Tailoring Recursion to Characterize Non-Deterministic Complexity Classes Over Arbitrary Structures

Olivier Bournez

Paulin de Naurois

Jean-Yves Marion

LORIA - Ecole des Mines-INPL, Nancy

Felipe Cucker

City University, Hong Kong

Computation over a Structure $\mathcal{K}$

A structure $\mathcal{K} = (\mathbb{K}, op_\imath, rel_\jmath)$ is given by:

- A domain $\mathbb{K}$.
- A finite set of operators $op_\imath$ over $\mathbb{K}$.
- A finite set of relations $rel_\jmath$ over $\mathbb{K}$.
- The equality relation $=$ over $\mathbb{K}$ is a relation of $\mathcal{K}$.

Computation over a Structure $\mathcal{K}$

A structure $\mathcal{K} = (\mathbb{K}, op_\imath, rel_\jmath)$ is given by:

- A domain $\mathbb{K}$.
- A finite set of operators $op_\imath$ over $\mathbb{K}$.
- A finite set of relations $rel_\jmath$ over $\mathbb{K}$.
- The equality relation $=$ over $\mathbb{K}$ is a relation of $\mathcal{K}$.

Examples:

- $\mathcal{R} = (\mathbb{R}, +, -, *, /, >, =, 0, 1)$
- $\mathcal{B} = (\{0, 1\}, =, 0, 1)$

BSS-TM over $\mathcal{K}$

Similar to a Turing machine (TM), with the properties:

- Its tape cells hold arbitrary elements of $\mathbb{K}$.
- It has Computation nodes for computing operators.
- It has Branch nodes for computing relations.
- It has Shift nodes for moving the head.

A TM computes a function from $\mathbb{K}^*$ to $\mathbb{K}^*$.

$\mathbb{K}^*$ denotes lists of element of $\mathbb{K}$.

Polynomial time functions over $\mathcal{K}$

A function $f : (\mathbb{K}^*)^n \to (\mathbb{K}^*)^m$ is in class $\mathrm{FPTIME}_{\mathcal{K}}$
iff
f is computable in polynomial time.

That is,
there is a polynomial p and a BSS-TM M, such that

- M computes f
- M stops in $p(|\overline{w}|)$ steps, $\forall \overline{w} \in \mathbb{K}^*$.

$|\overline{w}|$ is the length of the list $\overline{w} \in \mathbb{K}^*$.

Complexity Theory over $\mathcal{K}$

- Over the structure $\mathcal{B} = (\{0,1\}, =, 0, 1)$, we compute $f : \{0,1\}^* \to \{0,1\}^*$, corresponds to classical complexity and $\mathrm{P}_{\mathcal{B}} = \mathrm{P}$.
- Over the structure $\mathcal{R} = (\mathbb{R}, +, -, *, /, >, =, 0, 1)$ corresponds to the original Blum Shub and Smale 89 paper.

Classical Complexity Characterisations

Machine independent characterizations of complexity classes:

- Polynomial time FPTIME corresponds to safe recursive functions. *(Bellantoni-Cook 92)*
- Polynomial hierarchy PH corresponds to safe recursive decision sets with predicative recursion. *Bellantoni 94*
- Polynomial space $\mathrm{FPSPACE}$ corresponds to safe recursive functions with substitutions. *(Leivant-Marion 95)*

Complexity Theory over $\mathcal{K}$

- Polynomial time $\mathrm{FPTIME}_{\mathcal{K}}$ corresponds to safe recursive functions.
- Parallel polynomial time $\mathrm{PAR}_{\mathcal{K}}$ corresponds to safe recursive functions with substitutions.
- Polynomial hierarchy $\mathrm{PH}_{\mathcal{K}}$ corresponds to safe recursive decision sets with predicative minimization.
- Polynomial alternating time $\mathrm{PAT}_{\mathcal{K}}$ corresponds to safe recursive decision sets with predicative substitution

Analogous results with digital hierarchy.

(BCdNM-02-04)

Polynomial time functions $\mathrm{FPTIME}_{\mathcal{K}}$

Over any structure $\mathcal{K}$, the set of safe recursive functions over $\mathcal{K}$ is exactly $\mathrm{FPTIME}_{\mathcal{K}}$.

Safe Recursive functions over a structure $\mathcal{K}$ (1/2)

Two types of arguments,“safe”and “normal”

$$f(\overline{x};\overline{y})$$

The set of safe recursive functions over $\mathcal{K}$ is the smallest set of functions containing basic safe functions

- structure operators and relations
- projections
- list destructors : **hd** and **tl**
- list constructor : **cons**
- Boolean selection : **if** $x = 1$ **then** y **else** z

Safe Recursive functions over $\mathcal{K}$ (2/2)

and closed under both schemas

- safe composition

$$f(\overline{x};\overline{y}) = g(h_1(\overline{x};);h_2(\overline{x};\overline{y}))$$

- safe recursion

$$f(\epsilon,\overline{x};\overline{y}) = g(\overline{x};\overline{y})$$
$$f(a.\overline{z},\overline{x};\overline{y}) = g(\overline{z},\overline{x};f(\overline{z},\overline{x};\overline{y}),\overline{y})$$

About the exponential function

$$\exp(\overline{x}) = \mathbf{1}^{2^{|\overline{x}|}}$$

$$\begin{aligned}
\text{append}(\epsilon; \overline{y}) &= \overline{y} \\
\text{append}(a.\overline{x}; \overline{y}) &= \text{cons}(\mathbf{1}, \text{append}(\overline{x}; \overline{y})) \\
\exp(\epsilon) &= \mathbf{1} \\
\exp(a.\overline{x}) &= \text{append}(\exp(\overline{x}); \exp(\overline{x}))
\end{aligned}$$

exp is not a Safe Recursive definition because the recursive call $\exp(\overline{x})$ must be safe.

Complexity Theory over $\mathcal{K}$

- Polynomial time $\mathrm{FPTIME}_{\mathcal{K}}$ corresponds to safe recursive functions.

$\Rightarrow$ Parallel polynomial time $\mathrm{PAR}_{\mathcal{K}}$ corresponds to safe recursive functions with substitutions.

- Polynomial hierarchy $\mathrm{PH}_{\mathcal{K}}$ corresponds to safe recursive decision sets with predicative minimization.

$\Rightarrow$ Polynomial alternating time $\mathrm{PAT}_{\mathcal{K}}$ corresponds to safe recursive decision sets with predicative substitution

About space...

There is no valid notion of polynomial space over arbitrary structures.

Theorem [Michaux 89]: Over $(\mathbb{R}, 0, 1, =, +, -, *, <)$, any computable function can be computed in constant working space.

The whole picture

$$
\begin{array}{ccccccccc}
 & & \mathrm{DPAT}_{\mathcal{R}} & & & & \mathrm{PAT}_{\mathcal{R}} & & \\
 & \nearrow & & \searrow & & \nearrow & & \searrow & \\
\mathrm{DPH}_{\mathcal{R}} & & & & \mathrm{PAR}_{\mathcal{R}} & & & & \mathrm{PEXP}_{\mathcal{R}} \\
 & \searrow & & \nearrow & & \searrow & & \nearrow & \\
 & & \mathrm{PH}_{\mathcal{R}} & & & & \mathrm{EXP}_{\mathcal{R}} & &
\end{array}
$$

- Over $\mathcal{R}$, Inclusions $\mathrm{PAR}_{\mathcal{R}} \subset \mathrm{PAT}_{\mathcal{R}}$ and $\mathrm{PAR}_{\mathcal{R}} \subset \mathrm{EXP}_{\mathcal{R}}$ are known to be strict.
- Over the booleans, $PH = DPH$, and $PSPACE = PAR = PAT = DPAT$

Class $\mathrm{DNP}_{\mathcal{K}}$

A problem $P \subset \mathbb{K}^*$ is in class $\mathrm{DNP}_{\mathcal{K}}$ if there exists a polynomial p and a machine M, so that for all $\overline{w} \in \mathbb{K}^*$,

- $\overline{w} \in P$ iff $\exists \overline{y} \in \{\mathbf{0}, \mathbf{1}\}^*$ M accepts $(\overline{w}, \overline{y})$,
- M stops in $p(|\overline{w}|)$ steps.

Safe Recursion with Predicative Minimization

The set of safe recursive functions with predicative minimization over $\mathcal{K}$ is the smallest set of functions containing the safe recursive functions, and closed under the operations

- safe composition
- predicative minimization

$$f(\overline{x};\overline{y}) = \mu\bar{b}(h(\overline{x};\overline{y},\bar{b})) = \begin{cases} \mathbf{1} & \text{if there is } \bar{b} \in \mathbb{K}^* \text{ s.t. } h(\overline{x};\overline{a},\bar{b}) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Predicative Minimization Capture $\mathrm{PH}_{\mathcal{K}}$

Over any structure $\mathcal{K}$, a decision problem belongs to $\mathrm{PH}_{\mathcal{K}}$
iff
the characteristic function is a recursive function with predicative minimization.

Conclusion

Machine independent characterizations over arbitrary structures that subsume previous ones.

- Polynomial time $\mathrm{FPTIME}_{\mathcal{K}}$ corresponds to safe recursive functions.
- Parallel polynomial time $\mathrm{PAR}_{\mathcal{K}}$ corresponds to safe recursive functions with substitutions.
- Polynomial hierarchy $\mathrm{PH}_{\mathcal{K}}$ corresponds to safe recursive decision sets with predicative minimization.
- Polynomial alternating time $\mathrm{PAT}_{\mathcal{K}}$ corresponds to safe recursive decision sets with predicative substitution

Analogous result with digital hierarchy.

Polynomial Alternating Time

Decision problem S is in $\mathrm{PAT}_{\mathcal{K}}$ iff

- there exists a polynomial time BSS machine M
- there exist a polynomial p
- $\overline{x} \in S \Leftrightarrow \exists \overline{y_1} \forall \overline{y_2} \ldots \mathrm{Q}\overline{y_{p(|\overline{x}|)}},\ \mathrm{Q} \in \{\exists, \forall\},\ :$
  M accepts on input $\overline{x}, \overline{y_1}, \ldots, \overline{y_{p(|\overline{x}|)}}$