et = te = t$. As noted in Example 7 :

$$\Gamma(H) = \mathfrak{S}(\{1, 2\}), \qquad \Gamma'(H) = \mathfrak{S}(\{\{1\}, \{2, 3\}\}).$$

Note that the canonical isomorphism between $\Gamma(H)$ and $\mathcal{H}(e)$ is simply given

by $g \in \Gamma(H) \mapsto g \cdot e \in H$.

2. Counting fixed points

Consider the problem of counting the number of elements of the set $\mathrm{Fix}_G(h,k)$ where G is a finite group and $h,k \in G$. If $\mathrm{Fix}_G(h,k)$ is non-empty, it contains an element γ such that $h\gamma k = \gamma$, or equivalently $h = \gamma k^{-1}\gamma^{-1}$. So for any $g \in \mathrm{Fix}_G(h,k)$ we have:

$$hgk = g \Leftrightarrow \gamma k^{-1}\gamma^{-1}gk = g \Leftrightarrow \gamma^{-1}gk = k\gamma^{-1}g.$$

This means that $g \in \gamma C_G(k)$ where $C_G(k)$ is the centralizer of k in G. Because the other inclusion is obvious, we get a description of $|\mathrm{Fix}_G(h,k)|$: either h and k^{-1} are conjugated in which case there are $|C_G(k)|$ fixed points, or they are not, and there are no fixed points. In the case of a monoid, this reasoning mostly breaks: we crucially used the invertibility property, which monoids lack. The Schützenberger groups seem to be ideal candidates to get back some of this invertibility. In this section we clarify the role of the Schützenberger groups for counting fixed points, how to give meaning to "$h\times_H$ and $\times_H k$ are in the same conjugacy class", and how to factorize our previous remark over all the $\mathcal{H}$-classes of the same $\mathcal{J}$-class.

As the bijections between $\mathcal{L}$ (and $\mathcal{R}$) classes will play a major role in the remainder of this section, we introduce the following notation.

Notation 11. Given R, R' two $\mathcal{R}$-classes in the same $\mathcal{J}$-class, we say that (λ, λ') is a *left Green pair* with respect to (R, R') if:

- $\lambda R = R'$ and $\lambda' R' = R$.
- $(\lambda\lambda')\times_R = \mathrm{Id}_R$ and $(\lambda'\lambda)\times_{R'} = \mathrm{Id}_{R'}$

Similarly, given two $\mathcal{L}$-classes L, L' in the same $\mathcal{J}$-classes, (ρ, ρ') is a *right Green pair* with respect to (L, L') if:

- $L\rho = L'$ and $L'\rho' = L$.
- $\times_L(\rho\rho') = \mathrm{Id}_L$ and $\times_{L'}(\rho'\rho) = \mathrm{Id}_{L'}$

Using Green pairs, one can transport the problem of counting fixed points in an arbitrary $\mathcal{H}$-class to a reference $\mathcal{H}$-class.

Proposition 12. *Let $H_1, H_2 \subset J$ be two $\mathcal{H}$-classes contained in the same $\mathcal{J}$-class. Then there exists $\lambda, \lambda', \rho, \rho'$ such that:*

- (λ, λ') *is a left Green pair with respect to* $(\mathcal{R}(H_1), \mathcal{R}(H_2))$,
- (ρ, ρ') *is a right Green pair with respect to* $(\mathcal{L}(H_1), \mathcal{L}(H_2))$.

and the maps $x \mapsto \lambda' x \rho'$ *and* $x \mapsto \lambda x \rho$ *are reciprocal bijections between* H_2 *and* H_1. *Moreover, let* $(h, k) \in {}_M\mathrm{Stab}(\mathcal{R}(H_2)) \times \mathrm{Stab}_M(\mathcal{L}(H_2))$ *and define* $(h', k') = (\lambda' h \lambda, \rho k \rho')$. *Then the maps restrict to bijection between the sets* $\mathrm{Fix}_{H_2}(h, k) = \{a \in H_2 \mid hak = a\}$ *and* $\mathrm{Fix}_{H_1}(h', k') = \{a \in H_1 \mid h'ak' = a\}$.

Proof. The first part is exactly [**?** , Proposition 2.3 (a)]. The second part is immediately deduced from [**?** , Proposition 2.3 (c)]. Still, for completeness, we provide a proof in our very simple (and less general) setup. Let a_1 be an element of H_1 and denote by $a_2 = \lambda a_1 \rho$. Then:

$$ha_2k = a_2 \Leftrightarrow \lambda' h a_2 k \rho' = \lambda' a_2 \rho' \Leftrightarrow (\lambda' h \lambda) a_1 (\rho k \rho') = a_1 \Leftrightarrow h' a_1 k' = a_1$$

so these bijections restrict to $\mathrm{Fix}_{H_1}(h', k')$ and $\mathrm{Fix}_{H_2}(h, k)$. □

Keeping in mind our computational goals, transporting the problem of counting fixed points from H_2 to H_1 is helpful, as for the price of 4 monoid multiplications, we can use a lot of precomputations specific to a particular $\mathcal{H}$-class, avoiding the repetition of multiple similar computations for each $\mathcal{H}$-class.

The question is now to determine the fixed points in a single $\mathcal{H}$-class, using our previous remark on conjugacy. Let us first clarify the idea of elements of the left and right Schützenberger groups being in the same conjugacy class.

Proposition 13. *Given and* $\mathcal{H}$*-class* H, $a \in H$ *and* $g \in \Gamma(H)$, *we define* $\tau_a(g)$ *as the unique element of* $\Gamma'(H)$ *such that* $g \cdot a = a \cdot \tau_a(g)$. *Then* $\tau_a : \Gamma(H) \longrightarrow \Gamma'(H)$ *is an anti-isomorphism*[2]. *Moreover* τ_a *gives rise to a bijection between the conjugacy classes of* $\Gamma(H)$ *and* $\Gamma'(H)$ *that is independent of the choice of* a.

Proof. The first part is known since [**?**]. We want to check that for $a \in H, g \in \Gamma(H)$, the conjugacy class of $\tau_a(g)$ is defined independently of a. Take any $a, b \in H$. By definition of $\Gamma(H)$, there exist some $h \in \Gamma(H)$ such that $b = h \cdot a$. So :

$$b \cdot \tau_a(g) = (h \cdot a) \cdot \tau_a(g) = h \cdot (g \cdot a) = hgh^{-1} \cdot (h \cdot a) = b \cdot \tau_b(hgh^{-1}).$$

[2]Note that some authors equip the right Schützenberger group with reversed composition, and thus obtain an isomorphism instead of anti-isomorphism.

Since $\Gamma'(H)$ acts freely, this means that $\tau_a(g) = \tau_b(h)\tau_b(g)\tau_b(h)^{-1}$ and thus $\tau_a(g)$ is conjugated with $\tau_b(g)$, which proves that the conjugacy class of $\tau_a(g)$ is indeed defined independently of a. Finally, as τ_a is a group morphism, the image are of two conjugated elements are conjugated, meaning that τ_a does indeed induces bijection between the conjugacy classes of the left and right Schützenberger groups, independently of the choice of a. □

In the next proposition, we formalize the idea of searching the fixed points as some centralizer, but in the context of a monoid.

Proposition 14. *Let H be a $\mathcal{H}$-class, $a \in H$ and $(h,k) \in {}_M\operatorname{Stab}(\mathcal{R}(H)) \times \operatorname{Stab}_M(\mathcal{L}(H))$. Then*

$$|\operatorname{Fix}_H(h,k)| = \begin{cases} |C_{\Gamma'(H)}(\times_H k)| & \text{if } \tau_a(h\times_H)^{-1} \in \overline{\times_H k} \\ 0 & \text{otherwise} \end{cases}$$

where $\overline{\times_H k}$ is the conjugacy class of $\times_H k$ in $\Gamma'(H)$ and $C_{\Gamma'(H)}(\times_H k)$ is the centralizer in $\Gamma'(H)$ of $\times_H k$.

Proof. For simplicity, we commit an abuse of notation by denoting $h\times_H$ as h and $\times_H k$ as k. Let a be any element of H.

$$\begin{aligned} \operatorname{Fix}_H(h,k) &= \{b \in H \mid hbk = b\} \\ &= \{a \cdot g \mid g \in \Gamma'(H) \text{ and } ha \cdot gk = a \cdot g\} \\ &= \{a \cdot g \mid g \in \Gamma'(H) \text{ and } a \cdot \tau_a(h)gk = a \cdot g\} \\ &= \{a \cdot g \mid g \in \Gamma'(H) \text{ and } \tau_a(h)gk = g\}. \end{aligned}$$

The last equality comes from the fact that $\Gamma'(H)$ acts freely so we can simplify the a. Suppose that $\operatorname{Fix}_H(h,k)$ is non-empty and let $\gamma \in \Gamma'(H)$ such that $\tau_a(h)\gamma k = \gamma$. Then, for any $g \in \Gamma'(H)$:

$$\tau_a(h)gk = g \Leftrightarrow g^{-1}\tau_a(h)gk = e \Leftrightarrow g^{-1}\gamma k^{-1}\gamma^{-1}gk \ = e \Leftrightarrow [\gamma^{-1}g, k] = e$$

where $[\cdot,\cdot]$ is the commutation bracket. This means that

$$\operatorname{Fix}_H(h,k) = \{a \cdot g \mid g \in \gamma C_{\Gamma'(H)}(k)\}.$$

Note that because, again, $\Gamma'(H)$ acts freely, $\operatorname{Fix}_H(h,k)$ has the same cardinality as $C_{\Gamma'(H)}(k)$ and that, from Proposition 13 this is independent from the choice of a which proves the result. □

Example 15. Consider $a = [1\ 2\ 2\ 3] \in T_4$ and $H = \mathcal{H}(a)$. We have $\operatorname{Im} a = \{1, 2, 3\}$ and $\ker a = \{\{1\}, \{2, 3\}, \{4\}\}$. Notice that H is not a group since $a^2 = [1\ 2\ 2\ 2] \notin H$. Considering the Schützenberger groups as symmetric groups on the image and kernel common to all elements of H as in Example 7, we have $\Gamma(H) = \mathfrak{S}(\operatorname{Im} a)$ and $\Gamma'(H) = \mathfrak{S}(\ker a)$.

Let us first check for fixed points under the action of $h = [1\ 2\ 3\ 4]$ on the left and $k = [2\ 1\ 1\ 4]$ on the right. Seen as an element of $\Gamma(H)$, h corresponds to $\mathrm{Id}_{\operatorname{Im} a}$ and k corresponds to $(\{2, 3\}\ \{1\})$ in $\Gamma'(H)$. Since we have $\tau_a(h) = \mathrm{Id}_{\ker a}$, it follows that $|\operatorname{Fix}_H(h, k)| = 0$.

If we now take h to be $[1\ 3\ 2\ 4]$, the corresponding element in $\Gamma(H)$ is $(2\ 3)$ and $\tau_a(h) = (\{2, 3\}\ \{4\})$. Since $(\{2, 3\}\ \{4\})$ and $(\{2, 3\}\ \{1\})$ are conjugated in $\mathfrak{S}(\ker a)$, the set of fixed points is non-empty. Their centralizers have cardinal 2 and one can indeed check that $[2\ 3\ 3\ 1]$ and $[3\ 2\ 2\ 1]$ are the only fixed points in H.

Putting together the previous results, we get the following corollary on the cardinality of $\operatorname{Fix}_J(h, k)$.

Corollary 16. *Let J be a $\mathcal{J}$-class, a any element of J and denote by $H_0 = \mathcal{H}(a), R_0 = \mathcal{R}(a), L_0 = \mathcal{L}(a)$. Let h, k be any elements of M. We denote by:*

- *(λ_R, λ'_R) a left Green pair with respect to (R_0, R) for each $\mathcal{R}$-class $R \subset J$,*
- *(ρ_L, ρ'_L) a right Green pair with respect to (L_0, L) for each $\mathcal{L}$-class $L \subset J$,*
- *$S_{\mathcal{R}}(h) = \{R \subset J \mid R$ is a $\mathcal{R}$-class and $hR = R\}$,*
- *$S_{\mathcal{L}}(k) = \{L \subset J \mid R$ is a $\mathcal{L}$-class and $Lk = L\}$*

Denoting the set of conjugacy classes of $\Gamma'(H_0)$ as C, we further define two vectors:

- $r_J(h) = (|C_{\Gamma'(H)}(g)| \cdot |\{R \in S_{\mathcal{R}}(h) \mid \tau_a(\lambda'_R h \lambda_R) \in \bar{g}\}|)_{\bar{g} \in C}$,
- $l_J(k) = (|\{L \in S_{\mathcal{L}}(k) \mid \rho'_L k \rho_L \in \bar{g}\}|)_{\bar{g} \in C}$.

Then $\operatorname{Fix}_J(h, k)$ has cardinality the dot product of $r_J(h)$ with $l_J(k)$.

II. Modules: character table and Cartan matrix

We will discuss the representation theory of a monoid M over $\mathbf{k}$ using the language of modules, for which we assume a level a familiarity from the reader. For a full introduction (and more!) to the representation theory of finite monoids in module theoretic terms, we refer to *Representation theory of finite monoids* from Benjamin Steinberg [**?**]. For a comprehensive guide to the wider theory of algebra representations we refer to the very comprehensive *Representation theory of finite groups and associative algebras* [**?**] from Charles W. Curtis and Irving Reiner, or, for a gentle introduction to the subject to *Introduction to representation theory* from Pavel Etingof *et al.* [**?**].

In the remainder of this paper, $\mathbf{k}$ is a perfect field of null characteristic and M is still a finite monoid. Furthermore, we suppose that $\mathbf{k}$ is "big enough", meaning that if not algebraically closed, at least a splitting field for the characteristic polynomials of the elements of M seen as linear maps on $\mathbf{k}M$.

As previously mentioned, we will refer to representations in terms of modules, let us recall the language that we use. A representation of M over $\mathbf{k}$ will be a $\mathbf{k}$-vector space V equipped with a linear action of the algebra $\mathbf{k}M$. As is usage, whenever the action is on the left we will say that V is a $\mathbf{k}M - \mathrm{mod}$ and a $\mathrm{mod} - \mathbf{k}M$ if the action is on the right. If M, M' are two finite, possibly different monoids, a $\mathbf{k}M - \mathrm{mod} - \mathbf{k}M'$ is simply simultaneously a $\mathbf{k}M - \mathrm{mod}$ and a $\mathrm{mod} - \mathbf{k}M'$. For a monoid M, we denote by M^{op} the *opposite* monoid, with multiplication defined by $m \cdot_{M^{op}} m' = m'm$. We will use liberally the fact that a $\mathbf{k}M - \mathrm{mod} - \mathbf{k}M'$ is naturally a $\mathbf{k}M \otimes \mathbf{k}M'^{op} - \mathrm{mod}$ and a $\mathbf{k}(M \times M'^{op}) - \mathrm{mod}$, and reciprocally. In the totality of this paper, we assume that the modules are finite dimensional as vector spaces over $\mathbf{k}$. Because of this, the Jordan-Hölder Theorem applies and the set of composition factors counted with multiplicities of a module is independent of the choice of a composition series. If S is a $\mathbf{k}M$-module and S is a simple $\mathbf{k}M$-module, we denote by $[V : S]$ the multiplicity of S as a composition factor of V.

In this section, we deal with monoid representation theory, with the goal in mind to compute the character table of M. Using the Clifford-Munn-Ponizovskii, this can largely be reduced to group representation theory. Stated differently, the representation theory of a monoid M is an extension of the representation theory of certain groups embedded in M. The groups in question are precisely the groups of Definition 9. In the first part of this section, we use this fact to find a description of a $\mathcal{L}$-class containing an idempotent e quotiented by its radical as a product of simple $\mathbf{k}G_e$-modules and simple $\mathbf{k}M$-modules. In the second part, we translate

this decomposition in terms of characters, which gives us the formula we seek. Finally, we recall and discuss the formula for the Cartan matrix from Thiéry [**?**].

1. On modules

Note that if we choose an element $a \in M$ and denote by L its $\mathcal{L}$-class and H its $\mathcal{H}$-class, we can equip $\mathbf{k}L$ with a $\mathbf{k}M-\mathrm{mod}-\mathbf{k}\Gamma'(H)$ structure. $\mathbf{k}L$ is already a $\mathrm{mod}-\mathbf{k}\Gamma'(H)$ by definition of $\Gamma'(H)$. We can also make it into a $\mathbf{k}M-\mathrm{mod}$ by setting, for every $m \in M$ and $l \in L$:

$$m \cdot l = \begin{cases} ml \text{ if } ml \in L \\ 0 \text{ otherwise} \end{cases}.$$

This is well-defined, as $ml \notin L$ implies that $l >_{\mathcal{L}} ml$ and so for every $m' \in M$, $l >_{\mathcal{L}} m'ml \notin L$: once fallen out of L, we cannot climb back in.

We have previously stated that the representation theory of monoids is an extension of the representation theory of some subgroups. This mainly expressed using the two following functors.

Definition 17. Let $e \in M$ be an idempotent, $\mathcal{L}(e)$ its $\mathcal{L}$-class, G_e the associated maximal subgroup. We define the two following maps :

$$\mathrm{Ind}_{G_e}^{M} : \begin{cases} \mathbf{k}G_e-\mathrm{mod} \longrightarrow \mathbf{k}M-\mathrm{mod} \\ V \longmapsto \mathbf{k}\mathcal{L}(e) \otimes_{\mathbf{k}G_e} V \end{cases}$$

$$N_e : \begin{cases} \mathbf{k}M-\mathrm{mod} \longrightarrow \mathbf{k}M-\mathrm{mod} \\ V \longmapsto \{v \in V \,|\, eMv = 0\} \end{cases}.$$

The idempotents and their maximal subgroups play a central role in the theory. One can show (see for instance [**?** , Proposition 1.14]) that if e, f are two idempotents in the same $\mathcal{J}$-class, there are some $x, x' \in M$ such that $xx' = e$ and $x'x = f$ and that $G_e \cong G_f$. A $\mathcal{J}$-class containing an idempotent is called a *regular* $\mathcal{J}$-class.

We are almost ready to state the Clifford-Munn-Ponizovskii theorem, which is the central piece connecting group and monoid representation theory. We will need the notion of *apex* of a $\mathbf{k}M$-module. A proof of the Clifford-Munn-Ponizovskii Theorem can be found in [**?** , Section 5.2].

Definition 18. Let V be a $\mathbf{k}M-\mathrm{mod}$, we denote its annihilator in M by $\mathrm{Ann}_M(V) = \{m \in M \,|\, mV = \{0\}\}$. This is clearly a two-sided ideal of M and as such is a

union of $\mathcal{J}$-classes. A regular $\mathcal{J}$-class J is said to be the *apex* of V if $\mathrm{Ann}_M(V) = I_J$ where $I_J = \{J \not\leq_{\mathcal{L}} \mathcal{J}(s) \mid s \in M\}$. If $e \in J$ is an idempotent, we also say that V has apex e.

Théorème 19 (Clifford-Munn-Ponizovskii)**.** *Let M be a finite monoid, $e \in M$ an idempotent and $\mathbf{k}$ be a field.*

1. *There is a bijection between isomorphism classes of simple M-modules with apex e and isomorphism classes of simple G_e-modules given by :*

$$V \longmapsto V^{\#} = \mathrm{Ind}_{G_e}^M(V)/\operatorname{rad}(\mathrm{Ind}_{G_e}^M(V)).$$

   The reciprocal bijection is given by $S \longmapsto eS$.
2. $\operatorname{rad}(\mathrm{Ind}_{G_e}^M(V)) = N_e(\operatorname{rad}(\mathrm{Ind}_{G_e}^M(V)))$
3. *Every simple M-module has an apex.*
4. *Every composition factor of $\mathrm{Ind}_{G_e}^M(V)$ with the exception of $V^{\#}$ has an apex strictly $\mathcal{J}$-greater than e. Moreover, $V^{\#}$ has apex e and is a factor of multiplicity one.*

This allows us the following description of the $\mathcal{L}$-class of an idempotent e.

Proposition 20. *Let $e \in M$ be an idempotent and G_e be the maximal subgroup at e. Let Irr_e be a set of representatives of the isomorphism classes of simple $\mathbf{k}G_e$-modules. Then:*

$$\mathbf{k}\mathcal{L}(e) = \bigoplus_{V \in \mathrm{Irr}_e} \mathrm{Ind}_{G_e}^M(V) \otimes_{\mathbf{k}} V^*$$

where V^* is the dual of V.

Proof. By definition, $\mathrm{Ind}_{G_e}^M(V) = \mathcal{L}(e) \otimes_{\mathbf{k}G_e} V$. Now, since direct sum and tensor product over a ring with identity commute :

$$\bigoplus_{V \in \mathrm{Irr}_e} \mathrm{Ind}_{G_e}^M(V) \otimes V^* = \bigoplus_{V \in \mathrm{Irr}_e} \mathbf{k}\mathcal{L}(e) \otimes_{G_e} V \otimes V^* = \mathbf{k}\mathcal{L}(e) \otimes_{G_e} \left(\bigoplus_{V \in \mathrm{Irr}_e} V \otimes V^* \right)$$

Because $\mathbf{k}$ is of null characteristic, $\mathbf{k}G_e$ is semisimple. By the Wedderburn-Artin theorem, $\mathbf{k}G_e = \bigoplus_{V \in \mathrm{Irr}_e} V \otimes V^*$ so :

$$\bigoplus_{V \in \mathrm{Irr}_e} \mathrm{Ind}_{G_e}^M(V) \otimes V^* = \mathbf{k}\mathcal{L}(e) \otimes_{\mathbf{k}G_e} \mathbf{k}G_e = \mathcal{L}(e)$$

since $\mathbf{k}G_e$ is a ring with identity. □

Note that this puts in relation three kinds of modules : the simple $\mathbf{k}G_e$-modules, which are well understood, $\mathbf{k}\mathcal{L}(e)$ which is understood as well because it is a combinatorial module[3], and finally the modules $\mathrm{Ind}_{G_e}^{M}(V)$ which contain, in a sense, the simple $\mathbf{k}M$-modules that we are after. According to the Clifford-Munn-Ponizovskii Theorem, we still need to remove the radical of each $\mathrm{Ind}_{G_e}^{M}(V)$ factor. Proposition 23 puts the radical in a form similar to Theorem 19 while Proposition 24 does exactly this. Lemma 21 and its Corollary are technical results on radicals used in the proof of Proposition 23.

Lemma 21. *Let A, B be two finite dimensional algebras over a perfect field $\mathbf{k}$. Then:*

$$\mathrm{rad}(A \otimes B) = \mathrm{rad}(A) \otimes B + A \otimes \mathrm{rad}(B).$$

While we haven't be able to find a source for that claim it seems to be folklore in the algebra representation community. For the sake of completeness, we reproduce a proof communicated to us by Pr. Pierre-Guy Plamondon[4].

Proof. Because $\mathbf{k}$ is perfect, Wedderburn's Principal Theorem applies, and we get the decompositions $A = A' \oplus \mathrm{rad}(A)$, $B = B' \oplus \mathrm{rad}(B)$, with A' and B' semisimple algebras. To prove the result, we show that $\mathrm{rad}(A) \otimes B + A \otimes \mathrm{rad}(B)$ is a nil radical and that

$$A \otimes B \big/ \mathrm{rad}(A) \otimes B + A \otimes \mathrm{rad}(B)$$

is semisimple. Let us first show that the quotient is semisimple. We have:

$$\begin{aligned} A \otimes B &= (A' \oplus \mathrm{rad}(A)) \otimes (B' \oplus \mathrm{rad}(B)) \\ &= A' \otimes B' \oplus A' \otimes \mathrm{rad}(B) \oplus \mathrm{rad}(A) \otimes B' \oplus \mathrm{rad}(A) \otimes \mathrm{rad}(B). \end{aligned}$$

On the other hand, the same decompositions give us

$$A \otimes \mathrm{rad}(B) \oplus \mathrm{rad}(A) \otimes B = A' \otimes \mathrm{rad}(B) \oplus \mathrm{rad}(A) \otimes B' \oplus \mathrm{rad}(A) \otimes \mathrm{rad}(B).$$

Finally,

$$A \otimes B \big/ A \otimes \mathrm{rad}(B) \oplus \mathrm{rad}(A) \otimes B = A' \otimes B'$$

which, since A', B' are semisimple, is also semisimple. $A \otimes \mathrm{rad}(B) \oplus \mathrm{rad}(A) \otimes B$ is also nil, because $\mathrm{rad}(B)$ and $\mathrm{rad}(A)$ are, so, indeed, $\mathrm{rad}(A \otimes B) = A \otimes \mathrm{rad}(B) \oplus \mathrm{rad}(A) \otimes B$. □

[3] That is, the multiplication of an element of the basis of the module by an element of M is either another element of the basis or 0.

[4] Pr. Pierre-Guy Plamondon, Laboratoire de Mathématiques de Versailles, Université Paris-Saclay, website.

From Lemma 21, we get the following Corollary by recalling that if V is an A-module, $\mathrm{rad}_A(V) = \mathrm{rad}(A) \cdot V$.

Corollary 22. *Let A, B be two finite dimensional unitary algebras over a perfect field. If $V_A \otimes V_B$ is an $A - \mathrm{mod} - B$ (or equivalently an $A \otimes B^{op} - \mathrm{mod}$), then*

$$\mathrm{rad}_{A\otimes B^{op}}(V_A \otimes V_B) = \mathrm{rad}_A(V_A) \otimes B + A \otimes \mathrm{rad}_B(V_B).$$

This allows us to identify the radical of $\mathbf{k}\mathcal{L}(e)$.

Proposition 23. *Let M be a finite monoid, $e \in S$ an idempotent G_e be the maximal subgroup at e and $\mathbf{k}$ be a perfect field. Then:*

$$\mathrm{rad}_{\mathbf{k}M\otimes\mathbf{k}G_e^{op}}(\mathbf{k}\mathcal{L}(e)) = N_e(\mathbf{k}\mathcal{L}(e)).$$

Proof. Using Lemma 21, for V a simple G_e-module, we have that :

$$\begin{aligned}
&\mathrm{rad}_{\mathbf{k}M\otimes\mathbf{k}G_e^{op}}(\mathrm{Ind}_{G_e}^{S}(V) \otimes_{\mathbf{k}} V^*) \\
&\qquad = \mathrm{rad}_{\mathbf{k}M} \mathrm{Ind}_{G_e}^{M}(V) \otimes V^* + \mathrm{Ind}_{G_e}^{M}(V) \otimes \mathrm{rad}_{\mathbf{k}G_e^{op}}(V^*) \\
&\qquad \overset{(1)}{=} \mathrm{rad}_{\mathbf{k}M} \mathrm{Ind}_{G_e}^{M}(V) \otimes V^* \\
&\qquad \overset{(2)}{=} N_e(\mathrm{Ind}_{G_e}^{M}(V)) \otimes V^*
\end{aligned}$$

where equality (1) comes from the simplicity of V^* as a $\mathbf{k}G_e^{op}$-module and (2) is the second point of Theorem 19.

Since radical and direct sums commute, denoting by Irr_e a set of representatives of the isomorphism classes of simple $\mathbf{k}G_e$-modules, we know that:

$$\mathrm{rad}_{\mathbf{k}M\otimes\mathbf{k}G_e^{op}}(\mathcal{L}(e)) = \bigoplus_{V\in\mathrm{Irr}_e} N_e(\mathrm{Ind}_{G_e}^{M}(V)) \otimes V^*.$$

It remains to be seen why

$$\bigoplus_{V\in\mathrm{Irr}_e} N_e(\mathrm{Ind}_{G_e}^{M}(V)) \otimes V^* = N_e(\mathbf{k}\mathcal{L}(e)).$$

It is clear the direct sum on the left is a subset of the set on the right. For the other inclusion, we see that if V, V' are $\mathbf{k}M$-modules, $N_e(V \oplus V') = N_e(V) \oplus N_e(V')$. Given the proposition 20, it is enough to show that $N_e(\mathrm{Ind}_{G_e}^{M}(V)) \otimes V^* = N_e(\mathrm{Ind}_{G_e}^{M}(V) \otimes V^*)$. Let $x \in \mathrm{Ind}_{G_e}^{M}(V) \otimes V^*$ be such that for every

$m \in M, emx = 0$. x can be written as $\sum_i (\sum_j x_{i,j} b_j) \otimes b'_i$ where $\{b_j\}_j$ is a basis of $\mathrm{Ind}_{G_e}^M(V)$ and $\{b'_i\}_i$ is a basis of V^*. For every $m \in M$, we have :

$$em \cdot x = em \cdot \sum_i (\sum_j x_{i,j} b_j) \otimes b'_i = \sum_i (em \cdot \sum_j x_{i,j} b_j) \otimes b'_i = 0$$

that is, for every b'_i we get $em \cdot \sum_j x_{i,j} b_j = 0$ so $\sum_j x_{i,j} b_j \in N_e(\mathrm{Ind}_{G_e}^M(V))$ which means $x \in N_e(\mathrm{Ind}_{G_e}^M(V)) \otimes V^*$. □

Proposition 24. *Let $e \in M$ be an idempotent and G_e be the maximal subgroup at e. Let Irr_e be a set of representatives of the isomorphism classes of simple $\mathbf{k}G_e$-modules. Then:*

$$\mathbf{k}\mathcal{L}(e)/\operatorname{rad}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(\mathbf{k}\mathcal{L}(e)) \cong \bigoplus_{V \in \mathrm{Irr}_e} V^{\#} \otimes V^*.$$

Proof. From Proposition 23, we have a decomposition of $\operatorname{rad}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(\mathcal{L}(e))$ as a direct sum adapted to the decomposition of $\mathbf{k}\mathcal{L}(e)$ as $\bigoplus_{V \in \mathrm{Irr}_e} \mathrm{Ind}_{G_e}^S(V) \otimes_{\mathbf{k}} V^*$. So:

$$\mathbf{k}\mathcal{L}(e)/\operatorname{rad}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(\mathbf{k}\mathcal{L}(e)) \cong \bigoplus_{V \in \mathrm{Irr}_e} (\mathrm{Ind}_{G_e}^S(V) \otimes_{\mathbf{k}} V^*)/(N_e(\mathrm{Ind}_{G_e}^S(V)) \otimes V^*).$$

From Theorem 19, we know that

$$0 \longrightarrow N_e(\mathrm{Ind}_{G_e}^S(V)) \longrightarrow \mathrm{Ind}_{G_e}^S(V) \longrightarrow V^{\#} \longrightarrow 0$$

is a short exact sequence. Since $N_e(\mathrm{Ind}_{G_e}^S(V)) \otimes V^*$ a submodule of $\mathrm{Ind}_{G_e}^S(V) \otimes V^*$ and because tensor product over a field is exact we have a short exact sequence:

$$0 \longrightarrow N_e(\mathrm{Ind}_{G_e}^S(V)) \otimes V^* \longrightarrow \mathrm{Ind}_{G_e}^S(V) \otimes V^* \longrightarrow V^{\#} \otimes V^* \longrightarrow 0$$

which proves the result. □

2. On characters

One of the major features of the finite group representation theory is the fact that all the information on a representation can be summarized in its *character*. This (partially) carries over to monoid representation theory, as we shall see in this section where we reformulate the results of the previous section in terms of characters.

Definition 25. If V is a finite dimension $\mathbf{k}M$-module, its *character* is the map from M to $\mathbf{k}$ defined by $\chi^V_{\mathbf{k}M} : m \longmapsto \mathrm{Tr}(v \mapsto m \cdot v)$.

We recall the following well-known facts about characters. Proofs for fact 2 and 3 are respectively (ii) and (iii) of [**?** , Proposition 7.12][5].

Proposition 26. 1. *Let V be a $\mathbf{k}M$-module. We have $\chi^V_{\mathbf{k}M} = \chi^{V^*}_{\mathbf{k}M^{op}}$.*

2. *Consider the short exact sequence of $\mathbf{k}M$-modules:*

$$0 \longrightarrow A \longrightarrow B \longrightarrow B/A \longrightarrow 0.$$

Then $\chi^{B/A}_{\mathbf{k}M} = \chi^B_{\mathbf{k}M} - \chi^A_{\mathbf{k}M}$.

3. *Consider M, M' two finite monoids, V a $\mathbf{k}M$-module and W a $\mathbf{k}M'^{op}$-module. Then $\chi^{V\otimes W}_{\mathbf{k}M\otimes\mathbf{k}M'} = \chi^V_{\mathbf{k}M}\chi^W_{\mathbf{k}M'}$.*

The previous properties are simply extensions of similar properties on groups, and their proof is similar. From groups, we also keep in the case of monoids the linear independence of irreducible characters (see [**?** , Theorem 7.7] for reference):

Proposition 27. *The irreducible characters $\{\chi^S_{\mathbf{k}M} \mid S$ is a simple $\mathbf{k}M - \mathrm{mod}\}$ are linearly independent as $\mathbf{k}$ valued functions.*

This, together with the second point in the Proposition 26, has a nice consequence. As we are interested in finite dimensional module over finite monoids, those modules have a composition series. Say that a $\mathbf{k}M$-module V, has S as a composition factor with multiplicity $[V : S]$ for any simple $\mathbf{k}M$-module S. Then:

$$\chi^V_{\mathbf{k}M} = \sum_S [V : S]\chi^S_{\mathbf{k}M}.$$

In that way, since characters of the simple modules are linearly independent, the character of a module can be seen as a record of its composition factors.

The question of where to compute characters is worth asking: in the case of groups, one needs only to compute the character for a transversal of conjugacy classes to get its value everywhere. The case of monoids was described for the first time by McAlister in [**?**].

Definition 28. We say that two elements m, m' in M are in the same *generalized conjugacy class* or *character equivalency class* if for every $\mathbf{k}M$-module V, $\chi^V_M(m) = \chi^V_M(m')$. We note C_M the set of generalized conjugacy classes.

[5] Note that for Fact 3, our reference deals only with the case $M = M'$, but the proof is the same.

Proposition 29. *([? , Proposition 2.5]) Let $\mathcal{E} = \{e_1, \dots, e_n\}$ be idempotent representatives of the regular $\mathcal{J}$-classes of M and for each e_i let $\mathcal{C}_i = \{c_{i,1}, \dots, c_{i,m_i}\}$ be representatives of the conjugacy classes of G_{e_i}. Then the set $\mathcal{C}_M = \bigcup_{e_i \in \mathcal{E}} \mathcal{C}_i$ is a set of representatives of character equivalency classes of M.*

We can now recall the definition of the character table of a monoid.

Definition 30. Let Irr_M be the set of isomorphism classes of simple $\mathbf{k}M$-modules and C_M as in definition 28. The *character table* of M over $\mathbf{k}$ is the (square) matrix defined by :

$$X(M) = (\chi^V_{\mathbf{k}M}(m))_{V \in \mathrm{Irr}_M, m \in C_M}.$$

Moreover, if $e \in M$ is an idempotent, we define $X_e(M)$ as the matrix obtained by extracting from $X(M)$ only the lines corresponding to simple modules with apex e.

Finally, we can apply the language of characters to Proposition 24, which yield a formula for computing the character table of M over $\mathbf{k}$ given the character tables of the groups G_e over k.

Proposition 31. *Let $e \in M$ be an idempotent, G_e be the maximal subgroup at e. We have the formula for $X_e(M)$:*

$$X_e(M) = {}^tX(G_e)^{-1} \cdot \left(\chi^{\mathbf{k}\mathcal{L}(e)}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(m,g) - \chi^{N_e(\mathbf{k}\mathcal{L}(e))}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(m,g) \right)_{g \in C_{G_e}, m \in C_M}$$

where the dot is the matrix product.

Proof. First, we have, because of Proposition 26-2, we have:

$$\chi^{\mathbf{k}\mathcal{L}(e)/N_e(\mathbf{k}\mathcal{L}(e))}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}} = \chi^{\mathbf{k}\mathcal{L}(e)}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(m,g) - \chi^{N_e(\mathbf{k}\mathcal{L}(e))}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(m,g)$$

Then, from Proposition 24, we know that:

$$\begin{aligned}
\chi^{\mathbf{k}\mathcal{L}(e)/N_e(\mathbf{k}\mathcal{L}(e))}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}}(m,g) &= \chi^{\bigoplus_{V \in \mathrm{Irr}_e} V^\# \otimes V^*}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}} \\
&= \sum_{V \in \mathrm{Irr}_e} \chi^{V^\# \otimes V^*}_{\mathbf{k}M \otimes \mathbf{k}G_e^{op}} \\
&= \sum_{V \in \mathrm{Irr}_e} \chi^{V^\#}_{\mathbf{k}M}(m) \chi^V_{\mathbf{k}G_e}(g)
\end{aligned}$$

This last sum is clearly the dot product between the column of $X(G_e)$ indexed by g and the column of $X_e(M)$ indexed by m. That is, the coefficient in position (g,m) of $\chi^{\mathcal{L}(e)/\mathrm{rad}(\mathcal{L}(e))}_{M-G_e}$ is equal to the coefficient in position (g,m) of ${}^tX(G_e) \cdot X_e(M)$, which, together with Proposition 26(ii), proves the equality. □

3. One step beyond: The Cartan matrix

We are, at last, in measure to state the formula from Thiéry for the Cartan Matrix. Without getting into the specific details, the Cartan matrix can be seen as measure of how "not semisimple" the algebra of the monoid is. We use a non standard definition of the Cartan matrix, first given in [**?** , Definition 2.6]. A formal proof that this is equivalent to the usual definition can be found in [**?** , Corolary 7.28].

Definition 32. Let $\{S_1, \ldots, S_n\}$ be a set of representatives of the isomorphism classes of simple $\mathbf{k}M$-modules. The simple $\mathbf{k}M \otimes \mathbf{k}M^{op}$ modules are the $S_i \otimes S_j^*$ for all $i, j \in [\![1, n]\!]$. Denote by $[\mathbf{k}M : S_i \otimes S_j^*]$ the multiplicity of $S_i \otimes S_j^*$ as a composition factor of $\mathbf{k}M$.

The Cartan matrix of $\mathbf{k}M$ is defined by:

$$C(\mathbf{k}M) = ([\mathbf{k}M : S_i \otimes S_j^*])_{i,j}.$$

In other words, the Cartan matrix is a recording of the multiplicities of the composition factors of $\mathbf{k}M$ as a $\mathbf{k}M \otimes \mathbf{k}M^{op}$ module. But so is its character! The difference being that the character of $\mathbf{k}M$ as it is computed is expressed in the basis of the character equivalency classes of $M \times M^{op}$ while the Cartan matrix is expressed directly in the basis of the simple modules. Since the basis change between the two is precisely given by the character table and hence, we have the Thiéry's Formula for the Cartan matrix.

Proposition 33. *The Cartan matrix is given by the formula:*

$$C(\mathbf{k}M) = {}^tX_M^{-1} B X_M^{-1}$$

where $B = (|\{s \in M \mid msm'\}|)_{m,m' \in C_M}$

III. Some explicit computations

In this section, M is a fixed submonoid of T_n. We explain how to combinatorially compute the characters of three interesting modules: $\mathbf{k}J$ for a $\mathcal{J}$-class J, $\mathbf{k}M$ as $\mathbf{k}M - \mathrm{mod} - \mathbf{k}M$ and $\mathbf{k}\mathcal{L}(e)$ for a $\mathcal{L}$-class containing an idempotent e as a $\mathbf{k}M - \mathrm{mod} - \mathbf{k}G_e$. For clarity, we will designate these characters as *bicharacters*.

We begin by discussing the computational hypotheses we make before presenting the algorithms themselves. In a third subsection, we detail how to compute the radical of $\mathbf{k}\mathcal{L}(e)$ to apply our formula from Proposition 31 to compute the character table of a monoid. This latter case is not as tidy as the former ones, as we don't have a purely combinatorial way of computing the radical.

1. Computational hypotheses

In this section we discuss the computational hypotheses necessary for the algorithms in the next section. This section is based on the work [**?**] in which East, Egri-Nagy, Mitchell and Péresse provide efficient algorithms for all basic computational questions on finite semigroups (which include monoids). Although we limit our scope to the case of transformation monoids, methods described [**?**] allow the algorithms described below to be applied to other interesting classes of monoids. Moreover, they can theoretically be applied to any finite monoid using a Cayley embedding in a full transformation monoid. In general however, this is very inefficient and not feasible in practice.

Following the authors of [**?**], we make the following fundamental assumptions that we can compute:

- Assumption I : a product of two elements of the monoid.
- Assumption II : the image and kernel of a transformation (note that we do not explicitly use this assumption, but that it is necessary for the algorithms of [**?**] that we do use).
- Assumption III : Green's pairs.
- Assumption IV : Given $h \in {}_M\,\mathrm{Stab}(H)$ compute the corresponding element in $\Gamma(H)$ (understood as a permutation group of the image common to all elements of H as seen in Example 7), and similarly on the right.

Not only do we directly need to be able to do these computations for our own algorithm, but they are also prerequisite for the algorithms from [**?**]. As such, we refer to the top of Section 5.2 of [**?**] on how to realize these computations in the case of transformation monoids.

We, again, refer to [**?**] for the specific algorithms meeting our computational prerequisites.

- Computing the Schützenberger groups: [**?** , Algorithm 4]
- Checking membership of an element in a Green's class: [**?** , Algorithms 7 & 8].
- Finding idempotents: [**?** , Algorithm 10]. This algorithm also allows for finding the regular $\mathcal{J}$-classes.

- Decomposing the monoid in $\mathcal{R}, \mathcal{L}$ and $\mathcal{J}$-classes : [**?** , Algorithm 11] and its discussion. Note that by storing this decomposition, we can, given an element of the monoid, find the classes that contain it.

- Obtaining a representative of a Green's class: this is given by the data structure representing the Green's classes described at the top of [**?** , Section 5.4].

Finally, we require the following points that, although they are not described in [**?**], are easily obtained from it.

- Computing a set C_M of character equivalency representatives: given Proposition 29, this can be done in four steps:
  1. compute a set $\mathcal{E}$ of idempotent representatives of the regular $\mathcal{J}$-classes,
  2. compute $\Gamma(\mathcal{H}(e))$ for each $e \in \mathcal{E}$,
  3. compute a set C_e of representatives of the conjugacy classes of $\Gamma(\mathcal{H}(e))$ for each $e \in \mathcal{E}$, using for instance the procedure described in [**?**],
  4. for each $e \in \mathcal{E}$ and $c \in C_e$ compute the corresponding element of $\mathcal{H}(e)$ as in Example 10.

- Computing τ_a as in Proposition 13 : given $g \in \Gamma(H)$, $\tau_a(g)$ is simply, seen as an element of $\mathfrak{S}(\ker a)$:

$$a^{-1}\{i\} \mapsto (g \cdot a)^{-1}\{g \cdot a(i)\},$$

  which can be computed in $O(n)$. Note that this is a special case of the application described in [**?** , Proposition 3.11 (a)]

- Testing that two elements g, g' in $\Gamma'(\mathcal{H}(a))$ are conjugated : $\Gamma'(H)$ is represented as a subgroup of $\mathfrak{S}(\ker a)$ and known procedures, such as the one described in [**?**], can be used.

- Computing the cardinality of a conjugacy class of a Schützenberger group: for instance, the computer algebra system GAP uses the method described in [**?**].

2. Combinatorial bicharacter computing: 3 applications.

We are now ready to present the algorithm for fixed-point counting, keeping in mind that we want first to use the formula from Section II.2 to compute the character table of the monoid and further to compute the Cartan matrix of the monoid. In the cases we are interested in, we use the formalism of character computing, since, as stated in the Lemma below, computing the characters of so called *combinatorial modules* is actually counting fixed points.

Lemma 34. *Let M, M' be two finite monoids and (V, B) a finite dimensional $\mathbf{k}M-\mathrm{mod}-\mathbf{k}M'$ space equipped with a basis B. If the actions of M, M' on (V, B) are* combinatorial*, meaning for any $(m, b, m') \in M \times B \times M'$, $mbm' \in (B \cup \{0\})$, then:*

$$\chi^V_{\mathbf{k}M\otimes\mathbf{k}M'^{op}} = |\{b \in B \,|\, mbm' = b\}|.$$

Proof. In the basis B, the matrix of the linear map $x \mapsto mxm'$ is a $\{0, 1\}$-matrix, with for every $b \in B$ exactly one 1 in the b-th column, that 1 being on the b-th row if $mbm' = b$. Thus, the trace counts the number of fixed points. □

Note that we have already defined a structure of combinatorial $\mathbf{k}M-\mathrm{mod}-\mathbf{k}G_e$ on $(\mathbf{k}\mathcal{L}(e), \mathcal{L}(e))$ for any idempotent e. In the same way, $\mathbf{k}J$ for a $\mathcal{J}$-class J, can be equipped with a structure of $\mathbf{k}M-\mathrm{mod}-\mathbf{k}M$ by setting for every $(m, j) \in M \times J$:

$$m \cdot j = \begin{cases} mj \text{ if } mj \in J \\ 0 \text{ otherwise} \end{cases} \text{ and } j \cdot m = \begin{cases} jm \text{ if } jm \in J \\ 0 \text{ otherwise} \end{cases}.$$

As before, this is well defined: firstly because the actions on the left and on the right commute (because the monoid's law is associative by assumption) and secondly because either $m \leq_{\mathcal{L}} m'$ or $m \leq_{\mathcal{R}} m'$ imply $m \leq_{\mathcal{J}} m'$ so, as in Section II.1, if ml or lm has "fallen to 0", it can't "climb back up" to J.

This structure makes $(\mathbf{k}J, J)$ into a combinatorial module and we may apply our fixed points counting methods to compute its character.

Algorithm 35 (Computing the bicharacter of a $\mathcal{J}$-class)**.** Keeping the assumptions and notations of the previous Paragraph 1, we get from Corollary 16 an algorithm to compute the bicharacter of $\mathbf{k}J$ as a $\mathbf{k}M-\mathrm{mod}-\mathbf{k}M$:

- Input : A $\mathcal{J}$-class J, a set of representatives of the character equivalency classes C_M.
- Output : A matrix $(|\{m \in J \,|\, hmk = m\}|)_{(h,k)\in C_M^2}$

1. Preparations:
   (a) Choose $a \in J$ and define $H = \mathcal{H}(a)$.
   (b) Compute Green's pairs (λ_R, λ'_R) (respectively (ρ_L, ρ'_L)) for $(\mathcal{R}(a), R)$ (resp. $(\mathcal{L}(a), L)$) for all $\mathcal{R}$-class $R \subset J$ (resp. $\mathcal{L}$-class $L \subset J$).
   (c) Compute the set C of conjugacy classes of $\Gamma'(H)$.
2. For each character equivalency representative $h \in C_M$, initialize $r_J(h)$ and $l_J(h)$ to both be $(0)_{\bar{g}\in C}$.

(a) For each $\mathcal{R}$-class $R \subset J$, test if $h\lambda_R a \in R$. If so, denoting by $\bar{g}$ the conjugacy class of $\tau_a((\lambda'_R h \lambda_R)\times_H)$ in $\Gamma'(H)$, increment $r_J(h)$ by $|C_{\Gamma'(H)}(g)|$ at position $\bar{g}$.
(b) For each $\mathcal{L}$-class $L \subset J$, test if $a\rho_R h \in L$. If so, denoting by $\bar{g}$ the conjugacy class of $\times_H(\rho'_L h \rho_L)$ in $\Gamma'(H)$, increment $r_J(h)$ by 1 at position $\bar{g}$.

3. Compute the matrix $\chi = (r_J(h) \cdot l_J(k))_{(h,k)\in C_M^2}$ using the previously computed vectors and return χ.

Example 36. An *aperiodic* monoid is a monoid where all $\mathcal{H}$-classes are singletons. Let us apply the algorithm we just described in the case of a $\mathcal{J}$-class J with trivial $\mathcal{H}$-classes. Several simplifications occur : first, we don't need to check for the conjugacy class, as there is only one. Secondly, the conjugacy class has cardinality one. Consider the vectors $r_J = (|S_{\mathcal{R}}(h)|)_{h\in C_M}$ and $r_J = (|S_{\mathcal{L}}(h)|)_{h\in C_M}$ with $S_{\mathcal{L}}(h)$ and $S_{\mathcal{R}}(h)$ defined as in Corollary 16. The bicharacter is simply the matrix product of r_j^T with l_J. The particular case of this algorithm for aperiodic monoid is described in [**?** , Section .1]

Algorithm 37 (Computing the bicharacter of $\mathbf{k}M$)**.** If we consider $(\mathbf{k}M, M)$ as a combinatorial $\mathbf{k}M - \mathrm{mod} - \mathbf{k}M$, we immediately have that:

$$\chi^{\mathbf{k}M}_{\mathbf{k}M\otimes\mathbf{k}M'^{op}} = \sum_{J\in\mathcal{J}} \chi^{\mathbf{k}J}_{\mathbf{k}M\otimes\mathbf{k}M'^{op}}$$

and we can therefore compute the bicharacter of the whole monoid M: we first compute a set C_M of representatives of the character equivalency classes and we the iterate Algorithm 35 over all $\mathcal{J}$-classes and sum the results.

The final useful example is the case of counting fixed points in a single regular $\mathcal{L}$-class, for the purpose of computing the character table of the monoid.

Algorithm 38 (Computing the bicharacter of a $\mathcal{L}$-class)**.** Let e be an idempotent en let $L = \mathcal{L}(e)$. In this example $\mathbf{k}L$ is still a combinatorial module, but it has the particularity, compared with the other two examples, that the monoids on the left and right are not the same. However, as the maximal subgroup at e, G_e, is a subsemigroup of M, the same results apply at no extra costs.

We can simply adapt the algorithm of Algorithm 35. Since an element of C_{G_e} acts "as itself" on the right, we don't need to keep track of the action of the right with a vector r_L as we did previously.

1. Initialize χ to $(0)_{(h,k)\in C_M\times C}$

2. For each $h \in C_M$, for each $\mathcal{H}$-class H, test if $h\lambda_H a \in H$. If so, denoting by k the conjugacy class of $\lambda'_H h \lambda_H$ in G_e, increment χ by $|C_{G_e}(h)|$ at position (h, k).
3. Return χ

3. *Computing* $N_e(\mathbf{k}L)$

We are now almost in position to use the formula of Proposition 31: the character tables of the groups are supposed to be given, as we dispose of efficient group algorithms in the literature to compute them, from Algorithm 38 we now know how the efficiently compute the bicharacter of $\mathbf{k}\mathcal{L}(e)$ as a $\mathbf{k}M \otimes \mathbf{k}G_e^{op}$-module for some idempotent $e \in M$. It remains to compute the bicharacter of $N_e(\mathbf{k}\mathcal{L}(e))$ as a $\mathbf{k}M \otimes \mathbf{k}G_e^{op}$-module, which we discuss now.

Let $L = \mathcal{L}(e)$. Recall that, by definition, $N_e(\mathbf{k}L) = \{x \in \mathbf{k}L \,|\, eMx = 0\}$. Taking L as a basis for $\mathbf{k}L$, we can form a matrix with rows indexed by $M \times L$ and columns indexed by L, with the coefficient at $((m, l), l') = 1$ if $eml = l'$ and 0 otherwise. Computing the kernel of this matrix yields a basis for $N_e(\mathbf{k}L)$ but is extremely inefficient as the number of rows is many times the cardinality of the monoïd.

Notice first that for any $m \in M$, $em \leq_{\mathcal{R}} e$ so we can consider only the elements of M that are $\mathcal{R}$-smaller than e. Conversely, recall that the structure of $\mathbf{k}M$-module on M is defined by $m \cdot l = ml$ if $ml \in L$ and 0 otherwise and that this latter case happens if $ml \leq_{\mathcal{L}} l$. Since if $m \leq_{\mathcal{L}} l$ implies $ml \leq_{\mathcal{L}} l$ we have that the (m, l)-th row of the matrix is null and that we may omit it. This shows that we need only to consider the element of M that are not $\mathcal{L}$-below e. Together with the previous point, this means that the similarly defined matrix but whose rows are only indexed by $\mathcal{R}(e) \times L$ has the same kernel. This is good news, as we may now exploit the structure of the $\mathcal{J}$-class given by Green's Lemma to further reduce the dimension of this matrix. Indeed, another consequence of Green's Lemma is the so called "Location Theorem" from Clifford and Miller. A proof can be found in [**?** , Theorem 1.11].

Lemma 39 (Location Theorem)**.** *Let r, l be two elements in the same $\mathcal{J}$-class. We have:*

$$rl = \begin{cases} \gamma \in \mathcal{R}(r) \cap \mathcal{L}(l) \text{ if } \mathcal{L}(r) \cap \mathcal{R}(l) \text{ contains an idempotent,} \\ \gamma <_{\mathcal{J}} l, r \text{ otherwise.} \end{cases}$$

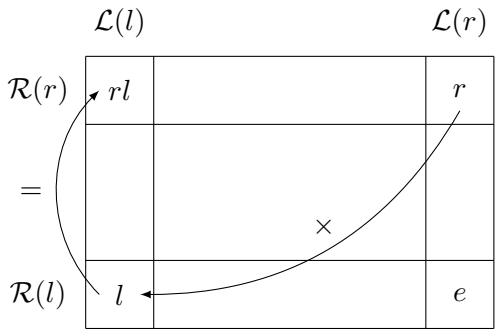


Figure 1: Location Theorem
Since there is an idempotent e in $\mathcal{L}(r) \cap \mathcal{R}(l)$, rl stays in the same $\mathcal{J}$-class, in $\mathcal{L}(l) \cap \mathcal{R}(r)$.

Lemma 40. *Let $e \in M$ be an idempotent, $R = \mathcal{R}(e)$ its $\mathcal{R}$-class and R' another $\mathcal{R}$-class of $\mathcal{J}(e)$. Let (λ, λ') be a left Green's pair for (L, L'). Then $(\lambda e, e\lambda')$ is a left Green's pair for (R, R').*

Similarly, if $L = \mathcal{L}(e)$, L' is a $\mathcal{L}$-class of $\mathcal{J}(e)$ and (ρ, ρ') is a right Green's pair for (L, L'), then $(e\rho, \rho' e)$ is a right Green's pair for (L, L')

Proof. Let g be any element of $\mathcal{H}(e)$ and $g' = \lambda g$. Since e is idempotent, $\mathcal{H}(e)$ is a group with identity e so we have $e\lambda'\lambda eg = e\lambda'\lambda g = eg = g$ and $\lambda ee\lambda' g' = \lambda eg = \lambda g = g'$ which, from Green's Lemma, make $(\lambda e, e\lambda')$ a left Green's pair for (L, L'). A similar argument applies for the second part of the proposition. □

Remark. This lemma means that for a regular $\mathcal{J}$-class J and for any two $\mathcal{L}$-class (or $\mathcal{R}$-class) it contains, we may choose a corresponding Green's pair among the elements of those two classes.

Proposition 41. *Let $e \in M$ be an idempotent and $H = \mathcal{H}(e), L = \mathcal{L}(e), R = \mathcal{R}(e)$ and $J = \mathcal{J}(e)$. For each $\mathcal{R}$-class $R' \subset J$, we choose a left Green's pair $(l, l') \in J^2$. We denote by $\mathfrak{L}$ the set of all l for the chosen left Green's pairs. We define $\mathfrak{R}$ similarly. Then N_e is the set of solutions of :*

$$\forall r \in \mathfrak{R}, \forall g \in H, \quad \sum_{l \in \mathfrak{L}} \mathbb{1}_H(rl) x_{l(rl)^{-1}g} = 0$$

Proof. Consider an element $a \in R$. a can be written in a unique way as gr, with $g \in H$ and $r \in \mathfrak{R}$ corresponding to $\mathcal{L}(a)$. Similarly, an element b in L as a unique

decomposition as $l\gamma, l \in \mathfrak{L}, \gamma \in H$. For an element $x \in \mathbf{k}L$ we note:

$$x = \sum_{l \in \mathfrak{L}, \gamma \in H} x_{l\gamma} l\gamma$$

its decomposition over the basis L.

We want to find the equations that describe $\ker(gr\times_L)$ (where $gr\times_L$ is the linear map on $\mathbf{k}L$ obtained by extending the monoid's multiplication by linearity). From the Location Theorem, we get that $\mathrm{Im}\,(gr\times_L) \subset \mathbf{k}H$. For $k \in H$, denote by $f_{k,gr}$ the k-th coordinate function of $gr\times_L$. Because $gr\times_L$ acts combinatorially on $\mathbf{k}L$, we have :

$$f_{k,gr}(x) = \sum_{l \in \mathfrak{L}, \gamma \in H} \mathbb{1}_{\{k\}}(grl\gamma) x_{l\gamma}$$

Note that $x_{l\gamma}$ appears in the sum if and only if $grl\gamma = k$. From the Location Theorem, and because we chose $l \in L, r \in R$, we have $grl\gamma = k$ if and only if $rl \in H$ and $\gamma = (rl)^{-1} g^{-1} k$ and thus the equation becomes:

$$f_{k,gr}(x) = \sum_{l \in \mathfrak{L}} \mathbb{1}_H(rl) x_{l(rl)^{-1} g^{-1} k}.$$

For x to be in $\ker(gr\times_L)$, x must cancel simultaneously $f_{k,gr}$ for all $k \in H$. We now have a set of equations for $ker(gr\times_L)$, and we can deduce that the set of equations

$$\forall r \in \mathfrak{R}, \forall g, k \in H, \quad f_{k,gr}(x) = \sum_{l \in \mathfrak{L}, \gamma \in H} \mathbb{1}_H(rl) x_{l(rl)^{-1} g^{-1} k} = 0$$

describes $N_e(\mathbf{k}L)$. However, the equation system is redundant as the equation $f_{k,gr}(x) = 0$ is the same for all pairs (g, gk') with $k' \in H$. Removing the duplicates equations gives the system announced in the proposition. □

Example 42 (N_e in the case of an aperiodic monoid)**.** As in Example 36, let us consider the case of a $\mathcal{J}$-class with trivial $\mathcal{H}$-classes. In this case, we have $L = \mathfrak{L}, R = \mathfrak{R}$ and $H = \{1_H\}$, so the equations become:

$$\forall r \in R, \quad \sum_{l \in L} \mathbb{1}_H(rl) x_l.$$

Again from the Location Theorem, we have that $\mathbb{1}_H(rl) = 1$ if and only if there is an idempotent in $\mathcal{L}(r) \cap \mathcal{R}(l)$. So if we form a matrix A with rows indexed by L and columns indexed by R, and with coefficients 1 at $(\mathcal{L}(r), \mathcal{R}(l))$ if $\mathcal{L}(r) \cap \mathcal{R}(l)$ contains an idempotent and 0 otherwise, the above equations

becomes :

$$(x_l)_{l\in L}^T A = 0,$$

that is, in the case of a $\mathcal{H}$-trivial $\mathcal{J}$-class, $N_e(\mathbf{k}L)$ is the left kernel of the eggbox picture seen as a $\{0,1\}$-matrix.

Note that given this set of equations, we can compute the character $\chi_{\mathbf{k}M\otimes\mathbf{k}G_e^{op}}^{N_e(\mathbf{k}\mathcal{L}(e))}$ from the formula in Proposition 31 using classical linear algebra algorithms to find a basis of $N_e(\mathbf{k}\mathcal{L}(e))$ and then computing the value of the character at any $(m,g) \in C_M \times C_{G_e}$ by iterating over the basis vectors, applying (m,g) as a linear map and computing the relevant coefficient in the image vector.

IV. Performance, computational complexity and benchmarks

To begin with, in this section, we discuss the challenges and choices we have been led to make to measure the performance of our algorithms. In a second part, we present the experimental results as well as, when possible, their complexity analysis. We discuss performance for the computation of the number of fixed points, the character table and finally the Cartan matrix. At the end of this section, in Figures 3, 4 and 5, we give experimental measures of the complexity of our algorithms as a function of cardinality.

1. Challenges, experimental choices and methodology

Monoids being as diverse as they are, a meaningful analysis of the time and space complexities of the above algorithms is difficult, as many relevant metrics (such as the number of Green classes of a given type) cannot be straightforwardly computed. Moreover, in some sense these metrics vary a lot: two transformation monoid acting on the same number of points with the same number of generators can have vastly different Green structures. Although we can provide some time complexities in terms of number of $\mathcal{L},\mathcal{R}$-classes and cardinality of $\mathcal{H}$-classes (as we do below), the real test of viability is to see if the algorithm effectively terminates in practice. Thus, we provide timings and memory usage measures for the computation of the three main objects of our discussion: the bicharacter, the character table and the Cartan matrix.

The performance measures provided for these new algorithms (as well as the computation results presented hereafter) all come from an implementation using the computer algebra system GAP. All performance measures are realized on a laptop equipped with an Intel Core i7-10850H @ 2.7GHz (on one core) and 16 GB memory. Our specific implementation, as well as the test cases used and the

raw data, are publicly available on our git repository[6]. As the following section will show, these algorithms are, on our test machine, memory limited. We have made some test computations on a machine equipped with 128 GB of RAM. Even then, due to the rapid explosion of memory requirements, we cannot compute the character table of $\mathcal{T}_8$ using these methods, and it seems that monoids containing $\mathfrak{S}_{12}$ generally fail to reach the end of the bicharacter computation. Since in the cases where this more powerful machine enables more computation by bypassing memory limitations, computation time can reach the hour range, we choose to focus on the smaller scale tests allowed by our smaller machine.

We consider three families of monoids implemented by the package GAP Semigroups [**?**]: transformation monoids, which we have introduced before, partition monoids, and partial permutation monoids.

Definition 43. A *partition* P of a set S is a collection of subsets of S such that $\bigcup P = S$ and for all $p, p' \in P, p \cap p' = \emptyset$. The elements of P are called its *blocks*. Given $s \in S, P(s)$ is the unique block of P containing s.

Let P, Q be two set partitions of the set $S = [\![-n, -1]\!] \cup [\![1, n]\!]$. The *partition product* of P by Q is the partition PQ where $i, j \in S$ are in the same block if:

- $i, j > 0$ and are in the same P-block.
- $i, j < 0$ and are in the same Q-block.
- $i > 0$ and $j < 0$ and there exists $k \in [\![-n, -1]\!]$ such that $k \in P(i)$ and $k \in Q(j)$.
- i and j are related in the transitive closure of the relation given by the previous point.

This defines **the** *partition monoid* $\mathcal{P}_n$ (with identity $\{\{i, -i\} \mid i \in [\![1, n]\!]\}$). **A** partition monoid (of rank n) is a submonoid of $\mathcal{P}_n$.

Definition 44. A *partial permutation* of $[\![1, n]\!]$ is an injective partial function from $[\![1, n]\!]$ to itself. Equipped with the identity function and the partial map composition, this defines the *inverse symmetric monoid* $\mathcal{I}_n$. A partial permutation monoid (of rank n) is a submonoid of $\mathcal{I}_n$.

We test our functions on the families $\mathcal{T}_n, \mathcal{P}_n$ and $\mathcal{I}_n$, with numeric values provided for our canonical example $\mathcal{T}_n$ in Tables 2, 3 and 4. However, many

[6] `github.com/ZoltanCoccyx/monoid-character-table`

computer algebra systems, including GAP, are smart enough to detect that the Schützenberger groups are actually symmetric groups and thus use some non-general algorithms that could not be used on a typical finite monoid. This has the potential to falsify our measures (and indeed probably does, given Figures 3-(a), 4-(a) and 5-(a)). To mitigate this issue, we provide timings for randomly chosen finite monoids. The question of picking a "generic" monoid is entirely outside the scope of this paper. We simply choose a monoid $R(m, n)$ of rank m with n generators by picking uniformly n element of $\mathcal{T}_m, \mathcal{P}_m$ or $\mathcal{I}_m$. The set of (rank, number of generators) pairs used is given in Table 1. For reasons discussed hereafter, we have chosen to have "enough" generators to have non-trivial $\mathcal{J}$-structure. Experimentally, we note that the resources in time and memory used by two transformation monoids acting of the same rank with the same number of generators can differ by up to an order of magnitude. Thus, for each pair (m, n), we measure performance on 10 randomly chosen test cases. To experimentally evaluate the complexity as a function of cardinality, we do a linear regression of the logarithm of the (time and memory) measures against the logarithm of the cardinality. In the case of transformation monoids and partition monoid, the data follows a relatively tight distribution (see Figures 3 and 4). To mitigate threshold effects, we weight the sample values linearly (and not logarithmically) so that the higher values have more importance. By contrast, the randomly generated partial permutation monoids give more dispersed samples. Given this dispersion, considering only (or with a heavy weight) the highest values makes for an overestimation of necessary resources in most cases. Thus, in that case, we do the linear regression with equal weights on all samples. This does not dramatically change the measured complexity (we refer to the actual code as the final arbiter), but we advise caution when using these figures. For ease of discussion, we will call *structured monoids* the non-random ones present in our tests.

2. Performances and experimental results.

In the case of Algorithm 35, we can give some analysis of the time complexity in terms of the Green structure of the particular $\mathcal{J}$-class Algorithm 35 is applied to.

Proposition 45. *Consider a $\mathcal{J}$-class J containing n_L $\mathcal{L}$-classes, n_R $\mathcal{R}$-classes, containing a $\mathcal{H}$-class H with n_C conjugacy classes in $\Gamma'(H)$, and let C_M be a set of representatives of the character equivalence classes, as before. Then, the Algorithm 35 does:*

- *n_C computations of conjugacy classes of $\Gamma'(H)$ cardinality (if we precompute*

Type	Bicharacter	Character table and Cartan matrix
Transformation	(4,3), (5,3), (5,4), (6,5), (7,6), (8,8), (9,8)	(4,3), (5,3), (5,4), (6,5), (7,6)
Partition	(4,3), (4,6), (5,10), (5,15), (6,18), (7,20), (8,20), (9,25), (10,30), (12,40)	(4,3), (4,6), (5,10), (5,15), (6,18), (7,20), (8,20), (9,25), (10,30), (12,40)
Partial Permutation	(3,6), (3,9), (4,8), (4,12), (5,15), (6,20), (7,21), (7,25), (7,30), (8,25), (8,30)	(3,6), (3,9), (4,8), (4,12), (5,15), (6,20), (7,21), (7,25), (7,30)

Table 1: Rank and number of generators for random monoids

those cardinalities to be able to do a lookup in step 2-a of Algorithm 35, instead of computing it on the fly),

- $O(|C_M|(n_L + n_R))$ *monoid multiplications, Green class membership tests and conjugacy class of* $\Gamma'(H)$ *membership tests,*
- $O(|C_M|n_R)$ *computations of* τ_a*,*
- $O(|C_M|n_L)$ *conjugacy class of* $\Gamma'(H)$ *cardinality lookups,*
- $n_C|C_M|^2$ *integer multiplications.*

Proof. This simply results from an inspection of Algorithm 35, with the caveat that we precompute the cardinalities of the conjugacy classes of $\Gamma'(H)$. □

In the case of Algorithm 37 applied to an arbitrary monoid, such an analysis is mostly meaningless because we would need to express the number of Green classes and understand their breakdown. However, because of the vast variety of possible monoids, there is no meaningful way to do it from simple parameters such as the rank or the number of generators.

Note that we do not provide a cumulative formula for the complexity of Algorithm 35 as, for instance, the complexity of a conjugacy class membership test heavily depends on the algorithm used by the computer algebra system, that can itself vary depending on the characteristics of the Schützenberger groups. This

makes the task of providing a meaningful evaluation of the global complexity of the algorithm quite difficult, mainly because expressing the complexity of those "elementary" operations of monoid multiplications, membership testing, etc… in terms of the same parameters is not straightforward. However, we can at least compare this to the naive algorithm of testing if every element of J is a fixed point which demands $O(n_L n_R |H|^2 |C_M|^2)$ monoid multiplications: as long as the complexity of the more complex operations of Green class or conjugacy class membership testing remains limited in terms of monoid multiplications, our complexity is better. For instance, in the case of the monoid $\mathcal{T}_n$, all the required operations can be done on $O(n)$, making Algorithm 35 (and, in turn, Algorithm 37) more efficient than the naive algorithm, as can be seen in Table 2, with a sublinear (with respect to cardinality) measured complexity (Figure 3).

Monoid	Cardinality	Coefficients	Naive	Ours
$\mathcal{T}_3$	27	6^2	29 ms	8 ms
$\mathcal{T}_4$	256	11^2	92 ms	32 ms
$\mathcal{T}_5$	3125	18^2	1.44 s	84 ms
$\mathcal{T}_6$	46656	29^2	53.0 s	0.30 s
$\mathcal{T}_7$	823543	44^2	>30 min	1.54 s
$\mathcal{T}_8$	16777216	66^2	$\cdots$	8.65 s
$\mathcal{T}_9$	387420489	96^2	$\cdots$	58.2 s

Table 2: Computation time of the regular representation bicharacter.

As shown in Table 3, the computation of the character table takes much longer. This is because, to compute the radical of $\mathbf{k}\mathcal{L}(e)$ for an idempotent e, we must solve a linear system of size $|\mathcal{R}(e)| \times |\mathcal{L}(e)|$, which necessitates $O(|\mathcal{R}(e)|^2 |\mathcal{L}(e)|)$ arithmetic operations. In the case of the full transformation semigroup $\mathcal{T}_n$, if e has k elements in its image, $|\mathcal{L}(e)| = k! \times \binom{n}{k}$, while $|\mathcal{R}(e)| = k! \times S(n,k)$ where $S(n,k)$ is a Stirling number of the second kind, which gives $|\mathcal{R}(e)| \sim k^n$. The size of that linear system becomes rapidly intractable. Moreover, once we have a basis of $N_e(\mathbf{k}L)$ of cardinality d, we still have to compute the C_M^2 character values in $O(d^2)$ operations each. Experiments indicate that the computation time of the character tables of the maximal subgroups is small in comparison to all radical related computations. As can be seen in Figures 3-(b, c), 4-(b, c) and 5-(b, c) the limiting factor is memory (the test on random monoids fails for the random transformation monoids of the form $R(9,8)$ by exceeding the 16 GB memory capacity of our testing machine). Although computation requirements are close to linear in the cardinality, the cardinality tends to be more than exponential in the rank, limiting these methods to small ranks.

Monoid	Cardinality	Coefficients	Ours
$\mathcal{T}_3$	27	6^2	80 ms
$\mathcal{T}_4$	256	11^2	182 ms
$\mathcal{T}_5$	3125	18^2	1.30 s
$\mathcal{T}_6$	46656	29^2	17.6 s
$\mathcal{T}_7$	823543	44^2	5.93 min

Table 3: Computation time of the character table.

Finally, for the computation of the Cartan Matrix, the previous timings show that the vast majority of the computation time is spent computing the character table of the monoid. As the computation of the combinatorial bicharacter is more than a hundred times faster than the computation of the character table, this is a clear invitation to improve in particular the computation of the character of the radical of the $\mathcal{L}$-classes. In Table 4, we show some timings for that computation, and a comparison with Sage's generalist algorithm (based on the Peirce decomposition of the monoid algebra) for the computation of the Cartan Matrix: despite its limitations, our specialized algorithm allows for the handling of larger objects. Indeed, our algorithm has near linear performance with respect to cardinality, while Sage's has roughly cubic complexity.

Monoid	Coefficients	Sage's	Ours
$\mathcal{T}_3$	6^2	575 ms	42 ms
$\mathcal{T}_4$	11^2	5.23 min	146 ms
$\mathcal{T}_5$	18^2	>2h	1.29 s
$\mathcal{T}_6$	29^2	$\cdots$	17.7 s
$\mathcal{T}_7$	44^2	$\cdots$	5.48 min

Table 4: Computation time of the Cartan matrix.
In the case $\mathcal{T}_5$, Sage's algorithm was interrupted before the end of the computation.

Again, memory fails before time for $\mathcal{T}_8$ and onward. For the transformation monoids of the form $R(9, 8)$, using the regression, we can predict a computation time of around 6 hours on our testing machine if it was not memory limited.

An example of a Cartan matrix obtained using our Algorithms and Thiéry's formula is pictured in Figure 2.

Generally, our algorithms achieve (at worst) near linear measured complexity. The fixed point counting is most efficient in the structured monoids, where there are few, big $\mathcal{J}$-class with big Schützenberger groups. In the contrary, on "sparse" monoid with many small $\mathcal{J}$-class, the efficiency of the algorithm drops.

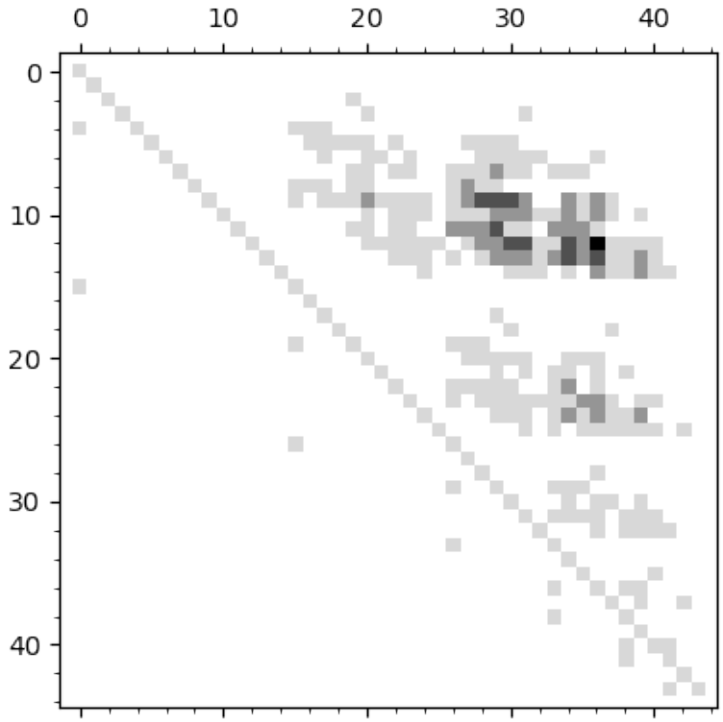
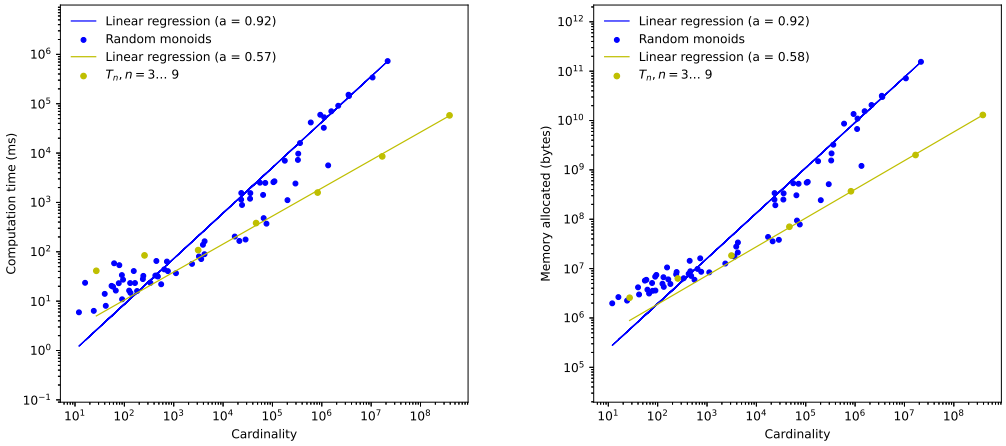
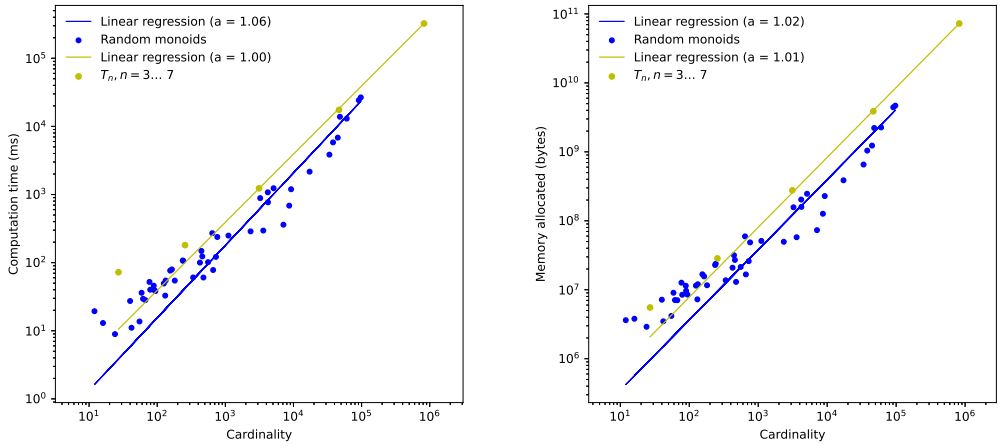


Figure 2: Cartan Matrix of $\mathcal{T}_7$
For legibility, the entries are represented as gray values. The entries are integers from 0 (in white) to 4 (the single black pixel).

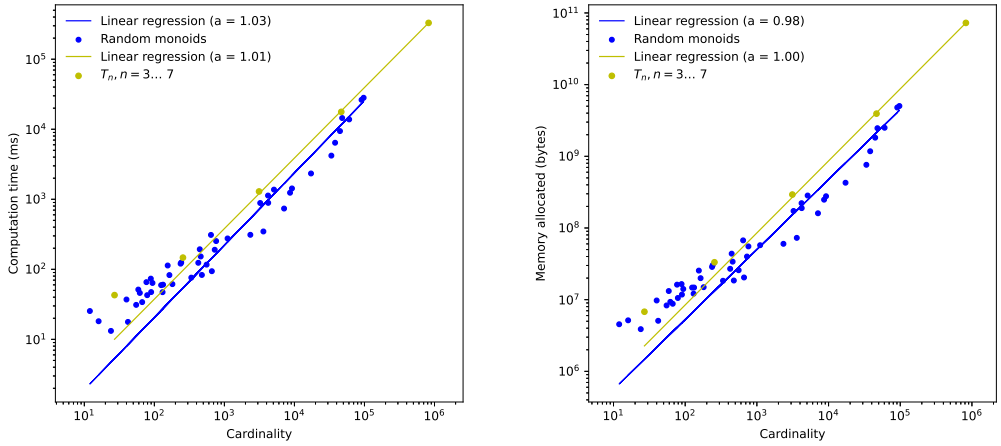
This most apparent in Figure 5, where the blue clusters are (roughly) semigroups of the same rank with the same number of generators. The partial permutation semigroups generated that way tends to have either (at our scale) low tens *or* multiple thousands $\mathcal{J}$-classes (without intermediate values). The latter kind tends to be much more expensive to deal with, despite implementing "lazy" fixed point counting on small $\mathcal{J}$-classes and $\mathcal{J}$-classes with trivial Schützenberger groups to avoid costly computations. The situation balances out (or even inverses itself, see Figure 4) when it comes to computing the character table and the Cartan matrix. This seems to come from the fact that bigger $\mathcal{J}$-classes mean higher dimensional radical and bigger Schützenberger groups mean less sparse basis vectors in the radical. Still, the measured complexity remains near linear in our experiments.



(a) Bicharacter

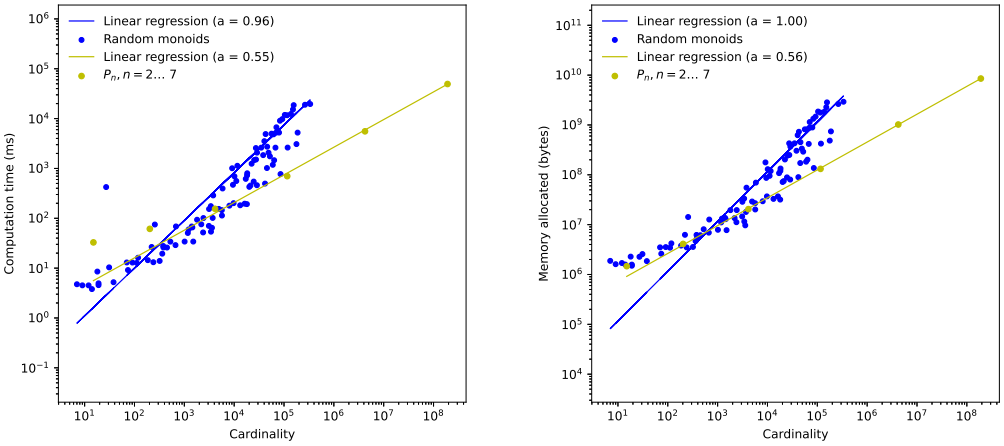


(b) Character table

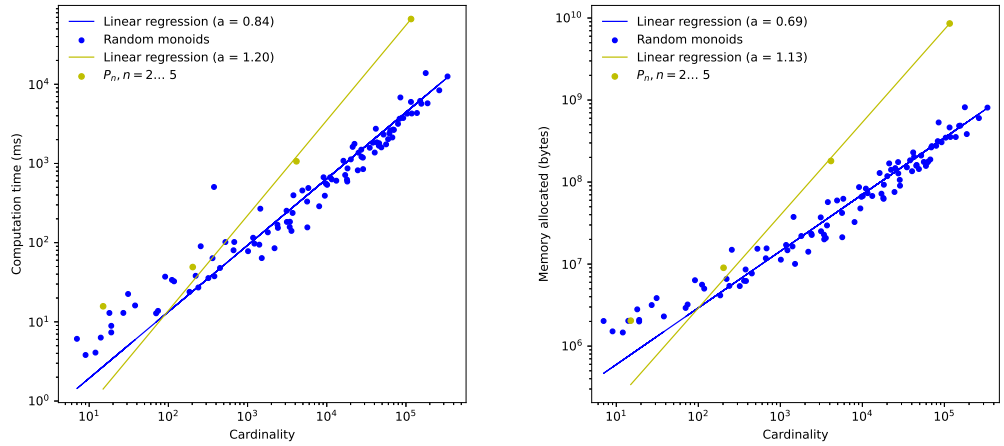


(c) Cartan Matrix

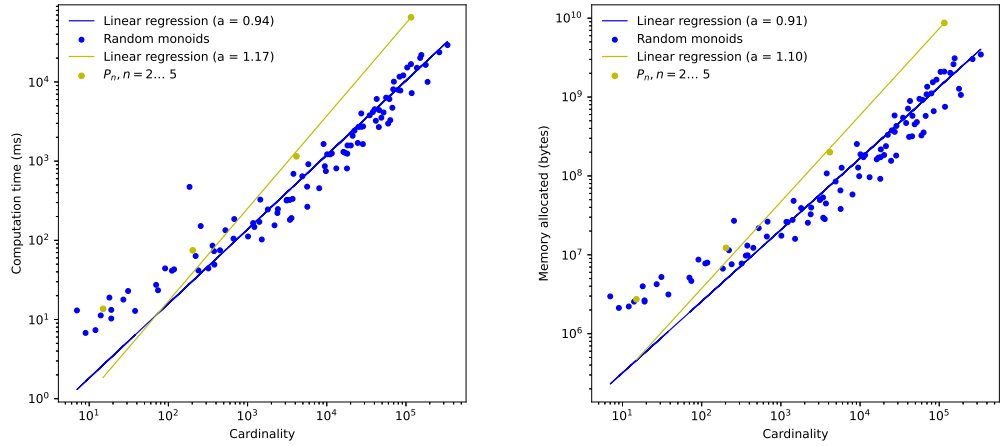
Figure 3: Time and memory usage: transformation monoids



(a) Bicharacter

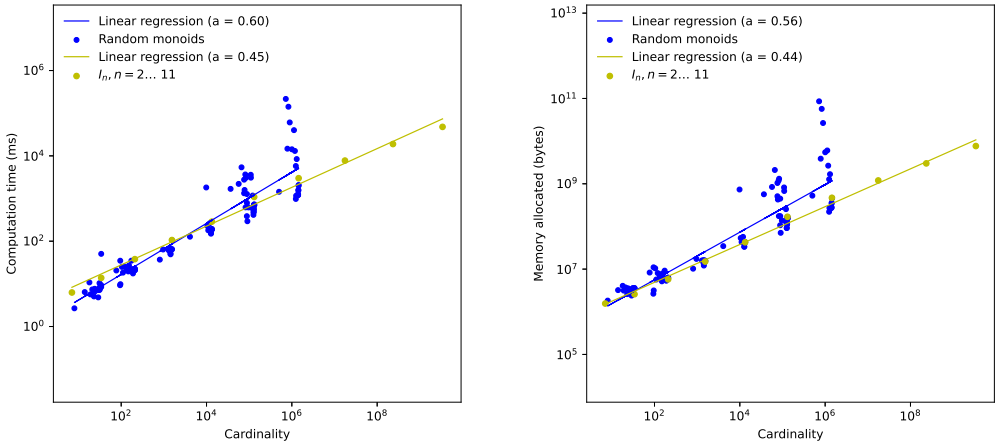


(b) Character table

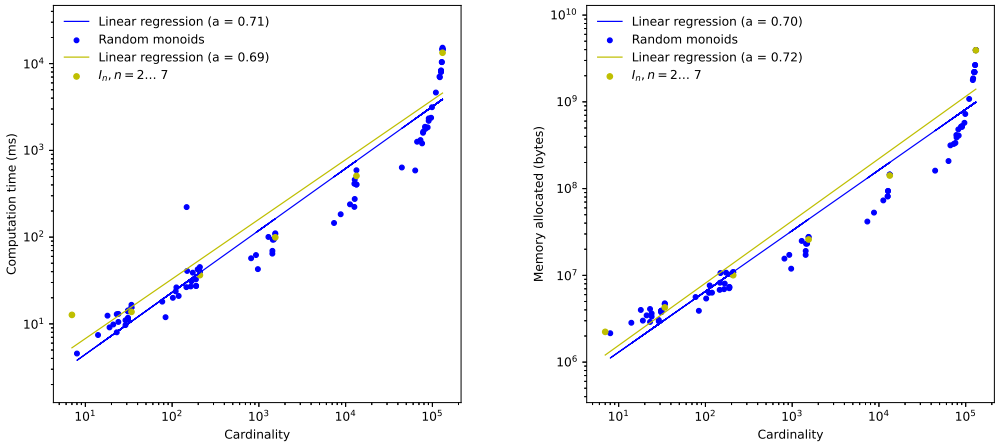


(c) Cartan Matrix

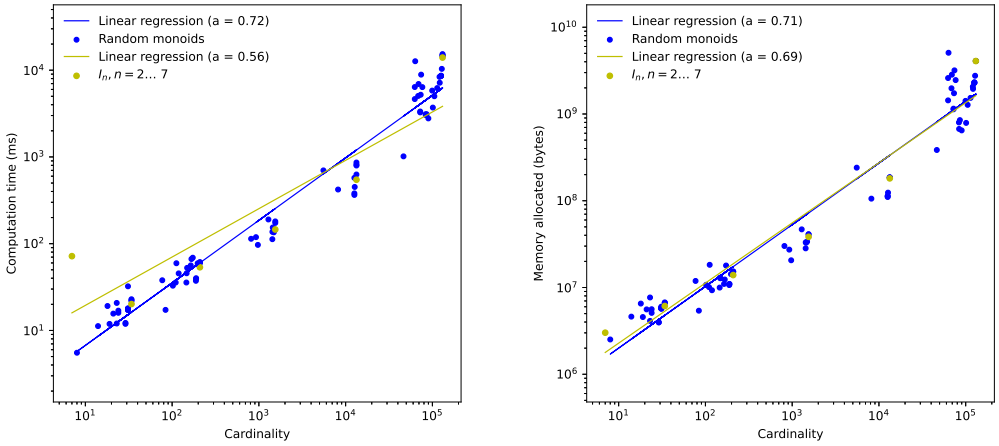
Figure 4: Time and memory usage: partition monoids



(a) Bicharacter



(b) Character table



(c) Cartan Matrix

Figure 5: Time and memory usage: partial permutation monoids

Conclusion and perspectives

The methods presented in this paper provide a new tool for the computational exploration of finite monoids representation theory. We give a method to compute the character table of a finite monoid in the general case as well as a method for the computation of the Cartan matrix. In the latter case, although general algorithms for any finite dimensional algebra already exist, by specializing to monoid algebras, we achieve vastly shorter computation times, thus making the question tractable for bigger monoids. Although we have presented the methods in details only for transformation monoids, the underlying formulas are true in general for finite monoids, and it remains computationally applicable whenever the hypotheses of Section III are verified. We also invite the reader to consult and test our implementation, available on our GitHub repository[7]. As this paper is inspired by the combinatorial research on monoid representation theory which have seen renewed activity in recent years, we hope that providing this effective tool will allow for the observation of new phenomena.

This work has two natural continuations: improving and expanding. For the improvement part, we have noted that by far the most inefficient part of our algorithm is the computation of the radical of the $\mathcal{L}$-classes. It happens to be the only point where linear algebra is necessary and combinatorics are seemingly not enough. We can ask whether this step could be replaced by a combinatorial computation. Some experiments show that, even in relatively small and very regular cases (T_5 for instance) the basis we find for the radical by solving the equation system described in Proposition 41 does not have easily understandable structure, once the common denominator of the coefficients is eliminated. It therefore seems unlikely to us that a general method for computing the radical of an entirely combinatorial nature exists. However, we remain optimistic that in very regular cases (again, T_n), the issue lies with us not finding the method rather than it not existing. More modestly, in a general context, we could try to exploit further the structure of the equations that define the radical to reduce the size of the system, which is a major bottleneck.

Another improvement, although perhaps less impactful, could be made by exploiting redundancy: it can happen that two $\mathcal{L}$-classes L_1, L_2 of a submonoid M of T_n are contained in the same $\mathcal{L}$-class L of T_n. Thus, in step $2-b$ of Algorithm 35 (for instance), instead of visiting each $\mathcal{L}$ of M, we could visit each $\mathcal{L}$ of T_n that contains a $\mathcal{L}$-class of M and count them with some multiplicity. Although this probably would not lead to great improvements in efficiency, this has the

[7] github.com/ZoltanCoccyx/monoid-character-table

advantage of making, in some sense, T_n the worst case scenario, allowing for a finer complexity analysis.

As for extending this work, the natural path seem to adapt these methods for fields of finite characteristic. At this point it appears to us that this question is tractable as the theory remains essentially the same, although it is somewhat difficult to implement in practice. The main hurdle arises, again, when computing the radical of a $\mathcal{L}$-class: an equivalent of Proposition 23 would have to take into account the role of the radical of the maximal subgroup algebra, which can be non-trivial in positive characteristic. This would translate in needing to effectively compute this radical. Although algorithms are available (for instance in GAP), this is a theoretically difficult and computationally expensive problem, considerably reducing the maximum size of a tractable problem. While modular representation theory is known to be a difficult subject in groups it seems that, again, the situation is not much more complicated for monoids than it is for groups as it is standard practice to reduce monoid theoretic questions to group theoretic ones. Treating modular group representation theory as a black box coming with already existing algorithms (much as we did here for null characteristic group representation theory as a matter of fact), we hope to be able to provide a modular version of our algorithms along with an implementation in the near future.

Acknowledgements

The research work devoted to this project was funded by a PhD grant from the French *Ministère de la recherche et de l'enseignement supérieur*, in the form of a *Contrat Doctoral Spécifique Normalien* attributed for a PhD in the STIC (*Sciences et Technologies de l'Information et de la Communication*) doctoral school of Paris-Saclay University, in the LISN (*Laboratoire Interdisciplinaire des Sciences du Numérique*) under the supervision of Pr. Nicolas Thiéry.

Annex: Given P, Q two set partitions on $\{\pm 1, \pm 2, \dots, \pm n\}$, one can create a new partition P, Q on the same set in the following way:

- if $i, j < 0$ are in the same P-block, they are PQ-related,
- if $i, j > 0$ are in the same Q-block, they are PQ-related,
- if $i < 0$, $j > 0$ and there exists $k \in [\![1, n]\!]$ such that i, k are in the same P-block, $-k, j$ are in the same Q-block, then i, j are PQ-related,
- the partition PQ is the set of classes of the equivalence relation generated by the three previous PQ relations.

The *partition monoid* P_n is the set of all set partitions on $\{\pm 1, \pm 2, \dots, \pm n\}$ equipped with the previous composition law, with identity element $\{\{i, -i\} \mid i \in [\![1, n]\!]\}$. Since the maximal subgroups at the idempotents of P_n are symmetric groups, we can meaningfully label the coefficients of the character table and of the Cartan matrix by integer partitions.

	1^5	$2^1 1^3$	$2^2 1^1$	$3^1 1^2$	$3^1 2^1$	$4^1 1^1$	5^1	1^4	$2^1 1^2$	2^2	$3^1 1^1$	4^1	1^3	$2^1 1^1$	3^1	1^2	2^1	1^1	1^1
5^1	1	1	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.
$4^1 1^1$	4	2	.	1	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.
$3^1 2^1$	5	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.
$3^1 1^2$	6	.	2	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.
$2^2 1^1$	5	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.	.
$2^1 1^2$	4	2	.	1	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.
1^5	1	1	1	1	1	1	1	.	.	.	.	.	.	.	.	.	.	.	.
4^1	15	7	3	3	1	1	.	1	1	1	1	1	.	.	.	.	.	.	.
$3^1 1^1$	45	9	1	.	.	1	.	3	1	1	.	1	.	.	.	.	.	.	.
2^2	30	2	2	2	1	.	.	2	.	2	1	.	.	.	.	.	.	.	.
$2^1 1^2$	41	1	2	1	1	1	1	3	1	1	.	1	.	.	.	.	.	.	.
1^4	15	5	1	3	1	1	.	1	1	1	1	1	.	.	.	.	.	.	.
3^1	75	21	11	3	3	1	.	10	4	2	1	.	1	1	1	.	.	.	.
$2^1 1^1$	109	23	5	1	1	1	1	17	3	1	1	1	2	.	1	.	.	.	.
1^3	75	1	9	3	1	1	.	10	2	2	1	.	1	1	1	.	.	.	.
2^1	51	19	11	6	4	3	1	14	6	6	2	2	4	2	1	1	1	.	.
1^2	160	32	.	7	1	.	.	31	5	5	1	1	6	.	.	1	1	.	.
1^1	151	47	19	13	5	5	1	37	13	5	4	1	10	4	1	3	1	1	.
1^1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 6: Character table of P_5.
For readability, zeroes are replaced with dots.

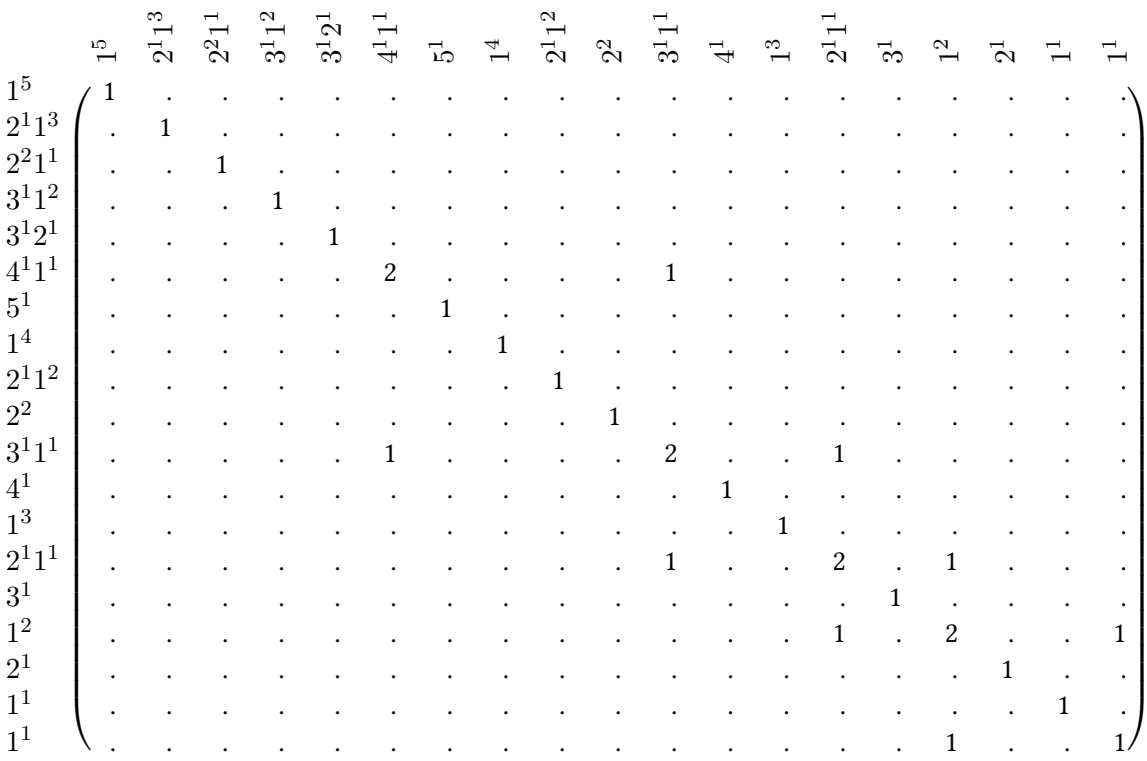


Figure 7: Cartan Matrix of the P_5.
For readability, zeroes are replaced with dots.