June 2013

Semigroups and one-way functions

Jean-Camille Birget

To Stuart Margolis on his 60th birthday.

CS Dept., Rutgers U. (Camden Campus), Camden, New Jersey

Goal:

Find semigroups (and groups) whose elements represent *computational devices* or *computable functions*.

1. Thompson-Higman groups and monoids:
   They represent all *finite functions* and
   acyclic digital circuits.

2. Monoids of polynomial-time computable functions:
   Their properties depend on P-vs.-NP.

   Study complexity classes through functions and semigroups (instead of only as sets of languages).

arXiv:1306.1447 [math.GR] 6 Jun 2013

Preliminary definitions

Fix a finite alphabet A.

A^* = set of all finite words over A.

View A^* as the rooted, regular, infinite, oriented **tree**, directed away from the root.

A^ω = set of all ω-words over A (the **ends** of A^*),
with the Cantor space topology.

Def. $R \subseteq A^*$ is **right ideal** iff $R\,A^* \subseteq R$.

Def. C ($\subseteq A^*$) **generates** a right ideal R iff
R is the intersection of all right ideals that contain C.

Equivalently, $R = CA^*$.

In A^ω, the open sets of the Cantor space are of the form $CA^\omega = \mathsf{ends}(CA^*)$.

Def. A right ideal R is **essential** iff
R intersects every right ideal of A^*.

I.e., $\mathsf{ends}(R)$ is *dense* in A^ω.

Def. $C \subseteq A^*$ is a **prefix code** (prefix-free code) iff no element of C is a *prefix* of another element of C.

(Shannon-Fano coding, 1948; Huffman, 1951.)

"Prefix": any initial segment of a word.

Def. A prefix code C is **maximal** iff C is not a strict subset of another prefix code.

Fact. A right ideal R has a *unique minimal* (for $\subseteq$) generating set C;
this minimum C is a prefix code.

Fact. A prefix code C is *maximal* iff CA^* is an *essential* right ideal.

Def. (end-equivalence):
For right ideals $R', R \subseteq A^*$: $R' \cong R$ iff

R' and R intersect the same right ideals iff

$\mathsf{ends}(R)$ and $\mathsf{ends}(R')$ "are the same up to density", i.e.,

$\overline{\mathsf{ends}(R)} = \overline{\mathsf{ends}(R')}$, where overlining denotes closure in the Cantor set topology.

Def. A **right ideal homomorphism** of A^* is a function $\varphi : R_1 \to A^*$ such that R_1 is a right ideal of A^*, and for all $x_1 \in R_1$ and all $w \in A^*$:

$$\varphi(x_1 w) = \varphi(x_1)\, w.$$

Notation: Domain $R_1 = \mathsf{Dom}(\varphi)$,
image set $= \mathsf{Im}(\varphi)$.

Fact. $\mathsf{Dom}(\varphi)$ and $\mathsf{Im}(\varphi)$ are right ideals.

Fact. φ acts as a continuous partial function on A^ω.

Def. $\mathcal{RM}^{\mathsf{fin}}_{|A|}$ is the set of all *right-ideal morphisms*, whose domains are *finitely generated* right ideals of A^* (i.e., the ends of the domain are a clopen set).

Fact. If $\mathsf{Dom}(\varphi)$ is finitely generated then $\mathsf{Im}(\varphi)$ is also finitely generated.

Prop. (R. Thompson, G. Higman, E. Scott, for *groups*) Every $\varphi \in \mathcal{RM}^{\mathsf{fin}}_{|A|}$ has a **unique** maximal end-equivalent extension (within $\mathcal{RM}^{\mathsf{fin}}_{|A|}$).

This max. extension is denoted by $\mathsf{max}(\varphi)$.

Definition of the **Higman-Thompson monoid** $M_{k,1}$:

$M_{k,1} = \{\mathsf{max}(\varphi) \ : \ \varphi$ is a right-ideal morphism between finitely generated right ideals of $A^*\}$.
($k = |A|$).

Multiplication: *function composition* followed by *maximal essentially equal extension.*
(This is associative.)

Prop. $M_{k,1}$ is the faithful action of $\mathcal{RM}_k^{\mathsf{fin}}$ on A^ω.

Definition of the **Higman-Thompson group**:

$G_{k,1} = \{\mathsf{max}(\varphi) \ : \ \varphi$ is a right-ideal **iso**morphism between finitely generated **essential** right ideals of $A^*\}$.

Prop. $G_{k,1}$ is the faithful action on A^ω of the isomorphisms between finitely generated essential right ideals.

Properties of $M_{k,1}$

$M_{k,1}$ is **congruence-simple**.

$G_{k,1}$ is **simple** iff k is even.

$G_{k,1}$ is the **group of units** (invertible elements) of $M_{k,1}$.

$M_{k,1} \hookrightarrow \mathcal{O}_k$ (Cuntz algebra).

$M_{k,1}$ contains all finite monoids,
$G_{k,1}$ contains all finite groups.

The Green relations of a monoid M:

Let $s, t \in M$.

$t \le_{\mathcal{J}} s$ iff $MtM \subseteq MsM$
iff $(\exists x, y \in M)\ t = xsy$. ($t$ is a two-sided multiple of s)

$t \le_{\mathcal{R}} s$ iff $tM \subseteq sM$
iff $(\exists y \in M)\ t = sy$. ($t$ is a right multiple of s)

$t \le_{\mathcal{L}} s$ iff $Mt \subseteq Ms$ contain t
iff $(\exists x \in M)\ t = xs$. ($t$ is a left multiple of s)

$t \equiv_{\mathcal{D}} s$ iff $(\exists p_1 \in M)\ t \equiv_{\mathcal{R}} p_1 \equiv_{\mathcal{L}} s$
iff $(\exists p_2 \in M)\ t \equiv_{\mathcal{L}} p_2 \equiv_{\mathcal{R}} s$.

Prop. ($\mathcal{J}$): $M_{k,1}$ is $\mathcal{J}^0$-simple
(the only ideals are $\mathbf{0}$ and $M_{k,1}$ itself).

Prop. ($\mathcal{D}$): $M_{k,1}$ has $k-1$ non-zero $\equiv_{\mathcal{D}}$-classes.
In particular, $M_{2,1}$ is $\mathcal{D}^0$-simple ("0-bisimple").

For all non-zero $\varphi, \psi \in M_{k,1}$:

$\psi \equiv_{\mathcal{D}} \varphi$ iff

$|\mathsf{imC}(\psi)| \equiv |\mathsf{imC}(\varphi)| \mod k-1.$

Prop. $M_{k,1}$ is regular (i.e., $\forall f \exists f' : f f' f = f$).

Prop. $\psi \leq_{\mathcal{R}} \varphi$ iff
$\mathsf{ends}(\mathsf{Im}(\psi)) \subseteq \mathsf{ends}(\mathsf{Im}(\varphi))$ iff
for some end-equivalent restrictions Ψ, Φ :

$\mathsf{imC}(\Psi) \subseteq \mathsf{imC}(\Phi)$.

Def. $\mathsf{mod}\varphi$ is the partition on $\mathsf{ends}(\mathsf{Dom}(\varphi))$, defined by
$u \equiv_{\mathsf{mod}\varphi} v$ iff $\varphi(u) = \varphi(v)$.

Prop. $\psi \leq_{\mathcal{L}} \varphi$ iff
$\mathsf{ends}(\mathsf{Dom}(\psi)) \subseteq \mathsf{ends}(\mathsf{Dom}(\varphi))$, and
$\mathsf{mod}\psi$ is coarser than $\mathsf{mod}\varphi$ on $\mathsf{ends}(\mathsf{Dom}(\psi))$

Prop. $<_{\mathcal{R}}$-chains and $<_{\mathcal{L}}$-chains are **dense**.
(If $x < y$ then $\exists z : \; x < z < y$.)

Prop. $M_{k,1}$ is *finitely generated.*

Prop. (Thompson, Higman): $G_{k,1}$ is *finitely presented.*

Open question: Is $M_{k,1}$ (not) *finitely presented* ?

Theorem.

Over any **finite generating set** Γ of $M_{k,1}$:

The *word problem* of $M_{k,1}$ is in P.

Deciding the Green relations of $M_{k,1}$ is in P.

Input: $\psi, \varphi \in M_{k,1}$, given by words over Γ.

Question: $\psi \le_{\mathcal{J}} \varphi$? (or $\le_{\mathcal{R}}$, $\le_{\mathcal{L}}$, $\equiv_{\mathcal{D}}$)

<u>Connection with combinational circuits</u>
(acyclic digital circuits)

$M_{2,1}$ acts (partially) on the set of bit-strings $\{0,1\}^*$;
so the elements of $M_{2,1}$ are boolean functions.

We now use a "**circuit-like**" generating set $\Gamma \cup \tau$;

Γ is any finite generating set of $M_{k,1}$ (generalized *gates*),

τ consists of the *position transpositions* on strings;
$\tau \ = \ \{\tau_{i,i+1} : i \geq 1\} \ \ (\ \subset G_{k,1})$

$$\tau_{i,i+1}: \quad x_1 \ \ldots \ x_{i-1} \ x_i \ x_{i+1} \ x_{i+2} \ \ldots \quad \longmapsto$$
$$x_1 \ \ldots \ x_{i-1} \ x_{i+1} \ x_i \ x_{i+2} \ \ldots$$

$\tau_{i,i+1}$ undefined on short words.

(*wire-crossing*).

Theorem.
For every combinational circuit C
there is a word w over $\Gamma \cup \tau$ such that:
(1) C and w represent the same function,
(2) $|w| \leq c \cdot |C|$. (c is a const.)

Conversely:
If $f : A^m \to A^n$ is represented by $w \in (\Gamma \cup \tau)^*$
then f has a combinational circuit C with

$$|C| \leq c \cdot |w|^2.$$

Decision problems over a "circuit-like" generating set $\Gamma \cup \tau$

Theorem.
The word problem of $M_{k,1}$ over $\Gamma \cup \tau$ is coNP-complete (similar to the circuit equivalence problem).

Theorem. Over $\Gamma \cup \tau$:
deciding $\le_{\mathcal{R}}$ is Π_2^{P}-complete
(similar to the surjectiveness problem for circuits);
deciding $\le_{\mathcal{L}}$ is coNP-complete
(similar to the injectiveness problem for circuits).

$\mathsf{coNP} = \{L : \overline{L} \in \mathsf{NP}\}$.

$\Sigma_2^{\mathsf{P}} = \mathsf{NP}^{\mathsf{NP}} =$ all languages accepted by polyn.-time *nondet.* Turing machines, with oracle in NP (or equivalently, with oracle in coNP).

$\Pi_2^{\mathsf{P}} = (\mathsf{coNP})^{\mathsf{NP}} =$ all languages accepted by polyn.-time *co-nondet.* Turing machines, with oracle in NP (or equivalently, with oracle in coNP).

Monoids of polyn.-time functions

Motivation:

Use (finitely generated) semigroups to study NP and one-way functions.

Definition scheme:

A partial function $f : A^* \to A^*$ is called **"one-way"** iff
(1) $f(x)$ is *"easy"* to compute (knowing f and x),
(2) knowing f and $y \in \mathsf{Im}(f)$, it is *"difficult"* to find any $x \in A^*$ such that $f(x) = y$.

(Old idea, William Stanley Jevons 1873; ex. of multiplication vs. factorization. Diffie & Hellman 1976, discr. log.)

The function semigroup fP

We fix an alphabet A (typically, $\{0, 1\}$).

Def. A partial function $f : A^* \to A^*$ is *polynomially balanced* iff there exists polynomials p, q such that for all $x \in \mathsf{Dom}(f)$: $|f(x)| \le p(|x|)$ and $|x| \le q(|f(x)|)$.

Def. fP = set of partial functions $f : A^* \to A^*$ such that
- $x \longmapsto f(x)$ is computable in det. polyn. time;
- f is polynomially balanced.

The first property implies **Dom**$(f) \in \mathsf{P}$.

Prop. fP is closed under composition.

Def. (worst-case one-way function; not "cryptographic"):
A function f is **one-way** iff $f \in \mathsf{fP}$, but there does *not* exist any deterministic polyn.-time algorithm which,

– on input $y \in A^*$,
– finds any $x \in A^*$ such that $f(x) = y$ when $y \in \mathsf{Im}(f)$.
(There is no requirement in when $y \notin \mathsf{Im}(f)$.)

Prop. (well known, 1980s or 1970s):
One-way functions exist iff $\mathsf{P} \neq \mathsf{NP}$.

Lemma. (Definition of "inverse"): The following are equivalent for partial functions $f, f' : A^* \to A^*$.

- For all $y \in \mathsf{Im}(f)$, $f'(y)$ is defined and $f(f'(y)) = y$.
(Thus, $\mathsf{Im}(f) \subseteq \mathsf{Dom}(f')$.)
- $f \cdot f'|_{\mathsf{Im}(f)} = \mathsf{id}|_{\mathsf{Im}(f)}$.
- $f \cdot f' \cdot f = f$.

Such an f' is called an **inverse** of f.

How any inverse f' of f is made:
(1) Choose $\mathsf{Dom}(f')$ arbitrarily, with $\mathsf{Im}(f) \subseteq \mathsf{Dom}(f')$.
For every $y \in \mathsf{Im}(f)$, choose $f'(y)$ to be any $x \in f^{-1}(y)$.
($f'|_{\mathsf{Im}(f)}$ is the *choice function* of f'.)

(2) For every $y \in \mathsf{Dom}(f') - \mathsf{Im}(f)$, choose $f'(y)$ arbitrarily in A^*.

Then $f f' f = f$. Any inverse of f arises in this way.

Prop. fP is *regular* iff one-way functions do *not* exist.

Prop.
(1) If $f \in \mathbf{fP}$ then $\mathsf{Im}(f) \in \mathsf{NP}$.
(2) For every language $L \in \mathsf{NP}$ there exists $f_L \in \mathsf{fP}$ such that $L = \mathsf{Im}(f_L)$.

Proof. (2) Let M_L be a non-det. polyn.-time Turing machine accepting L. Define

$$f_L(x, s) = x \quad \text{iff}$$

M_L, with choice sequence s, accepts x;
$f_L(x, s)$ is undefined otherwise. $\square$

Prop. If $f \in \mathsf{fP}$ is regular then $\mathsf{Im}(f) \in \mathsf{P}$.

Thm. (JCB 2011) If $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$ then there exist surjective one-way functions.
Consequence: For $f \in \mathsf{fP}$, $\mathsf{Im}(f) \in \mathsf{P}$ is not equivalent to f being regular (if $\Pi_2^{\mathsf{P}} \neq \Sigma_2^{\mathsf{P}}$).

Prop. (regular $\mathcal{L}$- and $\mathcal{R}$-orders):
If $f, r \in \mathsf{fP}$ and r is *regular* with an inverse $r' \in \mathsf{fP}$ then:

- $f \le_{\mathcal{R}} r$ iff $f = rr'f$ iff $\mathsf{Im}(f) \subseteq \mathsf{Im}(r)$.
- $f \le_{\mathcal{L}} r$ iff $f = fr'r$ iff $\mathsf{mod}f \le \mathsf{mod}r$.

The $\mathcal{D}$-relation:

It is not known which infinite languages in P can be mapped onto each other by maps in fP.

Are all regular elements of fP with infinite image in the $\mathcal{D}$-class of $\mathsf{id}|_{A^*}$?

Prop. Let $P \subseteq A^*$ be a prefix code in P, and let $p_0 \in P$. All *regular* elements $f \in \mathsf{fP}$ with $\mathsf{Im}(f)$ of the form

$$L_P \ = \ (P - \{p_0\})\, A^* \ \cup \ p_0\,(p_0 A^* \cup \overline{PA^*})$$

are in the $\mathcal{D}$-class of $\mathsf{id}|_{A^*}$.

L_P is an "approximation" of the right ideal PA^*, since

$$(P - \{p_0\})\, A^* \ \subset \ L_P \ \subset \ PA^*.$$

In general, P is infinite, in P; so, $P - \{p_0\}$ is "almost" P.

Lemma.
(1) $L \in \mathsf{P}$ implies $LA^* \in \mathsf{P}$.
(2) Let R be a right ideal in P, let P be the prefix code P of R (i.e., $R = PA^*$); then $P \in \mathsf{P}$.

Def.
$\mathcal{RM}^{\mathsf{P}}_{|A|} \ = \ \{f \in \mathsf{fP} : f$ is a right ideal morphism of $A^*\}$.

If f is a right ideal morphism, $\mathsf{Dom}(f)$ is a right ideal.

$\mathcal{RM}^{\mathsf{fin}}_{|A|} \ \subset \ \mathcal{RM}^{\mathsf{P}}_{|A|}$.

Prop. $\mathcal{RM}^{\mathsf{P}}_{|A|}$ is $\mathcal{J}^0$-simple.

Proof. Let $(v \leftarrow u)$ denote $uz \mapsto vz$ (for all $z \in A^*$). So, $(\varepsilon \leftarrow \varepsilon) = \mathsf{id}|_{A^*}$. For $f \neq \mathbf{0}$, let $f(x_0) = y_0$. Then $(\varepsilon \leftarrow \varepsilon) = (\varepsilon \leftarrow y_0) \circ f \circ (x_0 \leftarrow \varepsilon)$. $\square$

Prop. fP is not $\mathcal{J}^0$-simple.
It has regular continuous (prefix-order preserving) elements in different non-0 $\mathcal{J}$-classes.

Prop. Every regular $f \in \mathcal{RM}^{\mathsf{P}}_2$ is "close" to an element of fP belonging to the $\mathcal{D}$-class of $\mathsf{id}|_{A^*}$.

Restrict f from $\mathsf{Im}(f) = PA^*$, with $p_0 \in P$, to

$$L \ = \ (P - \{p_0\})\, A^* \ \cup \ p_0\,(p_0 A^* \cup \overline{PA^*}) \ ;$$

then

$$\mathsf{Im}(f) - p_0 A^* \ \subset \ L \ \subset \ \mathsf{Im}(f).$$

Prop. The $\mathcal{D}$-class of id in $\mathcal{RM}^{\mathsf{P}}_2$ is $\mathcal{H}$-trivial.

Def. The *polyn.-time Thompson-Higman monoid* $\mathcal{M}_2^{\mathsf{P}}$ consists of the *end-equivalence classes* of elements of $\mathcal{RM}_2^{\mathsf{P}}$.

$\mathcal{M}_2^{\mathsf{P}}$ is the faithful action of $\mathcal{RM}_2^{\mathsf{P}}$ on A^ω.

The Thompson-Higman monoid $M_{k,1}$ is a submonoid of $\mathcal{M}_{|A|}^{\mathsf{P}}$ (where $k = |A|$).

Padding arguments:
Time-complexity is defined as a function of the input length. By making inputs longer, without changing the essential difficulty of a problem, one obtains a new (but "similar") problem with lower time-complexity.

Padding can mean, e.g., to replace x by all words of the form xw for $w \in A^n$.

This padding preserves end-equivalence.

The padding argument implies that $\mathcal{M}_2^{\mathsf{P}} = \mathcal{M}_2^{\mathsf{rec}}$, i.e., the faithful action on A^ω of $\mathcal{RM}_2^{\mathsf{rec}}$. Here, $\mathcal{RM}_2^{\mathsf{rec}}$ = all right-ideal morphisms that are recursive partial functions, with recursive domain, recursively balanced.

Prop. $\mathcal{M}_2^{\mathsf{P}}$ is regular and $\mathcal{D}^0$-simple (hence $\mathcal{J}^0$-simple).

One can define a *Thompson group* of polynomial-time functions by taking the group of units of $\mathcal{M}_2^{\mathsf{P}}$.

Embedding fP into $\mathcal{RM}_2^{\mathsf{P}}$

Def. fP uses the alphabet $\{0, 1\}$; let $\#$ be a new letter. For any $f \in \mathsf{fP}$, define $f_\# : \{0, 1, \#\}^* \to \{0, 1, \#\}^*$ by

$$\mathsf{Dom}(f_\#) \ = \ \mathsf{Dom}(f)\,\#\,\{0, 1, \#\}^*, \text{ and}$$
$$f_\#(x\#w) \ = \ f(x)\,\#\,w,$$

for all $x \in \mathsf{Dom}(f)$ ($\subseteq \{0, 1\}^*$), and all $w \in \{0, 1, \#\}^*$.

Prop.
(1) For any $L \subseteq \{0, 1\}^*$, $L\#$ is a prefix code in $\{0, 1, \#\}^*$.
(2) $f \in \mathsf{fP}$ iff $f_\# \in \mathcal{RM}_3^{\mathsf{P}}$

Def. Encoding from $\{0, 1, \#\}$ to $\{0, 1\}$:

$$\mathsf{code}(0) = 00, \quad \mathsf{code}(1) = 01, \quad \mathsf{code}(\#) = 11.$$

Def. We define $f^C : \{0, 1\}^* \to \{0, 1\}^*$ by

$$\mathsf{Dom}(f^C) \ = \ \mathsf{code}(\mathsf{Dom}(f)\,\#)\ \{0, 1\}^*, \text{ and}$$
$$f^C(\mathsf{code}(x\#)\,v) \ = \ \mathsf{code}(f(x)\,\#)\ v,$$

for all $x \in \mathsf{Dom}(f)$ ($\subseteq \{0, 1\}^*$), and all $v \in \{0, 1\}^*$.

Prop. $f \in \mathsf{fP}$ iff $f^C \in \mathcal{RM}_2^{\mathsf{P}}$.

Prop.
(1) $f \in \mathsf{fP} \mapsto f^C \in \mathcal{RM}_2^{\mathsf{P}}$ is an injective monoid homomorphism.
(2) f is regular in fP iff f^C is regular in $\mathcal{RM}_2^{\mathsf{P}}$.

Embeddings:

$$\mathsf{fP} \ \stackrel{C}{\hookrightarrow} \ \mathcal{RM}_2^{\mathsf{P}} \ \subset \ [\mathsf{id}]^0_{\mathcal{J}(\mathsf{fP})} \ \subset \ \mathsf{fP}.$$

Here, $[\mathsf{id}]^0_{\mathcal{J}(\mathsf{fP})}$ is the Rees quotient of the $\mathcal{J}$-class of the identity id of fP.

fP embeds into its $\mathcal{J}$-class of the identity (plus zero).

Evaluation maps

Turing machine evaluation function

$$\mathsf{eval}_{\mathsf{TM}}(w, x) \ = \ f_w(x)$$

where f_w is the input-output (partial) function described by the word (program) w.
$\mathsf{eval}_{\mathsf{TM}}$ is the I/O map of the universal Turing machines, or of TM interpreters.

Evaluation function for acyclic circuits

$$\mathsf{eval}_{\mathsf{circ}}(C, x) = f_C(x),$$

where f_C is the input-output map of a circuit C.
(Assume f_C is length-preserving, i.e., $|f_C(x)| = |x|$.)

Levin's universal one-way function (1980s):

$$\mathsf{ev}_{\mathsf{Levin}}(C, x) \ = \ (C, f_C(x)),$$

Then, $\mathsf{ev}_{\mathsf{Levin}} \in \mathsf{fP}$.

Thm. (L. Levin) If one-way functions exist then $\mathsf{ev}_{\mathsf{Levin}}$ is a one-way function.

Evaluation maps for fP:

Use programs with *built-in polyn.-time counter*, for time complexity, and for balancing. (1970's, Hartmanis, Lewis, Stearns, et al.)

First attempt: For fP we define

$$\mathsf{ev_{poly}}(w, x) = (w, f_w(x)),$$

where w is any polynomial program, and $f_w \in \mathsf{fP}$.

But $\mathsf{ev_{poly}}$ is *not* in fP :
complexity on input (w, x) is $> c\,|w| \cdot p_w(|x|)$,
and balancing function is $> c\,(|w| + p_w(|x|))$;
the degree of p_w depends on w.

For a fixed polynomial q, let

$$\mathsf{fP}^{(q)} = \{ f_w \in \mathsf{fP}^{(q)} : \text{ for all } x \in \mathsf{Dom}(f), \\ w \text{ has time-complexity } T_w(|x|) \le q(|x|) \text{ and} \\ \text{input-balance } |x| \le q(|f_w(x)|) \}.$$

Let

$$\mathsf{ev}_{(q)}(w, x) = (w, f_w(x)),$$

where w is any q-polynomial program.

Encoding:

$$\mathsf{ev}^C_{(q)}(\mathsf{code}(w\#)\, x) = \mathsf{code}(w\#)\, f_w(x).$$

When f_w is a right ideal morphism, $\mathsf{ev}^C_{(q)}$ is also a right ideal morphism.

Prop. Suppose q satisfies $q(n) > c\, n^2 + c$ (for an appropriate constant $c > 1$ that depends on the model of computation). Then

$\mathsf{ev}^C_{(q)} \in \mathsf{fP}^{(q)}$, and

$\mathsf{ev}^C_{(q)}$ is a one-way function if one-way functions exist.

For any fixed word $v \in \{0,1\}^*$ we define

$$\pi_v : x \in \{0,1\}^* \longmapsto v\, x \ ;$$

and for any fixed integer $k > 0$ we define

$$\pi'_k : z\, x \in \{0,1\}^* \longmapsto x, \text{ where } |z| = k$$
$(\pi_k(t)$ undefined if $|t| < k)$.

π_v is a composite of the maps π_0 and π_1.
π'_k is the kth power of π'_1.

We define the padding map,

$$\mathsf{expand}(w, x) \ = \ (\mathsf{e}(w), \ (0^{|x|^2}, \ x))$$

where $\mathsf{e}(w)$ is such that

$$f_{\mathsf{e}(w)}(0^k, \ x) \ = \ (0^k, \ f_w(x)), \ \text{ for all } k.$$

Encoding:

$$\mathsf{expand}(\mathsf{code}(w)\ 11\ x) \ = \ \mathsf{code}(\mathsf{ex}(w))\ 11\ 0^{|x|^2}\ 11\ x,$$

now with $\mathsf{ex}(w)$ such that

$$f_{\mathsf{ex}(w)}(0^k\ 11\ x) \ = \ 0^k\ 11\ f_w(x) \quad \text{for all } k \geq 0.$$

We define a repeated padding map,

$$\mathsf{reexpand}(\mathsf{code}(\mathsf{ex}(w))\ 11\ 0^k\ 11\ x) \ = \ \mathsf{code}(\mathsf{ex}(w))\ 11\ 0^{k^2}\ 11\ x,$$

with $\mathsf{ex}(w)$ as above.

Unpadding map:

$$\mathsf{contr}(\mathsf{ex}(w),\ (0^{|y|^2},\ y)) \ = \ (w,\ y)$$

(undefined on other inputs).

Encoding:

$$\mathsf{contr}(\mathsf{code}(\mathsf{ex}(w))\ 11\ 0^{|y|^2}\ 11\ y) \ = \ w\ 11\ y$$

(undefined on other inputs).

Repeated unpadding:

$$\mathsf{recontr}(\mathsf{code}(\mathsf{ex}(w))\ 11\ 0^{k^2}\ 11\ y) \\ = \ \mathsf{code}(\mathsf{ex}(w))\ 11\ 0^{k}\ 11\ y$$

(undefined on other inputs).

Prop. fP is finitely generated.

Proof. The following is a generating set of fP:

$\{\mathsf{expand},\ \mathsf{reexpand},\ \mathsf{contr},\ \mathsf{recontr},\ \pi_0,\ \pi_1,\ \pi_1',\ \mathsf{ev}^C_{(q_2)}\}$,
where $q_2(n) = c\,n^2 + c$.

For any $f_w \in \mathsf{fP}^{(q)}$, let m be an integer $\geq \log_2$ of the sum of the degrees and the positive coefficients of q.

$$f_w(x) \ = \ \pi'_{2\,|w|+2} \circ \mathsf{contr} \circ \mathsf{recontr}^m \circ \mathsf{ev}^C_{(q_2)} \circ \mathsf{reexpand}^m \circ \mathsf{expand} \circ \pi_{\mathsf{code}(w)\,11}(x).$$

Now we have two ways to describe a function by a word.

Prop. (Program vs. generator string).
The maps $s \mapsto w$ and $w \mapsto s$ are in fP, where
s is over the generators of fP,
w is a polynomial program,
with $\Pi s = f_w$.
(Compiler maps.)

Prop. fP is *not* finitely presented. Its word problem is co-r.e. but not r.e.

(Undecidability of word probl.:
The problem $L \stackrel{?}{=} A^*$ for *context-free languages* is undecidable. Context-free languages are in P.)

Q. Is $\mathcal{RM}_2^{\mathsf{P}}$ finitely generated?
The maps $\pi_0,\ \pi_1,\ \pi_1'$, $\mathsf{reexpand}$, contr, $\mathsf{recontr}$ are in $\mathcal{RM}_2^{\mathsf{P}}$.
There exists an evaluation map that works just for $\mathcal{RM}_2^{\mathsf{P}}$.
But the first padding map expand is not in $\mathcal{RM}_2^{\mathsf{P}}$.

Prop. fP is finitely generated by regular elements.

Proof. Use $E_{(q)}(w,x) = (w, f_w(x), x)$; clearly, $E_{(q)}$ is not one-way. But $\mathsf{ev}_{(q)}$ can be expressed as a composition of $E_{(q)}$ and the other generators. $\square$

Prop. There are elements of fP that are non-regular (if $\mathsf{P} \neq \mathsf{NP}$), whose product is regular.

Reductions

The usual reduction between partial functions:

$f_1 \preccurlyeq f_2$ iff
$(\exists \beta, \alpha$, polyn.-time computable$)$ $[\, f_1 = \beta \circ f_2 \circ \alpha \,]$.

"f_1 is *simulated* by f_2"

For languages, recall polyn.-time *many-to-one reduction*:

$L_1 \preccurlyeq_{\mathsf{m:1}} L_2$ iff
$(\exists$ polyn.-time computable function $\alpha)(\forall x \in A^*)$
$[\, x \in L_1 \;\;\Leftrightarrow\;\; \alpha(x) \in L_2 \,]$.

Fact. $L_1 \preccurlyeq_{\mathsf{m:1}} L_2$ with α as above iff
$L_1 = \alpha^{-1}(L_2)$ iff
$\chi_{L_1} = \chi_{L_2} \circ \alpha$ (i.e., χ_{L_1} is simulated by χ_{L_2}).

For monoids $M_0 \le M_1$ in general:
simulation is $\le_{\mathcal{J}(M_0)}$ within M_1 (submonoid $\mathcal{J}$-order, using multipliers in the submonoid M_0).

We want an "inversive reduction" such that
if a one-way function f_1 reduces to a function $f_2 \in \mathsf{fP}$,
then f_2 is also one-way.

Idea:

f_1 reduces "inversively" to f_2 iff

(1) f_1 is simulated by f_2, and

(2) the "easiest inverses" of f_1 are simulated by the "easiest inverses" of f_2.

(The "easiest inverses" are the "minimal inverses" for the simulation preorder. But do minimal inverses exist?)

Def. (inversive reduction).

$f_1 \leqslant_{\sf inv} f_2$ ("f_1 *reduces inversively* to f_2") iff

(1) $f_1 \preccurlyeq f_2$, and

(2) for every inverse f_2' of f_2 there exists an inverse f_1' of f_1 such that $f_1' \preccurlyeq f_2'$.

Here, f_1, f_2, f_1', f_2' range over all partial functions on strings.

The relation $\leqslant_{\sf inv}$ can be defined on monoids.

Assume $M_0 \le M_1 \le M_2$, with f_1, f_2 ranging over M_1, inverses f_1', f_2' ranging over M_2, and simulation being $\le_{\mathcal{J}(M_0)}$ (i.e., multipliers are in M_0).

We should assume that M_1 is regular within M_2, to avoid empty ranges for the quantifiers $(\forall f_2')(\exists f_1')$ (otherwise, $f_1 \leqslant_{\sf inv} f_2$ is trivially equivalent to $f_1 \preccurlyeq f_2$, when f_2 has no inverse in M_2).

Prop. $\leqslant_{\mathsf{inv}}$ is transitive and reflexive (pre-order).

Prop. If $f_1 \leqslant_{\mathsf{inv}} f_2$, $f_2 \in \mathsf{fP}$, and f_2 is *regular*, then $f_1 \in \mathsf{fP}$ and f_1 is regular.
Contrapositive: If $f_1, f_2 \in \mathsf{fP}$ and f_1 is *one-way*, then f_2 is one-way.

Prop. The evaluation map $\mathsf{ev}^C_{(q_2)}$ is *complete* in fP with respect to inversive reduction.
Proof. For any $f_w \in \mathsf{fP}$ with q-polynomial program w,

$$f_w(x) \;=\; \pi'_{2\,|w|+2} \circ \mathsf{contr} \circ \mathsf{recontr}^m \circ \mathsf{ev}^C_{(q_2)} \circ \mathsf{reexpand}^m \circ \mathsf{expand} \circ \pi_{\mathsf{code}(w)\,11}(x).$$

Let e' be any inverse of $\mathsf{ev}^C_{(q_2)}$. Then for any string of the form $\mathsf{code}(w)\,11\,y$ with $y \in \mathsf{Im}(f_w)$ we have:

$$\mathsf{e}'(\mathsf{code}(w)\,11\,y) \;=\; \mathsf{code}(w)\,11\,x_i\,,$$
for some $x_i \in f_w^{-1}(y)$.

So e' simulates the inverse of f_w, defined by $f'_w(y) = x_i$, where x_i is as above (when $y \in \mathsf{Im}(f_w)$). $\square$

Prop. Levin's critical map $\mathsf{ev}_{\mathsf{Levin}}$ is $\leqslant_{\mathsf{inv}}$-complete in $\mathsf{fP}_{\mathsf{lp}}$ (length-preserving partial functions in fP).

Levin's map $\mathsf{ev}_{\mathsf{Levin}}$ is $\leqslant_{\mathsf{inv,T}}$-complete in fP, where
$\leqslant_{\mathsf{inv,T}}$ is polynomial inversive *Turing reduction.*

Prop. For each $f \in \mathsf{fP}$ there exists $\ell_f \in \mathsf{fP}_{\mathsf{lp}}$ such that $f \leqslant_{\mathsf{inv,T}} \ell_f$.

Inversification of any simulation:

For any $\preccurlyeq_X$, define $f_1 \leqslant_{\mathsf{inv,X}} f_2$ iff

$f_1 \preccurlyeq_X f_2$, and

$(\forall$ inverse f_2' of $f_2)$ $(\exists$ inverse f_1' of $f_1)$ $f_1' \preccurlyeq_X f_2'$.

Prop. If $\preccurlyeq_X$ is transitive then $\leqslant_{\mathsf{inv,X}}$ is transitive.

Prop. For every $f, r \in \mathcal{RM}_2^{\mathsf{P}}$ with r regular and f non-empty, we have $r \leqslant_{\mathsf{inv}} f$.

Prop. The $\equiv_{\mathcal{D}}$-relation is a refinement of $\leqslant_{\mathsf{inv}}$-equivalence.

The polynomial hierarchy

The classical polynomial hierarchy for languages:

$\Sigma_1^{\mathsf{P}} = \mathsf{NP}$, $\Pi_1^{\mathsf{P}} = \mathsf{coNP}$; and for $k > 0$:

$\Sigma_{k+1}^{\mathsf{P}} = \mathsf{NP}^{\Sigma_k^{\mathsf{P}}}$,

i.e., all languages accepted by non-det. Turing machines with oracle in Σ_k^{P} (equivalently, with oracle in Π_k^{P});

$\Pi_{k+1}^{\mathsf{P}} = (\mathsf{coNP})^{\Sigma_k^{\mathsf{P}}}$ $(= \mathsf{co}(\mathsf{NP}^{\Sigma_k^{\mathsf{P}}}))$;

$\mathsf{PH} = \bigcup_k \Sigma_k^{\mathsf{P}}$ $(\subseteq \mathsf{PSpace})$.

Polynomial hierarchy for functions:

$\mathsf{fP}^{\Sigma_k^{\mathsf{P}}}$ consists of all *polynomially balanced* partial functions (on A^*) computed by *det.* polyn.-time Turing machines with *oracle* in Σ_k^{P} (equivalently, with oracle in Π_k^{P}).

$\mathsf{fP}^{\mathsf{PH}}$ consists of all polynomially balanced partial functions (on A^*) computed by det. polyn.-time Turing machines with oracle in PH.

fPSpace consists of all polynomially balanced partial functions (on A^*) computed by det. polyn.-*space* Turing machines.

Prop. Every $f \in \mathsf{fP}$ has an inverse in $\mathsf{fP}^{\mathsf{NP}}$.
Every $f \in \mathsf{fP}^{\Sigma_k^{\mathsf{P}}}$ has an inverse in $\mathsf{fP}^{\Sigma_{k+1}^{\mathsf{P}}}$.
The monoids $\mathsf{fP}^{\mathsf{PH}}$ and $\mathsf{fPSpace}$ are regular.

Proof. The following is an inverse of f:

$$f'(y) = \begin{cases} \mathsf{min}(f^{-1}(y)) & \text{if } y \in \mathsf{Im}(f), \\ y & \text{otherwise,} \end{cases}$$

where min refers to dictionary order. □

If $\mathsf{P} = \mathsf{NP}$ then $\mathsf{P} = \mathsf{PH}$ and $\mathsf{fP}^{\mathsf{PH}} = \mathsf{fP}$; so $\mathsf{fP}^{\mathsf{PH}}$ is a "minimal" regular extension of fP.

Prop.
For each $k \geq 1$, $\mathsf{fP}^{\Sigma_k^{\mathsf{P}}}$ is finitely generated, but not finitely presented. The word problem is co-r.e. but not r.e.
$\mathsf{fPSpace}$ is also finitely generated, but not finitely presented. The word problem is co-r.e. but not r.e.
The monoid $\mathsf{fP}^{\mathsf{PH}}$ is not finitely generated, unless the polyn. hierarchy collapses.