Polynomial-Time Computability in Analysis: A Survey

Ker-I Ko
Stony Brook University, New York
Tsinghua University, Beijing
Outline

1  Computational Models
   Church’s thesis in computational analysis?

2  Complexity Hierarchy of Numerical Operations
   Applying NP-theory to analysis

3  Applications to Computational Geometry
   P-time computable Jordan domains

4  Applications in Complex Analysis
   Julia sets, conformal mappings
Computational Theory of Real Analysis

Constructive Analysis        Bishop, Bridges, Ishihara, ⋅⋅⋅
Intuitionistic Logic

Recursive Analysis (Computable Analysis)
Recursion Theory

Russian School      Šanin, Moschovakis, Ceitin, ⋅⋅⋅
Polish School       Grzegorczyk, Mostowski
                    Lacombe, Pour-El, Richards
                    Weihrauch, ⋅⋅⋅
Polynomial-Time Analysis

Complexity Theory

Turing machine model Ko, Friedman, Weihrauch, Müller Rettinger, Zheng, Cook, Braverman, ···
Real-valued circuit model Hoover
Algebraic model Blum, Shub, Smale, Cucker, ···
Information-based complexity theory Traub, Wozniakowski, ···

Numerical Analysis

Classical analysis, Arithmetic complexity theory
Interval analysis, Scientific computing
Relationship between these theories

Computability Theory ↔ NP-complete Theory ↔ Analysis of Algorithms

Computable Analysis ↔ Polynomial-Time Theory of Analysis ↔ Numerical Analysis
Example: *Roots of Polynomials*

**Bishop:** Fundamental Theorem of Algebra has a *constructive* proof.

**Specker:** All roots of a computable polynomial function are *computable.*

The mapping from coefficients to roots is *computable.*

**Ko-Friedman:** All roots of a polynomial-time computable polynomial function are *polynomial-time computable.*

**Neff:** The mapping from coefficients to roots is in *NC.*

**Schönhage:** The mapping from coefficients to roots is computable in *time* $O(n^3\phi(n))$.

**Smale:** Newton’s method runs in polynomial time on *average.*
**Warning** They may use different models.

⇒ There is **no Church’s Thesis** in computational analysis.

The models of the following theories are **consistent**:

- *Recursive analysis* (Polish school)
- *Polynomial-time analysis* (Turing machine model)
- *Discrete NP-completeness theory*
- *Classical numerical analysis* (e.g., interval analysis)
Real Numbers

A real number is an infinite object, and has no finite representations.

Basic representation: Cauchy functions with a fixed converging rate

\[ \varphi_x : \mathbb{N} \to \mathbb{D} \text{ with } |\varphi_x(n) - x| \leq \frac{1}{2^n}. \]

\( \mathbb{D} \): dyadic rationals

\( x \) is computable if \( \exists \) a computable \( \varphi_x \).

\( x \) is P-time computable if \( \exists \) a P-time computable \( \varphi_x \).
Other Representations?

Dedekind cuts: $L_x = \{d \in \mathbb{D} : d < x\}$

Binary expansions: $b_x : \mathbb{N}^+ \to \{0, 1\}$ and $b_x(0) \in \mathbb{Z}$, with
$$x = \sum_{n=0}^{\infty} b_x(n) \cdot 2^{-n}.$$ 

Continued fractions: $c_x : \mathbb{N} \to \mathbb{N}^+$ with
$$x = c_x(0) + \cfrac{1}{c_x(1) + \cfrac{1}{c_x(2) + \cfrac{1}{\ldots}}}$$

For computable real numbers, these representations are equivalent to Cauchy function representation.

For \textit{P-time computable} real numbers, they are \textit{not} equivalent.
Real Numbers as Discrete Objects

\( P^R \): Set of P-time computable real numbers
\( NP^R \): Set of NP-time computable real numbers
\( \#P^R, \#PSPACE^R, \ldots \)

What are the relations between these complexity classes?

General Observation
Representations of real numbers behave like selective sets or sparse sets.

\[ P^R = NP^R \iff P_1 = NP_1 \]

\[ \#P^R =? \#NP^R \text{ (YES if } NP = UP \) \]
Real Functions

**Representation of** $f : \mathbb{R} \rightarrow \mathbb{R}$:

Type-2 function with a fixed converging rate

$\Phi_f : \Psi \times \mathbb{N} \rightarrow \mathbb{D}$, with $|\Phi_f(\varphi_x, n) - f(x)| \leq \frac{1}{2^n}$

$\Psi$: set of Cauchy functions $\varphi_x$

**Computational Model** for type-2 functions:

Oracle Turing machine

$f$ is computable if $\Phi_f$ is computable by an oracle TM $M$

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

$f : [0, 1] \rightarrow \mathbb{R}$ is P-time computable if $M^{\varphi_x}(n)$ halts in time $n^{O(1)}$ for every oracle $\varphi_x$ with $x \in [0, 1]$. 
**Compute** $f(x) = x^2$:

**Input** $n$ (the output precision)

**Oracle** $\varphi_x$ (representation of a real $x$)

**Algorithm**

1. Compute required input precision $m$ from $n$
   
   ($n \mapsto m$ is called modulus function);

2. Ask oracle to get a rational $r$ with $|r - x| \leq 2^{-m}$;

3. Compute $s \leftarrow r^2$;

4. Output first $n$ bits of $s$.

**Note:** Modulus function may also depend on $x$. So, Steps (1) and (2) may be repeated to find the right $m$. 

An **Alternative** type-1 representation
(with an **additional** continuity requirement)

$(\varphi_f, m_f)$ where $\varphi_f : \mathbb{D} \times \mathbb{N} \rightarrow \mathbb{D}$, $m_f : \mathbb{N} \rightarrow \mathbb{N}$,

with $|\varphi_f(d, n) - f(d)| \leq 2^{-n}$, and

$|x - y| \leq 2^{-m_f(n)} \implies |f(x) - f(y)| \leq 2^{-n}$

$f$ is **computable** iff $\varphi_f, m_f$ are computable

$f$ is **P-time computable** iff $\varphi_f$ is P-time computable, and $m_f$ is a polynomial function.
Warning
In this model, comparison of two real numbers is noncomputable.

- $\exists$ oracle TM $M$ such that $M^\varphi_x,\varphi_y(0) = \begin{cases} 1 & \text{if } x < y, \\ 0 & \text{if } x > y, \\ \uparrow & \text{if } x = y. \end{cases}$

- No oracle TM: $M^\varphi_x,\varphi_y(0) = \begin{cases} 1 & \text{if } x \neq y, \\ 0 & \text{if } x = y. \end{cases}$

- The problem of determining whether a given polynomial function (represented by its coefficients) has multiple roots is undecidable.
Numerical Operators

\( F : C[0, 1] \rightarrow \mathbb{R} \) is a type-3 function.

We can use Oracle TM as a computational model.

\( F \) is computable if \( \exists \) oracle TM \( M \) such that
\[
| M^{\Phi_f(n)} - F(f) | \leq 2^{-n}.
\]
(In the computation, \( M \) may ask the oracle to find an approximate value of \( f(x) \) by asking the oracle for the value of \( \Phi^d_f(n) \), where \( d \approx x \).)
P-Time Computable Operators?

**Weak form:** Consider only P-time invariance

If $f$ is P-time computable, what is the complexity of $F(f)$?

**Strong form** [Kawamura-Cook, 2010]

Use regular functions as representations of $f$, a more general notion of P-time computable operator can be defined.

Many known results about P-time computability of numerical operators in the weak form can be extended to the strong form.
Outline

1 Computational Models
   Church’s thesis in computational analysis?

2 Complexity Hierarchy of Numerical Operations
   Applying NP-theory to analysis

3 Applications to Computational Geometry
   P-time computable Jordan domains

4 Applications in Complex Analysis
   Julia sets, conformal mappings
A **Complexity Hierarchy** of Numerical Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Complexity Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differentiation</td>
<td>Noncomputable</td>
</tr>
<tr>
<td>Integral Eq (with local Lipschitz cond)</td>
<td>EXPSPACE-Complete</td>
</tr>
<tr>
<td>Ordinary Diff Eq (with Lipschitz cond)</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>Integration</td>
<td>#P-complete</td>
</tr>
<tr>
<td>Minimax</td>
<td>NP(^{NP})-complete</td>
</tr>
<tr>
<td>Maximization</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Roots (of 2-dim. functions)</td>
<td>between UP and NP</td>
</tr>
<tr>
<td>Fixed Points (of 2-dim. functions)</td>
<td>PPAD-complete</td>
</tr>
<tr>
<td>Roots (of 1-1 functions)</td>
<td>P-complete</td>
</tr>
<tr>
<td>Differentiation ((f') has poly. modulus)</td>
<td>P</td>
</tr>
</tbody>
</table>
Maximization:

What is the complexity of finding 
\[ \max\{x_1, x_2, \ldots, x_K\} \]?

Depending on the representation of \( x_1, x_2, \ldots, x_K \).

(1) Explicit Representation:
\( x_1, x_2, \ldots, x_K \) are given as input (input size \( n \approx K \)):

Input: \( 38, 25, 19, 55, \ldots, 49 \)

find max

Complexity: \( \ln P \) (needs \( K - 1 \) comparisons)
(2) **Oracle Representation:**

$x_1, x_2, \ldots, x_K$ are given by an oracle function $\Phi$ 

$(\Phi(i) = x_i)$

**Oracle:**

<table>
<thead>
<tr>
<th></th>
<th>38</th>
<th>25</th>
<th>19</th>
<th>55</th>
<th>⋯</th>
<th>49</th>
</tr>
</thead>
</table>

**Input:** $K$  
(input size $n = \lceil \log K \rceil$)

**Complexity:** **Exponential time**  
(must ask the oracle $\Phi$ for $K \approx 2^n$ times)
(3) **Machine Representation:**

$x_1, x_2, \ldots, x_K$ are presented by a polynomial-time algorithm $A$ that computes the function $\Phi$

**Input:**

\[
\begin{array}{c}
A \\
\downarrow \downarrow \cdots \downarrow \\
38, 25, 19, 55, \ldots, 49 \quad \text{(hidden input, size = $K$)}
\end{array}
\]

\[
(n = \text{size}(A) \approx [\log K]^{O(1)})
\]

**Complexity:** In NP; NP-complete for some $A$

(Actually, the following variation is in NP: Given $A$ and an integer $M$, determine whether $M < \max\{\Phi(1), \ldots, \Phi(K)\}$.)

- Most **NP-ccomplete** optimization problems can be viewed in this form.
Traveling Salesman:

Input: Graph $G$ with $n$ vertices; weight $w : E \rightarrow \mathbb{N}^+$

Question: Find the min-weight Hamiltonian tour of $G$

- There are $K = (n - 1)!$ different Hamiltonian tours of $G$, and they can be enumerated as $H_1, H_2, \ldots, H_K$.

- Now, Traveling Salesman can be restated as follows:
  
  Find the minimum of the output from $A_G$: 
\(A_G:\) For \(i = 1, 2, \ldots, K\), identify \(i\) with a Hamiltonian tour \(H_i\) and output \(\Phi(i) = \text{total weight of } H_i\).
Numerical Maximization

Given \( f : [0, 1] \rightarrow \mathbb{R} \) (as an oracle), find \( \max_{0 \leq x \leq 1} f(x) \).

**Discretize** this problem:

**Assumption**: Function \( f \) has a polynomial modulus:

\[
|x - y| \leq 2^{-nc} \Rightarrow |f(x) - f(y)| \leq 2^{-n}
\]

With this assumption, the discretized problem becomes:

Find the maximum value of

\[
f\left(\frac{1}{2nc}\right), f\left(\frac{2}{2nc}\right), \ldots, f\left(\frac{2^{n}c}{2nc}\right)
\]

(For convenience, we use \( c = 1 \) in the following discussion.)
Representation of $f$:

(1) Explicit representation

Function values $f\left(\frac{1}{2n^c}\right)$, $f\left(\frac{2}{2n^c}\right)$, \ldots, $f\left(\frac{2^n}{2n^c}\right)$ are given as input.

**Complexity:** Polynomial in input size, exponential in output precision $n$

- This is the common practice of Computational Geometry (with $n$ input points, instead of $2^n$ points).
(2) Oracle representation
Function \( f \) is given by an oracle. The maximization algorithm may ask for \( f(r) \) for any rational number \( r \).

**Complexity:** Exponential in the output precision \( n \).

- This is used in some theoretical study of numerical analysis (e.g., Information-Based Complexity Theory of [Traub et al.]).
(3) Machine representation
Function $f$ is assumed to be computable by a machine $M_f$ in polynomial time (polynomial in output precision $n$), and the maximization algorithm may simulate $M_f$ on any input $r$.

Complexity: NP-complete.

Note: The Machine representation approach is equivalent to the model in the Turing Machine-Based P-Time Theory of Analysis.

**Theorem** [Ko, Friedman]
$P = NP \iff$ For every polynomial-time computable function $f : [0, 1] \rightarrow \mathbb{R}$, $\max f \in P$. 
Ordinary Differential Equations (IVP)

\[ y'(x) = f(x, y(x)), \; 0 \leq x \leq 1, \]
\[ y(0) = 0. \]

• \( \exists \) computable \( f \): all solutions \( y \) are not computable on \([0, \delta]\) for all \( \delta > 0 \).  
  
  Pour-El, Richards

• \( f \) computable, solution \( y \) unique \( \implies \) \( y \) computable.

• \( \exists \) P-time computable \( f \): solution \( y \) is unique, but complexity of \( y \) is arbitrary high.  
  
  Miller
Lipschitz Condition

\( f \in Lip(\alpha) : \ (\forall x \in [0, 1]) \ (\forall y_1, y_2 \in [-1, 1]) \)
\[
|f(x, y_1) - f(x, y_2)| \leq \alpha \cdot |y_1 - y_2|.
\]

- \( f \) P-time computable, \( f \in Lip(\alpha) \implies y \) P-space computable. 
  \text{Ko}

- (\exists P-time computable \( f \)): \( f \in Lip(\alpha) \), \( y \) is P-space complete. 
  \text{Ko, Kawamura}

- The mapping \( f \mapsto y \) is P-space complete. 
  \text{Kawamura, Cook}
**Volterra Integral Equations** (of the 2nd kind)

\[ y(x) = f(x) + \int_0^x K(x, s, y(s)) \, ds, \quad 0 \leq x \leq 1, \]

with \( K \in Lip_3(\alpha) \): 

\[ |K(x, s, y_1) - K(x, s, y_2)| \leq \alpha \cdot |y_1 - y_2| \]

- If \( \alpha \) is independent of \( x \), then this problem is P-space complete. \textbf{Ko, Kawamura}

- If \( \alpha \leq 2^{nO(1)} \) for \( x \leq 1 - 2^{-n} \), then \( y \) is EXP-space computable. \textbf{Ko}

- Under the above local Lipschitz condition, this problem is EXP-space complete. \textbf{Kawamura}
Outline

1  Computational Models
   Church’s thesis in computational analysis?

2  Complexity Hierarchy of Numerical Operations
   Applying NP-theory to analysis

3  Applications to Computational Geometry
   P-time computable Jordan domains

4  Applications in Complex Analysis
   Julia sets, conformal mappings
Subsets of $\mathbb{R}^2$

Computable sets of real numbers?
Again, there does not seem to be a Church’s Thesis.

For discrete $A \subseteq \{0, 1\}^*$, $A$ is computable if

$$
\chi_A(x) = \begin{cases} 
1 & \text{if } x \in A, \\
0 & \text{if } x \notin A
\end{cases}
$$
is computable.

**Try:** For $S \subseteq \mathbb{R}^2$, $S$ is computable if

$$
\chi_S(z) = \begin{cases} 
1 & \text{if } z \in S \\
0 & \text{if } z \notin S
\end{cases}
$$
is computable.
Warning The function $\chi_S$ is not computable for nontrivial $S$ (i.e., $S \neq \emptyset$, $S \neq \mathbb{R}^2$).

For an oracle TM, let

$$Err_n(M) = \{z : M^z(n) \neq \chi_S(z)\}.$$  

P-time Approximable (Measurable) Sets

$\exists$ P-time oracle TM $M$: $\mu(Err_n(M)) \leq 2^{-n}$.

P-time Recognizable Sets

$\exists$ P-time oracle TM $M$:

$$z \in Err_n(M) \Rightarrow \delta(z, \partial S) \leq 2^{-n}.$$
Strongly P-time Recognizable Sets

\[ \exists \text{ P-time oracle TM } M: \]
\[ z \in Err_n(M) \Rightarrow \delta(z, \partial S) \leq 2^{-n} \text{ and } z \notin S. \]

P-time Computable Sets [Weihrauch, ⋯]

\[ \exists \text{ P-time oracle TM } M: \]
\[ z \in Err_n(M) \Rightarrow 2^{-n} < \delta(z, S) \leq 2 \cdot 2^{-n}. \]

P = NP \iff the above two classes are equivalent.

P-time Computable Sets wrt Hausdorff Distance

\[ \exists \text{ P-time oracle TM } M: \]
[Braverman, Yampolsky]
\[ \delta_{\text{HAUS}}(S, \{z \ M^z(n) = 1\}) \leq 2^{-n} \]

All of the above definitions are not equivalent.
Jordan Domains

A Jordan domain is a singly-connected set whose boundary is a Jordan curve $\Gamma$ (the image of a mapping $f : [0, 1] \rightarrow \mathbb{R}^2$).

Computable Curves — still no unique definition

Monotonically Computable: $f$ is one-to-one
Retraceably Computable: $f$ is not necessarily one-to-one

Gu, Lutz, Mayordomo

Normalizably Computable: Length of $f[0, t]$ is proportional to $t$, for $0 < t < 1$ (if $leng(\Gamma)$ is finite).

Rettinger, Zheng
Continuous Computational Geometry

Goals: Resolve the numerical non-robustness problem
Deal with more general geometric objects
Allow efficient implementation of traditional algorithms
E.g. Exact Geometry Computation (EGC)

Yap, Melhorn, ...

Jordan Domain-Based Approach

General Question
Given a two-dimensional domain $S$ whose boundary is a P-time computable Jordan curve, what is the complexity of the related problems?
If $\partial S$ is $P$-computable, then it has polynomial modulus.

So, $\partial S$ is represented by an implicit polygon of $2^p(n)$ vertices.
Complexity of **Jordan Domains** $S$

<table>
<thead>
<tr>
<th>Property</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>Noncomputable (fractal)</td>
</tr>
<tr>
<td>Length of $\partial S$</td>
<td>Noncomputable (fractal)</td>
</tr>
<tr>
<td>Shortest Paths in $S$</td>
<td>between #P and PSPACE</td>
</tr>
<tr>
<td>Pancake Cutting</td>
<td>#P-complete</td>
</tr>
<tr>
<td>Membership ($x \in S$?)</td>
<td>between UP and #P</td>
</tr>
<tr>
<td>Circumscribed Rectangle</td>
<td>NP(^{NP})-complete</td>
</tr>
<tr>
<td>Distance of $x$ from $S$</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Convex Hull</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>
Outline

1 Computational Models
   Church’s thesis in computational analysis?

2 Complexity Hierarchy of Numerical Operations
   Applying NP-theory to analysis

3 Applications to Computational Geometry
   P-time computable Jordan domains

4 Applications in Complex Analysis
   Julia sets, conformal mappings
Analytic Functions

If $f$ is real analytic and P-time computable, then integral $\int_0^x f$, derivative $f'(x)$, maximum value $\max f(x)$, and roots $\{x : f(x) = 0\}$ are all P-time computable.

Parallel Complexity

If $f$ is analytic and is NC (or LOG-space) computable, then integral, derivative, maximum value and roots of $f$ are all NC (or LOG-space, resp.) computable.
Zeroes of an Analytic Function $f$
on a Jordan domain $S$

Assumptions

- $f$ is analytic on $S \cup \partial S$
- $f(z) > 0$ on $\partial S$
- $f$ and $\partial S$ are NC computable
Quadrature Method

(1) Compute the number of zeroes

\[ n = \frac{1}{2\pi i} \int_{\partial S} \frac{f'(z)}{f(z)} dz \]  
(by principle of argument)

(2) Compute the Newton sums

\[ s_p = \sum_{i=1}^{n} z_i^p = \frac{1}{2\pi i} \int_{\partial S} z^p \frac{f'(z)}{f(z)} dz \]

(3) Compute the associated polynomial

\[ p_n(z) = \prod_{i=1}^{n} (z - z_i) \]  
(by Newton’s identity and \( s_p, \ p = 1, \ldots, n \))

(4) Solve the associated polynomial equation  \[ \text{[Neff]} \]

All the above calculations can be parallelized.
Some problems related to Membership Problem

- Computing Winding Number of a closed curve
- Computing Single-Valued Analytic Branch of a multi-valued function

**Square Root Problem**

On a complex domain, \( \sqrt{z} = \sqrt{|z|} \cdot e^{i \text{arg}(z)/2} \) has 2 single-valued, analytic branches:

\[ \sqrt{z} = \sqrt{|z|} \text{ or } \sqrt{|z|} \cdot e^{i\pi} \]
Logarithm Problem

On a complex domain,
\[ \log z = \log |z| + i \arg(z) \]
has \( \infty \) single-valued analytic branches:
\[ \arg(z) = \cdots, -4\pi, -2\pi, 0, 2\pi, 4\pi, \cdots \]
corresponding to
\[ \arg(z_0) = \cdots, 0, 2\pi, 4\pi, 6\pi, 8\pi, \cdots \]
Analytic Branch Problem
Given a P-time computable closed Jordan curve $\Gamma$, what is the complexity of finding a single-valued analytic branch of $\log z$ or $\sqrt{z}$ on $S = \text{Int}(\Gamma)$?

Equivalent Problem: Given $\Gamma$, what is the complexity of computing a continuous argument function $h(z) \in \text{arg}(z)$ on $S$?

$$\log z \equiv h(z) - h(z_0) \quad \sqrt{z}, \equiv \frac{h(z) - h(z_0)}{2\pi} \mod 2$$

If $z$ and $z_0$ are on the boundary of $S$,
$$h(z) \approx \text{winding number about } z.$$
## Complexity

<table>
<thead>
<tr>
<th>Problem</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winding Number</td>
<td>#P</td>
<td>#P</td>
</tr>
<tr>
<td>Logarithm</td>
<td>#P</td>
<td>#P</td>
</tr>
<tr>
<td>Square root</td>
<td>⊕P</td>
<td>MP</td>
</tr>
<tr>
<td>Membership</td>
<td>UP</td>
<td>MP</td>
</tr>
</tbody>
</table>

NP: \( \{ x \mid (\exists y) R(x, y) \} \), where \( R \in P \)

\#P: \( f(x) = \text{number of } y \text{ such that } R(x, y) \)

⊕P (Parity P): \( f(x) \) is odd

MP (Midbit P): the middle bit of \( f(x) = 1 \).
**Analytic Continuation**

Assume that $f$ is an analytic function defined on a domain $S$. Then, the power series of $f$ at any $z \in S$ can be computed from that of $f$ at a starting point $z_0$.

**Complexity?**
Depends on geometric properties of $\partial S$?
Julia Sets

For a function $f : \mathbb{C} \to \mathbb{C}$, define

$$K(f) = \{ z \in \mathbb{C} | (\exists C > 0)(\forall n) |f^n(z)| \leq C \},$$

$$J(f) = \text{boundary of } K(f).$$

- $\exists$ P-time computable $f : \mathbb{C} \to \mathbb{C}$ such that $J(f)$ encodes the halting problem of the universal Turing machine.
- Membership problem of $J_f$ for a hyperbolic polynomial $f$ is P-time computable [Rettinger, Weihrauch, Braverman, Yampolsky]
- A special group of functions: $f_c(z) = z^2 + c, z, c \in \mathbb{C}$. For most $c$ (including all $c$ outside the Mandelbrot set), $f_c$ is hyperbolic.
Conformal Mappings

Given a Jordan domain $S$, what is the complexity of the Riemann mapping from $S$ to the unit disk (relative to the complexity of $S$)?

- Under some restrictions on the boundary of $S$, the complexity is $\#P$-complete (if $S$ is P-time computable). [Braverman, Yampolsky, Rettinger]

- **Open Question:** In the general case, when it is only known that $\partial S$ is P-time computable, is the complexity still $\#P$?
Thank You