AJSW 2010 (Obergurgl), August 3, 2010

A rewriting characterization of the polynomial space functions

Naohi Eguchi

School of Information Science

Japan Advanced Institute of Science and Technology, Japan

Outline

1. Complexity classes.
2. A machine-independent characterization of the polynomial time functions.

   (S. Bellantoni & S. Cook, 1992)
3. A rewriting characterization of the polynomial time functions based on 2.

   (A. Beckmann & A. Weiermann, 1996)
4. This talk: A rewriting characterization of the polynomial space functions. (E., 2010)

Def Let $T : \mathbb{N} \to \mathbb{N}$ be a function.

1. $\mathbf{F}_{\mathrm{TIME}}(T(n)) := \bigcup_k \{f : \mathbb{N}^k \to \mathbb{N} \mid f(\vec{m})$ is computable by a deterministic Turing machine in a step bounded by $T(\max|\vec{m}|)\}$.
2. $\mathbf{F}_{\mathrm{SPACE}}(T(n)) := \bigcup_k \{f : \mathbb{N}^k \to \mathbb{N} \mid f(\vec{m})$ is computable by a deterministic Turing machine in a space bounded by $T(\max|\vec{m}|)\}$.

F_{P}, $\mathsf{F}_{\mathsf{PSPACE}}$ and $\mathsf{F}_{\mathsf{EXP}}$

1. $\mathsf{F}_{\mathsf{P}} := \bigcup_k \mathsf{F}_{\mathsf{TIME}}(O(n^k))$.
2. $\mathsf{F}_{\mathsf{EXP}} := \bigcup_k \mathsf{F}_{\mathsf{TIME}}(2^{O(n^k)})$.
3. $\mathsf{F}_{\mathsf{PSPACE}} := \bigcup_k \mathsf{F}_{\mathsf{SPACE}}(O(n^k))$.

1. $\mathbf{F}_{\mathbf{P}} := \bigcup_k \mathbf{F}_{\mathbf{TIME}}(O(n^k))$.
2. $\mathbf{F}_{\mathbf{EXP}} := \bigcup_k \mathbf{F}_{\mathbf{TIME}}(2^{O(n^k)})$.
3. $\mathbf{F}_{\mathbf{PSPACE}} := \bigcup_k \mathbf{F}_{\mathbf{SPACE}}(O(n^k))$.

Well known:

1. $\mathbf{F}_{\mathbf{P}} \subseteq \mathbf{F}_{\mathbf{PSPACE}} \subseteq \mathbf{F}_{\mathbf{EXP}}$.
2. $\mathbf{F}_{\mathbf{P}} \neq \mathbf{F}_{\mathbf{EXP}}$.
3. $\mathbf{F}_{\mathbf{P}} = \mathbf{F}_{\mathbf{PSPACE}} \Rightarrow \mathbf{P} = \mathbf{NP}$.

Open: $\mathbf{F}_{\mathbf{P}} \neq \mathbf{F}_{\mathbf{PSPACE}}$.

1. $\mathbf{F}_{\mathbf{P}} := \bigcup_k \mathbf{F}_{\mathbf{TIME}}(O(n^k))$.
2. $\mathbf{F}_{\mathbf{EXP}} := \bigcup_k \mathbf{F}_{\mathbf{TIME}}(2^{O(n^k)})$.
3. $\mathbf{F}_{\mathbf{PSPACE}} := \bigcup_k \mathbf{F}_{\mathbf{SPACE}}(O(n^k))$.

Well known:

1. $\mathbf{F}_{\mathbf{P}} \subseteq \mathbf{F}_{\mathbf{PSPACE}} \subseteq \mathbf{F}_{\mathbf{EXP}}$.
2. $\mathbf{F}_{\mathbf{P}} \neq \mathbf{F}_{\mathbf{EXP}}$.
3. $\mathbf{F}_{\mathbf{P}} = \mathbf{F}_{\mathbf{PSPACE}} \Rightarrow \mathbf{P} = \mathbf{NP}$.

Open: $\mathbf{F}_{\mathbf{P}} \neq \mathbf{F}_{\mathbf{PSPACE}}$.

This talk: A rewriting characterization of $\mathbf{F}_{\mathbf{PSPACE}}$.

Outline

1. Complexity classes.
2. A machine-independent characterization of the polynomial time functions.

   (S. Bellantoni & S. Cook, 1992)
3. A rewriting characterization of the polynomial time functions based on 2.

   (A. Beckmann & A. Weiermann, 1996)
4. This talk: A rewriting characterization of the polynomial space functions. (E., 2010)

Several machine-independent approaches to complexity classes (Implicit Computational Complexity):

- Function algebras.
- Proof theory.
- Descriptive complexity.

Function algebra

Several machine-independent approaches to complexity classes (Implicit Computational Complexity):

- Function algebras.
- Proof theory.
- Descriptive complexity.

A function algebra:

-consists of initial functions and operations.

Function algebra

Several machine-independent approaches to complexity classes (Implicit Computational Complexity):

- Function algebras.
- Proof theory.
- Descriptive complexity.

A function algebra:

-consists of initial functions and operations (composition, recursion).

Several machine-independent approaches to complexity classes (Implicit Computational Complexity):

- Function algebras.
- Proof theory.
- Descriptive complexity.

A function algebra:

-consists of initial functions and operations.

-generates a class of functions from the initial functions by means of the operations.

Rewrite systems for primitive recursive functions

(Cichon & Weiermann, 1997):

$$f(y+1,\vec{x}) = h(y,\vec{x},f(y,\vec{x}))$$

Rewrite systems for primitive recursive functions

(Cichon & Weiermann, 1997):

$$f(y+1,\vec{x}) = h(y,\vec{x},f(y,\vec{x}))$$

$$\Downarrow$$

$$f(S(y),\vec{x}) \to h(y,\vec{x},f(y,\vec{x}))$$

Rewriting approach

Rewrite systems for primitive recursive functions

(Cichon & Weiermann, 1997):

$$f(y+1,\vec{x}) = h(y,\vec{x},f(y,\vec{x}))$$
$$\Downarrow$$
$$f(S(y),\vec{x}) \to h(y,\vec{x},f(y,\vec{x}))$$

Let $\mathcal{R}_{\mathcal{A}}$: (schematic) TRS for a function algebra $\mathcal{A}$.

the step complexity of $\mathcal{A}$

$\approx$ the derivational complexity of $\mathcal{R}_{\mathcal{A}}$

A function algebra for $\mathbf{F_P}$:

Def $\mathcal{B}$ is the smallest class containing certain initial functions and closed under safe composition and safe recursion on notation (SRN),

A function algebra for $\mathbf{F_P}$:

Def $\mathcal{B}$ is the smallest class containing certain initial functions and closed under safe composition and safe recursion on notation (SRN),

where

-safe composition: a restriction of composition,

A function algebra for $\mathbf{F_P}$:

Def $\mathcal{B}$ is the smallest class containing certain initial functions and closed under safe composition and safe recursion on notation (SRN),
where
-safe composition: a restriction of composition,
-safe recursion: a restriction of primitive recursion,

 Bellantoni & Cook

A function algebra for $\mathbf{F_P}$:

Def $\mathcal{B}$ is the smallest class containing certain initial functions and closed under safe composition and safe recursion on notation (SRN),
where
-safe composition: a restriction of composition,
-safe recursion: a restriction of primitive recursion,
-recursion on notation: recursion on binary strings.

A function algebra for $\mathbf{F_P}$:

Def $\mathcal{B}$ is the smallest class containing certain initial functions and closed under safe composition and safe recursion on notation (SRN), where

-safe composition: a restriction of composition,

-safe recursion: a restriction of primitive recursion.

Theorem $\mathcal{B} = \mathbf{F_P}$. (Bellantoni-Cook, 1992)

Outline

1. Complexity classes.
2. A machine-independent characterization of the polynomial time functions.

   (S. Bellantoni & S. Cook, 1992)
3. A rewriting characterization of the polynomial time functions based on 2.

   (A. Beckmann & A. Weiermann, 1996)
4. This talk: A rewriting characterization of the polynomial space functions. (E., 2010)

A rewriting characterization of $\mathbf{F_P}$:

Theorem Let $\mathcal{R}$: a finite TRS obtained from $\mathcal{B}$. $\mathbf{dh}(f(\vec{m}), \xrightarrow{\mathsf{i}}_{\mathcal{R}})$ is bounded by a polynomial of $\max|\vec{m}|$. (Beckmann & Weiermann, 1996)

$$\mathbf{dh}(t, \to) := \max\{l \mid \exists (t_1, \ldots, t_l) t = t_1 \to \cdots \to t_l\}$$

A rewriting characterization of $\mathbf{F_P}$:

Theorem Let $\mathcal{R}$: a finite TRS obtained from $\mathcal{B}$. $\mathbf{dh}(f(\vec{m}), \xrightarrow{\mathsf{i}}_{\mathcal{R}})$ is bounded by a polynomial of $\max |\vec{m}|$. (Beckmann & Weiermann, 1996)

$$\mathbf{dh}(t, \rightarrow) := \max\{l \mid \exists (t_1, \ldots, t_l) t = t_1 \rightarrow \cdots \rightarrow t_l\}$$

An application:

Theorem $\mathbf{F_P}$ is closed under a more general form of SRN. (Beckmann & Weiermann, 1996)

Outline

1. Complexity classes.
2. A machine-independent characterization of the polynomial time functions.

   (S. Bellantoni & S. Cook, 1992)
3. A rewriting characterization of the polynomial time functions based on 2.

   (A. Beckmann & A. Weiermann, 1996)
4. This talk: A rewriting characterization of the polynomial space functions. (E., 2010)

Introduce:

1. A function algebra N such that $N = \mathbf{F}_{\mathbf{PSPACE}}$.
2. At the same time, a schematic TRS $\mathcal{R}_N$ which is complete for $\mathbf{F}_{\mathbf{PSPACE}}$:

Introduce:

1. A function algebra N such that $N = \mathbf{F}_{\mathbf{PSPACE}}$.
2. At the same time, a schematic TRS $\mathcal{R}_N$ which is complete for $\mathbf{F}_{\mathbf{PSPACE}}$:
   - Every function representable as a (completely defined) finite subsystem of $\mathcal{R}_N$ belongs to $\mathbf{F}_{\mathbf{PSPACE}}$. (Soundness)
   - Every function from $\mathbf{F}_{\mathbf{PSPACE}}$ is representable as a finite restriction of $\mathcal{R}_N$. (Completeness)

Motivation

Recall:

the step complexity of $\mathcal{A}$

$\approx$ the derivational complexity of $\mathcal{R}_{\mathcal{A}}$

Recall:

the step complexity of $\mathcal{A}$

$\approx$ the derivational complexity of $\mathcal{R}_{\mathcal{A}}$

Then:

the space complexity of $\mathcal{A} \approx$? complexity of $\mathcal{R}_{\mathcal{A}}$

Recall:

the step complexity of $\mathcal{A}$

$\approx$ the derivational complexity of $\mathcal{R}_{\mathcal{A}}$

Then:

the space complexity of $\mathcal{A} \approx$? complexity of $\mathcal{R}_{\mathcal{A}}$

I. Oitavem et al. have shown that the following complexity measure suffices:

$\mathbf{m}(t, \to) = \max\{\mathbf{lh}(s) | t \to^* s\}$ for a suitable auxiliary measure **lh** s.t. $\mathbf{lh}(s) \approx \mathbf{size}(s)$.

N is based on the Bellantoni-Cook principle:

$$f(\underbrace{x_1, \ldots, x_k}_{\text{normal}}; \underbrace{x_{k+1}, \ldots, x_{k+l}}_{\text{safe}}) \in N$$

- Normal: already used as recursion parameters.
- Safe: not yet used.

N is based on the Bellantoni-Cook principle:

$$f(\underbrace{x_1, \ldots, x_k}_{\text{normal}}; \underbrace{x_{k+1}, \ldots, x_{k+l}}_{\text{safe}}) \in N$$

- Normal: already used as recursion parameters.
- Safe: not yet used.

Idea:

to weaken the power of recursion schema.

Safe recursion (SR)

$$f(0, \vec{x}; \vec{y}) = g(\vec{x}; \vec{y})$$
$$f(z+1, \vec{x}; \vec{y}) = h(z, \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y}))$$

Safe recursion on notation (SRN)

$$f(0, \vec{x}; \vec{y}) = g(\vec{x}; \vec{y})$$
$$f(zi, \vec{x}; \vec{y}) = h_i(z, \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y})) \quad (i = 0, 1)$$

Safe recursion on notation (SRN)

$$f(0,\vec{x};\vec{y}) = g(\vec{x};\vec{y})$$
$$f(zi,\vec{x};\vec{y}) = h_i(z,\vec{x};\vec{y},f(z,\vec{x};\vec{y})) \quad (i=0,1)$$

Safe nested recursion on notation (SNRN)

$$f(0,\vec{x};\vec{y}) = g(\vec{x};\vec{y})$$
$$f(zi,\vec{x};\vec{y}) = h_i(z,\vec{x};\vec{y},f(z,\vec{x};\vec{\varphi}_i(z,\vec{x};\vec{y},f(z,\vec{x};\vec{y})))) \quad (i=0,1)$$

complexity class	recursion
$\mathbf{F}_{\mathbf{P}}$	SRN (Bellantoni-Cook '92)
$\mathbf{F}_{\mathbf{PSPACE}}$	?
$\mathbf{F}_{\mathbf{EXP}}$	SNRN (Arai-E. '09)

complexity class	recursion
$\mathbf{F}_{\mathbf{P}}$	SRN (Bellantoni-Cook '92)
$\mathbf{F}_{\mathbf{PSPACE}}$	SNRN
$\mathbf{F}_{\mathbf{EXP}}$	SNRN (Arai-E. '09)

complexity class	recursion
$\mathbf{F}_{\mathbf{P}}$	SRN (Bellantoni-Cook '92)
$\mathbf{F}_{\mathbf{PSPACE}}$	SNRN
$\mathbf{F}_{\mathbf{EXP}}$	SNRN (Arai-E. '09)

However: Initial functions are suitably changed.

complexity class	recursion
$\mathbf{F}_{\mathbf{P}}$	SRN (Bellantoni-Cook '92)
$\mathbf{F}_{\mathbf{PSPACE}}$	SNRN
$\mathbf{F}_{\mathbf{EXP}}$	SNRN (Arai-E. '09)

However: Initial functions are suitably changed.

Def $\quad \mathcal{N}$ is the smallest class containing certain initial functions and closed under safe composition and SNRN.

Def $\mathcal{R}_N$ is the schematic TRS obtained from N.

Theorem

1. Every function representable as a completely defined finite subsystem of $\mathcal{R}_N$ belongs to $\mathbf{F}_{\mathbf{PSPACE}}$.
2. Every function from $\mathbf{F}_{\mathbf{PSPACE}}$ is representable as a finite restriction of $\mathcal{R}_N$.

Theorem Every function representable as a completely defined finite subsystem of $\mathcal{R}_N$ belongs to $\mathbf{F}_{\mathbf{PSPACE}}$.

Theorem Every function representable as a completely defined finite subsystem of $\mathcal{R}_N$ belongs to $\mathbf{F}_{\mathbf{PSPACE}}$.

Lemma Let $\mathcal{R}$: a completely defined finite subsystem of $\mathcal{R}_N$.
Then, $\max\{\mathbf{lh}(s) \mid f(\vec{m};\vec{n}) \xrightarrow{\mathsf{i}}{}^{*}_{\mathcal{R}} s\}$ is bounded by $p(\max \mathbf{lh}(\vec{m},\vec{n}))$ for some polynomial p, where $\mathbf{lh}(s) \approx \mathbf{size}(s)$.

Theorem Every function from $\mathbf{F}_{\mathbf{PSPACE}}$ is representable as a finite restriction of $\mathcal{R}_N$.

Theorem Every function from $\mathbf{F}_{\mathbf{PSPACE}}$ is representable as a finite restriction of $\mathcal{R}_{N}$.

Proof sketch.

1. Reduce the polynomial space Turing computations to the exponential step Register-machine computations.
2. Simulate the exponential step computations within N.

Summary

Introduce:

1. A function algebra $\boldsymbol{N}$ such that $N = \mathbf{F}_{\mathbf{PSPACE}}$.
2. At the same time, a schematic TRS $\boldsymbol{\mathcal{R}}_N$ which is complete for $\mathbf{F}_{\mathbf{PSPACE}}$:
   - Every function representable as a (completely defined) finite subsystem of $\boldsymbol{\mathcal{R}}_N$ belongs to $\mathbf{F}_{\mathbf{PSPACE}}$. (Soundness)
   - Every function from $\mathbf{F}_{\mathbf{PSPACE}}$ is representable as a finite restriction of $\boldsymbol{\mathcal{R}}_N$. (Completeness)

Conclusions

1. Rewriting techniques can be applied to space complexity classes.
2. The obtained system $\mathcal{R}_N$ is simpler. (e.g. the termination of $\mathcal{R}_N$ is easily shown by a LPO)
3. Some exponential step computations can be simulated by $\mathcal{R}_N$.
4. Hence, non-deterministic polynomial time computations might be simulated by $\mathcal{R}_N$.

Further researches

complexity class	path order
$\mathbf{F}_{\mathbf{P}}$	POP* (Avanzini-Moser '08)
$\mathbf{F}_{\mathbf{PSPACE}}$	?
$\mathbf{F}_{\mathbf{EXP}}$	EPO* (Avanzini-E., WST '10)