Analytic Functions and Small Complexity Classes

Holger Thies [1]

The University of Tokyo

June 15, 2016

[1] Joint work with Florian Steinberg

Motivation

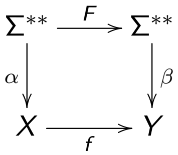
- Analytic Functions have been studied in real complexity theory in terms of polynomial time computability
- Implemented a data-type for analytic functions in iRRAM
- Many operations are done by some transformation on each coefficient of the power series
- This seems to be well-suited for parallelization
- How can we model this theoretically?

Representations

A representation for a set X is a partial surjective function $\rho : \Sigma^{**} \to X$, that is, objects are encoded by string functions.

Representations

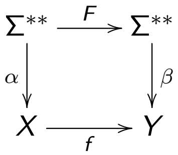
A representation for a set X is a partial surjective function $\rho : \Sigma^{**} \to X$, that is, objects are encoded by string functions.
How to compute with representations?

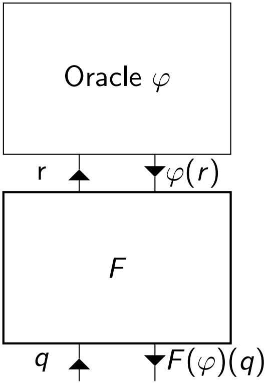


Representations

A representation for a set X is a partial surjective function $\rho : \Sigma^{**} \to X$, that is, objects are encoded by string functions.

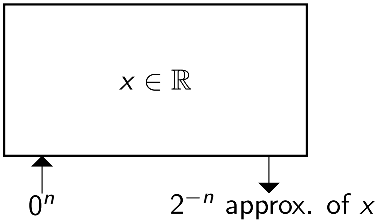
How to compute with representations?





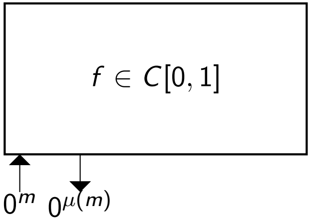
Example

Define a $\rho_{\mathbb{R}}$-name of $x \in \mathbb{R}$ by letting $\rho_{\mathbb{R}}(0^n)$ encode some $d \in \mathbb{D}$ such that $|d - x| \leq 2^{-n}$.



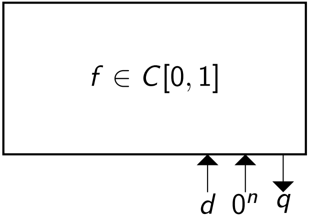
Example

Define a $\delta_{\square}$-name of a function $f \in C[0,1]$ as a pair $< \mu, \phi >$ where μ encodes the modulus of continuity (in unary) and ϕ is such that $|\phi(d, 0^n) - f(d)| \leq 2^{-n}$ for all $d \in \mathbb{D} \cap [0,1]$



Example

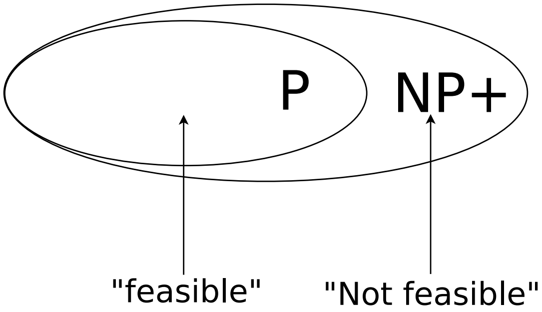
Define a $\delta_{\square}$-name of a function $f \in C[0,1]$ as a pair $< \mu, \phi >$ where μ encodes the modulus of continuity (in unary) and ϕ is such that $|\phi(d, 0^n) - f(d)| \leq 2^{-n}$ for all $d \in \mathbb{D} \cap [0,1]$



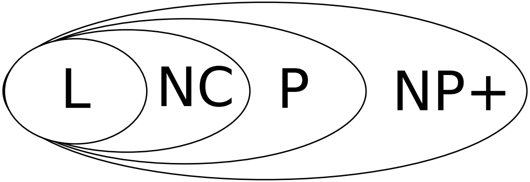
Type-2 Complexity Theory

- Let Σ^{**} be the set of length-monotone string-functions, i.e. s.t. $|x| \leq |y| \Rightarrow |\varphi(x)| \leq |\varphi(y)|$.
- Define $|\varphi| : \mathbb{N} \to \mathbb{N}$ by $|\varphi|(|u|) = |\varphi(u)|$
- Bound running time by second order polynomials $P(|\varphi|)(|x|)$
- Example: $P(L, n) = 2L(L(L(n)^4 + 2L(n)) + 2) + n + 10$
- Can define complexity classes FP^2, $\#\mathrm{P}^2$ and $\mathrm{FPSPACE}^2$
- Can also define complexity classes P^2, NP^2 and PSPACE^2 by considering functions $\varphi : \Sigma^{**} \to (\Sigma^* \to \{0,1\})$
- Type-2 classes are easy to seperate

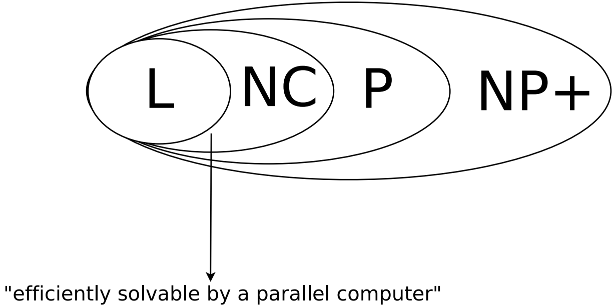
Small Complexity Classes



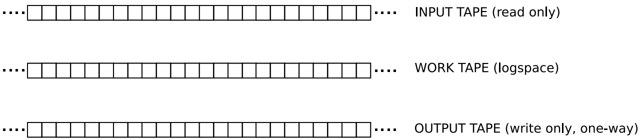
Small Complexity Classes



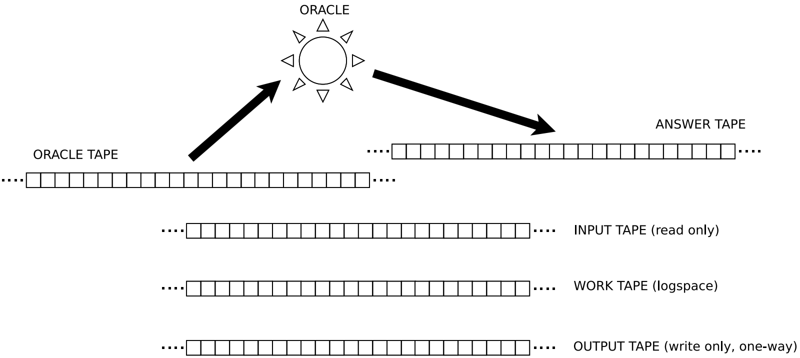
Small Complexity Classes



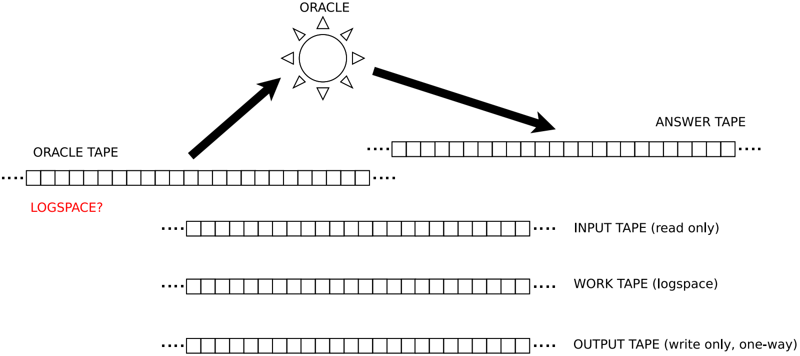
Small Complexity Classes



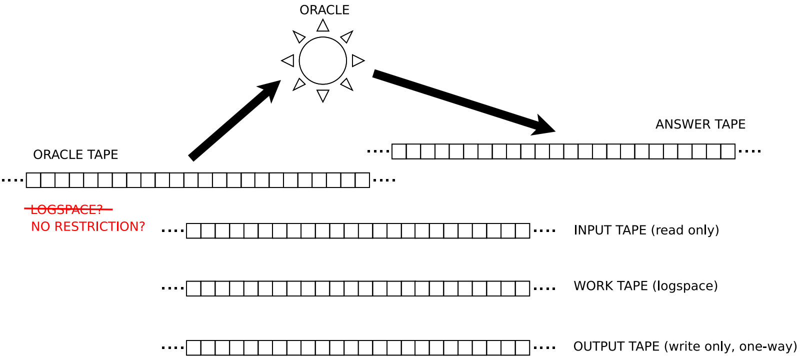
Small Complexity Classes



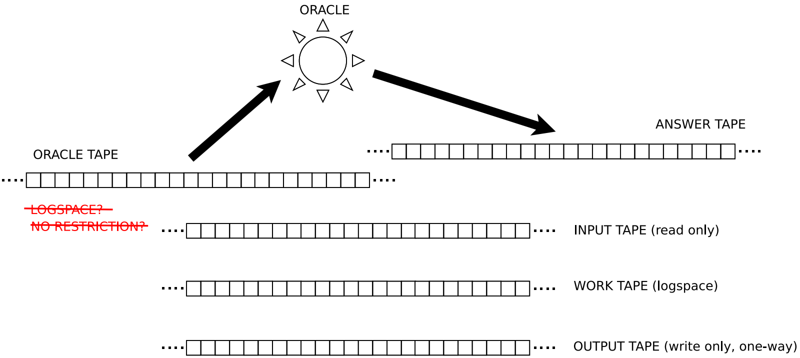
Small Complexity Classes



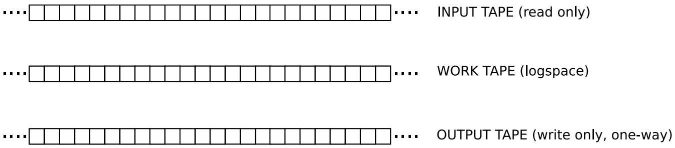
Small Complexity Classes



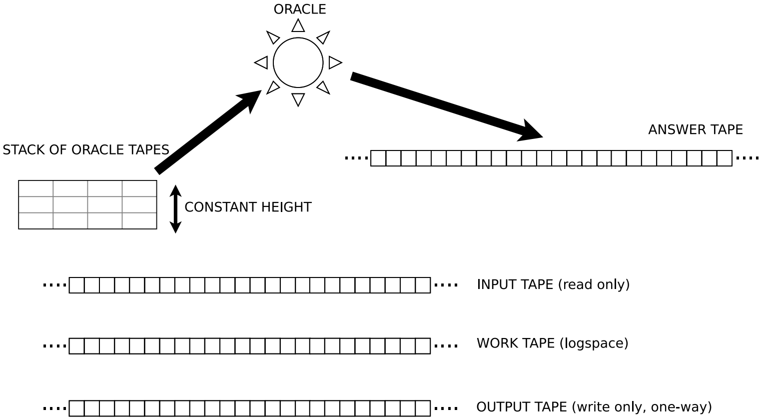
Small Complexity Classes



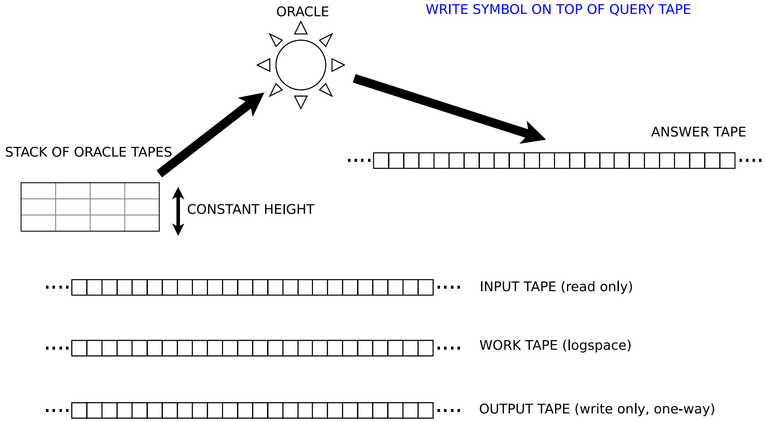
The stack model (Kawamura and Ota)



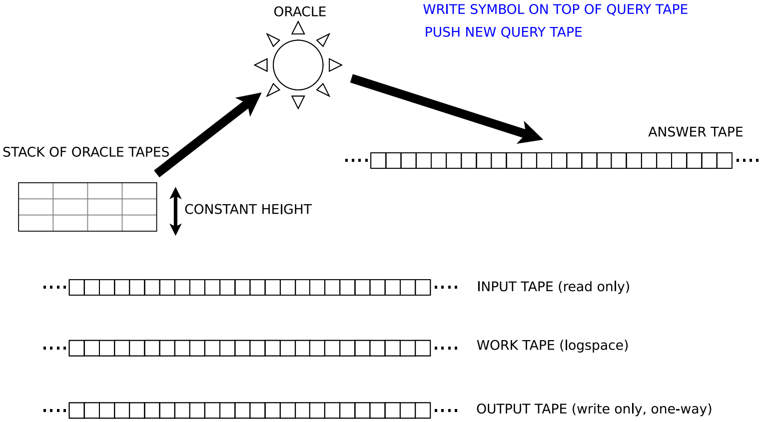
The stack model (Kawamura and Ota)



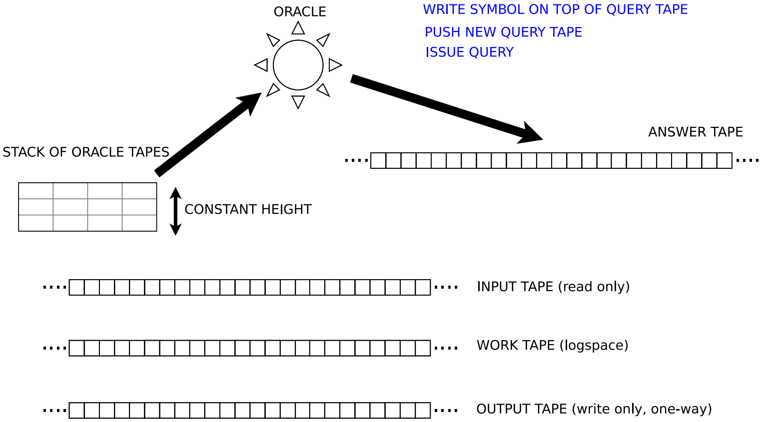
The stack model (Kawamura and Ota)



The stack model (Kawamura and Ota)



The stack model (Kawamura and Ota)



Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f, x) \mapsto f(x)$ is $([\delta_{\square}, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})-$FL computable.

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f,x) \mapsto f(x)$ is $([\delta_\square, \rho_\mathbb{R}^{|[0,1]}], \rho_\mathbb{R})$−FL computable.

`Input:`	1^n
`Stack:`	-
`Oracle Output:`	ε
`Work:`	ε
`Output:`	ε

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f,x) \mapsto f(x)$ is $([\delta_{\square}, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})$−FL computable.

`Input:`	1^n
`Stack:`	$\varepsilon;\varepsilon;\varepsilon$
`Oracle Output:`	ε
`Work:`	ε
`Output:`	ε

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f, x) \mapsto f(x)$ is $([\delta_\square, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})$−FL computable.

`Input:`	1^n
`Stack:`	1^{n+1};ε;ε
`Oracle Output:`	ε
`Work:`	ε
`Output:`	ε

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f, x) \mapsto f(x)$ is $([\delta_\Box, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})-$FL computable.

`Input:`	1^n
`Stack:`	$\varepsilon;\varepsilon$
`Oracle Output:`	1^m
`Work:`	ε
`Output:`	ε

$|x-y| \leq 2^{-m} \Rightarrow |f(x) - f(y)| \leq 2^{-(n+1)}$

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f, x) \mapsto f(x)$ is $([\delta_\square, \rho_\mathbb{R}^{|[0,1]}], \rho_\mathbb{R})$−FL computable.

`Input:`	1^n
`Stack:`	$1^m;\varepsilon$
`Oracle Output:`	1^m
`Work:`	ε
`Output:`	ε

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f, x) \mapsto f(x)$ is $([\delta_{\square}, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})$−FL computable.

`Input:`	1^n	
`Stack:`	ε	$\lvert x - d \rvert \leq 2^{-m}$
`Oracle Output:`	d	
`Work:`	ε	
`Output:`	ε	

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f,x) \mapsto f(x)$ is $([\delta_\square, \rho_\mathbb{R}^{|[0,1]}], \rho_\mathbb{R})$−FL computable.

`Input:`	1^n
`Stack:`	$d, 1^{n+1}$
`Oracle Output:`	d
`Work:`	ε
`Output:`	ε

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f,x) \mapsto f(x)$ is $([\delta_{\square}, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})$−FL computable.

`Input:`	1^n
`Stack:`	-
`Oracle Output:`	q
`Work:`	ε
`Output:`	ε

$|q - f(d)| \leq 2^{-(n+1)}$

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f,x) \mapsto f(x)$ is $([\delta_\square, \rho_\mathbb{R}^{|[0,1]}], \rho_\mathbb{R})$−FL computable.

`Input:`	1^n
`Stack:`	-
`Oracle Output:`	q
`Work:`	ε
`Output:`	q

$|q - f(x)| \leq 2^{-n}$

Example

The function Apply : $C[0,1] \times [0,1] \to \mathbb{R}$, $(f,x) \mapsto f(x)$ is $([\delta_{\Box}, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})$−FL computable.

Similarly,Apply^c : $C[0,1] \times [0,1] \to [0,1]$ $(f,x) \mapsto f^c(x)$ is $([\delta_{\Box}, \rho_{\mathbb{R}}^{|[0,1]}], \rho_{\mathbb{R}})$−FL computable for constant $c \in \mathbb{N}$ using a stack of size $2c+1$.

Complexity of Operators

Fact

For general polynomial time computable functions, many important operators have been shown to be computationally hard. For example

- *Polynomial time computable functions may have noncomputable derivatives. (Ko 1983)*
- *Parametric maximization is NP-hard. (Ko/Friedman (1982))*
- *Integration is #P-hard. (Friedman (1984))*

Thus, we can not expect to have efficient algorithms for those operators unless we restrict the possible functions.

Analytic Function

An analytic function is a function locally given by a complex power series.

Definition (Analytic Function)

$f : D \to \mathbb{C}$, $D \subseteq \mathbb{C}$ is analytic if for any $x_0 \in D$ the Taylor-series

$$T(x) := \sum_{n=0}^{\infty} a_n (x - x_0)^n$$

converges to $f(x)$ for x in a neighborhood of x_0.

Some non-uniform results

$$a_m = \frac{f^{(m)}(x_0)}{m!}, \ f(x) = \sum_{m=0}^{\infty} a_m (x - x_0)^k \quad \text{for } x \in B(x_0, R)$$

Theorem (Pour-El, Richards, Ko, Friedman, Müller (1987/1989))

f is (polytime) computable iff $(a_m)_{m \in \mathbb{N}}$ is.

Some non-uniform results

$$a_m = \frac{f^{(m)}(x_0)}{m!}, \ f(x) = \sum_{m=0}^{\infty} a_m (x - x_0)^k \quad \text{for } x \in B(x_0, R)$$

Theorem (Pour-El, Richards, Ko, Friedman, Müller (1987/1989))

f is (polytime) computable iff $(a_m)_{m \in \mathbb{N}}$ is.

From that polynomial time computability of the derivative and the anti-derivative of a function follows immediately.

Some non-uniform results

$$a_m = \frac{f^{(m)}(0)}{m!}, \quad f(x) = \sum_{m=0}^{\infty} a_m x^k \quad \text{for } x \in B(0, R)$$

Theorem (Müller (1995))

- *The operator* $f \mapsto (a_m)_{m \in \mathbb{N}}$ *is not computable.*
- *The evaluation operator* $((a_m)_{m \in \mathbb{N}}, x) \mapsto f(x)$ *is not computable.*

Some non-uniform results

$$a_m = \frac{f^{(m)}(0)}{m!}, \quad f(x) = \sum_{m=0}^{\infty} a_m x^k \quad \text{for } x \in B(0, R)$$

Theorem (Müller (1995))

- *The operator* $f \mapsto (a_m)_{m \in \mathbb{N}}$ *is not computable.*
- *The evaluation operator* $((a_m)_{m \in \mathbb{N}}, x) \mapsto f(x)$ *is not computable.*

However, if we supply some additional (discrete) information those operators become computable.

How to represent analytic functions?

Lemma

Let $f : \overline{B}(0,1) \to \mathbb{R}$ *be real analytic and* $(a_n)_{n\in\mathbb{N}}$ *its power series around* 0.

Then there exists an $k, A \in \mathbb{N}$ *such that*

1. $\sqrt[k]{2}$ *is a lower bound on the radius of convergence*
2. $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$

How to represent analytic functions?

Lemma

Let $f : \overline{B}(0,1) \to \mathbb{R}$ *be real analytic and* $(a_n)_{n\in\mathbb{N}}$ *its power series around* 0*.*
Then there exists an $k, A \in \mathbb{N}$ *such that*

1. $\sqrt[k]{2}$ *is a lower bound on the radius of convergence*
2. $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$

A and k can be used to make a tail estimate on

$$\left| \sum_{n\geq N} a_n z^n \right|$$

How to represent analytic functions?

Representation 1

A (length-monotone) function $\varphi : \Sigma^* \to \Sigma^*$ *is a name for a power series* $(a_k)_{k\in\mathbb{N}}$ *iff it is a concatenation of the following*

1. *An integer A encoded in binary*
2. *An integer k encoded in unary*
3. *A name for a sequence* $(a_k)_{k\in\mathbb{N}}$

Such that $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$ *for all* $n \in \mathbb{N}$.

How to represent analytic functions?

Representation 2

A (length-monotone) function $\varphi : \Sigma^* \to \Sigma^*$ *is a name for an analytic function* $f : \bar{B}(0,1) \to \mathbb{R}$ *iff it is a concatenation of the following*

1. *An integer* A *encoded in binary,*
2. *An integer* k *encoded in unary,*
3. *A name for the function* f

Such that f *extends analytically to* $B(0, \sqrt[k]{2})$ *and* $|f(z)| \leq A$ *for all* $z \in B(0, \sqrt[k]{2})$

Analytic Functions and Computational Complexity

Theorem (Kawamura, Rösnick, Müller, Ziegler (2013))

With the previous two representations the following operations can be performed in polynomial time

1. *evaluation*
2. *addition and multiplication*
3. *differentiation and anti-differentiation*
4. *parametric maximization*

Further, when identifying an analytic function with its power series, the operators that compute one representation from the other are polynomial-time computable.

Analytic Functions and Small Complexity Classes

Consider functions complex analytic on the closed unit disc.

Representation 1

Integers A, k and the series sequence $(a_k)_{k \in \mathbb{N}}$.
$|a_n| \leq A \cdot 2^{-\frac{n}{k}}$ *for all* $n \in \mathbb{N}$.

Representation 2

Integers A, k, and name for function f.
f *extends analytically to* $B(0, \sqrt[k]{2})$ *and* $|f(z)| \leq A$ *for all* $z \in B(0, \sqrt[k]{2})$

Those two representations are logspace equivalent.

Representation 1 $\Rightarrow$ Representation 2

Given A, k s.t. $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$ for all $n \in \mathbb{N}$. Need A', k' s.t. for all $z \in B(0, \sqrt[k']{2})$ $|f(z)| \leq A$

Representation 1 $\Rightarrow$ Representation 2

Given A, k s.t. $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$ for all $n \in \mathbb{N}$. Need A', k' s.t. for all $z \in B(0, \sqrt[k']{2})$ $|f(z)| \leq A$

Let $k' = 2k$ and $A' = 4kA$
For $z \in B(0, \sqrt[k']{2})$

$$f(z) = \sum_{n=0}^{\infty} a_n z^n \leq A \sum_{n=0}^{\infty} 2^{-\frac{n}{k}} \cdot 2^{\frac{n}{2k}} \leq 4Ak$$

Representation 1 $\Rightarrow$ Representation 2

$n \mapsto n + \log_2(A) + 2\log_2(k) + 5$ is a modulus of continuity for the function.

Representation 1 $\Rightarrow$ Representation 2

$n \mapsto n + \log_2(A) + 2\log_2(k) + 5$ is a modulus of continuity for the function.

Further, we need to evaluate the function.

Note that the following is logspace computable

1. Addition of polynomially (in n) many n-bit integers
2. Multiplication of polynomially (in n) many n-bit integers
3. Compute x^m with precision polynomial in n where m is an integer of length polynomial in n
4. Composition of a constant number of logspace computable functions

Representation 1 $\Rightarrow$ Representation 2

$n \mapsto n + \log_2(A) + 2\log_2(k) + 5$ is a modulus of continuity for the function.
Further, we need to evaluate the function.
Note that the following is logspace computable

1. Addition of polynomially (in n) many n-bit integers
2. Multiplication of polynomially (in n) many n-bit integers
3. Compute x^m with precision polynomial in n where m is an integer of length polynomial in n
4. Composition of a constant number of logspace computable functions

Thus $\sum_{j=0}^{poly(n)} a_j x^j$ is logspace computable.
We need to evaluate $N \approx n \cdot k + \log(A)$ terms.

Representation 2 $\Rightarrow$ Representation 1

Given A, k s.t. for all $z \in B(0, \sqrt[k']{2})$ $|f(z)| \leq A$
Need A', k' such that $|a_m| \leq A \cdot 2^{-\frac{m}{k}}$ for all $m \in \mathbb{N}$.
By Cauchy's integral formula $|a_m| = \frac{f^{(m)}(0)}{m!} \leq A \cdot 2^{-\frac{n}{k}}$ for all $n \in \mathbb{N}$.
Thus, we can just set $A' = A$ and $k' = k$.

Representation 2 $\Rightarrow$ Representation 1

Given A, k s.t. for all $z \in B(0, \sqrt[k']{2})$ $|f(z)| \leq A$
Need A', k' such that $|a_m| \leq A \cdot 2^{-\frac{m}{k}}$ for all $m \in \mathbb{N}$.
By Cauchy's integral formula $|a_m| = \frac{f^{(m)}(0)}{m!} \leq A \cdot 2^{-\frac{n}{k}}$ for all $n \in \mathbb{N}$.
Thus, we can just set $A' = A$ and $k' = k$.
Further we need to compute the coefficients for the power series around 0.
Approximating the derivatives can be done by first approximating the function by a polynomial and then differentiating this polynomial (Müller).
To get the coefficient a_m the function has to be evaluated at $2m + 1$ equidistant points with polynomial precision.
Note that Computing factorials and binomial coefficients of polynomial size is logspace computable.

Further Operations

Similarly, the following operations on analytic functions are computable in logartihmic-space

1. Addition, Subtraction, Multiplication of two analytic functions
2. Computing the d-fold derivative
3. Computing the d-fold anti-derivative

P-completeness

Kawamura and Ota also define the notions of reductions and completeness. For example they give the following uniform version of a theorem by Ko

Theorem

For the set M of bijective functions in $C[0,1]$, define the $\delta_{\Box INV}$ representation by adding a modulus of continuity for the inverse function to the $\delta_{\Box}$ representation.
The function $Inv : M \to C[0,1]$, $Inv(f) \mapsto f^{-1}$ is $(\delta_{\Box INV}, \delta_{\Box})$-FP-complete.

Future Work: P-completeness

Kawamura and Ota also define the notions of reductions and completeness.

Future Work: P-completeness

Kawamura and Ota also define the notions of reductions and completeness. For example they give the following uniform version of a theorem by Ko

Theorem

For the set M of bijective functions in $C[0,1]$, define the $\delta_{\square INV}$ representation by adding a modulus of continuity for the inverse function to the $\delta_{\square}$ representation.
The function $Inv : M \to C[0,1]$, $Inv(f) \mapsto f^{-1}$ is $(\delta_{\square INV}, \delta_{\square})$-FP-complete.

Future Work: P-completeness

Kawamura and Ota also define the notions of reductions and completeness. For example they give the following uniform version of a theorem by Ko

Theorem

For the set M of bijective functions in $C[0,1]$, define the $\delta_{\square INV}$ representation by adding a modulus of continuity for the inverse function to the $\delta_{\square}$ representation.
The function $Inv : M \to C[0,1]$, $Inv(f) \mapsto f^{-1}$ is $(\delta_{\square INV}, \delta_{\square})$-FP-complete.

Which operations on analytic functions are P-hard?

- Initial Value Problem?
- Maximization?

Conclusion

- Presented Kawamura and Ota's model for logspace computability in analysis.
- In this model many operations on analytic functions are logspace computable, when considering representations that have previously been considered for polynomial time computability.
- Open Problems: Parametrized Maximization, Ordinary Differential Equations
- Connection to parallelization in exact real arithmetic
- Implementations