Part I

Group radical

Group radical

The group radical of a finite monoid M is the smallest submonoid $D(M)$ of M containing the idempotents and closed under weak conjugation: if $sts = s$ and $d \in D(M)$, then $sdt, tds \in D(M)$.

Computation of the radical

Initialisation : $D(M) = E(M)$

For each d in $D(S)$

For each weakly conjugate pair (s,t)
  add sdt and tds to $D(S)$
add $D(S)d$ to $D(S)$.

Time complexity in $0(|S|^3)$.

Part II

Syntactic ordered monoid

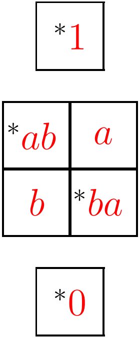
If P is a subset of a monoid M, the syntactic preorder $\leqslant_P$ is defined on M by $u \leqslant_P v$ iff, for all $x, y \in M$,

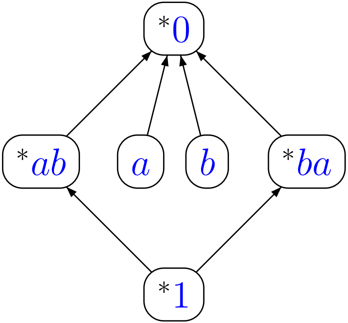
$$xvy \in P \Rightarrow xuy \in P$$

Denote by $\bar{P}$ the complement of P. Then $u \not\leqslant_P v$ iff there exist $x, y \in M$ such that

$$xuy \in \bar{P} \text{ and } xvy \in P$$

The syntactic ordered monoid of ab in B_2^1





An algorithm for the syntactic preorder

Let G be the graph with $M \times M$ as set of vertices and edges of the form $(ua, va) \to (u, v)$ or $(au, av) \to (u, v)$.

We have seen that $u \not\leqslant_P v$ iff there exist $x, y \in M$ such that

$$xuy \in \bar{P} \text{ and } xvy \in P$$

Therefore, $u \not\leqslant_P v$ iff the vertex (u, v) is reachable in G from some vertex of $\bar{P} \times P$.

The algorithm (2)

(1) Label each vertex (u, v) as follows:

$$\begin{cases} (0,1) & \text{if } u \notin P \text{ and } v \in P \quad [u \not\leqslant_P v] \\ (1,0) & \text{if } u \in P \text{ and } v \notin P \quad [v \not\leqslant_P u] \\ (1,1) & \text{otherwise} \end{cases}$$

(2) Do a depth first search (starting from each vertex labeled by $(0, 1)$) and set to 0 the first component of the label of all visited vertices.

Constraint propagation

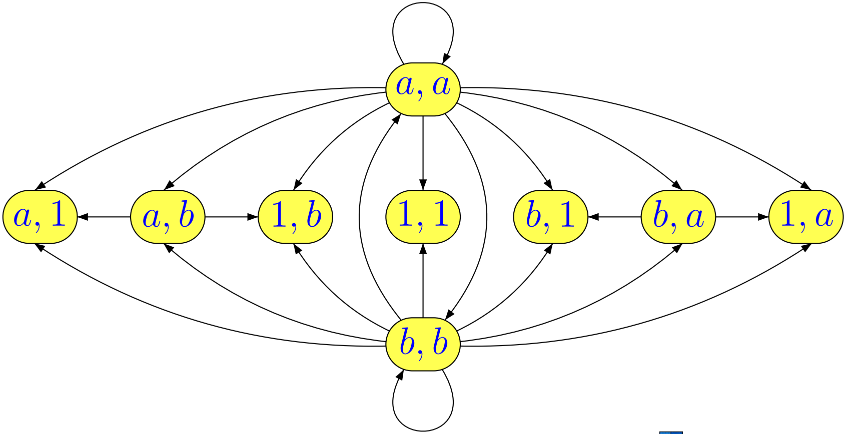
(3) Do a depth first search (starting from each vertex labeled by $(0, 0)$ or $(1, 0)$) and set to 0 the second component of the label of all visited vertices.

(4) The label of each vertex now encodes the syntactic preorder of P as follows:

$$\begin{cases} (1,1) & \text{if } u \sim_P v \\ (1,0) & \text{if } u \leqslant_P v \\ (0,1) & \text{if } v \leqslant_P u \\ (0,0) & \text{if } u \text{ and } v \text{ are incomparable} \end{cases}$$

Computation of the syntactic preorder

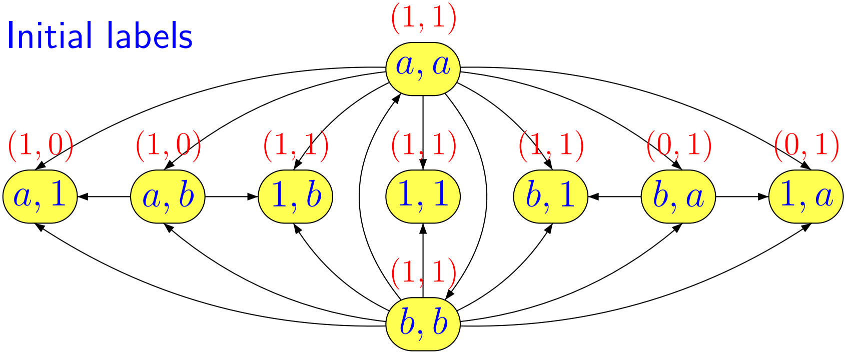
Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.



Computation of the syntactic preorder

Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.

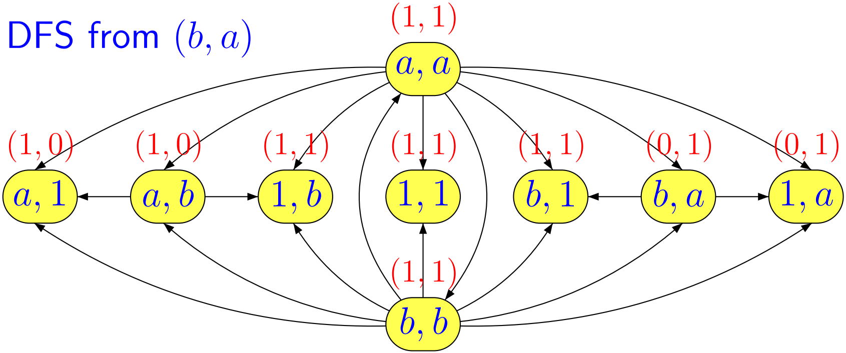
Initial labels



Computation of the syntactic preorder

Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.

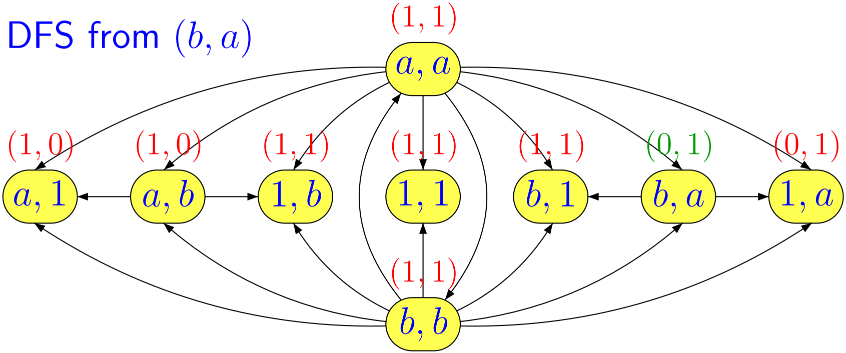
DFS from (b, a)



Computation of the syntactic preorder

Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.

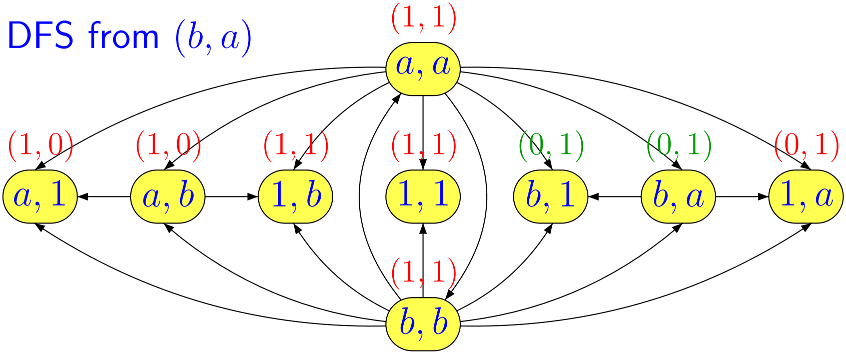
DFS from (b, a)



Computation of the syntactic preorder

Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.

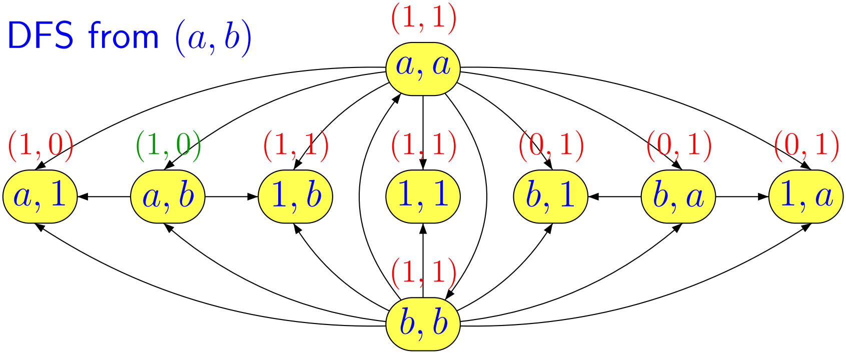
DFS from (b, a)



Computation of the syntactic preorder

Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.

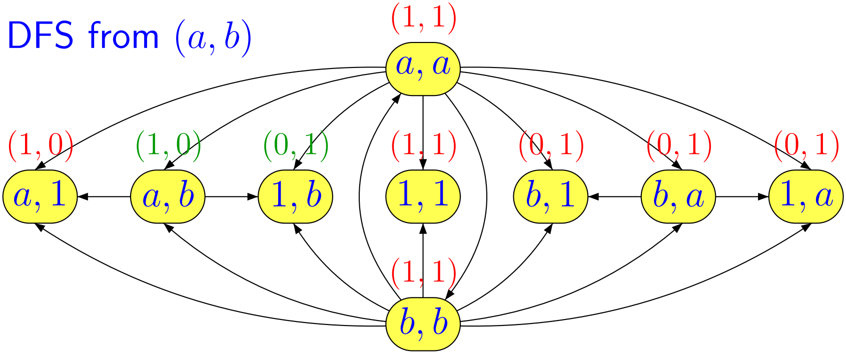
DFS from (a, b)



Computation of the syntactic preorder

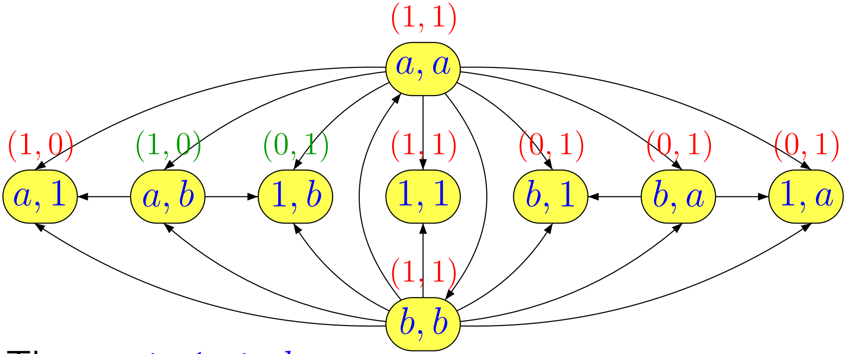
Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.

DFS from (a, b)



Computation of the syntactic preorder

Let $M = \{1, a, b\}$ with $aa = ba = a$ and $ab = bb = b$. Let $P = \{a\}$.



Thus $a \leqslant_P 1 \leqslant_P b$

Complexity of the algorithm

The syntactic preorder can be computed in $O(|A||M|^2)$ time and space.

Aperiodicity

Theorem (Cho-Huynh 1991)

Testing aperiodicity of a deterministic n-state automaton is P-space complete.

Proposition

One can test in $O(|A||S|)$-time whether an A-generated finite semigroup S is aperiodic.

It suffices to test whether the $\mathcal{H}$-classes are trivial.

Other varieties

Proposition

One can test in $O(|A||S|)$*-time whether an* A*-generated finite semigroup* S *is* $\mathcal{R}$*-trivial [*$\mathcal{L}$*-trivial,* $\mathcal{J}$*-trivial, commutative, idempotent, nilpotent, a group, a block-group].*

Testing a set of identities

This is a difficult problem for several reasons:

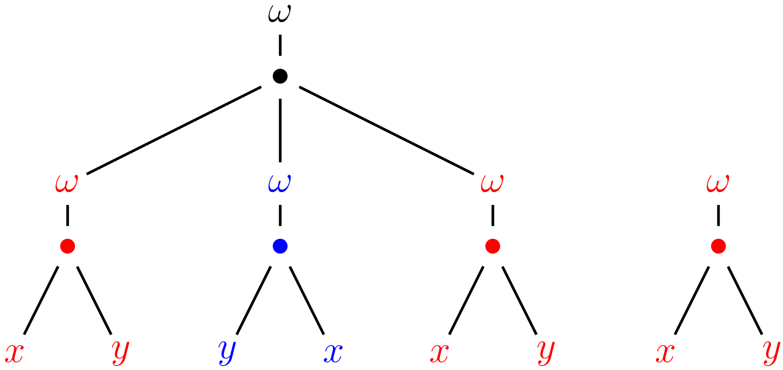
• It may happen that testing whether a set of identities is satisfied is much easier than testing whether any of the individual identities is satisfied.

• Identities for finite semigroups are profinite identities. The operations x^ω and $x^{\omega-1}$ are frequently needed, but other operators might be needed.

• There might be some tricky tree pattern-matching problems to solve.

Tree pattern-matching problems

A simple example: the variety $\mathbf{DS}$ is defined by the identity $((xy)^\omega(yx)^\omega(xy)^\omega)^\omega = (xy)^\omega$



Semigroup theory might help. . .

Proposition

One can test in $O(|A||S|)$-time whether an A-generated finite semigroup S belongs to $\mathbf{DS}$.

Indeed, a semigroup belongs to $\mathbf{DS}$ iff every regular $\mathcal{D}$-class is union of groups. Therefore, it suffices to test whether the number of regular $\mathcal{H}$-classes is equal to the number of idempotents.

Part III

New directions

A stamp is a morphism from a finitely generated free monoid onto a finite monoid. An ordered stamp is a stamp onto an ordered monoid.

$$\varphi : A^* \to M$$

Stable subsemigroup

Let $\varphi : A^* \to M$ be a stamp and let $Z = \varphi(A)$. Then Z belongs to the monoid $\mathcal{P}(M)$ of subsets of M.

Since $\mathcal{P}(M)$ is finite, Z has an idempotent power. The stability index of φ is the least positive integer such that $\varphi(A^s) = \varphi(A^{2s})$.

The set $\varphi(A^s)$ is a subsemigroup of M called the stable semigroup of φ and the monoid $\varphi(A^s) \cup \{1\}$ is called the stable monoid of φ.

Applications to logic

Theorem (McNaughton-Paper 1971, Schützenberger 1965)

A language is $\mathbf{FO}[<]$*-definable iff its syntactic semigroup is aperiodic.*

Theorem (Barrington, Compton, Straubing, Thérien 1992)

A language is $\mathbf{FO}[< + \mathrm{MOD}]$*-definable iff the stable semigroup of its syntactic stamp is aperiodic.*

A bit of logic

To each nonempty word u is associated a structure

$$\mathcal{M}_u = (\{1, 2, \ldots, |u|\}, <, (\mathbf{a})_{a \in A})$$

where $\mathbf{a}$ is interpreted as the set of integers i such that the i-th letter of u is an a, and $<$ as the usual order on integers.

If $u = abbaab$, then $\mathrm{Dom}(u) = \{1, 2, 3, 4, 5, 6\}$, $\mathbf{a} = \{1, 4, 5\}$ and $\mathbf{b} = \{2, 3, 6\}$.

Modular predicates

Let $d > 0$ and $r \in \mathbb{Z}/d\mathbb{Z}$. We define two new symbols (the modular symbols):

- The unary symbol MOD_r^d:

$$\mathrm{MOD}_r^d(n) = \{i < n \mid i \bmod d = r\}$$

- A constant symbol m for the last position in a word

Fragments of first order logic

$\mathbf{FO}[<]$ denotes the set of first order formulas in the signature $\{<, (\mathbf{a})_{a \in A}\}$.

$\mathbf{FO}[< + \text{MOD}]$ denotes the logic obtained by adjoining all modular symbols.

Fragments of first order logic

$\mathbf{FO}[<]$ denotes the set of first order formulas in the signature $\{<, (\mathbf{a})_{a \in A}\}$.

$\mathbf{FO}[< + \text{MOD}]$ denotes the logic obtained by adjoining all modular symbols.

$\mathbf{\Sigma_1}$ denotes the set of existential formulas:

$$\exists x_1 \cdots \exists x_n \ \varphi(x_1, \ldots, x_n)$$

where φ is quantifier-free.

$\mathcal{B}\mathbf{\Sigma_1}$ denotes the set of Boolean combinations of $\mathbf{\Sigma_1}$-formulas.

Some examples

The formula $\exists x\ \mathbf{a}x$ is interpreted as:

> *There exists an integer x such that, in u, the letter in position x is an a.*

This defines the language A^*aA^*.

Some examples

The formula $\exists x \ \mathbf{a}x$ is interpreted as:

> *There exists an integer x such that, in u, the letter in position x is an a.*

This defines the language A^*aA^*.

The formula $\exists x \ \exists y \ (x < y) \wedge \mathbf{a}x \wedge \mathbf{b}y$ defines the language $A^*aA^*bA^*$.

Some examples

The formula $\exists x\ \mathbf{a}x$ is interpreted as:

> *There exists an integer x such that, in u, the letter in position x is an a.*

This defines the language A^*aA^*.

The formula $\exists x\ \exists y\ (x < y) \wedge \mathbf{a}x \wedge \mathbf{b}y$ defines the language $A^*aA^*bA^*$.

The formula $\exists x\ \forall y\ (x < y) \vee (x = y) \wedge \mathbf{a}x$ defines the language aA^*.

Simple languages

A simple language is a language of the form

$$A^*a_1A^*a_2A^* \cdots \; a_kA^*$$

where $d > 0$, $k \geqslant 0$ and $a_1, a_2, \ldots, a_k \in A$.

A modular simple language is a language of the form

$$(A^d)^*a_1(A^d)^*a_2(A^d)^* \cdots \; a_k(A^d)^*$$

where $d > 0$, $k \geqslant 0$ and $a_1, a_2, \ldots, a_k \in A$.

Logical description of simple languages

The language $A^*a_1A^*a_2A^* \cdots\ a_kA^*$ can be defined by the Σ_1-formula

$$\exists x_1\ \ldots\ \exists x_k\ (x_1 < \ldots < x_k) \wedge (\mathbf{a}_1 x_1 \wedge \cdots \wedge \mathbf{a}_k x_k)$$

Logical description of simple languages

The language $A^*a_1A^*a_2A^* \cdots\ a_kA^*$ can be defined by the Σ_1-formula

$$\exists x_1 \ \ldots \ \exists x_k \ (x_1 < \ldots < x_k) \wedge (\mathbf{a}_1x_1 \wedge \cdots \wedge \mathbf{a}_kx_k)$$

The language $(A^d)^*a_1(A^d)^*a_2(A^d)^* \cdots\ a_k(A^d)^*$ can be defined by the Σ_1-formula

$$\exists x_1 \ \ldots \ \exists x_k \ (x_1 < \ldots < x_k) \wedge (\mathbf{a}_1x_1 \wedge \cdots \wedge \mathbf{a}_kx_k) \wedge$$
$$(\mathrm{MOD}_0^d\, x_1 \wedge \mathrm{MOD}_1^d\, x_2 \wedge \cdots \wedge \mathrm{MOD}_{k-1}^d\, x_k \wedge \mathrm{MOD}_{k-1}^d\, m)$$

First order

Theorem (McNaughton-Paper 1971, Schützenberger 1965)

A language is $\mathbf{FO}[<]$*-definable iff its syntactic semigroup is aperiodic.*

Theorem (Barrington, Compton, Straubing, Thérien 1992)

A language is $\mathbf{FO}[< + \mathrm{MOD}]$*-definable iff the stable semigroup of its syntactic stamp is aperiodic.*

Existential formulas (Σ_1)

Proposition

A language is definable in $\Sigma_1[<]$ iff it is a finite union of simple languages.

Proposition

A language is definable in $\Sigma_1[< + \mathrm{MOD}]$ iff it is a finite union of modular simple languages.

Algebraic characterization

Theorem (Thomas 1982, Perrin-Pin 1986)

A language is definable in $\mathbf{\Sigma_1}[<]$ *iff its ordered syntactic monoid satisfies the identity* $x \leqslant 1$.

Theorem (Chaubard, Pin, Straubing 2006)

A language is definable in $\mathbf{\Sigma_1}[< + \mathrm{MOD}]$ *iff the stable ordered monoid of its ordered syntactic stamp satisfies the identity* $x \leqslant 1$.

lm-morphisms

A morphism $f : A^* \to B^*$ is length-multiplying (lm for short) if there exists an integer k such that the image of each letter of A is a word of B^k.

For instance, if $A = \{a, b\}$ and $B = \{a, b, c\}$, the morphism defined by $\varphi(a) = abca$ and $\varphi(b) = cbba$ is length-multiplying.

lm-identities

Let u, v be two words on the alphabet B. A morphism $\varphi : A^* \to M$ satisfies the lm-identity $u = v$ if, for every lm-morphism $f \colon B^* \to A^*$, $\varphi \circ f(u) = \varphi \circ f(v)$.

For instance, $\varphi : A^* \to M$ satisfies the lm-identity $xyx = xy$ if for any pair of words of the same length x, y of A^*, $\varphi(xyx) = \varphi(xy)$.

lm-identities

Let u, v be two words on the alphabet B. A morphism $\varphi : A^* \to M$ satisfies the lm-identity $u = v$ if, for every lm-morphism $f\colon B^* \to A^*$, $\varphi \circ f(u) = \varphi \circ f(v)$.

For instance, $\varphi : A^* \to M$ satisfies the lm-identity $xyx = xy$ if for any pair of words of the same length x, y of A^*, $\varphi(xyx) = \varphi(xy)$.

If M is ordered, we say that φ satisfies the lm-identity $u \leqslant v$ if, for every lm-morphism $f\colon B^* \to A^*$, $\varphi \circ f(u) \leqslant \varphi \circ f(v)$.

Characterization by lm-identities

Theorem (Thomas 1982, Perrin-Pin 1986)

A language is definable in $\mathbf{\Sigma_1}[<]$ *iff its ordered syntactic monoid satisfies the identity* $x \leqslant 1$.

Theorem (Chaubard, Pin, Straubing 2006)

A language is definable in $\mathbf{\Sigma_1}[< + \mathrm{MOD}]$ *iff its ordered syntactic stamp satisfies the* lm*-identities* $x^{\omega-1}y \leqslant 1$ *and* $yx^{\omega-1} \leqslant 1$.

Boolean combination of existential formulas

Theorem (Thomas 1982)

A language is definable in $\mathcal{B}\boldsymbol{\Sigma}_{\mathbf{1}}[<]$ *iff it is a Boolean combination of simple languages.*

Theorem (Chaubard, Pin, Straubing 2006)

A language is definable in $\mathcal{B}\boldsymbol{\Sigma}_{\mathbf{1}}[< + \mathrm{MOD}]$ *iff it is a Boolean combination of modular simple languages.*

Algebraic characterization

Theorem (Simon 1972, Thomas 1982)

A language is definable in $\mathcal{B}\boldsymbol{\Sigma}_{\mathbf{1}}[<]$ *iff its syntactic monoid is* $\mathcal{J}$*-trivial.*

Theorem (Chaubard, Pin, Straubing 2006)

A language is a Boolean combination of modular simple languages iff its syntactic stamp belongs to the lm-variety $\mathbf{J} * \mathbf{MOD}$.

Derived category of a stamp $\varphi : A^* \to M$

Let $\pi_n(u) = |u| \bmod n$.

$$\begin{tikzcd} & A^* \arrow[dl, "\varphi"'] \arrow[dr, "\pi_n"] & \\ M & & \mathbb{Z}/n\mathbb{Z} \end{tikzcd}$$

Let $C_n(\varphi)$ be the category whose objects are elements of $\mathbb{Z}/n\mathbb{Z}$ and whose arrows from i to j are the triples (i, m, j) where $j - i \in \pi_n(\varphi^{-1}(m))$.

Composition is given by
$(i, m_1, j)(j, m_2, k) = (i, m_1m_2, k)$.

A decidable characterization

Theorem (Chaubard, Pin, Straubing 2006)

Let φ be a stamp of stability index s. Then φ belongs to $\mathbf{J} * \mathbf{MOD}$ *iff* $C_s(\varphi)$ *is in* $g\mathbf{J}$.

No characterization by lm-identities is known at the moment.

What would be useful in GAP 4...

- Define stamps as a basic object.

- Compute stable semigroups and monoids of stamps.

- Test for length-preserving and length-multiplying identities.

- Compute derived categories

References I

- V. Froidure and J.-E. Pin, Algorithms for computing finite semigroups, in *Foundations of Computational Mathematics*, F. Cucker et M. Shub (éd.), Berlin, 1997, pp. 112–126, Springer.
- Semigroupe, C programme, available at http://www.liafa.jussieu.fr/~jep/Logiciels/Semigroupe/semigroupe.html
- L. Chaubard, J.-E. Pin and H. Straubing, First order formulas with modular predicates, in *Proceedings of LICS'06*, 2006.