# Some New Approaches to Characterizing Computable Analysis by Analog Computation

Kerry Ojakian[1] (with Manuel L. Campagnolo[2])

[1]SQIG/IT Lisbon and IST, Portugal
ojakian@math.ist.utl.pt

[2]DM/ISA, Lisbon University of Technology and SQIG/IT Lisbon

Computability in Europe, 2007

# Outline

1. Introduction
2. Technical Framework
3. Results
   - Real Recursive Functions
   - General Purpose Analog Computer
4. Aspects of the proofs
5. Conclusion

# Outline

1. Introduction
2. Technical Framework
3. Results
   - Real Recursive Functions
   - General Purpose Analog Computer
4. Aspects of the proofs
5. Conclusion

# Characterizing Computable Analysis.

## Previous Work

- Results of the form **C**$(\mathbb{R}) =$ "Analog″.
- Real Recursive Functions introduced by C. Moore 1996.
  Function algebras with operations like this:

  *Solve a differential equation and keep the result.*

  Modified by Bournez and Hainry 2005, 2006.
- General Purpose Analog Computer (GPAC) introduced by Shannon 1941.
  Analog circuit model with gates that solve differential equations.
  Extended by Graça, Bournez, Campagnolo, Hainry 2007.

# Characterizing Computable Analysis.

## Previous Work

- Results of the form $\mathbf{C}(\mathbb{R}) =$ "Analog″.
- Real Recursive Functions introduced by C. Moore 1996.
  Function algebras with operations like this:

  *Solve a differential equation and keep the result.*

  Modified by Bournez and Hainry 2005, 2006.
- General Purpose Analog Computer (GPAC) introduced by Shannon 1941.
  Analog circuit model with gates that solve differential equations.
  Extended by Graça, Bournez, Campagnolo, Hainry 2007.

# Characterizing Computable Analysis.

## Previous Work

- Results of the form $\mathbf{C}(\mathbb{R}) =$ "Analog".
- Real Recursive Functions introduced by C. Moore 1996.
  Function algebras with operations like this:

  *Solve a differential equation and keep the result.*

  Modified by Bournez and Hainry 2005, 2006.
- General Purpose Analog Computer (GPAC) introduced by Shannon 1941.
  Analog circuit model with gates that solve differential equations.
  Extended by Graça, Bournez, Campagnolo, Hainry 2007.

# Motivation

- Church-Turing style thesis.
  - Many distinct models of computation on the reals: Computable Analysis, Real Recursive Functions, GPAC, BSS machines, Neural Networks, Dynamic Systems, ...
  - How are the models distinct?
  - What kinds of modifications make them equal?
- Applications in discrete complexity theory.
  - Can hard separation questions (e.g. P versus NP) be reduced to relevant questions in analysis?
  - For example, can discrete separations be reduced to "analytic" separations?
  - Work in this direction: Costa and Mycka 2006, 2007

# Motivation

- Church-Turing style thesis.
  - Many distinct models of computation on the reals: Computable Analysis, Real Recursive Functions, GPAC, BSS machines, Neural Networks, Dynamic Systems, ...
  - How are the models distinct?
  - What kinds of modifications make them equal?
- Applications in discrete complexity theory.
  - Can hard separation questions (e.g. P versus NP) be reduced to relevant questions in analysis?
  - For example, can discrete separations be reduced to "analytic" separations?
  - Work in this direction: Costa and Mycka 2006, 2007

# Outline

1 Introduction

2 Technical Framework

3 Results
- Real Recursive Functions
- General Purpose Analog Computer

4 Aspects of the proofs

5 Conclusion

# Computable Analysis

## Definition

- $f(x) \in \mathbf{C}(\mathbb{R})$:

  *There is a computable function $F^x(n)$ with an oracle for the real number $x$ such that $|f(x) - F^x(n)| \leq 1/n$.*

- $\mathbf{E}(\mathbb{R})$: Like $\mathbf{C}(\mathbb{R})$, replacing computable by elementary computable.

# Computable Analysis

## Definition

- $f(x) \in \mathbf{C}(\mathbb{R})$:

  *There is a computable function $F^x(n)$ with an oracle for the real number $x$ such that $|f(x) - F^x(n)| \leq 1/n$.*

- $\mathbf{E}(\mathbb{R})$: Like $\mathbf{C}(\mathbb{R})$, replacing computable by elementary computable.

# Approximation and Completion

Goal: $\mathbf{C}(\mathbb{R}) = \mathrm{A(LIM)}$, broken into 2 steps:

- (Approximation) $\mathbf{C}(\mathbb{R}) \approx \mathrm{A}$.
- (Completion) $\mathbf{C}(\mathbb{R}) = \mathrm{A(LIM)}$.

# Approximation Relation

## Definition

$\mathcal{A} \preceq^{\mathcal{E}} \mathcal{B}$ iff

*For every $f(\bar{x}) \in \mathcal{A}$, and every $\alpha(\bar{x}, \bar{y}) \in \mathcal{E}$, there is $f^*(\bar{x}, \bar{y}) \in \mathcal{B}$ such that $|f(\bar{x}) - f^*(\bar{x}, \bar{y}))| \leq \alpha(\bar{x}, \bar{y})$.*

$\mathcal{A} \approx^{\mathcal{E}} \mathcal{B}$ iff $\mathcal{A} \preceq^{\mathcal{E}} \mathcal{B}$ and $\mathcal{B} \preceq^{\mathcal{E}} \mathcal{A}$.

## Lemma (Transitivity)

*If $\mathcal{A} \preceq^{\mathcal{E}} \mathcal{B}$ and $\mathcal{B} \preceq^{\mathcal{E}} \mathcal{C}$ then $\mathcal{A} \preceq^{\mathcal{E}} \mathcal{C}$.*

# Completion Operation

## Definition

LIM is the operation:

- Input: $f(t, \bar{x})$
- Output: $F(\bar{x}) = \lim_{t \to \infty} f(t, \bar{x})$, if the limit exists and $F \preceq^{1/t} f$, for positive $t$

## Definition

If OP is an operation and $\mathcal{F}$ a set of functions, then $\mathcal{F}(\text{OP})$ is:

$$\mathcal{F} \cup \{\text{OP}(f) \mid f \in \mathcal{F}\}$$

Thus $\mathcal{F}(\text{LIM})$ is a "completion" of $\mathcal{F}$.

# Completion Operation

**Definition**

LIM is the operation:

- Input: $f(t, \bar{x})$
- Output: $F(\bar{x}) = \lim_{t \to \infty} f(t, \bar{x})$, if the limit exists and $F \preceq^{1/t} f$, for positive $t$

**Definition**

If OP is an operation and $\mathcal{F}$ a set of functions, then $\mathcal{F}(\text{OP})$ is:

$$\mathcal{F} \cup \{\text{OP}(f) \mid f \in \mathcal{F}\}$$

Thus $\mathcal{F}(\text{LIM})$ is a "completion" of $\mathcal{F}$.

# Outline

1. Introduction
2. Technical Framework
3. Results
   - Real Recursive Functions
   - General Purpose Analog Computer
4. Aspects of the proofs
5. Conclusion

# Outline

1 Introduction

2 Technical Framework

3 Results
- Real Recursive Functions
- General Purpose Analog Computer

4 Aspects of the proofs

5 Conclusion

# Function Algebras

**Definition**

Suppose $\mathcal{B}$ is a set of functions (i.e. the basic functions) and $\mathcal{O}$ is a set of operations. Then FA$[\mathcal{B}; \mathcal{O}]$ is the smallest set of functions containing $\mathcal{B}$ and closed under $\mathcal{O}$.

Basic Functions:

- Constant functions: $0, 1, -1, \pi$
- Projection functions "P" (example: $U(x, y) = x$)
- $\theta(x) = \begin{cases} 0, & x < 0; \\ x^3, & x \geq 0. \end{cases}$

# Function Algebras

## Definition

Suppose $\mathcal{B}$ is a set of functions (i.e. the basic functions) and $\mathcal{O}$ is a set of operations. Then FA[$\mathcal{B}$; $\mathcal{O}$] is the smallest set of functions containing $\mathcal{B}$ and closed under $\mathcal{O}$.

Basic Functions:

- Constant functions: $0, 1, -1, \pi$
- Projection functions "P" (example: $U(x, y) = x$)
- $\theta(x) = \begin{cases} 0, & x < 0; \\ x^3, & x \geq 0. \end{cases}$

# The Operations

"comp": Composes the given functions.

**Definition**

LI is the operation:

- Input: Functions: $\overrightarrow{\mathbf{g}}(\bar{x})$, $\overrightarrow{\mathbf{f}}(y, \bar{u}, \bar{x})$, linear in $\bar{u}$.
- Output: $h_1(y, \bar{x})$ where $(h_1, \ldots, h_n)$ is the solution to the IVP:

$$\overrightarrow{\mathbf{h}}(0, \bar{x}) = \overrightarrow{\mathbf{g}}(\bar{x})$$
$$\frac{\partial}{\partial y}\overrightarrow{\mathbf{h}} = \overrightarrow{\mathbf{f}}(y, \overrightarrow{\mathbf{h}}, \bar{x})$$

# The Operations

"comp": Composes the given functions.

## Definition

LI is the operation:

- Input: Functions: $\overrightarrow{\mathbf{g}}(\bar{x})$, $\overrightarrow{\mathbf{f}}(y, \bar{u}, \bar{x})$, linear in $\bar{u}$.
- Output: $h_1(y, \bar{x})$ where $(h_1, \ldots, h_n)$ is the solution to the IVP:

$$\overrightarrow{\mathbf{h}}(0, \bar{x}) = \overrightarrow{\mathbf{g}}(\bar{x})$$
$$\frac{\partial}{\partial y}\overrightarrow{\mathbf{h}} = \overrightarrow{\mathbf{f}}(y, \overrightarrow{\mathbf{h}}, \bar{x})$$

# Elementary Computability.

Let $\mathcal{L}$ abbreviate $\mathrm{FA}[0, 1, -1, \pi, \theta, \mathrm{P}; \mathrm{comp}, \mathrm{LI}]$.
Let $\mathcal{L}^{\mathrm{a}}$ abbreviate $\mathrm{FA}[0, 1, -1, \mathrm{P}; \mathrm{comp}, \mathrm{LI}]$.

**Theorem**

- *(Approximation)* $\mathbf{E}(\mathbb{R}) \approx \mathcal{L} \approx \mathcal{L}^{\mathrm{a}}$
- *(Completion)* $\mathbf{E}(\mathbb{R}) = \mathcal{L}(LIM) = \mathcal{L}^{\mathrm{a}}(LIM)$
- *(Alternative Completion)* $\mathbf{E}(\mathbb{R}) = \mathcal{L}(dLIM)$

**Definition**

dLIM is the operation:

- Input: $f(t, \bar{x})$
- Output: $F(\bar{x}) = \lim_{t\to\infty} f(t, \bar{x})$, if $|\frac{\partial}{\partial t} f| \leq 1/2^t$ for $t \geq 1$.

# Elementary Computability.

Let $\mathcal{L}$ abbreviate $\text{FA}[0, 1, -1, \pi, \theta, \text{P}; \text{comp}, \text{LI}]$.
Let $\mathcal{L}^{\text{a}}$ abbreviate $\text{FA}[0, 1, -1, \text{P}; \text{comp}, \text{LI}]$.

## Theorem

- *(Approximation)* $\mathbf{E}(\mathbb{R}) \approx \mathcal{L} \approx \mathcal{L}^{\text{a}}$
- *(Completion)* $\mathbf{E}(\mathbb{R}) = \mathcal{L}(LIM) = \mathcal{L}^{\text{a}}(LIM)$
- *(Alternative Completion)* $\mathbf{E}(\mathbb{R}) = \mathcal{L}(dLIM)$

## Definition

dLIM is the operation:

- Input: $f(t, \bar{x})$
- Output: $F(\bar{x}) = \lim_{t \to \infty} f(t, \bar{x})$, if $|\frac{\partial}{\partial t} f| \leq 1/2^t$ for $t \geq 1$.

# Elementary Computability.

Let $\mathcal{L}$ abbreviate $\text{FA}[0, 1, -1, \pi, \theta, \text{P}; \text{comp}, \text{LI}]$.
Let $\mathcal{L}^{\text{a}}$ abbreviate $\text{FA}[0, 1, -1, \text{P}; \text{comp}, \text{LI}]$.

## Theorem

- *(Approximation)* $\mathbf{E}(\mathbb{R}) \approx \mathcal{L} \approx \mathcal{L}^{\text{a}}$
- *(Completion)* $\mathbf{E}(\mathbb{R}) = \mathcal{L}(LIM) = \mathcal{L}^{\text{a}}(LIM)$
- *(Alternative Completion)* $\mathbf{E}(\mathbb{R}) = \mathcal{L}(dLIM)$

## Definition

dLIM is the operation:

- Input: $f(t, \bar{x})$
- Output: $F(\bar{x}) = \lim_{t\to\infty} f(t, \bar{x})$, if $|\frac{\partial}{\partial t} f| \leq 1/2^t$ for $t \geq 1$.

# Computability.

Bournez and Hainry 2005, 2006.

**Theorem**

- *(Approximation)* $\mathbf{C}(\mathbb{R}) \approx FA[0, 1, \theta, P; comp, CLI, UMU]$
- *(Completion)*

$$\begin{aligned} \mathbf{C}(\mathbb{R}) &= FA[0, 1, \theta, P; comp, CLI, UMU](LIM) \\ &= FA[0, 1, \theta, P; comp, CLI, UMU](dLIM) \end{aligned}$$

# Outline

1 Introduction
2 Technical Framework
3 Results
- Real Recursive Functions
- General Purpose Analog Computer
4 Aspects of the proofs
5 Conclusion

# GPAC Generability

## Definition

Let PI be the operation:

- Input: $n-1$ polynomials: $\overrightarrow{\mathbf{p}}(\bar{y}, t)$, a polynomial $q(x)$, and numbers $\alpha_1, \ldots, \alpha_{n-1} \in \mathbb{R}$.
- Output: $y_1(t, x)$ where: $(y_1, ..., y_n)$ is the solution of IVP:

$$\overrightarrow{\mathbf{y}}(0) = (\alpha_1, ..., \alpha_{n-1}, q(x))$$
$$\frac{\partial}{\partial t}\overrightarrow{\mathbf{y}} = \overrightarrow{\mathbf{p}}(\overrightarrow{\mathbf{y}}, t)$$

## Definition

For $X \subseteq \mathbb{R}$, let $\text{GPAC}_X$ be the set of functions generated by PI using polynomials with coefficients from $X$ and initial conditions from $X$.

# Result

Let $\mathcal{CR}$ be the set of computable reals.
Graça, Bournez, Campagnolo and Hainry 2007.

## Theorem

- *(Approximation)* $\mathbf{C}(\mathbb{R}) \approx^{\mathcal{E}} GPAC_{\mathcal{CR}}$
- *(Completion)* $\mathbf{C}(\mathbb{R}) = GPAC_{\mathcal{CR}}(\mathcal{E}{-}LIM)$

# Outline

1. Introduction
2. Technical Framework
3. Results
   - Real Recursive Functions
   - General Purpose Analog Computer
4. Aspects of the proofs
5. Conclusion

# Lifting.

Example: $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}$.

- Reduce to the classic claim: $\mathbf{E}(\mathbb{N}) = \mathrm{FA}_{\mathbb{N}}$. How?
- Define function classes on $\mathbb{Q}$, as bridges:
  - Continuous layer: $\mathrm{FA}_{\mathbb{Q}}(\mathrm{ctn})$
  - Discontinuous layer: $\mathrm{FA}_{\mathbb{Q}}(\mathrm{disctn})$, $\mathbf{E}(\mathbb{Q})$
- $\mathrm{FA}_{\mathbb{Q}}(\mathrm{ctn}) \approx \mathcal{L}$.
  $\{\mathbf{E}(\mathbb{Q}), \mathrm{FA}_{\mathbb{Q}}(\mathrm{disctn})\}$ relates to $\{\mathbf{E}(\mathbb{N}), \mathrm{FA}_{\mathbb{N}}\}$.
- $\{\mathbf{E}(\mathbb{R}), \mathrm{FA}_{\mathbb{Q}}(\mathrm{ctn})\} \approx \{\mathbf{E}(\mathbb{Q}), \mathrm{FA}_{\mathbb{Q}}(\mathrm{disctn})\}$, for functions with modulus.
- Apply transitivity of approximation.

# Lifting.

Example: $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}$.

- Reduce to the classic claim: $\mathbf{E}(\mathbb{N}) = \mathrm{FA}_{\mathbb{N}}$. How?
- Define function classes on $\mathbb{Q}$, as bridges:
  - Continuous layer: $\mathrm{FA}_{\mathbb{Q}}(\mathrm{ctn})$
  - Discontinuous layer: $\mathrm{FA}_{\mathbb{Q}}(\mathrm{disctn})$, $\mathbf{E}(\mathbb{Q})$
- $\mathrm{FA}_{\mathbb{Q}}(\mathrm{ctn}) \approx \mathcal{L}$.
  $\{\mathbf{E}(\mathbb{Q}), \mathrm{FA}_{\mathbb{Q}}(\mathrm{disctn})\}$ relates to $\{\mathbf{E}(\mathbb{N}), \mathrm{FA}_{\mathbb{N}}\}$.
- $\{\mathbf{E}(\mathbb{R}), \mathrm{FA}_{\mathbb{Q}}(\mathrm{ctn})\} \approx \{\mathbf{E}(\mathbb{Q}), \mathrm{FA}_{\mathbb{Q}}(\mathrm{disctn})\}$, for functions with modulus.
- Apply transitivity of approximation.

# Lifting.

Example: $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}$.

- Reduce to the classic claim: $\mathbf{E}(\mathbb{N}) = \text{FA}_{\mathbb{N}}$. How?
- Define function classes on $\mathbb{Q}$, as bridges:
  - Continuous layer: $\text{FA}_{\mathbb{Q}}(\text{ctn})$
  - Discontinuous layer: $\text{FA}_{\mathbb{Q}}(\text{disctn})$, $\mathbf{E}(\mathbb{Q})$
- $\text{FA}_{\mathbb{Q}}(\text{ctn}) \approx \mathcal{L}$.
  $\{\mathbf{E}(\mathbb{Q}), \text{FA}_{\mathbb{Q}}(\text{disctn})\}$ relates to $\{\mathbf{E}(\mathbb{N}), \text{FA}_{\mathbb{N}}\}$.
- $\{\mathbf{E}(\mathbb{R}), \text{FA}_{\mathbb{Q}}(\text{ctn})\} \approx \{\mathbf{E}(\mathbb{Q}), \text{FA}_{\mathbb{Q}}(\text{disctn})\}$, for functions with modulus.
- Apply transitivity of approximation.

# Lifting.

Example: $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}$.

- Reduce to the classic claim: $\mathbf{E}(\mathbb{N}) = \text{FA}_{\mathbb{N}}$. How?
- Define function classes on $\mathbb{Q}$, as bridges:
  - Continuous layer: $\text{FA}_{\mathbb{Q}}(\text{ctn})$
  - Discontinuous layer: $\text{FA}_{\mathbb{Q}}(\text{disctn})$, $\mathbf{E}(\mathbb{Q})$
- $\text{FA}_{\mathbb{Q}}(\text{ctn}) \approx \mathcal{L}$.
  $\{\mathbf{E}(\mathbb{Q}), \text{FA}_{\mathbb{Q}}(\text{disctn})\}$ relates to $\{\mathbf{E}(\mathbb{N}), \text{FA}_{\mathbb{N}}\}$.
- $\{\mathbf{E}(\mathbb{R}), \text{FA}_{\mathbb{Q}}(\text{ctn})\} \approx \{\mathbf{E}(\mathbb{Q}), \text{FA}_{\mathbb{Q}}(\text{disctn})\}$, for functions with modulus.
- Apply transitivity of approximation.

# Lifting.

Example: $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}$.

- Reduce to the classic claim: $\mathbf{E}(\mathbb{N}) = \text{FA}_{\mathbb{N}}$. How?
- Define function classes on $\mathbb{Q}$, as bridges:
  - Continuous layer: $\text{FA}_{\mathbb{Q}}(\text{ctn})$
  - Discontinuous layer: $\text{FA}_{\mathbb{Q}}(\text{disctn})$, $\mathbf{E}(\mathbb{Q})$
- $\text{FA}_{\mathbb{Q}}(\text{ctn}) \approx \mathcal{L}$.
  $\{\mathbf{E}(\mathbb{Q}), \text{FA}_{\mathbb{Q}}(\text{disctn})\}$ relates to $\{\mathbf{E}(\mathbb{N}), \text{FA}_{\mathbb{N}}\}$.
- $\{\mathbf{E}(\mathbb{R}), \text{FA}_{\mathbb{Q}}(\text{ctn})\} \approx \{\mathbf{E}(\mathbb{Q}), \text{FA}_{\mathbb{Q}}(\text{disctn})\}$, for functions with modulus.
- Apply transitivity of approximation.

# Exploiting Approximation.

Transitivity:

- A basic point in Lifting.
- A way to break down an argument into simpler pieces.
  Ex: Elimination of non-analytic functions:
  $\mathbf{E}(\mathbb{R}) \approx \mathcal{L} \approx \mathcal{L}^{\mathrm{a}}$ implies $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}^{\mathrm{a}}$.

  *Recall:* $\mathcal{L}$ *is* $FA[0, 1, -1, \pi, \theta, P; comp, LI]$.
  $\mathcal{L}^{\mathrm{a}}$ *is* $FA[0, 1, -1, P; comp, LI]$.
  *To show* $\mathcal{L} \preceq \mathcal{L}^{\mathrm{a}}$*:*

  - $\theta, \pi \preceq \mathcal{L}^{\mathrm{a}}$
  - $comp, LI \preceq \mathcal{L}^{\mathrm{a}}$

General tools:

- Re-usable tools
  Ex: Approximating an operation.

# Exploiting Approximation.

Transitivity:

- A basic point in Lifting.
- A way to break down an argument into simpler pieces.
  Ex: Elimination of non-analytic functions:
  $\mathbf{E}(\mathbb{R}) \approx \mathcal{L} \approx \mathcal{L}^{\mathrm{a}}$ implies $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}^{\mathrm{a}}$.

  *Recall:* $\mathcal{L}$ *is* $FA[0, 1, -1, \pi, \theta, P; comp, LI]$.
  $\mathcal{L}^{\mathrm{a}}$ *is* $FA[0, 1, -1, P; comp, LI]$.
  *To show* $\mathcal{L} \preceq \mathcal{L}^{\mathrm{a}}$*:*

  - $\theta, \pi \preceq \mathcal{L}^{\mathrm{a}}$
  - $comp, LI \preceq \mathcal{L}^{\mathrm{a}}$

General tools:

- Re-usable tools
  Ex: Approximating an operation.

# Exploiting Approximation.

Transitivity:

- A basic point in Lifting.
- A way to break down an argument into simpler pieces.
  Ex: Elimination of non-analytic functions:
  $\mathbf{E}(\mathbb{R}) \approx \mathcal{L} \approx \mathcal{L}^{\mathrm{a}}$ implies $\mathbf{E}(\mathbb{R}) \approx \mathcal{L}^{\mathrm{a}}$.

  *Recall:* $\mathcal{L}$ *is* $FA[0, 1, -1, \pi, \theta, P; comp, LI]$.
  $\mathcal{L}^{\mathrm{a}}$ *is* $FA[0, 1, -1, P; comp, LI]$.
  *To show* $\mathcal{L} \preceq \mathcal{L}^{\mathrm{a}}$*:*
  - $\theta, \pi \preceq \mathcal{L}^{\mathrm{a}}$
  - $comp, LI \preceq \mathcal{L}^{\mathrm{a}}$

General tools:

- Re-usable tools
  Ex: Approximating an operation.

# Outline

1 Introduction

2 Technical Framework

3 Results
- Real Recursive Functions
- General Purpose Analog Computer

4 Aspects of the proofs

5 Conclusion

# Conclusion

- Approximation and Completion as a way to organize the characterizations of computable analysis.
- Technical uses of approximation: Transitivity, eliminating non-analytic functions, lifting.

- Future Work:
  - Other complexity classes (e.g. polynomial time?).
  - Other modes of completion.
    - Discussed: LIM and dLIM.
    - Myca 2003: limits = zero-finding
  - With no explicit completion operation.

# Conclusion

- Approximation and Completion as a way to organize the characterizations of computable analysis.
- Technical uses of approximation: Transitivity, eliminating non-analytic functions, lifting.

- Future Work:
  - Other complexity classes (e.g. polynomial time?).
  - Other modes of completion.
    - Discussed: LIM and dLIM.
    - Myca 2003: limits $=$ zero-finding
  - With no explicit completion operation.

# For Further Reading I

See http://www.math.ist.utl.pt/~ ojakian/ :

M.L. Campagnolo and K. Ojakian.
Using approximation to relate computational classes over the reals.
To appear.

M. L. Campagnolo and K. Ojakian.
The elementary computable functions over the real numbers: Applying two new techniques.
To appear.