On low for speed oracles

Laurent Bienvenu (CNRS & Université de Bordeaux)
Rod Downey (Victoria University of Wellington)

Ghent-Leeds Virtual Logic Seminar

November 19, 2020

Lowness for speed

An important concept in complexity theory and crucial in computability theory: relativization.

Lowness for speed

An important concept in complexity theory and crucial in computability theory: relativization.

Add an oracle A (= infinite binary sequence) available on a special read-only tape, which can be used for computations.

Lowness for speed

An important concept in complexity theory and crucial in computability theory: relativization.

Add an oracle A (= infinite binary sequence) available on a special read-only tape, which can be used for computations.

Computability theory: $\mathbf{HP}^A$ (also called A'), $\mathbf{DNC}^A$, $\mathbf{MLR}^A$, ...

Lowness for speed

An important concept in complexity theory and crucial in computability theory: relativization.

Add an oracle A (= infinite binary sequence) available on a special read-only tape, which can be used for computations.

Computability theory: $\mathbf{HP}^A$ (also called A'), $\mathbf{DNC}^A$, $\mathbf{MLR}^A$, …

Complexity theory: $\mathbf{P}^A$, $\mathbf{NP}^A$, …

Lowness for speed

Relativization in complexity can shed light on important questions:

Theorem (Baker, Gill, Solovay)

There are oracles A such that $\mathbf{P}^A = \mathbf{NP}^A$, and oracles B such that $\mathbf{P}^B \neq \mathbf{NP}^B$

Lowness for speed

Allender proposed to study **lowness for speed** (inspired by computability theory):

Definition (Allender)

X is **low for speed (l.f.s)** if every *decidable* set/language L that can be computed with oracle X in time f can be computed without oracle in time $poly(f)$.

(model of computation: Turing machine with a dedicated tape; the machine may write n on this tape then query the oracle X as to whether $n \in X$).

Lowness for speed

Allender proposed to study **lowness for speed** (inspired by computability theory):

Definition (Allender)

X is **low for speed (l.f.s)** if every *decidable* set/language L that can be computed with oracle X in time f can be computed without oracle in time $poly(f)$.

(model of computation: Turing machine with a dedicated tape; the machine may write n on this tape then query the oracle X as to whether $n \in X$).

This basically amounts to collapsing all time classes simultaneously: $\mathbf{P}^A = \mathbf{P}$, $\mathbf{NP}^A = \mathbf{NP}$, $\mathbf{EXPTIME}^A = \mathbf{EXPTIME}$, ...

Lowness for speed

Allender proposed to study **lowness for speed** (inspired by computability theory):

Definition (Allender)

X is **low for speed (l.f.s)** if every *decidable* set/language *L* that can be computed with oracle *X* in time *f* can be computed without oracle in time $poly(f)$.

(model of computation: Turing machine with a dedicated tape; the machine may write *n* on this tape then query the oracle *X* as to whether $n \in X$).

This basically amounts to collapsing all time classes simultaneously: $\mathbf{P}^A = \mathbf{P}$, $\mathbf{NP}^A = \mathbf{NP}$, $\mathbf{EXPTIME}^A = \mathbf{EXPTIME}$, ...

Does such an *A* exist?

Lowness for speed

Allender proposed to study **lowness for speed** (inspired by computability theory):

Definition (Allender)

X is **low for speed (l.f.s)** if every *decidable* set/language *L* that can be computed with oracle *X* in time *f* can be computed without oracle in time *poly*(f).

(model of computation: Turing machine with a dedicated tape; the machine may write *n* on this tape then query the oracle *X* as to whether $n \in X$).

This basically amounts to collapsing all time classes simultaneously: $\mathbf{P}^A = \mathbf{P}$, $\mathbf{NP}^A = \mathbf{NP}$, $\mathbf{EXPTIME}^A = \mathbf{EXPTIME}$, ...

Does such an *A* exist? Obviously yes: take *A* to be in PTIME-computable! (note: *X* computable but EXPTIME-complete would not work, so lowness for speed is **not** closed under $\equiv_T$).

Lowness for speed

Much less obvious: is there a non-computable A that is l.f.s. ?

Lowness for speed

Much less obvious: is there a non-computable A that is l.f.s. ?

Theorem (Bayer, Slaman)

There exists A non-computable and computably enumerable that is l.f.s.

Lowness for speed

Much less obvious: is there a non-computable A that is l.f.s. ?

Theorem (Bayer, Slaman)
There exists A non-computable and computably enumerable that is l.f.s.

Proof is a priority argument. One constructs A to be sparse, so that at stage t there are few candidates for $A \upharpoonright t$, thus for a functional Φ one can try to simulate all possible Φ^A in parallel (+ some very nice twist to handle Friedberg-Muchnik requirements).

Lowness for speed

Three directions for the study of lowness for speed:

1. What are the c.e. sets in LFS?

Lowness for speed

Three directions for the study of lowness for speed:

1. What are the c.e. sets in LFS?
2. What is the situation outside c.e. sets? How big is the set LFS in terms of cardinality/category/measure? (category answered by Bayer and Slaman)

Lowness for speed

Three directions for the study of lowness for speed:

1. What are the c.e. sets in LFS?
2. What is the situation outside c.e. sets? How big is the set LFS in terms of cardinality/category/measure? (category answered by Bayer and Slaman)
3. Closing under $\equiv_T$: what are the X that are equivalent to some low for speed? (note: every degree contains a non low for speed). Are such X closed downwards? under join?

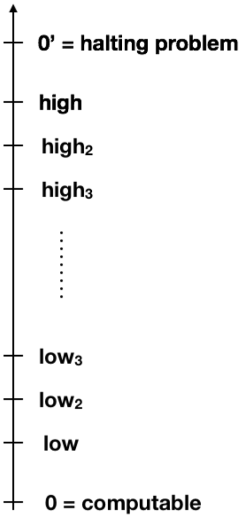
Within c.e. sets

Can we characterize the c.e. sets in LFS? Seems very hard, but one can get partial results.

One way to study LFS inside c.e. sets is with respect to the high/low hierarchy:

- A is *low* if $A' = \mathbf{0}'$; A is *low*$_n$ if $A^{(n)} = \mathbf{0}^{(n)}$.
- A is *high* if $A' = \mathbf{0}''$; A is *high*$_n$ if $A^{(n)} = \mathbf{0}^{(n+1)}$.

Strength


0' = halting problem
high
high2
high3
low3
low2
low
0 = computable

Within c.e. sets

We were able to prove:

Within c.e. sets

We were able to prove:

Theorem (BD)

If $A \geq_T \emptyset'$, then A is not l.f.s. (does not require A to be c.e.).

Within c.e. sets

We were able to prove:

Theorem (BD)
If $A \geq_T \emptyset'$, then A is not l.f.s. (does not require A to be c.e.).

Theorem (BD)
It is possible for A to be c.e., high and l.f.s. .

Within c.e. sets

We were able to prove:

Theorem (BD)

If $A \geq_T \emptyset'$, then A is not l.f.s. (does not require A to be c.e.).

Theorem (BD)

It is possible for A to be c.e., high and l.f.s. .

Theorem (BD)

If A is c.e., non-computable but low, it is necessarily **not** l.f.s. (!).

Within c.e. sets

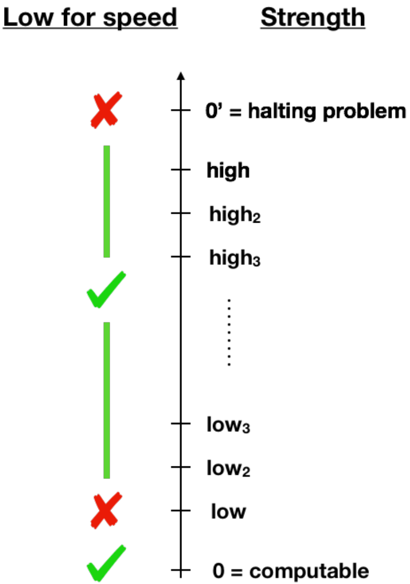
We were able to prove:

Theorem (BD)
If $A \geq_T \emptyset'$, then A is not l.f.s. (does not require A to be c.e.).

Theorem (BD)
It is possible for A to be c.e., high and l.f.s. .

Theorem (BD)
If A is c.e., non-computable but low, it is necessarily **not** l.f.s. (!).

Theorem (BD)
However, there is a c.e. set A which is non-computable, low_2, and l.f.s. .



Low for speed
Strength
0' = halting problem
high
high2
high3
low3
low2
low
0 = computable

Outside the c.e. world

How common are low for speed sets? Is the set LFS uncountable? co-meager? of measure 1?

Quick definitions

A set $\mathcal{U}$ of infinite binary sequence is **open** for the product topology if it can be written as:

$$\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$$

where *W* is a (countable) set of binary strings and $[\sigma]$ is the set of infinite binary sequences that start with σ.

Quick definitions

A set $\mathcal{U}$ of infinite binary sequence is **open** for the product topology if it can be written as:

$$\mathcal{U} = \bigcup_{\sigma \in W} [\sigma]$$

where W is a (countable) set of binary strings and $[\sigma]$ is the set of infinite binary sequences that start with σ.

We say that $\mathcal{U}$ is **effectively open** if W can be chosen to be computably enumerable (or computable)

Quick definitions

Effective Baire category:

Quick definitions

Effective Baire category:

- A sequence X is **weakly 1-generic** if for every dense effectively open set $\mathcal{U}$, we have that X is in $\mathcal{U}$.
- A sequence X is **1-generic** if for every effectively open set $\mathcal{U}$, we have $X \in \mathcal{U}$ or X is in the interior of the complement of $\mathcal{U}$.

Quick definitions

Effective measure theory:

Quick definitions

Effective measure theory:

- A sequence X is **Martin-Löf random** if for every sequence of uniformly effectively open sets $(\mathcal{U}_n)$ such that $\mu(\mathcal{U}_n) \leq 2^{-n}$, we have $X \notin \bigcap_n \mathcal{U}_n$
- A sequence X is **Schnorr random** if for every sequence of uniformly effectively open sets $(\mathcal{U}_n)$ such that $\mu(\mathcal{U}_n) = 2^{-n}$, we have $X \notin \bigcap_n \mathcal{U}_n$

The strange case of generics

So, back to our first question. Is the set LFS meager or co-meager?

The strange case of generics

So, back to our first question. Is the set LFS meager or co-meager?

Well...... it's complicated....

The strange case of generics

So, back to our first question. Is the set LFS meager or co-meager?

Well...... it's complicated....

Theorem (Bayer-Slaman)
LFS is meager **if and only if** $P \neq NP$.

The strange case of generics

So, back to our first question. Is the set LFS meager or co-meager?

Well...... it's complicated....

Theorem (Bayer-Slaman)
LFS is meager **if and only if** P $\neq$ NP.

So we might not know for a while whether LFS is meager or co-meager.

The strange case of generics

However,

Theorem (BD)
LFS contains a computably homeomorphic copy of the set of 1-generics (which is a co-meager set).

Therefore:

The strange case of generics

However,

Theorem (BD)
LFS contains a computably homeomorphic copy of the set of 1-generics (which is a co-meager set).

Therefore:

- LFS has size $2^{\aleph_0}$

The strange case of generics

However,

Theorem (BD)
LFS contains a computably homeomorphic copy of the set of 1-generics (which is a co-meager set).

Therefore:

- LFS has size $2^{\aleph_0}$
- Every non-computable c.e. set *computes* a l.f.s. set.

The strange case of generics

However,

Theorem (BD)
LFS contains a computably homeomorphic copy of the set of 1-generics (which is a co-meager set).

Therefore:

- LFS has size $2^{\aleph_0}$
- Every non-computable c.e. set *computes* a l.f.s. set.
- Almost every oracle (in the measure sense) *computes* a l.f.s. set.

The strange case of generics

However,

Theorem (BD)
LFS contains a computably homeomorphic copy of the set of 1-generics (which is a co-meager set).

Therefore:

- LFS has size $2^{\aleph_0}$
- Every non-computable c.e. set *computes* a l.f.s. set.
- Almost every oracle (in the measure sense) *computes* a l.f.s. set.
- There is a low Δ_2^0 set that is low for speed.

Randomness vs lowness for speed

Like for generics, one could expect a conditional behaviour of randoms w.r.t. lowness for speed, for example a dependance on the answer to $\mathrm{P} = \mathrm{BPP}$. This is not the case:

Randomness vs lowness for speed

Like for generics, one could expect a conditional behaviour of randoms w.r.t. lowness for speed, for example a dependance on the answer to $\mathrm{P} = \mathrm{BPP}$. This is not the case:

Theorem (BD)

If A is Schnorr random, it is not l.f.s.

Randomness vs lowness for speed

Like for generics, one could expect a conditional behaviour of randoms w.r.t. lowness for speed, for example a dependance on the answer to $\mathrm{P} = \mathrm{BPP}$. This is not the case:

Theorem (BD)
If A is Schnorr random, it is not l.f.s.

A Schnorr random can however be *equivalent* to a l.f.s. (take a l.f.s. of high degree).

Randomness vs lowness for speed

Like for generics, one could expect a conditional behaviour of randoms w.r.t. lowness for speed, for example a dependance on the answer to $\mathrm{P} = \mathrm{BPP}$. This is not the case:

Theorem (BD)
If A is Schnorr random, it is not l.f.s.

A Schnorr random can however be *equivalent* to a l.f.s. (take a l.f.s. of high degree). However, unlike for generics (assuming $\mathrm{P} \neq \mathrm{NP}$), the phenomenon disappears for Martin-Löf randomness. In fact:

Randomness vs lowness for speed

Like for generics, one could expect a conditional behaviour of randoms w.r.t. lowness for speed, for example a dependance on the answer to $\mathrm{P} = \mathrm{BPP}$. This is not the case:

Theorem (BD)
If A is Schnorr random, it is not l.f.s.

A Schnorr random can however be *equivalent* to a l.f.s. (take a l.f.s. of high degree). However, unlike for generics (assuming $\mathrm{P} \neq \mathrm{NP}$), the phenomenon disappears for Martin-Löf randomness. In fact:

Theorem (BD)
If A has Martin-Löf random **degree** (in fact, DNC degree is enough), it is not low for speed.

Proof inspired by Blum's speedup theorem.

Turing degrees and LFS

Some more results on the Turing degrees of l.f.s. sets.

Turing degrees and LFS

Some more results on the Turing degrees of l.f.s. sets.

Theorem (BD)

The Turing degrees of LFS are not closed downwards.

Proof: extend the earlier result to show that a low c.e. *degree* does not contain any l.f.s. set. Take a non-computable c.e. set X which is l.f.s. and apply Sack's splitting theorem to get a low c.e. Y with $0 <_T Y <_T X$.

Turing degrees and LFS

Some more results on the Turing degrees of l.f.s. sets.

Theorem (BD)

The Turing degrees of LFS are not closed downwards.

Proof: extend the earlier result to show that a low c.e. *degree* does not contain any l.f.s. set. Take a non-computable c.e. set X which is l.f.s. and apply Sack's splitting theorem to get a low c.e. Y with $0 <_T Y <_T X$.

How does lowness interact with minimality? We were able to prove

Theorem (BD)

There exists a minimal Turing degree which does not contain any l.f.s. set.

We conjectured that there is also a l.f.s. oracle of minimal Turing degree.......


BREAKING
NEWS

Turing degrees and LFS

A very recent result:

Theorem (Harrison-Trainor, Downey)
There exists a l.f.s. oracle of minimal Turing degree.

Thank you!