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).

Preliminary definitions

Fix a finite alphabet $A$.

$A^* = \text{set of all finite words over } A$.

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

$A^\omega = \text{set of all } \omega\text{-words over } A$ (the ends of $A^*$), with the Cantor space topology.

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

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

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

I.e., $\text{ends}(R)$ is dense in $A^\omega$. 
**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 $\text{ends}(R)$ and $\text{ends}(R')$ “are the same up to density”, i.e., $\overline{\text{ends}(R)} = \overline{\text{ends}(R')}$, 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 $R_1 = \text{Dom}(\varphi)$, image set = $\text{Im}(\varphi)$.

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

**Fact.** $\varphi$ acts as a continuous partial function on $A^\omega$.

**Def.** $\mathcal{RM}^{\text{fin}}_{|A|}$ 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 $\text{Dom}(\varphi)$ is finitely generated then $\text{Im}(\varphi)$ is also finitely generated.

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

This max. extension is denoted by $\text{max}(\varphi)$. 
Definition of the Higman-Thompson monoid $M_{k,1}$:

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

($k = |A|$).

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

Prop. $M_{k,1}$ is the faithful action of $\mathcal{RM}_{k}^{\text{fin}}$ on $A^\omega$.

Definition of the Higman-Thompson group:

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

Prop. $G_{k,1}$ is the faithful action on $A^\omega$ 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 \( M_{k,1} \).

\( M_{k,1} \) \( \hookrightarrow \) \( O_k \) (Cuntz algebra).

\( M_{k,1} \) contains all finite monoids, 
\( G_{k,1} \) contains all finite groups.

The Green relations of a monoid \( M \):

Let \( s, t \in M \).

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

\( t \leq_R s \) iff \( tM \subseteq sM \)
iff \( (\exists y \in M) \ t = sy. \ (t \text{ is a right multiple of } s) \)

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

\( t \equiv_D s \) iff \( (\exists p_1 \in M) \ t \equiv_R p_1 \equiv_L s \)
iff \( (\exists p_2 \in M) \ t \equiv_L p_2 \equiv_R s. \)
Prop. \((\mathcal{J})\): \(M_{k,1}\) is \(\mathcal{J}^0\)-simple
(the only ideals are \(0\) 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 \(\varphi, \psi \in M_{k,1}\):
\[
\psi \equiv_{\mathcal{D}} \varphi \quad \text{iff} \quad |\text{im}C(\psi)| \equiv |\text{im}C(\varphi)| \mod k - 1.
\]

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

Prop. \(\psi \leq_{\mathcal{R}} \varphi \quad \text{iff} \quad \text{ends}(\text{Im}(\psi)) \subseteq \text{ends}(\text{Im}(\varphi)) \quad \text{iff} \quad \text{for some end-equivalent restrictions } \Psi, \Phi : \text{im}C(\Psi) \subseteq \text{im}C(\Phi).

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

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

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 $\Gamma$ of $M_{k,1}$:
The word problem of $M_{k,1}$ is in $\mathcal{P}$.
Deciding the Green relations of $M_{k,1}$ is in $\mathcal{P}$.

Input: $\psi, \varphi \in M_{k,1}$, given by words over $\Gamma$.
Question: $\psi \leq_J \varphi$? (or $\leq_R, \leq_L, \equiv_D$)
Connection with combinational circuits
(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 $\Gamma \cup \tau$; $\Gamma$ is any finite generating set of $M_{k,1}$ (generalized gates), $\tau$ consists of the position transpositions on strings; $\tau = \{\tau_{i,i+1} : i \geq 1\}$ ($\subset G_{k,1}$)

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

$\tau_{i,i+1}$ 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 $f : A^m \rightarrow A^n$ 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 $\text{coNP}$-complete (similar to the circuit equivalence problem).

**Theorem.** Over $\Gamma \cup \tau$:
- deciding $\leq_R$ is $\Pi^P_2$-complete (similar to the surjectiveness problem for circuits);
- deciding $\leq_L$ is $\text{coNP}$-complete (similar to the injectiveness problem for circuits).

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

$\Sigma^P_2 = \text{NP}^{\text{NP}} = \text{all languages accepted by polyn.-time nondet. Turing machines, with oracle in } \text{NP} \text{ (or equivalently, with oracle in } \text{coNP}).$

$\Pi^P_2 = (\text{coNP})^{\text{NP}} = \text{all languages accepted by polyn.-time co-non} \text{det. Turing machines, with oracle in } \text{NP} \text{ (or equivalently, with oracle in } \text{coNP}).$
Monoids of polyn.-time functions

Motivation:
Use (finitely generated) semigroups to study $\text{NP}$ 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 $y \in \text{Im}(f)$, 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 $f\mathcal{P}$
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 $x \in \text{Dom}(f) : |f(x)| \leq p(|x|)$ and $|x| \leq q(|f(x)|)$.

Def. $f\mathcal{P} = \text{set of partial functions } f : A^* \to A^* \text{ such that}$
$\bullet \ x \mapsto f(x) \text{ is computable in det. polyn. time;}
\bullet \ f \text{ is polynomially balanced.}$

The first property implies $\text{Dom}(f) \in \mathcal{P}$.

Prop. $f\mathcal{P}$ is closed under composition.
Def. (worst-case one-way function; not “cryptographic”): A function $f$ is **one-way** iff $f \in f\mathcal{P}$, 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 $y \in \text{Im}(f)$.
  (There is no requirement in when $y \not\in \text{Im}(f)$.)

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

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

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

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

How any inverse $f'$ of $f$ is made:
1. Choose $\text{Dom}(f')$ arbitrarily, with $\text{Im}(f) \subseteq \text{Dom}(f')$.
   For every $y \in \text{Im}(f)$, choose $f'(y)$ to be any $x \in f^{-1}(y)$.
   ($f'|_{\text{Im}(f)}$ is the *choice function* of $f'$.)
2. For every $y \in \text{Dom}(f') - \text{Im}(f)$, choose $f'(y)$ arbitrarily in $A^*$.
   Then $ff'f = f$. Any inverse of $f$ arises in this way.

Prop. $f\mathcal{P}$ is **regular** iff one-way functions do not exist.
Prop. (1) If \( f \in \mathsf{fP} \) then \( \text{Im}(f) \in \mathsf{NP} \).
(2) For every language \( L \in \mathsf{NP} \) there exists \( f_L \in \mathsf{fP} \) such that \( L = \text{Im}(f_L) \).

Proof. (2) Let \( M_L \) be a non-det. polyn.-time Turing machine accepting \( L \). Define
\[
f_L(x, s) = x \quad \text{iff} \quad M_L, \text{ with choice sequence } s, \text{ accepts } x;
\]
\( f_L(x, s) \) is undefined otherwise. \( \square \)

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

Thm. (JCB 2011) If \( \Pi^P_2 \neq \Sigma^P_2 \) then there exist surjective one-way functions.

Consequence: For \( f \in \mathsf{fP}, \text{Im}(f) \in \mathsf{P} \) is not equivalent to \( f \) being regular (if \( \Pi^P_2 \neq \Sigma^P_2 \)).
Prop. (regular \(L\)- and \(R\)-orders):

If \(f, r \in fP\) and \(r\) is regular with an inverse \(r' \in fP\) then:

- \(f \leq R r\) iff \(f = rr'f\) iff \(\text{Im}(f) \subseteq \text{Im}(r)\).
- \(f \leq L r\) iff \(f = fr'r\) iff \(\text{mod} f \leq \text{mod} r\).

The \(D\)-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 \(D\)-class of \(\text{id}|_{A^*}\)?

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

\[
L_P = (P - \{p_0\}) A^* \cup p_0 (p_0 A^* \cup PA^*)
\]

are in the \(D\)-class of \(\text{id}|_{A^*}\).

\(L_P\) is an “approximation” of the right ideal \(PA^*\), since

\[
(P - \{p_0\}) A^* \subseteq L_P \subseteq PA^*.
\]

In general, \(P\) is infinite, in \(P\); so, \(P - \{p_0\}\) is “almost” \(P\).

Lemma.

(1) \(L \in P\) implies \(LA^* \in P\).

(2) Let \(R\) be a right ideal in \(P\), let \(P\) be the prefix code \(P\) of \(R\) (i.e., \(R = PA^*\)); then \(P \in P\).
Def. \( \mathcal{RM}_{|A|}^P = \{ f \in fP : f \text{ is a right ideal morphism of } A^* \} \).

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

\[ \mathcal{RM}_{|A|}^{\text{fin}} \subset \mathcal{RM}_{|A|}^P. \]

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

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

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 \( f \in \mathcal{RM}_2^P \) is “close” to an element of \( fP \) belonging to the \( \mathcal{D} \)-class of \( \text{id}|_{A^*} \).

Restrict \( f \) from \( \text{Im}(f) = PA^* \), with \( p_0 \in P \), to
\[
L = (P - \{p_0\}) A^* \cup p_0 \left(p_0 A^* \cup \overline{PA^*}\right);
\]
then
\[
\text{Im}(f) - p_0 A^* \subset L \subset \text{Im}(f).
\]

Prop. The \( \mathcal{D} \)-class of \( \text{id} \) in \( \mathcal{RM}_2^P \) is \( \mathcal{H} \)-trivial.
Def. The *polyn.-time Thompson-Higman monoid* \( \mathcal{M}_2^P \) consists of the *end-equivalence classes* of elements of \( \mathcal{RM}_2^P \). \( \mathcal{M}_2^P \) is the faithful action of \( \mathcal{RM}_2^P \) on \( A^\omega \).

The Thompson-Higman monoid \( M_{k,1} \) is a submonoid of \( \mathcal{M}_{|A|}^P \) (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 \( \mathcal{M}_2^P = \mathcal{M}_2^{\text{rec}} \), i.e., the faithful action on \( A^\omega \) of \( \mathcal{RM}_2^{\text{rec}} \). Here, \( \mathcal{RM}_2^{\text{rec}} \) = all right-ideal morphisms that are recursive partial functions, with recursive domain, recursively balanced.

Prop. \( \mathcal{M}_2^P \) 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 \( \mathcal{M}_2^P \).
Embedding \( f\mathcal{P} \) into \( \mathcal{RM}_2^P \)

**Def.** \( f\mathcal{P} \) uses the alphabet \( \{0, 1\} \); let \( \# \) be a new letter. For any \( f \in f\mathcal{P} \), define \( f\# : \{0, 1, \#\}^* \to \{0, 1, \#\}^* \) by

\[
\text{Dom}(f\#) = \text{Dom}(f) \# \{0, 1, \#\}^*, \quad \text{and} \quad f\#(x\#w) = f(x) \# w,
\]
for all \( x \in \text{Dom}(f) (\subseteq \{0, 1\}^*) \), and all \( w \in \{0, 1, \#\}^* \).

**Prop.**

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

**Def.** Encoding from \( \{0, 1, \#\} \) to \( \{0, 1\} \):

\[
\text{code}(0) = 00, \quad \text{code}(1) = 01, \quad \text{code}(\#) = 11.
\]

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

\[
\text{Dom}(f^C) = \text{code}(\text{Dom}(f) \#) \{0, 1\}^*, \quad \text{and} \quad f^C(\text{code}(x\#) v) = \text{code}(f(x) \#) v,
\]
for all \( x \in \text{Dom}(f) (\subseteq \{0, 1\}^*) \), and all \( v \in \{0, 1\}^* \).

**Prop.** \( f \in f\mathcal{P} \) iff \( f^C \in \mathcal{RM}_2^P \).

**Prop.**

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

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

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

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

*Turing machine evaluation function*

\[ \text{eval}_{\text{TM}}(w, x) = f_w(x) \]

where \( f_w \) is the input-output (partial) function described by the word (program) \( w \).

\( \text{eval}_{\text{TM}} \) is the I/O map of the universal Turing machines, or of TM interpreters.

*Evaluation function for acyclic circuits*

\[ \text{eval}_{\text{circ}}(C, x) = f_C(x), \]

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)*:

\[ \text{ev}_{\text{Levin}}(C, x) = (C, f_C(x)), \]

Then, \( \text{ev}_{\text{Levin}} \in \text{fP} \).

**Thm.** (L. Levin) If one-way functions exist then \( \text{ev}_{\text{Levin}} \) is a one-way function.
Evaluation maps for $\text{fP}$:

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

First attempt: For $\text{fP}$ we define

$$\text{ev}_{\text{poly}}(w, x) = (w, f_w(x)),$$

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

But $\text{ev}_{\text{poly}}$ is not in $\text{fP}$:
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
\[
\mathsf{fP}^{(q)} = \{ f_w \in \mathsf{fP}^{(q)} : \text{ for all } x \in \text{Dom}(f), \]
\[
\text{ } w \text{ has time-complexity } T_w(|x|) \leq q(|x|) \text{ and } \\
\text{input-balance } |x| \leq q(|f_w(x)|) \}.
\]

Let
\[
\text{ev}^{(q)}(w, x) = (w, f_w(x)),
\]
where $w$ is any $q$-polynomial program.

Encoding:
\[
\text{ev}^C_{(q)}(\text{code}(w\#) x) = \text{code}(w\#) f_w(x).
\]

When $f_w$ is a right ideal morphism, $\text{ev}^C_{(q)}$ 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
\[
\text{ev}^C_{(q)} \in \mathsf{fP}^{(q)}, \quad \text{and}
\]
\[
\text{ev}^C_{(q)} \text{ is a one-way function if one-way functions exist}.
\]
For any fixed word $v \in \{0, 1\}^*$ we define

$$\pi_v : x \in \{0, 1\}^* \mapsto vx ;$$

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

$$\pi'_k : zx \in \{0, 1\}^* \mapsto x, \text{ where } |z| = k$$

($\pi_k(t)$ undefined if $|t| < k$).

$\pi_v$ is a composite of the maps $\pi_0$ and $\pi_1$.

$\pi'_k$ is the $k$th power of $\pi'_1$.

We define the padding map,

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

where $e(w)$ is such that

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

Encoding:

$$\text{expand} (\text{code}(w) \ 11 \ x) =$$

$$\text{code} (\text{ex}(w)) \ 11 \ 0^{|x|^2} \ 11 \ x, \text{ now with } \text{ex}(w) \text{ such that}$$

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

We define a repeated padding map,

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

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

with $\text{ex}(w)$ as above.
Unpaddings map:

\[
\text{contr}(\text{ex}(w), (0^{|y|^2}, y)) = (w, y)
\]
(undefined on other inputs).

Encoding:

\[
\text{contr}(\text{code}(\text{ex}(w)) \ 11 \ 0^{|y|^2} \ 11 \ y) = w \ 11 \ y
\]
(undefined on other inputs).

Repeated unpaddings:

\[
\text{recontr}(\text{code}(\text{ex}(w)) \ 11 \ 0^{|y|^2} \ 11 \ y) = \text{code}(\text{ex}(w)) \ 11 \ 0^{|y|^2} \ 11 \ y
\]
(undefined on other inputs).
Prop. $fP$ is finitely generated.

Proof. The following is a generating set of $fP$:

$$\{\text{expand, reexpand, contr, recontr, } \pi_0, \pi_1, \pi'_1, \text{ ev}^C_{(q_2)}\},$$

where $q_2(n) = cn^2 + c$.

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

$$f_w(x) = \pi'_2|w|+2 \circ \text{contr} \circ \text{recontr}^m \circ \text{ev}^C_{(q_2)} \circ \text{reexpand}^m \circ \text{expand} \circ \pi_{\text{code}(w)11}(x).$$

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 $L = A^*$ for context-free languages is undecidable. Context-free languages are in $P$.)

25
Q. Is $\mathcal{RM}_2^P$ finitely generated?
The maps $\pi_0$, $\pi_1$, $\pi'_1$, reexpand, contr, recontr are in $\mathcal{RM}_2^P$. There exists an evaluation map that works just for $\mathcal{RM}_2^P$. But the first padding map expand is not in $\mathcal{RM}_2^P$.

**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 $ev(q)$ can be expressed as a composition of $E(q)$ and the other generators. □

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

The usual reduction between partial functions:

\[ f_1 \preceq f_2 \iff (\exists \beta, \alpha, \text{polyn.-time computable}) \left[ f_1 = \beta \circ f_2 \circ \alpha \right]. \]

“\( f_1 \) is simulated by \( f_2 \)”

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

\[ L_1 \preceq_{m:1} L_2 \iff (\exists \text{polyn.-time computable function } \alpha)(\forall x \in A^*) \left[ x \in L_1 \iff \alpha(x) \in L_2 \right]. \]

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

For monoids \( M_0 \leq M_1 \) in general:
simulation is \( \preceq_{J(M_0)} \) within \( M_1 \) (submonoid \( J \)-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 \( f_2 \in fP \),

then \( f_2 \) is also one-way.
Idea:
\( f_1 \) reduces “inversely” 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).**
\( f_1 \leq_{\text{inv}} f_2 \) ("\( f_1 \) reduces inversively to \( f_2 \)"") iff
1. \( f_1 \leq f_2 \), and
2. for every inverse \( f_2' \) of \( f_2 \) there exists an inverse \( f_1' \) of \( f_1 \) such that \( f_1' \leq f_2' \).

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

The relation \( \leq_{\text{inv}} \) can be defined on monoids.

Assume \( M_0 \leq M_1 \leq M_2 \), with \( f_1, f_2 \) ranging over \( M_1 \), inverses \( f_1', f_2' \) ranging over \( M_2 \), and simulation being \( \leq_J(M_0) \) (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, \( f_1 \leq_{\text{inv}} f_2 \) is trivially equivalent to \( f_1 \leq f_2 \), when \( f_2 \) has no inverse in \( M_2 \)).
Prop. \( \preceq_{\text{inv}} \) is transitive and reflexive (pre-order).

Prop. If \( f_1 \preceq_{\text{inv}} f_2, \ f_2 \in \mathcal{fP} \), and \( f_2 \) is regular, then \( f_1 \in \mathcal{fP} \) and \( f_1 \) is regular.
Contrapositive: If \( f_1, f_2 \in \mathcal{fP} \) and \( f_1 \) is one-way, then \( f_2 \) is one-way.

Prop. The evaluation map \( \text{ev}_{\langle q_2 \rangle} \) is complete in \( \mathcal{fP} \) with respect to inversive reduction.

Proof. For any \( f_w \in \mathcal{fP} \) with \( q \)-polynomial program \( w \),
\[
 f_w(x) = \pi'_{2|w|+2} \circ \text{contr} \circ \text{recontr}^m \circ \text{ev}_{\langle q_2 \rangle} \\
 \circ \text{reexpand}^m \circ \text{expand} \circ \pi_{\text{code}(w)11}(x).
\]
Let \( e' \) be any inverse of \( \text{ev}_{\langle q_2 \rangle} \). Then for any string of the form \( \text{code}(w)11y \) with \( y \in \text{Im}(f_w) \) we have:
\[
e'(\text{code}(w)11y) = \text{code}(w)11x_i,
\]
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 \( y \in \text{Im}(f_w) \)). \( \square \)

Prop. Levin’s critical map \( \text{ev}_{\text{Levin}} \) is \( \preceq_{\text{inv}} \)-complete in \( \mathcal{fP}_{\text{lp}} \) (length-preserving partial functions in \( \mathcal{fP} \)).

Levin’s map \( \text{ev}_{\text{Levin}} \) is \( \preceq_{\text{inv},T} \)-complete in \( \mathcal{fP} \), where \( \preceq_{\text{inv},T} \) is polynomial inversive Turing reduction.

Prop. For each \( f \in \mathcal{fP} \) there exists \( \ell_f \in \mathcal{fP}_{\text{lp}} \) such that \( f \preceq_{\text{inv},T} \ell_f \).
Inversification of any simulation:

For any \( \preceq_X \), define \( f_1 \preceq_{\text{inv},X} f_2 \) iff

\[
f_1 \preceq_X f_2, \text{ and } \forall \text{ inverse } f_2' \text{ of } f_2 \exists \text{ inverse } f_1' \text{ of } f_1 \text{ } f_1' \preceq_X f_2'.
\]

Prop. If \( \preceq_X \) is transitive then \( \preceq_{\text{inv},X} \) is transitive.

Prop. For every \( f, r \in \mathcal{RM}_2^P \) with \( r \) regular and \( f \) non-empty, we have \( r \preceq_{\text{inv}} f \).

Prop. The \( \equiv_D \)-relation is a refinement of \( \preceq_{\text{inv}} \)-equivalence.
The polynomial hierarchy

The classical polynomial hierarchy for languages:

\[ \Sigma_1^P = \text{NP}, \quad \Pi_1^P = \text{coNP} ; \quad \text{and for } k > 0 : \]

\[ \Sigma_{k+1}^P = \text{NP}^{\Sigma_k^P} , \]

i.e., all languages accepted by non-det. Turing machines with oracle in \( \Sigma_k^P \) (equivalently, with oracle in \( \Pi_k^P \));

\[ \Pi_{k+1}^P = (\text{coNP})^{\Sigma_k^P} \quad (= \text{co}(\text{NP}^{\Sigma_k^P})); \]

\[ \text{PH} = \bigcup_k \Sigma_k^P \quad (\subseteq \text{PSpace}). \]

**Polynomial hierarchy for functions:**

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

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

\( fP^{\text{Space}} \) consists of all polynomially balanced partial functions (on \( A^* \)) computed by det. polyn.-space Turing machines.
Prop. Every $f \in \mathbb{fP}$ has an inverse in $\mathbb{fP}^{\text{NP}}$.

Every $f \in \mathbb{fP}^{\Sigma_k}$ has an inverse in $\mathbb{fP}^{\Sigma_{k+1}}$.

The monoids $\mathbb{fP}^{PH}$ and $\mathbb{fP}^{\text{Space}}$ are regular.

Proof. The following is an inverse of $f$:

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

where $\min$ refers to dictionary order. □

If $\mathbb{P} = \text{NP}$ then $\mathbb{P} = \text{PH}$ and $\mathbb{fP}^{\text{PH}} = \mathbb{fP}$; so $\mathbb{fP}^{\text{PH}}$ is a “minimal” regular extension of $\mathbb{fP}$.

Prop.

For each $k \geq 1$, $\mathbb{fP}^{\Sigma_k}$ is finitely generated, but not finitely presented. The word problem is co-r.e. but not r.e.

$\mathbb{fP}^{\text{Space}}$ is also finitely generated, but not finitely presented. The word problem is co-r.e. but not r.e.

The monoid $\mathbb{fP}^{\text{PH}}$ is not finitely generated, unless the polyn. hierarchy collapses.