Consistency and Optimality

Computability in Europe, Sofia, 2011.

Moritz Müller

with Yijia Chen and Jörg Flum

S.-D. Friedman's question

Let $Q \subseteq \{0,1\}^*$ be a problem.

S.-D. Friedman's question

Let $Q \subseteq \{0, 1\}^*$ be a problem.

Let $T^* \supseteq T$ be arithmetical theories. Then T^* may verify faster algorithms for Q than T does.

S.-D. Friedman's question

Let $Q \subseteq \{0, 1\}^*$ be a problem.

Let $T^* \supseteq T$ be arithmetical theories. Then T^* may verify faster algorithms for Q than T does.

Question
Can $T^* = T \cup \{\mathit{Con}_T\}$ be understood as a "minimal" extension of T ?

S.-D. Friedman's question

Let $Q \subseteq \{0, 1\}^*$ be a problem.

Let $T^* \supseteq T$ be arithmetical theories. Then T^* may verify faster algorithms for Q than T does.

Question
Can $T^* = T \cup \{\mathit{Con}_T\}$ be understood as a "minimal" extension of T ?

Main Result
"Knowing Con_T means knowing some algorithm that is as fast as any algorithm T knows and knowing that it is that fast."

Optimality

A, B algorithms deciding $Q \subseteq \{0,1\}^*$.

A is as fast as B iff $t_A(x) \leq p(t_B(x) + |x|)$ for some polynomial p.

Optimality

A, B algorithms deciding $Q \subseteq \{0,1\}^*$.

A is as fast as B iff $t_A(x) \leq p(t_B(x) + |x|)$ for some polynomial p.

A is optimal iff it is as fast as any algorithm deciding Q,

> i.e. for all algorithms B deciding Q there is a polynomial p such that for all $x \in \{0,1\}^*$
>
> $t_A(x) \leq p(t_B(x) + |x|)$.

Open question

Is there a $Q \in \mathrm{NP} \setminus \mathrm{P}$ with an optimal algorithm?

Levin

NP search problems have optimal algorithms.

Krajíček, Pudlák, Sadowski

SAT has an optimal algorithm iff both SAT and TAUT have p-optimal proof systems.

Open question
Is there a $Q \in \mathsf{NP} \setminus \mathsf{P}$ with an optimal algorithm?

Levin
NP search problems have optimal algorithms.

Krajíček, Pudlák, Sadowski
SAT has an optimal algorithm iff both SAT and TAUT have p-optimal proof systems.

Blum, McCreight, Meyer
E-hard problems do not have optimal algorithms.

Theorem
There is a $Q \in \mathsf{E} \setminus \mathsf{P}$ with an optimal algorithm.

Theorem
Yes, if the Measure Hypothesis holds true.

Fast diagonal algorithms I

Observation
Assume $\mathcal{D}$ is a c.e. set of algorithms deciding Q. Then there is A deciding Q that is as fast as every $B \in \mathcal{D}$.

For $\mathcal{D} := \{B \mid B\ T\text{-provably decides}\ Q\}$ and c.e. T

...

Theories

Fix some decidable Q and A_0 deciding Q.

consider theories T in the language $\{+, \cdot, 0, 1, \leq\}$.

natural n is denoted by term $\dot{n}$.

Theories

Fix some decidable Q and A_0 deciding Q.

consider theories T in the language $\{+, \cdot, 0, 1, \leq\}$.

natural n is denoted by term $\dot{n}$.

B T-provably decides Q iff T proves "$\dot{B}$ decides Q".

"u decides Q" := u always halts and

$\forall x y y' z z' (\mathit{Run}(\dot{A}_0, x, y, z) \wedge \mathit{Run}(u, x, y', z') \to y = y')$

Fast diagonal algorithms II

Observation

Assume $\mathcal{D}$ is a c.e. set of algorithms deciding Q. Then there is A deciding Q that is as fast as every $B \in \mathcal{D}$.

For $\mathcal{D} := \{B \mid B\ T\text{-provably decides } Q\}$ and c.e. T

...get A as fast as any algorithm T-provably decides Q.

provided any $B \in \mathcal{D}$ decides Q.

T is sound for Q-decision: if B T-provably decides Q, then B decides Q.

T is sound for Q-decision: if B T-provably decides Q, then B decides Q.

T is complete for Q-decision: if B decides Q, then B T-provably decides Q.

T is sound for Q-decision: if B T-provably decides Q, then B decides Q.

T is complete for Q-decision: if B decides Q, then B T-provably decides Q.

Proposition

No c.e. theory is sound and complete for Q-decision.

T is sound for Q-decision: if B T-provably decides Q, then B decides Q.

T is complete for Q-decision: if B decides Q, then B T-provably decides Q.

Proposition
No c.e. theory is sound and complete for Q-decision.

Proposition
Q has an optimal algorithm $\iff$ there is a c.e. theory T that is sound and almost complete for Q-decision.

> for all B deciding Q there is A that T-provably decides Q and is as fast as B.

Fast diagonal algorithms III

Observation

Assume T is c.e. and sound for Q-decision.
Then there is A_T that is as fast as every B that T-provably decides Q and A_T decides Q.

Fast diagonal algorithms III

Observation
Assume T is c.e. and sound for Q-decision.
Then there is A_T that is as fast as every B that T-provably decides Q and A_T decides Q.

Lemma
Assume T is c.e. and consistent, Σ_1^0-complete.
Then there is A_T that is as fast as every B that T-provably decides Q and A_T decides Q.

Proof. Work with

$$\mathcal{D} := \{\text{“}B \text{ and } A_0 \text{ in parallel”} \mid B\ T\text{-provably decides } Q\}.$$

Fast diagonal algorithms III

Observation
Assume T is c.e. and sound for Q-decision.
Then there is A_T that is as fast as every B that T-provably decides Q and A_T decides Q.

Lemma
Assume $Q \notin \mathsf{P}$ and T is c.e. and Σ^0_1-complete.
Then there is A_T that is as fast as every B that T-provably decides Q and

T is consistent $\iff$ A_T decides Q.

Proof. Work with

$\mathcal{D} := \{$ "B and A_0 in parallel" $\mid B$ T-provably decides $Q\}$.

Characterization of Con_T.

Let $Q \notin \mathsf{P}$, decidable and let T_0 be a suitable finite, true theory.

Given a c.e. $T \supseteq T_0$ one can compute A_T such that

(1) A_T is as fast as every B that T-provably decides Q. Furthermore T_0 proves this.

(2) For every c.e. $T^* \supseteq T$:

T^* proves $\mathit{Con}_T \iff A_T$ T^*-provably decides Q

Characterization of *Con*$_T$.

Let $Q \notin$ P, decidable and let T_0 be a suitable finite, true theory.

Given a c.e. $T \supseteq T_0$ one can compute A_T such that

(1) A_T is as fast as every B that T-provably decides Q. Furthermore T_0 proves this.

(2) For every c.e. $T^* \supseteq T$:

T^* proves *Con*$_T \iff A_T$ T^*-provably decides Q

$\iff$ there is A that T^*-provably decides Q and T^* proves

$$\forall u \left(u \; T\text{-provably decides } Q \rightarrow \dot{A} \text{ is as fast as } u \right).$$

Special case I

Corollary
Assume ZFC is consistent. Then there is Q such that

(1) there is no algorithm that decides Q as fast as any algorithm deciding Q.

(2) there is an algorithm that decides Q as fast as any algorithm ZFC-provably deciding Q.

Special case II

Blum, Mc Creight, Meyer

If Q is E-hard, then there is a computable g such that

(1) if A decides Q, then so does $g(A)$.

(2) A is not as fast as $g(A)$.

Special case II

Blum, Mc Creight, Meyer
If Q is E-hard, then there is a computable g such that
(1) if A decides Q, then so does $g(A)$.
(2) A is not as fast as $g(A)$.

Corollary
There is a finite true T_1 such that for all consistent c.e. $T \supseteq T_1$:

$$T \nvdash \mathit{Con}_T.$$

Proof

Argue in T:

Con$_T$
$\rightarrow A_T$ decides Q (Lemma)
$\rightarrow g(A_T)$ decides Q (BMcCM)

Argue outside T:

$T \vdash$ *Con*$_T$
$\Rightarrow g(A_T)$ T-provably decides Q
$\Rightarrow A_T$ is as fast as $g(A_T)$ (Lemma)
$\Rightarrow A_T$ does not decide Q (BMcCM)
$\Rightarrow T$ is inconsistent (Lemma).

Thank you.