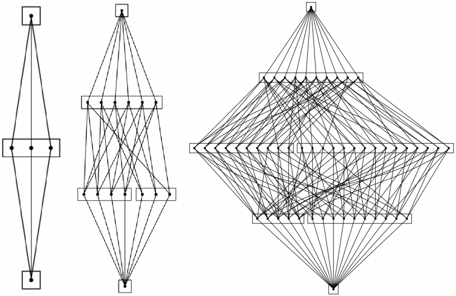
Computing a finite semigroup

J. D. Mitchell

School of Mathematics and Statistics, University of St Andrews

4th of February, 2014



Some questions?

Aim: to present some algorithms to compute finite semigroups.

We must address the following questions:

- how is the semigroup **given**?
- what are we trying to **compute**?
- what is the **complexity** of the algorithms?

Some things the talk is not about:

- programming issues;
- data structures;
- implementations;
- user interfaces.

GAP

A prelude to some answers

GAP is a free, open system for **computational discrete mathematics**, in particular group theory.

- **free** GAP is can be downloaded from `www.gap-system.org`
- **free** GAP (as of version 4.3) is released under the GPL.
- **open** the source code is completely available.
- **open** mechanism for third-party contributions, and distribution.

GAP runs on (almost) every platform.

Estimated to have **thousands** of users world-wide.

Why compute?

- perform low-level calculations such as **multiplication**, **inversion**, and so on;
- suggests new **theoretical results**;
- obtain **counter-examples**;
- gain more **detailed understanding** of the objects under consideration;
- perform more **intricate** calculations than possible by hand.

Even computing small examples by hand can exceed human patience.

Insert semigroup into computer...

How is a semigroup given?

There are 3 main ways to define a semigroup to a computer:

Cayley table: ...;

Finite presentation: words in generators and relations i.e.

$$\langle e, f \mid e^2 = e,\ efe = fe,\ f^2e = fe,\ f^3 = f,\ fef^2 = fe \rangle.$$

Generators: as a subsemigroup of a larger semigroup such as transformations, matrices, binary relations, partitions, and so on ...

In this talk, we will deal (almost) exclusively with the latter.

Enumeration of semigroups

n	number of semigroups	
0	1	
1	1	
2	4	
3	18	
4	126	(Forsythe '54)
5	1160	(Motzkin-Selfridge '56)
6	15 973	(Plemmons '66)
7	836 021	(Jürgensen-Wick '76)
8	1 843 120 128	(Satoh-Yama-Tokizawa '94)
9	52 989 400 714 478	(Distler-Kelsey '11)
10	12 418 001 077 381 302 684	(Distler-Kelsey '13)
11	??	(Everyone '13)

The semigroups of orders 1 to 8 are available in the GAP package Smallsemi available at `tinyurl.com/smallsemi`.

The semigroups of order 2, 3 and 4

	a	b
a	a	a
b	a	a

	a	b
a	a	b
b	b	a

	a	b
a	a	a
b	a	b

	a	b
a	a	a
b	b	b

	a	b	c
a	a	a	a
b	a	a	a
c	a	a	a

	a	b	c
a	a	a	c
b	a	a	c
c	c	c	a

	a	b	c
a	a	b	b
b	b	a	a
c	b	a	a

	a	b	c
a	a	a	a
b	a	a	a
c	a	a	b

	a	b	c
a	a	a	a
b	a	a	a
c	a	a	c

	a	b	c
a	a	a	a
b	a	a	a
c	a	b	c

	a	b	c
a	a	a	a
b	a	a	a
c	c	c	c

	a	b	c
a	a	a	a
b	a	a	b
c	a	b	c

	a	b	c
a	a	a	c
b	a	a	c
c	c	c	c

	a	b	c
a	a	b	a
b	b	a	b
c	a	b	c

	a	b	c
a	a	b	c
b	b	a	c
c	c	c	c

	a	b	c
a	a	a	a
b	a	b	a
c	a	a	c

	a	b	c
a	a	a	a
b	a	b	a
c	c	c	c

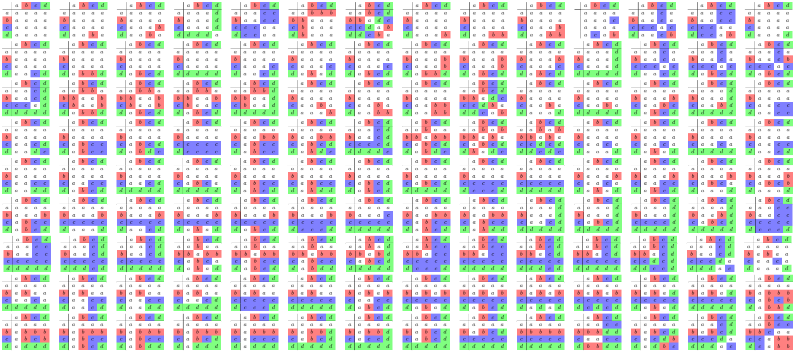
	a	b	c
a	a	a	a
b	a	b	b
c	a	b	c

	a	b	c
a	a	a	a
b	a	b	b
c	a	c	c

	a	b	c
a	a	a	a
b	a	b	c
c	c	c	c

	a	b	c
a	a	a	a
b	b	b	b
c	c	c	c

	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b



Fundamental tasks

What are we trying to compute anyway?

INPUT: a list of $x_1, \ldots, x_m$ (in the universe) generating a semigroup U.

OUTPUT/TEST:

- the size of U;
- membership in U;
- factorise elements over the generators;
- the number of idempotents ($x^2 = x$);
- the maximal subgroups;
- the ideal structural of U (i.e. Green's relations);
- is U a group? an inverse semigroup? a regular semigroup?

The universe

Transformations, partial perms, matrices, partitions...

The **full transformation monoid** is just the monoid of all transformations under composition of functions.

A **symmetric inverse monoid** is just the monoid of all partial permutations under composition of functions.

A **general linear monoid** of $n \times n$ matrices over a finite field.

A **partition monoid** is the monoid of partitions...

A **Rees 0-matrix semigroup** ...

Excluded middle

Exhaustive: store the elements

- be happy with relatively small semigroups
- SgpWin by Don McAlister (2006?)
- Semigroupe by Jean-Eric Pin (2009)

Non-exhaustive: don't store the elements

- Lallement-McFadden (1990)
- Monoid package for GAP3 by Linton-Pfeiffer-Robertson-Ruškuc (1997)
- Semigroups package for GAP4 by me (2013)

The limitations of exhaustive enumeration

n	# transformations	memory	unit
1	1	16	bits
2	4	16	bytes
3	27	162	bytes
4	256	2	kb
5	3 125	~ 30	kb
6	46 656	~ 546	kb
7	823 543	~ 10	mb
8	16 777 216	~ 256	mb
9	387 420 489	~ 6	gb
10	10 000 000 000	~ 186	gb
11	285 311 670 611	~ 6	tb
12	8 916 100 448 256	~ 194	tb
13	302 875 106 592 253	~ 7	pb

Storing the elements of a semigroup internally quickly becomes impractical.

A simple example

Let S be the semigroup generated by the transformations

$$x_1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 3 \end{pmatrix} \qquad \text{and} \qquad x_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 1 \end{pmatrix}.$$

- How many elements does S have?
- How many idempotents does S have?
- What are its maximal subgroups?

An exhaustive algorithm

S acting on itself by right multiplication

Input: A subsemigroup $S = \langle\, x_1, x_2, \ldots, x_m \,\rangle$ of a larger semigroup.

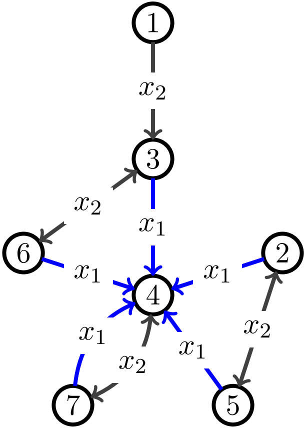
Output: The elements of S.

Suppose $x_1, x_2, \ldots, x_m$ are distinct. Here's the algorithm:

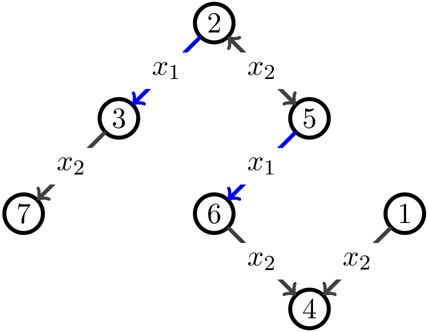
$X := [x_1, x_2, \ldots, x_m]$
for $y \in X$ **do**
    for $i \in \{1, \ldots, m\}$ **do**
        if $yx_i \notin X$ **then**
            append yx_i to X
        end if
    end for
end for
return X

Pay closer attention...

...and we've found the Cayley graphs and a presentation



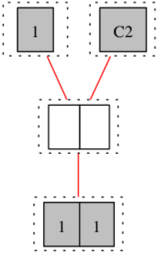
The right Cayley graph.



The left Cayley graph.

$$\langle e, f \mid e^2 = e,\ efe = fe,\ f^2e = fe,\ f^3 = f,\ fef^2 = fe\rangle ...$$

… and the Green's structure



What's not so great, is that the algorithm spends lots of time:

- checking $yx_i \notin X$;
- multiplying elements;
- uses too much memory.

The algorithm takes no advantage of the structure or representation of the semigroup.

Non-exhaustive

Overview

Let S be a finite regular semigroup and let $U = \langle x_1, \ldots, x_m \rangle \leq S$.

We consider S **known** and U **unknown**.

We don't want to find or store the elements of U.

We want to **decompose** S into **blocks** so that:

- the blocks have some **uniform structure**;
- the blocks are **easy to compute** from the given generators;
- the **structure** of S is used;
- we take advantage of **computational group theory**.

Blocks = Green's $\mathscr{R}$-classes

Actions and stabilisers

Suppose S **acts on the right** on a set Ω. The **natural induced right action** of S on the power set $\mathcal{P}(\Omega)$ is:

$$\Sigma \cdot s = \{\, \alpha \cdot s \;:\; \alpha \in \Sigma \,\} \qquad \text{for } \Sigma \subseteq \Omega \text{ and } s \in S$$

and we define $s|_\Sigma : \Sigma \longrightarrow \Sigma \cdot s$ by $s|_\Sigma : \alpha \mapsto \alpha \cdot s$.
The **stabiliser** of Σ under S is

$$\mathrm{Stab}_S(\Sigma) = \{\, s \in S^1 \;:\; \Sigma \cdot s = \Sigma \,\}.$$

Then the quotient of $\mathrm{Stab}_S(\Sigma)$ by the kernel of its action is isomorphic to

$$S_\Sigma = \{\, s|_\Sigma \;:\; s \in \mathrm{Stab}_S(\Sigma) \,\}$$

which is a subgroup of the symmetric group $\mathrm{Sym}(\Sigma)$ on Σ.
The **strongly connected component** (s.c.c.) of $\alpha \in \Omega$ is

$$\{\, \beta \in \Omega \;:\; \exists s, t \in S^1, \beta = \alpha \cdot s, \alpha = \beta \cdot t \,\}.$$

Schreier's Theorem for Semigroups Actions

Suppose that S **acts on the right** on a set Ω, and U is a subsemigroup of S.

If $\Sigma \subseteq \Omega$, then

$$S_\Sigma = \{\, s|_\Sigma \ :\ s \in \mathrm{Stab}_S(\Sigma) \,\}.$$

Proposition (Linton-Pfeiffer-Robertson-Ruškuc '98)

Let $\{\Sigma_1, \ldots, \Sigma_n\}$ be an s.c.c. of the action of U on $\mathcal{P}(\Omega)$. Then:

(i) *for every $i > 1$, there exist $u_i, v_i \in U$ such that $\Sigma_1 \cdot u_i = \Sigma_i$, $\Sigma_i \cdot v_i = \Sigma_1$, $(u_i v_i)|_{\Sigma_1} = \mathrm{id}_{\Sigma_1}$ and $(v_i u_i)|_{\Sigma_i} = \mathrm{id}_{\Sigma_i}$;*

(ii) *U_{Σ_i} and U_{Σ_j} are isomorphic as permutation groups;*

(iii) *$U_{\Sigma_1} = \langle\, (u_i s v_j)_{\Sigma_1} : 1 \leq i, j \leq n,\ s \in X,\ \Sigma_i \cdot s = \Sigma_j \,\rangle$.*

Action on $\mathscr{L}$-classes

Let S be a finite regular semigroup and let $U \leq S$. Then U acts freely on the $\mathscr{L}$-classes of S by right multiplication ($\mathscr{L}$ is a **right congruence**).

Proposition (East-Egri-Nagy-M-Péresse '13)

Let $x, y \in U$ be arbitrary and let $x' \in S$ such that $xx'x = x$. Then:

(i) $\{\, L_y^S \;:\; y \in R_x^U \,\}$ *is an s.c.c. of the action of* U *on* $S/\mathscr{L}$;

(ii) $L_x^S \cap R_x^U$ *is a group under* $s * t = sx't$ *that is isomorphic to* $U_{L_x^S}$;

(iii) $x\mathscr{R}^U y$ *implies* $U_{L_x^S} \cong U_{L_y^S}$;

(iv) $|R_x^U| = |U_{L_x^S}| \cdot |\{\, L_y^S \;:\; y \in R_x^U \,\}|$;

(v) *If* L_x^S *belongs to the s.c.c. of* L_y^S *under the action of* U *on* $S/\mathscr{L}$, *then* $|R_x^U| = |R_y^U|$.

So what?

For an $\mathscr{R}$-class of $U \leq S$, there are the actions of:

- U on the $\mathscr{L}$-classes of S by right multiplication;
- the group U_L where L is an $\mathscr{L}$-class of S.

Suppose that Ω is a set and $\lambda : S \longrightarrow \Omega$ such that:

(i) $|\Omega| = |S/\mathscr{L}|$;

(ii) $(x)\lambda = (y)\lambda$ if and only if $L_x^S = L_y^S$ for all $x, y \in S$;

(iii) S acts on Ω such that $(x \cdot u)\lambda = (x)\lambda \cdot u$ for all $x \in S$ and $u \in U$.

Then the actions of U on $S/\mathscr{L}$ and Ω are **isomorphic** via λ.

Transformation semigroups

Proposition

Let U be any subsemigroup of some T_n where $n \in \mathbb{N}$. Then:

(i) *the action of U on $T_n/\mathscr{L}$ is isomorphic to the action of U on $\mathcal{P}(\{1, 2, \ldots, n\})$ defined by*

$$X \cdot f = \{\, (x)f \,:\, x \in X \,\};$$

(ii) *if $L \in T_n/\mathscr{L}$, then U_L acts faithfully on $\operatorname{im}(x)$ for all $x \in L$.*

This is what:

(i) Subsets of $\{1, \ldots, n\}$ are easier to compute with than $T_n/\mathscr{L}$;

(ii) For example, if $x \in T_{10}$ and $|\operatorname{im}(x)| = 5$, then

$$|L| = 5! * S(10, 5) = 5103000.$$

It is much easier to compute the action of U_L on $\operatorname{im}(x)$ than on L.

Rees 0-matrix semigroups

Let $S = \mathcal{M}^0[T : I, J; P]$ be a regular **Rees 0-matrix semigroup** over a permutation group $G \leq S_n$ and $|J| \times |I|$ matrix $P = (p_{j,i})_{j \in J, i \in I}$.

Proposition

Let U be any subsemigroup of S. Then:

(i) *the action of U on $S/\mathscr{L}$ is isomorphic the action of U on $J \cup \{0\}$ defined by*

$$0 \cdot (j, g, k) = 0 \cdot 0 = 0 = i \cdot 0, \quad i \cdot (j, g, k) = \begin{cases} k & \textit{if } p_{i,j} \neq 0 \\ 0 & \textit{if } p_{i,j} = 0 \end{cases}$$

(ii) *if L is any $\mathscr{L}$-class of S, then U_L acts faithfully on $\{1, \ldots, n\}$ by*

$$m \cdot (i, g, j)|_L = m \cdot p_{j,i} g \quad \textit{for all} \quad m \in \{1, 2, \ldots, n\}$$

is faithful.

An algorithm

Let $U = \langle\, X \,\rangle$ be a subsemigroup of a finite regular semigroup S.

Green's $\mathscr{R}$-relation is a **left congruence** on S and so S acts by left multiplication on $\mathscr{R}$-classes.

find $(U)\lambda = \{(u)\lambda : u \in U\}$ $\triangleright$ **the standard orbit algorithm**
find the s.c.c.s of $(U)\lambda$ $\triangleright$ **standard graph algorithms**
$\mathfrak{R} \leftarrow X$ $\triangleright$ **$\mathscr{R}$-class reps**
for $r \in \mathfrak{R}$ **do**
    identify the s.c.c. of $(r)\lambda$ in $(U)\lambda$
    compute $U_{L_x^S}$ $\triangleright$ **if we didn't already**
    for $x \in X$ **do**
        if $(xr, y) \notin \mathscr{R}^U$ for any $y \in \mathfrak{R}$ **then**
            append xr to $\mathfrak{R}$
        end if
    end for
end for

Return

The output is:

- $\mathfrak{R}$ - the $\mathscr{R}$-class representatives of U;
- data structures for the $\mathscr{R}$-classes;

The latter lets us calculate/test:

size: $|U| = \sum_{x\in\mathfrak{R}} |R_x|$;

membership: $x \in U$ if and only $(x)\lambda \in (U)\lambda$ and $(x'y)|_{L_y^S} \in U_{L_y^S}$ for some $y \in \mathfrak{R}$;

factorisation: pay very very very close attention!

...