Semigroupe 2.0

Jean-Éric Pin

Jean-Eric.Pin@liafa.jussieu.fr
http://www.liafa.jussieu.fr/~jep

1 Presentation

Semigroupe is a C programme designed to compute finite semigroups. It runs on any machine equipped with a C compiler. So far, it has been tested successfully on Unix machines, on PC's and on Apple machines (Macintosh, PowerBook, etc.). The programme is extremely fast, but is not optimized in terms of space requirements. It is very likely that, if you try to push the programme to its limits, you will encounter memory problems rather than time problems.

The programme is based on algorithms designed by Véronique Froidure and the author [2]. Version 2.0 allows to produce outputs in LaTeX, but this feature is mainly useful for relatively small semigroups (say of size smaller than 300), otherwise the files generated by the programme are huge.

A reminder on the basic definitions of semigroup theory is given at the end of this document.

2 Installation

Read carefully the `README` file and follow the instructions. Please report any problem to `Jean-Eric.Pin@liafa.jussieu.fr`

3 Main Menu

After the prompt

```
Semigroupe Version 2.000000 *** Jean-Eric Pin, January 2009 ***
```

You are first asked:

```
Would you like a LaTeX output for this session (y/n) ?
```

You should answer by `y` or `n` (followed by `return`, like for any command).
If you answer no, you will have the choice between the following options:

```
 Give your choice :
 (1) Semigroup
 (2) Monoid
 (3) Ordered syntactic semigroup
 (4) Ordered syntactic monoid
 (5) Standard example
 (6) Read a file
 (7) Modify preferences
 (8) Quit Semigroupe
```

3.1 Options 1-4

Options (1)-(4) are self-explanatory. For instance, choose option (1) if you want to compute a finite semigroup (not a monoid). More details are given in Section 4 below.

3.2 Standard examples

See Section 5 for more details on this option.

3.3 Read a file

This option allows you to read a file as input. This file should be in the Examples folder specified in your Preference file. On a Unix system, the Preference file is called `.Semigroupe2Prefs` and should be in your home directory.

3.4 Modify preferences

This option lets you modify your Preferences.

```
 (1) Change language ?
 (2) Modify the path of the examples folder ?
 (3) Modify the path of the LaTeX examples folder ?
```

If you want to change the default language, the following languages are available: English, French, German, Italian, Portuguese and Spanish.
By default, the path for the examples folder is
`homedirectory/Documents/Semigroupe/Examples`
where `homedirectory` is the absolute path of your home directory.
The path of the LaTeX examples folder is
`homedirectory/Documents/Semigroupe/LaTeXFiles`

3.5 Quit Semigroupe

This is the way to quit the programme.

4 Computing semigroups

If you select one of the options 1–4, you will be asked

```
  Number of letters of the alphabet ?
```

that is, the number of generators of your semigroup. Since Semigroupe was designed with application to automata theory in mind, the number of generators should be $\leqslant 26$.

The semigroup is given as a subsemigroup of one of the following semigroups (called the *universe*), which are selected in the next menu.

```
Give the type of semigroup :
 (1) Transitions
 (2) Partial transitions
 (3) Boolean matrices
 (4) Max-Plus matrices
 (5) Min-Plus matrices
 (6) Tropical Max-Plus matrices
 (7) Tropical Min-Plus matrices
 (8) Projective Max-Plus matrices
 (9) Matrices with integer coefficients
```

These options have the following meaning:

(1) selects the semigroup of all transformations on the set $\{1, 2, \dots, n\}$,

(2) selects the semigroup of all partial transformations on the set $\{1, 2, \dots, n\}$,

(3) selects the semigroup of square Boolean matrices of size n,

(4) selects the semigroup of matrices over the semiring $(\mathbb{Z}, \max, +)$,

(5) selects the semigroup of matrices over the semiring $(\mathbb{Z}, \min, +)$,

(6) selects the semigroup of matrices over the semiring $(\{-\infty, 0, 1, \ldots, t\}, \max, +)$, for some threshold t,

(7) selects the semigroup of matrices over the semiring $\{0, 1, \ldots, t, +\infty\}, \min, +)$, for some threshold t,

(8) selects the semigroup of all projective matrices over the semiring $(\mathbb{Z}, \max, +)$,

(9) selects the semigroup all matrices over the semiring $\mathbb{Z}_{t,p}$, for some threshold t and some period p.

The next question is

```
Number of states of the automaton ?
```

If you chose Option (2) (partial transitions), do not count the sink state 0 in the number of states. Next a self-explanatory dialog permits to enter the generators. This time you can use the sink state 0 as in the following example:

```
Number of states of the automaton ? 3

1.a = 2
2.a = 3
3.a = 1

1.b = 1
2.b = 0
3.b = 3
```

Then you are asked to give an upper bound to the size of the semigroup you are computing. This value is used to estimate the size of the hash table used in the computation so it is important to give a real upper bound (otherwise the programme may crash). If you have no idea of the size, try a large number, like 50,000. Actually, Semigroupe is able to compute semigroups as large as 2,000,000 elements on a machine with 256 Mbytes of memory, and over 5,000,000 elements if you have over 1Gbyte available. But in general, you will not have to go that far.

You can also save your generators in a file.

```
Would-you like to save these generators (y/n) ? y
Give the name of the file : myTestFile
```

This feature is very useful since you can edit and modify this file for future use.
Then the computation starts and gives as output something like this

```
Generators :
a | 2  3  4  5  6  7  8  1
b | 1  2  3  4  5  6  8  0

*******************
Computation of the D-classes
Computation of the H-classes
Computation time for the elements 0s  2/100
Computation time for the D-classes 0s  1/100
Computation time for the H-classes 0s  0/100
Cumulative computation time 0s  4/100
Number of elements : 51481
Number of relations : 1603
Computation terminated. Maximal length of words : 55
D-classes 9
R-classes 256
L-classes 256
```

```
H-classes 12870

Number of idempotents : 256

bbaabbaabbaabb = 0
```

The generators are the generators of the semigroup you just defined. The sentence "Computation of the $\mathcal{D}$-classes" indicates that this computation is over. The next sentences give some indication on the computation time: the first line gives the time needed to compute the elements of the semigroup, and the next line gives the total amount of time to compute the elements and the Green relations. The number of elements is given, as well as the number of relations.

Several options are offered:

4.1 List of elements

This option provides the list of all elements of the semigroup. These elements are represented by words. If the universe is the semigroup T_n of all transformations on $\{1, 2, \ldots, n\}$, the value of each element (a transformation) is given. For instance, if you compute the semigroup F_2 of example 3 (`Monoid of partial functions Fn`, with $n = 2$), you will obtain the following output:

```
           |  1  2
------------------
*  1     1 |  1  2
G  2     a |  2  1
*  3     b |  1  1
*  4     c |  1  0
R  5    ac |  0  1
*  6    ba |  2  2
R  7    ca |  2  0
*  8   aca |  0  2
*M 9   bac |  0  0
```

A `G` indicates an element of a group, an `R` indicates a regular element and an `M` indicates an element of the minimal ideal. A star indicates an idempotent.
If the universe is a semigroup of matrices, each matrix is given line by line. For instance

```
(11) Monoid of unitriangular Boolean matrices of size n x n
...
(1) List of elements

          |  1  0  0 | 0  1  0 | 0  0  1 |
-----------------------------------------
*  1    1 |  1  0  0 | 0  1  0 | 0  0  1 |
*  2    a |  1  1  0 | 0  1  0 | 0  0  1 |
*  3    b |  1  0  1 | 0  1  0 | 0  0  1 |
*  4    c |  1  0  0 | 0  1  1 | 0  0  1 |
*  5   ab |  1  1  1 | 0  1  0 | 0  0  1 |
*M 6   ac |  1  1  1 | 0  1  1 | 0  0  1 |
*  7   bc |  1  0  1 | 0  1  1 | 0  0  1 |
   8   ca |  1  1  0 | 0  1  1 | 0  0  1 |
```

In this example, the generators are

$$a = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

4.2 List of relations

Provides a presentation of S. Actually the output is a confluent rewriting system defining S. The rules are given under the form $u = v$, but could be interpreted as $u \to v$. In each case, v is strictly smaller than u in the shortlex order (that is, either v is strictly shorter than u, or v and u have the same length, but v is before u in the lexicographic order). For instance, if S is the semigroup T_3 of all transformations on three elements generated by the three transformations a, b and c described below, the relations are

```
bb = 1
bc = ac
cc = c
aaa = 1
aab = ba
aba = b
baa = ab
bab = aa
bac = c
cac = cb
acaac = caac
caacb = caaca
caacab = caac
```

4.3 List of idempotents

This command gives, as expected, the list of idempotents. For instance, if $S = T_3$, the set of idempotents is

$$E(S) = \{1, c, acb, cba, aaca, acaa, caac, aacab, caaca, caacaa\}$$

4.4 Minimal ideal

Computes the minimal ideal I. For instance, if $S = T_3$,

$$I = \{caac, caaca, caacaa\}$$

4.5 Green's relations

Computes the Green's relations $\mathcal{D}$, $\mathcal{R}$ and $\mathcal{L}$. Actually, the programme computes the number of $\mathcal{D}$-classes, $\mathcal{R}$-classes and $\mathcal{L}$-classes and assign a number to each of these classes. Then it gives, for each element s, the number corresponding to the $\mathcal{D}$-class, the $\mathcal{R}$-class and the $\mathcal{L}$-class of s, respectively.

Remember that $\mathcal{J} = \mathcal{D}$ and that $\mathcal{H} = \mathcal{R} \cap \mathcal{L}$.

4.6 Computation of the inverses

An element $\bar{x}$ is a weak inverse of x if $\bar{x}x\bar{x} = \bar{x}$. It is an inverse if, furthermore, $x\bar{x}x = x$. This option computes both the inverses and weak inverses of each element.

4.7 Computation of a local submonoid

Given an idempotent e, compute the monoid eSe. For instance, if $S = T_3$ and $e = c$, $eSe = \{c, cb, caac, caaca\}$.

4.8 Computation of a right ideal

Computes the right ideal generated by a given element. For instance, if $S = T_3$ and $u = cc$, then $uS = \{c, ca, cb, caa, cab, cba, caac, caaca, caacaa\}$.

4.9 Computation of a left ideal

Computes the left ideal generated by a given element. For instance, if $S = T_3$ and $u = cc$, $Su = \{c, ac, cb, aac, acb, aacb, caac, caaca\}$.

4.10 Computation of an element

Given a word, computes its reduced form using the rewriting system described above. For instance, if $S = T_3$ and $u = baaababcbabbacba$, the reduced form is $caacaa$.

4.11 Computation of the kernel

Computes the kernel of S. For instance, if S the semigroup of example 22 (in the standard examples option), then

$$K(S) = \{baaaab, aaaaaab, baaaaaa, aaaaaaaa\}$$

4.12 Variety tests

Tests whether the semigroup belongs to a few standard varieties. These are the following varieties of finite semigroups

(1) commutative semigroups,
(2) idempotent semigroups,
(3) nilpotent semigroups,
(4) aperiodic semigroups,
(5) groups,
(6) $\mathcal{R}$-trivial semigroups (**R**),
(7) $\mathcal{L}$-trivial semigroups (**L**),
(8) $\mathcal{J}$-trivial semigroups,
(9) semigroups with commuting idempotents (**Ecom**),
(10) block-groups (**BG**),
(11) semigroups in which regular elements are idempotent (**DA**),
(12) the join of the varieties **R** and **L**. ($\mathbf{R} \vee \mathbf{L}$),

4.13 Do another computation

Ready for another run ?

4.14 Quit Semigroupe

The end !

5 Standard examples

Semigroupe offers a variety of examples. Trying these examples is an easy way to become familiar with the software.

5.1 Symmetric group S_n

A permutation on $\{1, 2, \ldots, n\}$ is a bijection from $\{1, 2, \ldots, n\}$ into itself. The set of all permutations on $\{1, 2, \ldots, n\}$ is a group, called the *symmetric group* on n elements under the multiplication defined by

$$fg = g \circ f$$

The symmetric group on n elements is generated by the two permutations

$$a = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 2 & 3 & \ldots & n & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 2 & 1 & 3 & \ldots & n \end{pmatrix}$$

The symmetric group $\mathfrak{S}_n$ has $n!$ elements.

5.2 Transformation monoid T_n

A transformation on $\{1, 2, \ldots, n\}$ is a total function from $\{1, 2, \ldots, n\}$ into itself. The set of all transformations on $\{1, 2, \ldots, n\}$ is a monoid, under the multiplication defined by

$$fg = g \circ f$$

This monoid, denoted by T_n, is generated by the three transformations

$$a = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 2 & 3 & \ldots & n & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 2 & 1 & 3 & \ldots & n \end{pmatrix}$$

$$c = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 1 & 2 & \ldots & n-1 & 1 \end{pmatrix}$$

The monoid T_n has n^n elements.

5.3 Monoid of partial functions F_n

The set of all partial functions from $\{1, 2, \ldots, n\}$ into itself a monoid, under the multiplication defined by

$$fg = g \circ f$$

This monoid, denoted by F_n, is generated by the four transformations

$$a = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 2 & 3 & \ldots & n & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 2 & 1 & 3 & \ldots & n \end{pmatrix}$$

$$c = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 1 & 2 & \ldots & n-1 & 1 \end{pmatrix} \qquad d = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 1 & 2 & \ldots & n-1 & - \end{pmatrix}$$

The monoid F_n has $(n+1)^n$ elements.

5.4 Monoid of injective partial functions I_n

The set of all injective functions from $\{1, 2, \ldots, n\}$ into itself a monoid, under the multiplication defined by

$$fg = g \circ f$$

This monoid, denoted by I_n, is generated by the transformations

$$a = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 2 & 3 & \ldots & n & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ 2 & 1 & 3 & \ldots & n \end{pmatrix}$$

$$c = \begin{pmatrix} 1 & 2 & \ldots & n-1 & n \\ 1 & 2 & \ldots & n-1 & - \end{pmatrix}$$

5.5 Monoid of the multiplication by n (in inverse binary)

This monoid is generated by the transformations a and b defined on $\{1, \ldots, n\}$ as follows:

$$q{\cdot}a = \begin{cases} q/2 & \text{if } q \text{ is even} \\ (q-1)/2 & \text{if } q \text{ is odd} \end{cases}$$

$$q{\cdot}b = \begin{cases} n + q - 1/2 & \text{if } q \text{ is even} \\ n + q/2 & \text{if } q \text{ is odd} \end{cases}$$

5.6 Monoid RB_n generated by the regular relations on $\{1, ..., n\}$

This monoid, denoted by RB_n contains all regular elements of B_n. It is generated by the four relations

$$a = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ 2 & 1 & 3 & \dots & n \end{pmatrix}$$

$$c = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 1 & 2 & \dots & n-1 & \{n, 1\} \end{pmatrix} \qquad d = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 1 & 2 & \dots & n-1 & - \end{pmatrix}$$

5.7 Monoid of order preserving functions on $\{1, ..., n\}$

The set of all order preserving functions from $\{1, 2, \dots, n\}$ into itself is a monoid, under the multiplication defined by

$$fg = g \circ f$$

This monoid, denoted by O_n, can be generated by n transformations of the form

$$a_i = \begin{pmatrix} 1 & 2 & \dots & i-1 & i & i+1 & \dots & n \\ 1 & 2 & \dots & i-1 & i+1 & i+1 & \dots & n \end{pmatrix}$$

for $0 \leqslant i \leqslant n-1$ and

$$a_0 = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 1 & 1 & \dots & n-2 & n-1 \end{pmatrix}$$

The monoid O_n has $\binom{2n-1}{n}$ elements.

5.8 Monoid of injective order preserving partial functions

The set of all order preserving partial functions from $\{1, 2, \dots, n\}$ into itself is a monoid, under the multiplication defined by

$$fg = g \circ f$$

This monoid, denoted by POI_n, can be generated by n transformations of the form

$$a_i = \begin{pmatrix} 1 & 2 & \dots & (n-i)-2 & (n-i)-1 & (n-i) & (n-i)+1 & \dots & n \\ 1 & 2 & \dots & (n-i)-2 & (n-i) & - & (n-i)+1 & \dots & n \end{pmatrix}$$

for $0 \leqslant i \leqslant n-1$. For $n = 4$, these generators are

$$a = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & - \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & - & 4 \end{pmatrix}$$

$$c = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & - & 3 & 4 \end{pmatrix} \qquad d = \begin{pmatrix} 1 & 2 & 3 & 4 \\ - & 1 & 2 & 3 \end{pmatrix}$$

The monoid POI_n has $\binom{2n}{n}$ elements.

5.9 Monoid $POPI_n$

This monoid is generated by the two partial functions

$$a = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & \dots & n-2 & n-1 & n \\ 1 & 2 & \dots & n-2 & n & - \end{pmatrix}$$

and contains $1 + \frac{n}{2}\binom{2n}{n}$ elements.

5.10 Group $\mathbb{Z}/n\mathbb{Z}$

This is the well-known cyclic group of order n, generated by the circular permutation

$$a = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & 1 \end{pmatrix}$$

5.11 Brandt semigroup BA_n

There are several equivalent definitions of this semigroup BA_n, called the Brandt aperiodic semigroup of dimension n. It is the syntactic semigroup of the language $(a_1a_2\cdots a_n)^+$ on the alphabet $\{a_1, a_2, \cdots, a_n\}$. It is also the semigroup of all square matrices of size n with 0-1 entries having at most one non-zero entry, under the usual multiplication of matrices. For instance,

$$BA_2 = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \right\}$$

Finally, it can be shown that BA_n is the transformation semigroup generated by the n partial functions a_i $(1 \leqslant i \leqslant n)$ defined by

$$a_i = \begin{pmatrix} 1 & 2 & \dots & i-1 & i & i+1 & \dots & n \\ - & - & \dots & - & i+1 & - & \dots & - \end{pmatrix}$$

where the value of $i+1$ is taken modulo n in the range $\{1, 2, \dots, n\}$. In particular, $n{\cdot}a_n = 1$. The size of BA_n is $n^2 + 2$.

5.12 Brandt monoid BA_n

This is the same semigroup as in the previous section, with an identity adjoined.

5.13 Monoid of triangular Boolean matrices of size $n \times n$

This monoid is generated by $\frac{n(n+1)}{2}$ boolean matrices : the $\frac{n(n-1)}{2}$ generators of U_n (see next subsection) and the n "subidentities" obtained from the identity matrix by replacing exactly one diagonal entry by a zero. For $n = 4$, these four extra matrices are

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

5.14 Monoid of unitriangular Boolean matrices of size $n \times n$

A Boolean matrix is said to be unitriangular if all its diagonal entries are ones and its subdiagonal entries are zeroes. The set U_n of all $n \times n$ unitriangular Boolean matrices form a monoid under the product of Boolean matrices. This monoid is generated by the $\frac{n(n-1)}{2}$ unitriangular matrices $U_{i,j}$ $(1 \leqslant i < j \leqslant n)$ having ones on the diagonal and exactly one extra one in position (i, j). For $n = 4$, these matrices are

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The monoid U_n is $\mathcal{J}$-trivial and contains $2^{\frac{n(n-1)}{2}}$ elements.

5.15 An ordered syntactic monoid

This is the monoid generated by the following generators:

	1	2	3
a	2	3	0
b	0	1	2

The initial state and unique final state is 1. The output generated by `Semigroupe` is given in Section 6.

5.16 Syntactic monoid of $(a(a\cdots(a(ab)^*b)^*\cdots b)^*b)^*$ (n times)

This monoid is generated by the two transformations

$$a = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & - \end{pmatrix} \qquad b = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ - & 1 & 2 & \dots & n-1 \end{pmatrix}$$

and contains $1 + \frac{n(n+1)(2n+1)}{6}$ elements.

5.17 Monoid generated by the matrices $\binom{1\ 0}{1\ 1}$ and $\binom{1\ 1}{0\ 1}$

The entries of these matrices belong to the semiring $\mathbb{Z}_{t,p}$, where the threshold t and the period p are specified in the next dialog.

5.18 A semigroup in LJ but not in $\mathbf{B}_1$

The variety $\mathbf{B}_1$ is the variety of semigroups corresponding to the so-called *dot-depth one* languages. These languages are boolean combinations of subsets of A^+ the form $u_0A^*u_1A^*u_2\cdots A^*u_k$, where u_0, u_1, u_2, $\ldots$, u_k are words. It was conjectured for some time that $\mathbf{B}_1$ was equal to $\mathbf{LJ}$, the variety of locally $\mathcal{J}$-trivial semigroups, before Knast found a counterexample.

Knast's counterexample is generated by the four transformations

$$a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 4 & - & - & - & - & 4 \end{pmatrix} \quad b = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ - & 2 & - & 2 & 5 & 5 & - \end{pmatrix}$$

$$c = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ - & - & 3 & 7 & - & 3 & 7 \end{pmatrix} \quad d = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ - & 6 & 6 & - & 6 & - & - \end{pmatrix}$$

This semigroup has 31 elements.

5.19 An example of transition semigroup

This example computes the transition semigroup generated by the two transformations

$$a = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,5\,3\,3\,5 \end{pmatrix} \qquad b = \begin{pmatrix} 1\,2\,3\,4\,5 \\ -\,2\,4\,2\,2 \end{pmatrix}$$

This semigroup has 8 elements.

5.20 An example of monoid of Boolean matrices

This example computes the monoid generated by the Boolean matrices

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

This monoid has 7 elements.

5.21 An example of semigroup of matrices with integer entries

This example computes the semigroup generated by the two matrices with entries in the semiring $\mathbb{Z}_{1,2}$.

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

This semigroup has 37 elements.

5.22 An example of semigroup of matrices with Max-Plus entries

This example computes the semigroup generated by the following matrices

$$\begin{pmatrix} 0 & -4 \\ -4 & -1 \end{pmatrix}, \begin{pmatrix} 0 & -3 \\ -3 & -1 \end{pmatrix}$$

This monoid has 37 elements.

6 An example computed by `Semigroupe`

This is the example
Monoid generated by the following generators:

	1	2	3
a	2	3	0
b	0	1	2

This monoid has a zero: $a^3 = 0$
Number of elements: 15
Number of relations: 9
Maximal length of the words: 4
Number of $\mathcal{D}$-classes: 4
Number of $\mathcal{R}$-classes: 7
Number of $\mathcal{L}$-classes: 7
Number of $\mathcal{H}$-classes: 15
Elements:

	1	2	3
$*\ 1$	1	2	3
a	2	3	0
b	0	1	2
a^2	3	0	0
$*\ ab$	1	2	0
$*\ ba$	0	2	3
b^2	0	0	1
$*\ a^3$	0	0	0
a^2b	2	0	0
ab^2	0	1	0
ba^2	0	3	0
b^2a	0	0	2
$*\ a^2b^2$	1	0	0
$*\ ab^2a$	0	2	0
$*\ b^2a^2$	0	0	3

Relations:

$$\begin{array}{ccccc} aba = a & bab = b & b^2b = 0 & a^3 = 0 & ba^2a = 0 \\ ba^2b = ab^2a & a^2b^2a = a^2b & ab^2aa = ba^2 & & \end{array}$$

Idempotents:

$$E(S) = \{1, ab, ba, a^3, a^2b^2, ab^2a, b^2a^2\}$$

Syntactic preorder:

$$\begin{array}{ccccc} 1 < ab & 1 < ba & a < a^2b & a < ba^2 & b < ab^2 \\ b < b^2a & a^2 < a^3 & ab < a^2b^2 & ab < ab^2a & ba < ab^2a \\ ba < b^2a^2 & b^2 < a^3 & a^2b < a^3 & ab^2 < a^3 & ba^2 < a^3 \\ b^2a < a^3 & a^2b^2 < a^3 & ab^2a < a^3 & b^2a^2 < a^3 & \end{array}$$

Minimal ideal:

$$I = \{a^3\}$$

$\mathcal{D}$-classes:

$^{*}1$

$^{*}ba$	b
a	$^{*}ab$

$^{*}b^2a^2$	b^2a	b^2
ba^2	$^{*}ab^2a$	ab^2
a^2	a^2b	$^{*}a^2b^2$

$^{*}a^3$

This monoid is not commutative, since $ab \neq ba$. This monoid is not idempotent, since $u \neq u^2$ for $u = a$. This monoid is not nilpotent. This monoid is aperiodic. This monoid is not a group. This monoid is neither $\mathcal{R}$-trivial nor $\mathcal{L}$-trivial. The idempotents of this monoid commute. This monoid is a block-group. This monoid is not in **DA** since the identity $(xy)^\omega(yx)^\omega(xy)^\omega = (xy)^\omega$ is not satisfied for $x = a$ and $y = b$. This monoid is not in **DS**, since the identity $((xy)^\omega(yx)^\omega(xy)^\omega)^\omega = (xy)^\omega$ is not satisfied for $x = a$ and $y = b$. This monoid is not in $\mathbf{R} \vee \mathbf{L}$ since the identity $(xy)^\omega x(zx)^\omega = (xy)^\omega(zx)^\omega$ is not satisfied for $x = a$, $y = b$ and $z = b$.

7 Finite semigroups

A *semigroup* is a set equipped with an internal associative operation which is usually written in a multiplicative form. A *monoid* is a semigroup with an identity element (usually denoted by 1). If S is a semigroup, S^1 denotes the monoid equal to S if S has an identity element and to $S \cup \{1\}$ otherwise. In the latter case, the multiplication on S is extended by setting $s1 = 1s = s$ for every $s \in S^1$. If S is a semigroup, the operation $*$ defined on S by $s * t = ts$ defines a new semigroup, called the *reverse* of S.

An element e of a semigroup S is *idempotent* if $e^2 = e$. A semigroup is *idempotent* if all its elements are idempotent. In this chapter, we will mostly use finite semigroups, in which idempotents play a key role. In particular, if s is an element of a finite semigroup, the subsemigroup generated by s contains a unique idempotent and a unique maximal subgroup, whose identity is the unique idempotent.

An *ideal* of a semigroup S is a non empty subset I of S such that, for all $x \in I$ and for all $s, t \in S$, $sxt \in I$. If S is finite, the intersection of all ideals is still an ideal, called the *minimal ideal* of S.

If s is an element of a finite semigroup, the unique idempotent power of s is denoted s^ω. If e is an idempotent of a finite semigroup S, the set

$$eSe = \{ese \mid s \in S\}$$

is a subsemigroup of S, called the *local subsemigroup* associated with e. This semigroup is in fact a monoid, since e is an identity in eSe.

A finite semigroup S is said to satisfy *locally* a property $\mathcal{P}$ if every local subsemigroup of S satisfies $\mathcal{P}$. For instance, S is *locally trivial* if, for every idempotent $e \in S$ and every $s \in S$, $ese = e$.

A *zero* is an element 0 such that, for every $s \in S$, $s0 = 0s = 0$. It is a routine exercise to see that there is at most one zero in a semigroup. A non-empty finite semigroup that contains a zero and no other idempotent is called *nilpotent*.

A *semiring* is a set k equipped with an addition and a multiplication. It is a commutative monoid with identity 0 for the addition and a monoid with identity 1 for the multiplication. Multiplication is distributive over addition and 0 satisfies $0x = x0 = 0$ for every $x \in k$. The simplest example of a semiring which is not a ring is the Boolean semiring $\mathbb{B} = \{0, 1\}$ defined by $0 + 0 = 0$, $0 + 1 = 1 + 1 = 1 + 0 = 1$, $1.1 = 1$ and $1.0 = 0.0 = 0.1 = 0$. Several other semirings are used by Semigroupe:

(1) The semiring $(\mathbb{N} \cup \{-\infty\}, \max, +)$,
(2) The semiring $(\mathbb{N} \cup \{+\infty\}, \min, +)$,
(3) The semiring $\{-\infty, 0, 1, \ldots, t\}, \max, +)$, for some threshold t,
(4) The semiring $\{0, 1, \ldots, t, +\infty\}, \min, +)$, for some threshold t,
(5) The semiring $(\mathbb{Z}, +, x)$,
(6) The semiring $\mathbb{N}_{t,p}$, for some threshold t and some period p: this semiring is the quotient of $\mathbb{N}$ under the congruence $t = t + p$. Thus $\mathbb{N}_{t,p} = \{0, 1, \ldots, t, t + 1, \ldots, t + p - 1\}$

For each $n > 0$, the set $M_n(k)$ of n by n matrices with entries in k is again a semiring for addition and multiplication of matrices induced by the operations in k.

7.1 Green's relations

Green's relations on a semigroup S are defined as follows. If s and t are elements of S, we set

$$\begin{array}{ll}
s \mathrel{\mathcal{L}} t & \text{if there exist } x, y \in S^1 \text{ such that } s = xt \text{ and } t = ys,\\
s \mathrel{\mathcal{R}} t & \text{if there exist } x, y \in S^1 \text{ such that } s = tx \text{ and } t = sy,\\
s \mathrel{\mathcal{J}} t & \text{if there exist } x, y, u, v \in S^1 \text{ such that } s = xty \text{ and } t = usv.\\
s \mathrel{\mathcal{H}} t & \text{if } s \mathrel{\mathcal{R}} t \text{ and } s \mathrel{\mathcal{L}} t.
\end{array}$$

For finite semigroups, these four equivalence relations can be represented as follows. The elements of a given $\mathcal{R}$-class (resp. $\mathcal{L}$-class) are represented in a row (resp. column). The intersection of an $\mathcal{R}$-class and an $\mathcal{L}$-class is an $\mathcal{H}$-class. Each $\mathcal{J}$-class is a union of $\mathcal{R}$-classes (and also of $\mathcal{L}$-classes). It is not obvious to see that this representation is consistent: it relies in particular on the fact that, in finite semigroups, the relations $\mathcal{R}$ and $\mathcal{L}$ commute. Thus one can introduce a fifth relation

$$\mathcal{D} = \mathcal{R} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{R}$$

One can show that, in a finite semigroup, $\mathcal{D} = \mathcal{J}$. In other words, $s \mathrel{\mathcal{J}} t$ if and only if there exists $r \in S$ such that $s \mathrel{\mathcal{R}} r$ and $r \mathrel{\mathcal{R}} t$, or equivalently, if there exists $u \in S$ such that $s \mathrel{\mathcal{L}} u$ and $u \mathrel{\mathcal{L}} t$.

The presence of an idempotent in an $\mathcal{H}$-class is indicated by a star. One can show that each $\mathcal{H}$-class containing an idempotent e is a subsemigroup of S, which is in fact a group with identity e. Furthermore, all $\mathcal{R}$-classes (resp. $\mathcal{L}$-classes) of a given $\mathcal{J}$-class have the same number of elements.

*a_1, a_2	*a_3, a_4	a_5, a_6
b_1, b_2	*b_3, b_4	*b_5, b_6

A $\mathcal{J}$-class.

In this figure, each row is an $\mathcal{R}$-class and each column is an $\mathcal{L}$-class. There are 6 $\mathcal{H}$-classes and 4 idempotents. Each idempotent is the identity of a group of order 2.

A $\mathcal{J}$-class containing an idempotent is called *regular*. One can show that in a regular $\mathcal{J}$-class, every $\mathcal{R}$-class and every $\mathcal{L}$-class contains an idempotent.

A semigroup S is *$\mathcal{L}$-trivial* (resp. *$\mathcal{R}$-trivial, $\mathcal{J}$-trivial, $\mathcal{H}$-trivial*) if two elements of S which are $\mathcal{L}$-equivalent (resp. $\mathcal{R}$-equivalent, $\mathcal{J}$-equivalent, $\mathcal{H}$-equivalent) are equal. See [3, 4] for more details.

7.2 Ordered semigroups

See [5, 6] for relevant definitions. A relation $\mathcal{R}$ on a semigroup S is *stable on the right* (resp. *left*) if, for every $x, y, z \in S$, $x \mathrel{\mathcal{R}} y$ implies $xz \mathrel{\mathcal{R}} yz$ (resp. $zx \mathrel{\mathcal{R}} zy$). A relation is *stable* if it is stable on the right and on the left. An *ordered semigroup* is a semigroup S equipped with a stable order relation $\leqslant$ on S. Ordered monoids are defined analogously.

An *order ideal* I of an ordered monoid $(M, \leqslant)$ is a subset of M such that if $x \in I$ and $y \leqslant x$ then $y \in I$.

Let A^* be a free monoid. Given a language P of A^* we define the *syntactic congruence* $\sim_P$ and the *syntactic preorder* $\leqslant_P$ as follows:

(1) $u \sim_P v$ if and only if for all $x, y \in A^*$, $xvy \in P \Leftrightarrow xuy \in P$,

(2) $u \leqslant_P v$ if and only if for all $x, y \in A^*, xvy \in P \Rightarrow xuy \in P$.

The monoid $A^*/{\sim_P}$ is called the *syntactic monoid* of P, and is denoted by $M(P)$. The monoid $A^*/{\sim_P}$, ordered with the stable order relation induced by $\leqslant_P$ is called the *ordered syntactic monoid* of P. The syntactic (ordered) monoid of a rational language is finite.

7.3 Varieties

A *variety of semigroups* is a class of semigroups closed under taking subsemigroups, quotients and direct products. A *variety of finite semigroups*, or *pseudovariety*, is a class of finite semigroups closed under taking subsemigroups, quotients and finite direct products. Varieties of ordered semigroups and varieties of finite ordered semigroups are defined analogously. Varieties of semigroups or ordered semigroups will be denoted by boldface capital letters, like $\mathbf{V}$.

An important variety of monoids is the variety of aperiodic monoids, defined by the identity $x^\omega = x^{\omega+1}$. Thus, a finite monoid M is *aperiodic* if and only if, for each $x \in M$, there exists $n \geqslant 0$ such that $x^n = x^{n+1}$. This also means that the cyclic subgroup of the submonoid generated by any element x is trivial or that in M the Green relation $\mathcal{H}$ is the equality relation. It follows that a monoid is aperiodic if and only if it is group-free: every subsemigroup which happens to be a group has to be trivial.

Another important variety is the variety $\mathbf{G}$ of all finite groups. This is indeed a variety because a submonoid of a finite group is a group.

7.4 Kernel

Recall that a relational morphim between monoids M and N is a relation $\tau : M \to N$ such that:

(1) $\tau(m)\tau(n) \subset \tau(mn)$ for all $m, n \in M$,

(2) $\tau(m)$ is non-empty for all $m \in M$,

(3) $1 \in \tau(1)$

Equivalently, τ is a relation whose graph

$$\text{graph}(\tau) = \{\ (m, n) \mid n \in m\tau\ \}$$

is a submonoid of $M \times N$ that projects onto M. The *kernel* of M, denoted by $K(M)$, is the intersection of the submonoids $\tau^{-1}(1)$ over all relational morphims s $\tau : M \to G$ into a group. this definition is not constructive, but a deep result of Ash [1] gives an algorithm to compute $K(M)$. The kernel of M

is the smallest submonoid of M closed under weak conjugation: if m is a *weak inverse* of n, that is, if $mnm = m$, then, for every $k \in K(M)$, $mkn \in K(M)$ and $nkm \in K(M)$.

The kernel was introduced as a tool to study decidability problems related to Malcev products. Let $\mathbf{V}$ be a variety of finite monoids. Let

$$\mathbf{V} \textcircled{M} \mathbf{G} = \{\ M \mid \text{ There is a relational morphism } \tau \text{ from } M \\ \text{onto a group } G \text{ such that } \tau^{-1}(1) \in \mathbf{V}\}$$

Then $\mathbf{V} \textcircled{M} \mathbf{G}$ is a variety, called the *Mal'cev product* of $\mathbf{V}$ and $\mathbf{G}$. The following consequence of Ash's theorem shows that if $\mathbf{V}$ is decidable, then $\mathbf{V} \textcircled{M} \mathbf{G}$ is decidable.

Theorem 7.1 *Let M be a monoid and let $\mathbf{V}$ be a variety. Then $M \in \mathbf{V} \textcircled{M} \mathbf{G}$ if and only if $K(M) \in \mathbf{V}$.*

References

[1] C.J. Ash, Inevitable Graphs: A proof of the type II conjecture and some related decision procedures, *Int. Jour. Alg. and Comp.* **1** (1991) 127–146.

[2] V. Froidure et J.-E. Pin, Algorithms for computing finite semigroups, in *Foundations of Computational Mathematics*, F. Cucker et M. Shub (éd.), Springer Verlag, Berlin, (1997), 112–126.

[3] G. Lallement, *Semigroups and combinatorial applications*, Wiley, New York, 1979.

[4] J.-E. Pin, *Variétés de langages formels*, Masson, Paris, 1984. English translation: *Varieties of formal languages*, Plenum, New-York, 1986.

[5] J.-E. Pin, A variety theorem without complementation, *Russian Mathematics (Izvestija vuzov.Matematika)* **39**, (1995), 80–90.

[6] J.-E. Pin, Syntactic semigroups, Chapter 10 in *Handbook of language theory*, G. Rozenberg and A. Salomaa eds., Springer, 1997, 679–746.

Index