Computabiltiy in the lattice of Equivalence Relations

Jean-Yves Moyen[1]   Jakob Grue Simonsen[1]
Jean-Yves.Moyen@lipn.univ-paris13.fr

[1]Datalogisk Institut
University of Copenhagen

Supported by the Marie Curie action "Walgo" program H2020-MSCA-IF-2014, number 655222 and the Danish Council for Independent Research *Sapere Aude* grant "Complexity via Logic and Algebra" (COLA).

April 22-23 2017

Motivations: the Lattice of Equivalences

The beginning: Rice's Theorem

Theorem (Rice, 1952)

Every non-trivial, extensional set of programs is undecidable.

Very powerful Theorem. One of the cornerstones of Computability.

Sketch of Proof.

`q'(x) = q(0); p(x)` computes the same thing as `p(x)` iff `q(0)` terminates. □

The beginning: Rice's Theorem

Theorem (Rice, 1952)

Every non-trivial, extensional set of programs is undecidable.

Very powerful Theorem. One of the cornerstones of Computability.

Sketch of Proof.

`q'(x) = q(0); p(x)` computes the same thing as `p(x)` iff `q(0)` terminates. □

Essentially, the question "do `p` and `q'` computes the same function?" is undecidable.

The Rereading: Rice's Equivalence

Essentially, the question "do `p` and `q` computes the same function?" is undecidable.

The Rereading: Rice's Equivalence

Essentially, the question "do `p` and `q` computes the same function?" is undecidable.
There is an underlying "extensional equivalence", or Rice's Equivalence: $\mathtt{p}\mathfrak{R}\mathtt{q}$ iff `p` and `q` compute the same function.

Theorem (Rice's Theorem, again)

Each (non-trivial) union of classes of $\mathfrak{R}$ is undecidable.

The Rereading: Rice's Equivalence

Essentially, the question "do `p` and `q` computes the same function?" is undecidable.
There is an underlying "extensional equivalence", or Rice's Equivalence: $\mathtt{p}\mathfrak{R}\mathtt{q}$ iff `p` and `q` compute the same function.

Theorem (Rice's Theorem, again)

Each (non-trivial) union of classes of $\mathfrak{R}$ is undecidable.

The set of equivalences between programs has a nice complete lattice structure.

Theorem (still Rice's Theorem)

Each (non-trivial) equivalence in the principal filter at $\mathfrak{R}$ is undecidable.

The object of study: the Lattice of Equivalences

Theorem (Rice's Theorem)

Each (non-trivial) equivalence in the principal filter at $\mathfrak{R}$ is undecidable.

- Rice's Theorem is expressed neatly in the language of Order Theory. Can we find something more if we dig deeper?

The object of study: the Lattice of Equivalences

Theorem (Rice's Theorem)

Each (non-trivial) equivalence in the principal filter at $\mathfrak{R}$ is undecidable.

- Rice's Theorem is expressed neatly in the language of Order Theory. Can we find something more if we dig deeper?
- There are $2^{\aleph_0}$ equivalences, so most of them are undecidable. $\mathfrak{R}$ is not really an exception.
- But, there are also many "easy to express" decidable equivalences (*e.g.*, having the same number of variables, of lines of code, ...)
- And it is not that easy to build undecidable out of the principal filter at $\mathfrak{R}$ that are undecidable. Yet, most of them are also undecidable... Notable success: Asperti-Rice Theorem, 2008.

The long term Plan and the Dream

- Systematic study of the set of Equivalences using various mathematical tools. Starting with Order Theory because we already know that interesting results (Rice's Theorem) have a nice expression in that language.
- Maybe, one of the equivalence is "`p` and `q` iff the implement the same algorithm." Thus we could start a scientifically sound Theory of Algorithms.

The long term Plan and the Dream

- Systematic study of the set of Equivalences using various mathematical tools. Starting with Order Theory because we already know that interesting results (Rice's Theorem) have a nice expression in that language.
- Maybe, one of the equivalence is "`p` and `q` iff the implement the same algorithm." Thus we could start a scientifically sound Theory of Algorithms.
- Wait, is "implementing the same Algorithm" really an Equivalence? (Blass, Derschowitz and Gurevich doubt it. . .)

The short term Plan: Order Theoretical Study

Study of the Lattice of Equivalences using tools from Order Theory.

- "Chains, Antichains, and Complements in Infinite Partition Lattices", AMSR:
  Complete characterisation of the possibles cardinals of these sets.
  Lattice of Equivalences over any set (non only countable ones).
  The Lattice structure is very rich!

The short term Plan: Order Theoretical Study

Study of the Lattice of Equivalences using tools from Order Theory.

- "Chains, Antichains, and Complements in Infinite Partition Lattices", AMSR:
  Complete characterisation of the possibles cardinals of these sets.
  Lattice of Equivalences over any set (non only countable ones).
  The Lattice structure is very rich!

  Let **C** be a maximal chain in the lattice of partitions over a set of cardinality $\aleph_{42}$. Under GCH, **C** contains $\aleph_{41}$, $\aleph_{42}$ or $\aleph_{43}$ partitions.

The short term Plan: Order Theoretical Study

Study of the Lattice of Equivalences using tools from Order Theory.

- "Chains, Antichains, and Complements in Infinite Partition Lattices", AMSR:
  Complete characterisation of the possibles cardinals of these sets.
  Lattice of Equivalences over any set (non only countable ones).
  The Lattice structure is very rich!
- "More intensional versions of Rice's Theorem", MS:
  Can we build Rice-like Theorems on another equivalence? Yes!

The short term Plan: Order Theoretical Study

Study of the Lattice of Equivalences using tools from Order Theory.

- "Chains, Antichains, and Complements in Infinite Partition Lattices", AMSR:
  Complete characterisation of the possibles cardinals of these sets.
  Lattice of Equivalences over any set (non only countable ones).
  The Lattice structure is very rich!
- "More intensional versions of Rice's Theorem", MS:
  Can we build Rice-like Theorems on another equivalence? Yes!

  Any decidable set that contains all the polytime **programs** must contain one program of each complexity ($n\log(n)$, $2^{2^{n^2}}$, $Ack(n,n)$, not multiple recursive, ...)

The short term Plan: Order Theoretical Study

Study of the Lattice of Equivalences using tools from Order Theory.

- "Chains, Antichains, and Complements in Infinite Partition Lattices", AMSR:
  Complete characterisation of the possibles cardinals of these sets.
  Lattice of Equivalences over any set (non only countable ones).
  The Lattice structure is very rich!
- "More intensional versions of Rice's Theorem", MS:
  Can we build Rice-like Theorems on another equivalence? Yes!
- Today:
  Since the Lattice itself is too big (uncountable), can we find subsets that are manageable and still keep the interesting properties?
  Can we find an approximation of the Lattice, in the same sense that $\mathbb{Q}$ approximate $\mathbb{R}$?

The Lattice of Equivalences

Refinment Ordering

- Isomorphism between Equivalences/Classes and Partitions/Blocks.
- $\mathcal{P} \leq \mathcal{Q}$ iff $\mathtt{x}\mathcal{P}\mathtt{y}$ implies $\mathtt{x}\mathcal{Q}\mathtt{y}$.
  That is, each block of $\mathcal{Q}$ is the union of one or more blocks of $\mathcal{P}$.

Refinment Ordering

- Isomorphism between Equivalences/Classes and Partitions/Blocks.
- $\mathcal{P} \leq \mathcal{Q}$ iff $\mathtt{x}\mathcal{P}\mathtt{y}$ implies $\mathtt{x}\mathcal{Q}\mathtt{y}$.
  That is, each block of $\mathcal{Q}$ is the union of one or more blocks of $\mathcal{P}$.
- Meet is easy: blocks of $\mathcal{P} \wedge \mathcal{Q}$ are (non-empty) intersections of one block of $\mathcal{P}$ and one of $\mathcal{Q}$.
- Join is more complicated... $\mathtt{x}(\mathcal{P} \vee \mathcal{Q})\mathtt{y}$ iff there exists $\mathtt{x}_1, \ldots, \mathtt{x}_n$ such that $\mathtt{x}\mathcal{P}\mathtt{x}_1\mathcal{Q}\ldots\mathcal{P}\mathtt{x}_n\mathcal{Q}\mathtt{y}$.

The Lattice of Equivalences

- The Lattice of Equivalences between programs is isomorphic to $\mathrm{Equ}(\mathbb{N})$, the Lattice of Equivalences between naturals numbers (or any other countable infinite set).
- The Lattice is complete, *i.e.* every set of equivalences has a meet and a join (not only the finite sets).

  Computability point of view: **every** set, whatever its own complexity (*e.g.* any Π^0_{14} set of equivalences has a join); computing these might be awfully complicated.
- The Lattice is complemented: every equivalence has at least one complement. (non-trivial equivalences have between $\aleph_0$ and $2^{\aleph_0}$ complements)

Today: searching for Sublattices

Can we find "natural" sublattices?
Preferably countable. Better if complete sublattices.

Intuition: since meet is extremely easy, it won't be our main problem (stability under meet will boil down to stability under intersection). However, join is union + transitive closure and will cause trouble.

Intuition: **complete** sublattice will be extremely difficult because we need to consider the meet of an **arbitrarily** set of equivalences and it's easy to get out of our sublattice.

"Natural" subset: defined by computability or complexity properties, *e.g.* is "the set of equivalences decidable in polynomial space" a sublattice?

Results

	Finite		Arithmetical		Arbitrary	
	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge/\vee$	complements
Automatic	Yes	Yes	No	Yes	No/Yes	N/A
Subrecursive	Yes	No†	?	No†	No	$\geq$ PSPACE
Σ_k^0	Yes	Yes	No	Yes	No	No
Π_k^0	Yes	No	Yes	No	No	?
Δ_k^0	Yes	No	No	No	No	Yes

†: for LOGSPACE or larger classes.

Results

	Finite $\wedge$	Finite $\vee$	Arithmetical $\wedge$	Arithmetical $\vee$	Arbitrary $\wedge/\vee$	complements
Automatic	Yes	Yes	No	Yes	No/Yes	N/A
Subrecursive	Yes	No†	?	No†	No	$\geq$ PSPACE
Σ_k^0	Yes	Yes	No	Yes	No	No
Π_k^0	Yes	No	Yes	No	No	?
Δ_k^0	Yes	No	No	No	No	Yes

†: for LOGSPACE or larger classes.

■ is closure under finite intersection.

Results

	Finite		Arithmetical		Arbitrary	
	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge/\vee$	complements
Automatic	Yes	Yes	No	Yes	No/Yes	N/A
Subrecursive	Yes	No†	?	No†	No	$\geq$ PSPACE
Σ_k^0	Yes	Yes	No	Yes	No	No
Π_k^0	Yes	No	Yes	No	No	?
Δ_k^0	Yes	No	No	No	No	Yes

†: for LOGSPACE or larger classes.

(green) is closure under finite intersection.

(orange) the intersection of a family of co-finite set can be anything.

Results

	Finite		Arithmetical		Arbitrary	
	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge/\vee$	complements
Automatic	Yes	Yes	No	Yes	No/Yes	N/A
Subrecursive	Yes	No[†]	?	No[†]	No	$\geq$ PSPACE
Σ^0_k	Yes	Yes	No	Yes	No	No
Π^0_k	Yes	No	Yes	No	No	?
Δ^0_k	Yes	No	No	No	No	Yes

[†]: for LOGSPACE or larger classes.

Results

	Finite		Arithmetical		Arbitrary	
	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge/\vee$	complements
Automatic	Yes	Yes	No	Yes	No/Yes	N/A
Subrecursive	Yes	No†	?	No†	No	$\geq$ PSPACE
Σ^0_k	Yes	Yes	No	Yes	No	No
Π^0_k	Yes	No	Yes	No	No	?
Δ^0_k	Yes	No	No	No	No	Yes

†: for LOGSPACE or larger classes.

Cherry-picking some proofs

Automatic Equivalences

Equivalences that are decidable by an automaton.

Problem: small changes in the model (number of tapes, heads, ...) actually change the class of languages recognised. Even a change of representation (interleaving inputs?) can change it.

Simple class: equivalences $\mathcal{E}$ such that $\{\, \mathtt{n}\square\mathtt{m} \ :\ n\mathcal{E}m \,\}$ is regular. It is a sublattice, but not a very interesting one ($\mathcal{E}$ must have only finitely many classes).

Undecidable Join

Theorem

There exists two LogSpace-*decidable equivalences whose join is undecidable.*

Sketch of Proof.

One-step transition: $\mathtt{c} \to \mathtt{c}'$ (deterministic TM).
Clocked one-step transition: $(\mathtt{n}, \mathtt{c}) \to (\mathtt{n}+1, \mathtt{c}')$.
Even one-step transition: $(2\mathtt{n}, \mathtt{c}) \to_{\text{even}} (2\mathtt{n}+1, \mathtt{c}')$.

Undecidable Join

Theorem

There exists two LogSpace*-decidable equivalences whose join is undecidable.*

Sketch of Proof.

One-step transition: $\mathtt{c} \to \mathtt{c}'$ (deterministic TM).
Clocked one-step transition: $(\mathtt{n}, \mathtt{c}) \to (\mathtt{n}+\mathtt{1}, \mathtt{c}')$.
Even one-step transition: $(\mathtt{2n}, \mathtt{c}) \to_{\text{even}} (\mathtt{2n}+\mathtt{1}, \mathtt{c}')$.

$\approx_{\text{even}}$, the transitive reflexive closure of $\to_{\text{even}}$, is LogSpace-decidable because we can't have $x \to_{\text{even}} y \to_{\text{even}} z$.
$(\mathtt{n}, \mathtt{c})(\approx_{\text{even}} \vee \approx_{\text{odd}})(\text{final state})$ iff the computation starting at $\mathtt{c}$ terminates, hence the join is not decidable. $\square$

Recursively enumerable Equivalences

Theorem

The set of recursively enumerable equivalences is a sublattice.

Closure under meet is easy (closure under intersection).

Closure under join.

$\mathcal{E}$ is recursively enumerable (Σ^0_1) iff $x\mathcal{E}y \Leftrightarrow \exists a / E(a, x, y)$, E decidable.

$x(\mathcal{E} \vee \mathcal{F})y$ iff
$\exists x_1, \ldots, x_n / x\mathcal{E}x_1\mathcal{F} \ldots \mathcal{E}x_n\mathcal{F}y$ iff
$\exists x_1, \ldots, x_n, a_0, \ldots, a_{n+1} / E(a_0, x, x_1)$ && $F(a_1, x_1, x_2)$ && … &&
$F(a_n, x_n, y)$ $\square$

Recursively enumerable Equivalences

Theorem

The join of a r.e. set of r.e. equivalences is a r.e. equivalence.

Idea.

We can enumerate all the equivalences needed to actually compute the join. □

Recursively enumerable Equivalences

Theorem

The join of a r.e. set of r.e. equivalences is a r.e. equivalence.

Idea.

We can enumerate all the equivalences needed to actually compute the join. □

Theorem

*There is a decidable set of recursively enumerable equivalences whose meet is **not** recursively enumerable.*

Idea.

The intersection of the formulae accepting i (for each i) is the set of tautologies. □

Subrecursive Complements

Theorem

Any PSPACE *(*EXPTIME*, …) equivalence has at least one* PSPACE *(*EXPTIME*, …) complement.*

Idea.

Consider $\mathcal{E}$, let $\mathcal{F}$ have only one non-singleton class, containing the smallest element of each class of $\mathcal{E}$. $m\mathcal{F}n$:

(i) if $m = n$, accept.

(ii) if $\forall k < m, k\overline{\mathcal{E}}n$, continue.

(iii) if $\forall k' < n, m\overline{\mathcal{E}}k'$, accept.

Subrecursive Complements

Theorem

Any PSPACE *(*EXPTIME*, . . .) equivalence has at least one* PSPACE *(*EXPTIME*, . . .) complement.*

Idea.

Consider $\mathcal{E}$, let $\mathcal{F}$ have only one non-singleton class, containing the smallest element of each class of $\mathcal{E}$. $m\mathcal{F}n$:
(i) if $m = n$, accept.
(ii) if $\forall k < m, k\overline{\mathcal{E}}n$, continue.
(iii) if $\forall k' < n, m\overline{\mathcal{E}}k'$, accept.
(ii) requires $O(|\mathtt{m}|)$ space and $O(2^{|\mathtt{m}|})$ time. □

Question

Do all PTIME equivalences have at least one PTIME complement?

Arithmetical Complements

Theorem

There exists a recursively enumerable equivalence who has no recursively enumerable complements.

Sketch of Proof.

E, r.e, $\overline{E}$ not r.e. and $\mathcal{E}$ with only non-singleton class E.
Classes of its complement, $\mathcal{F}$, intersect E in exactly one point.
$x \in \overline{E}$ iff $\exists e / e \neq x \;\&\&\; e \in E \;\&\&\; e\mathcal{F}x$. □

Conclusion

Conclusion

	Finite		Arithmetical		Arbitrary	
	$\wedge$	$\vee$	$\wedge$	$\vee$	$\wedge/\vee$	complements
Automatic	Yes	Yes	No	Yes	No/Yes	N/A
Subrecursive	Yes	No†	?	No†	No	$\geq$ PSPACE
Σ^0_k	Yes	Yes	No	Yes	No	No
Π^0_k	Yes	No	Yes	No	No	?
Δ^0_k	Yes	No	No	No	No	Yes

†: for LOGSPACE or larger classes.

None of these classes is very good at approximating the lattice :-(

The set of recursively enumerable equivalences might be the less bad candidate: it's a sublattice, and it's not too trivial.