Computation Paradigms in the Light of Hilbert’s Tenth Problem

Yuri Matiyasevich

Steklov Institute of Mathematics at St.Petersburg, Russia

http://logic.pdmi.ras.ru/~yumat
Statement of the Problem: Intuitive Notion of Algorithm


David Hilbert, *Mathematical Problems* [1900]

10. Determination of the Solvability of a Diophantine Equation. Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.
"Occasionally it happens that we seek the solution under insufficient hypotheses or in an incorrect sense, and for this reason do not succeed. The problem then arises: to show the impossibility of the solution under the given hypotheses, or in the sense contemplated. Such proofs of impossibility were effected by the ancients, for instance when they showed that the ratio of the hypotenuse to the side of an isosceles triangle is irrational. In later mathematics, the question as to the impossibility of certain solutions plays a preëminent part, and we perceive in this way that old and difficult problems, such as the proof of the axiom of parallels, the squaring of circle, or the solution of equations of the fifth degree by radicals have finally found fully satisfactory and rigorous solutions, although in another sense than that originally intended.

"It is probably this important fact along with other philosophical reasons that gives rise to conviction (which every mathematician shares, but which no one has as yet supported by a proof) that every definite mathematical problem must necessary be susceptible of an exact settlement, either in the form of an actual answer to the question asked, or by the proof of the impossibility of its solution and therewith the necessary failure of all attempts."

David Hilbert, Mathematical Problems [1900]
Martin Davis’s Conjecture: From Algorithms to Sets

**Martin Davis’ Conjecture.** Every effectively enumerable set is Diophantine.
Martin Davis’s Conjecture: From Algorithms to Sets

**Martin Davis’ Conjecture.** Every effectively enumerable set is Diophantine.

**Definition.** A set of natural numbers is effectively enumerable if it is the range of a computable function.
Martin Davis’s Conjecture: From Algorithms to Sets

**Martin Davis’ Conjecture.** Every effectively enumerable set is Diophantine.

**Definition.** A set of natural numbers is effectively enumerable if it is the range of a computable function

**Definition.** A function is computable if its graph is effectively enumerable
Martin Davis’s Conjecture: From Algorithms to Sets

**Martin Davis’ Conjecture.** Every effectively enumerable set is Diophantine.

**Definition.** A set of natural numbers is effectively enumerable if it is the range of a computable function.

**Definition.** A function is computable if its graph is effectively enumerable.

Davis Normal Form: Proof via Arithmetization

**Theorem (Davis Normal Form [1953]).** Every effectively enumerable set \( M \) has a representation of the form

\[
a \in M \iff \exists z \forall y \leq z \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m, y, z) = 0]
\]

where \( P \) is a polynomial with integer coefficients and \( \forall y \leq z \) is the bounded universal quantifier “for all \( y \) not greater than \( z \)”.

Davis Normal Form: Proof via Arithmetization

Theorem (Davis Normal Form [1953]). Every effectively enumerable set $M$ has a representation of the form

$$a \in M \iff \exists z \forall y \leq z \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m, y, z) = 0]$$

where $P$ is a polynomial with integer coefficients and $\forall y \leq z$ is the bounded universal quantifier “for all $y$ not greater than $z$”.

Theorem (Kurt Gödel [1931]). Every effectively enumerable set $M$ has an arithmetical representation.
Effectively enumerable Sets are effectively Diophantine

DPRM-theorem. The notions of a Diophantine set and the notion of an effectively enumerable set coincide.
Effectively enumerable Sets are effectively Diophantine

**DPRM-theorem.** The notions of a Diophantine set and the notion of an effectively enumerable set coincide.

Given an arbitrary effectively enumerable set \( \mathcal{M} \) presented in any standard form one can construct corresponding polynomial giving Diophantine representation of the set.
Effectively enumerable Sets are effectively Diophantine

**DPRM-theorem.** The notions of a Diophantine set and the notion of an effectively enumerable set coincide.

Given an arbitrary effectively enumerable set $M$ presented in any standard form one can construct corresponding polynomial giving Diophantine representation of the set.

1. Construction of an arithmetical formula with many bounded universal quantifiers;
Effectively enumerable Sets are effectively Diophantine

**DPRM-theorem.** The notions of a Diophantine set and the notion of an effectively enumerable set coincide.

Given an arbitrary effectively enumerable set $M$ presented in any standard form one can construct corresponding polynomial giving Diophantine representation of the set.

1. Construction of an arithmetical formula with many bounded universal quantifiers;
2. Transformation of this formula into Davis normal form with single bounded universal quantifier;
Effectively enumerable Sets are effectively Diophantine

**DPRM-theorem.** The notions of a Diophantine set and the notion of an effectively enumerable set coincide.

Given an arbitrary effectively enumerable set $M$ presented in any standard form one can construct corresponding polynomial giving Diophantine representation of the set.

1. Construction of an arithmetical formula with many bounded universal quantifiers;
2. Transformation of this formula into Davis normal form with single bounded universal quantifier;
3. Elimination of the single bounded universal quantifier at the cost of passing to exponential Diophantine equations;
Effectively enumerable Sets are effectively Diophantine

**DPRM-theorem.** The notions of a Diophantine set and the notion of an effectively enumerable set coincide.

Given an arbitrary effectively enumerable set $M$ presented in any standard form one can construct corresponding polynomial giving Diophantine representation of the set.

1. Construction of an arithmetical formula with many bounded universal quantifiers;
2. Transformation of this formula into Davis normal form with single bounded universal quantifier;
3. Elimination of the single bounded universal quantifier at the cost of passing to exponential Diophantine equations;
4. Elimination of the exponentiation.

\[ AX \rightarrow XB \]
Theorem (E. Kummer [1852])

In order to calculate $\beta_p$:
write $a$ and $b$ in base-$p$ notation and add them; the number of carries from digit to
digit performed during this addition is exactly $\beta_p$. 

<table>
<thead>
<tr>
<th></th>
<th>$a_6$</th>
<th>$a_5$</th>
<th>$a_4$</th>
<th>$a_3$</th>
<th>$a_2$</th>
<th>$a_1$</th>
<th>$a_0$</th>
</tr>
</thead>
</table>


Theorem (E. Kummer [1852])

In order to calculate $\beta_p$:

- Write $a$ and $b$ in base-$p$ notation and add them;
- The number of carries from digit to digit performed during this addition is exactly $\beta_p$.
Theorem (E. Kummer [1852])

In order to calculate $\beta_p$: write $a$ and $b$ in base-$p$ notation and add them; the number of carries from digit to digit performed during this addition is exactly $\beta_p$.
$\sum_{k=0}^{\infty} a_k p^k$

$$\binom{a + b}{b} = \frac{(a + b)!}{a! \cdot b!} = 2^{\beta_2} \cdot 3^{\beta_3} \cdot 5^{\beta_5} \cdots$$

**Theorem (E. Kummer [1852])** In order to calculate $\beta_p$: write $a$ and $b$ in base-$p$ notation and add them; the number of carries from digit to digit performed during this addition is exactly $\beta_p$. 

---

**Existential Arithmetization I: Turing Machines**

|   | $a_6$ | $a_5$ | $a_4$ | $a_3$ | $a_2$ | $a_1$ | $a_0$ |
Existential Arithmetization II: Register Machines

A register machine has a finite number of registers $R_1, \ldots, R_n$ each of which is capable of containing an arbitrarily large natural number. The machine performs a program consisting of finitely many instructions labeled by $S_1, \ldots, S_m$.

I. $S_k$: $R_l + +; \text{ goto } S_i$

II. $S_k$: if $R_l > 0$ then $R_l --; \text{ goto } S_i$ else goto $S_j$

III. $S_k$: STOP

Existential Arithmetization II: Register Machines

A register machine has a finite number of registers $R_1, \ldots, R_n$ each of which is capable of containing an arbitrarily large natural number. The machine performs a program consisting of finitely many instructions labeled by $S_1, \ldots, S_m$.

I. $S_k$: $R_l++$; goto $S_i$

II. $S_k$: if $R_l > 0$ then $R_l--$; goto $S_i$ else goto $S_j$

III. $S_k$: STOP

Existential Arithmetization III: Partial Recursive Functions

\[ C^n_q(a_1, \ldots, a_n) = q \]
Existential Arithmetization III: Partial Recursive Functions

\[ C_q^n(a_1, \ldots, a_n) = q \]

\[ U_k^n(a_1, \ldots, a_n) = a_k \]
Existential Arithmetization III: Partial Recursive Functions

\[
\begin{align*}
C_q^n(a_1, \ldots, a_n) &= q \\
U_k^n(a_1, \ldots, a_n) &= a_k \\
S(a) &= a + 1
\end{align*}
\]
Existential Arithmetization III: Partial Recursive Functions

\[ C^n_q(a_1, \ldots, a_n) = q \]
\[ U^n_k(a_1, \ldots, a_n) = a_k \]
\[ S(a) = a + 1 \]
\[ f(a_1, \ldots, a_n) = g(h_1(a_1, \ldots, a_n), \ldots, h_m(a_1, \ldots, a_n)) \]
Existential Arithmetization III: Partial Recursive Functions

\[
\begin{align*}
C^n_q(a_1, \ldots, a_n) & = q \\
U^n_k(a_1, \ldots, a_n) & = a_k \\
S(a) & = a + 1 \\
f(a_1, \ldots, a_n) & = g(h_1(a_1, \ldots, a_n), \ldots, h_m(a_1, \ldots, a_n)) \\
f(0, a_1, \ldots, a_n) & = g(a_1, \ldots, a_n), \\
f(a + 1, a_1, \ldots, a_n) & = h(a, f(a, a_1, \ldots, a_n), a_1, \ldots, a_n)
\end{align*}
\]
Existential Arithmetization III: Partial Recursive Functions

\[ C^n_q(a_1, \ldots, a_n) = q \]
\[ U^m_k(a_1, \ldots, a_n) = a_k \]
\[ S(a) = a + 1 \]
\[ f(a_1, \ldots, a_n) = g(h_1(a_1, \ldots, a_n), \ldots, h_m(a_1, \ldots, a_n)) \]
\[ f(0, a_1, \ldots, a_n) = g(a_1, \ldots, a_n), \]
\[ f(a + 1, a_1, \ldots, a_n) = h(a, f(a, a_1, \ldots, a_n), a_1, \ldots, a_n) \]
\[ f(a_1, \ldots, a_n) = \mu_y(g(y, a_1, \ldots, a_n) = 0) \]
Existential Arithmetization III: Partial Recursive Functions

\[ C_q^n(a_1, \ldots, a_n) = q \]
\[ U_k^n(a_1, \ldots, a_n) = a_k \]
\[ S(a) = a + 1 \]
\[ f(a_1, \ldots, a_n) = g(h_1(a_1, \ldots, a_n), \ldots, h_m(a_1, \ldots, a_n)) \]
\[ f(0, a_1, \ldots, a_n) = g(a_1, \ldots, a_n), \]
\[ f(a + 1, a_1, \ldots, a_n) = h(a, f(a, a_1, \ldots, a_n), a_1, \ldots, a_n) \]
\[ f(a_1, \ldots, a_n) = \mu_y(g(y, a_1, \ldots, a_n) = 0) \]

\[ f(a_1, \ldots, a_n) = b \iff \exists x_1 \ldots x_m[F(a_1, \ldots, a_n, b, x_1, \ldots, x_m) = 0] \]
Existential Arithmetization III: Partial Recursive Functions

\[ C^n_q(a_1, \ldots, a_n) = b \iff b - q = 0 \]
Existential Arithmetization III: Partial Recursive Functions

\[ C^q_n(a_1, \ldots, a_n) = b \iff b - q = 0 \]
\[ U^n_k(a_1, \ldots, a_n) = b \iff b - a_k = 0 \]
Existential Arithmetization III: Partial Recursive Functions

\[ C^n_q(a_1, \ldots, a_n) = b \iff b - q = 0 \]
\[ U^n_k(a_1, \ldots, a_n) = b \iff b - a_k = 0 \]
\[ S(a) = b \iff b - a - 1 = 0 \]
Existential Arithmetization III: Partial Recursive Functions

\[ C_q^n(a_1, \ldots, a_n) = b \iff b - q = 0 \]
\[ U_k^n(a_1, \ldots, a_n) = b \iff b - a_k = 0 \]
\[ S(a) = b \iff b - a - 1 = 0 \]
\[ f(a_1, \ldots, a_n) = g(h_1(a_1, \ldots, a_n), \ldots, h_m(a_1, \ldots, a_n)) \]
\[ f(a_1, \ldots, a_n) = b \iff \exists c_1 \ldots c_m \]
\[ g(c_1, \ldots, c_m) = b \]
\[ \& h_1(a_1, \ldots, a_n) = c_1 \]
\[ \vdots \]
\[ \& h_m(a_1, \ldots, a_n) = c_m \]
Existential Arithmetization III: Partial Recursive Functions

\[
\begin{align*}
  f(0, c) &= g(c), \\
  f(a + 1, c) &= h(a, f(a), c)
\end{align*}
\]
Existential Arithmetization III: Partial Recursive Functions

\[
\begin{align*}
    f(0, c) &= g(c), \\
    f(a + 1, c) &= h(a, f(a), c)
\end{align*}
\]

\[
\begin{bmatrix}
    f(0, c) \\
    f(1, c) \\
    \vdots \\
    f(a, c) \\
    f(a + 1, c)
\end{bmatrix}
= 
\begin{bmatrix}
    \begin{bmatrix}
        0 \\
        \vdots \\
        a - 1 \\
        a
    \end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
    g(c) \\
    f(0, c) \\
    \vdots \\
    f(a - 1, c) \\
    f(a, c)
\end{bmatrix}
\begin{bmatrix}
    c \\
    \vdots \\
    c \\
    c
\end{bmatrix}
\]
Existential Arithmetization III: Partial Recursive Functions

\[
\begin{bmatrix}
X \\
\vdots \\
x
\end{bmatrix} = \\
\begin{bmatrix}
g(c) \\
\vdots \\
h\left( \begin{bmatrix}
0 \\
\vdots \\
a
\end{bmatrix}, X, \begin{bmatrix}
c \\
\vdots \\
c
\end{bmatrix} \right) \\
\end{bmatrix}
\]
Existential Arithmetization III: Partial Recursive Functions

\[
\begin{bmatrix}
X \\
X
\end{bmatrix} = h\left(\begin{bmatrix}
0 \\
\vdots \\
a
\end{bmatrix}, X, \begin{bmatrix}
c \\
\vdots \\
c
\end{bmatrix}\right)
\]

\[
X = f\left(\begin{bmatrix}
0 \\
\vdots \\
a
\end{bmatrix}, \begin{bmatrix}
c \\
\vdots \\
c
\end{bmatrix}\right) = \begin{bmatrix}
f(0, c) \\
\vdots \\
f(a, c)
\end{bmatrix}, \quad x = f(a + 1, c)
\]
Existential Arithmetization III: Partial Recursive Functions

\[
f\left( \begin{bmatrix} a_{1,1} \\ \vdots \\ a_{l,1} \end{bmatrix}, \ldots, \begin{bmatrix} a_{1,n} \\ \vdots \\ a_{l,n} \end{bmatrix} \right) = \begin{bmatrix} f(a_{1,1}, \ldots, a_{1,n}) \\ \vdots \\ f(a_{l,1}, \ldots, a_{l,n}) \end{bmatrix}
\]

\[
f(A_1, \ldots, A_n) = B \iff \exists x_1 \ldots x_m [F(\bar{A}_1, \ldots, \bar{A}_n, \bar{B}, x_1, \ldots, x_m) = 0]
\]
Speeding up Diophantine Equations

Theorem (M.Davis [1973] after M.Blum [1967]).
Theorem (M.Davis [1973] after M.Blum [1967]). For every total computable function $\alpha(a, x)$ one can construct two one-parameter Diophantine equations

$$P_1(a, x_1, \ldots, x_k) = 0, \quad P_2(a, x_1, \ldots, x_k) = 0 \quad (*)$$

such that
Speeding up Diophantine Equations

Theorem (M.Davis [1973] after M.Blum [1967]). For every total computable function \( \alpha(a, x) \) one can construct two one-parameter Diophantine equations

\[
P_1(a, x_1, \ldots, x_k) = 0, \quad P_2(a, x_1, \ldots, x_k) = 0
\] (*)

such that

(i) for every value of the parameter \( a \) exactly one of these two equations has a solution;
Theorem (M.Davis [1973] after M.Blum [1967]). For every total computable function \( \alpha(a, x) \) one can construct two one-parameter Diophantine equations 

\[
P_1(a, x_1, \ldots, x_k) = 0, \quad P_2(a, x_1, \ldots, x_k) = 0 \tag{*}
\]

such that

(i) for every value of the parameter \( a \) exactly one of these two equations has a solution;

(ii) if Diophantine equations 

\[
Q_1(a, y_1, \ldots, y_l) = 0, \quad Q_2(a, y_1, \ldots, y_l) = 0 \tag{**}
\]

are solvable for the same values of the parameter \( a \) as, respectively, equations \((*)\),
Speeding up Diophantine Equations

Theorem (M. Davis [1973] after M. Blum [1967]). For every total computable function $\alpha(a, x)$ one can construct two one-parameter Diophantine equations

$$P_1(a, x_1, \ldots, x_k) = 0, \quad P_2(a, x_1, \ldots, x_k) = 0$$

such that

(i) for every value of the parameter $a$ exactly one of these two equations has a solution;

(ii) if Diophantine equations

$$Q_1(a, y_1, \ldots, y_l) = 0, \quad Q_2(a, y_1, \ldots, y_l) = 0$$

are solvable for the same values of the parameter $a$ as, respectively, equations (*), then one can construct a third pair of Diophantine equations

$$R_1(a, z_1, \ldots, z_m) = 0, \quad R_2(a, z_1, \ldots, z_m) = 0$$

such that

$$y_1 + \cdots + y_l > \alpha(a, z_1 + \cdots + z_m).$$
Speeding up Diophantine Equations

Theorem (M.Davis [1973] after M.Blum [1967]). For every total computable function $\alpha(a, x)$ one can construct two one-parameter Diophantine equations

$$P_1(a, x_1, \ldots, x_k) = 0, \quad P_2(a, x_1, \ldots, x_k) = 0 \quad (\ast)$$

such that

(i) for every value of the parameter $a$ exactly one of these two equations has a solution;

(ii) if Diophantine equations

$$Q_1(a, y_1, \ldots, y_l) = 0, \quad Q_2(a, y_1, \ldots, y_l) = 0 \quad (\ast\ast)$$

are solvable for the same values of the parameter $a$ as, respectively, equations (\ast), then one can construct a third pair of Diophantine equations

$$R_1(a, z_1, \ldots, z_m) = 0, \quad R_2(a, z_1, \ldots, z_m) = 0 \quad (\ast\ast\ast)$$

such that

- these equations are again solvable for the same values of the parameter $a$ as, respectively, equations (\ast);
Theorem (M. Davis [1973] after M. Blum [1967]). For every total computable function $\alpha(a, x)$ one can construct two one-parameter Diophantine equations

\[ P_1(a, x_1, \ldots, x_k) = 0, \quad P_2(a, x_1, \ldots, x_k) = 0 \quad (*) \]

such that

(i) for every value of the parameter $a$ exactly one of these two equations has a solution;

(ii) if Diophantine equations

\[ Q_1(a, y_1, \ldots, y_l) = 0, \quad Q_2(a, y_1, \ldots, y_l) = 0 \quad (***) \]

are solvable for the same values of the parameter $a$ as, respectively, equations ($*$), then one can construct a third pair of Diophantine equations

\[ R_1(a, z_1, \ldots, z_m) = 0, \quad R_2(a, z_1, \ldots, z_m) = 0 \quad (** *) \]

such that

\[ \begin{align*}
& \text{these equations are again solvable for the same values of the parameter } a \text{ as, respectively, equations } (*); \\
& \text{for all sufficiently large values of the parameter } a \text{ for every solution } y_1, \ldots, y_l \text{ of one of the equations } (***) \text{ there exists a solution } z_1, \ldots, z_m \text{ of the corresponding equation } (** *) \text{ such that} \\
& y_1 + \cdots + y_l > \alpha(a, z_1 + \cdots + z_m). 
\end{align*} \]
Diophantine Machines: Capturing Nondeterminism


\[ P(a, x_1, \ldots, x_m) \equiv 0 \]

**DPRM-theorem:** NDDMs are as powerful as, say, Turing machines, i.e., every set acceptable by a Turing machine is accepted by some NDDM, and, of course, *vice versa.*
Definition. A purely existential representation

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m T(a, x_1, \ldots, x_m) \]

is called single-fold if for given value of the parameter \( a \) there exists at most one choice of the values of \( x_1, \ldots, x_m \).
Unambiguity: Equations with Unique Solution

**Definition.** A purely existential representation

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m T(a, x_1, \ldots, x_m) \]

is called *single-fold* if for given value of the parameter \( a \) there exists at most one choice of the values of \( x_1, \ldots, x_m \).

**Theorem (Yu. Matiyasevich [1974] as an improvement of DPR[1961]).** Every effectively enumerable set \( \mathcal{M} \) has a single-fold exponential Diophantine representation

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m [E(a, x_1, x_2, \ldots, x_m) = 0] \]

where \( E \) is an exponential polynomial.
**Unambiguity: Equations with Unique Solution**

**Definition.** A purely existential representation

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m T(a, x_1, \ldots, x_m) \]

is called *single-fold* if for given value of the parameter \( a \) there exists at most one choice of the values of \( x_1, \ldots, x_m \).

**Theorem (Yu.Matiyasevich [1974] as an improvement of DPR[1961]).** Every effectively enumerable set \( \mathcal{M} \) has a single-fold exponential Diophantine representation

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m [E(a, x_1, x_2, \ldots, x_m) = 0] \]

where \( E \) is an exponential polynomial.

**Open Problem.** Does every effectively enumerable set \( \mathcal{M} \) have a single-fold Diophantine representation

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m [P(a, x_1, x_2, \ldots, x_m) = 0] \]

where \( P \) is a polynomial?
Unambiguity: Equations with Unique Solution

Open Problem. Does every effectively enumerable set $\mathcal{M}$ have a single-fold Diophantine representation

$$a \in \mathcal{M} \iff \exists x_1 \ldots x_m[P(a, x_1, x_2, \ldots, x_m) = 0]$$

where $P$ is a polynomial?
Unambiguity: Equations with Unique Solution

Open Problem. Does every effectively enumerable set $\mathcal{M}$ have a single-fold Diophantine representation

$$a \in \mathcal{M} \iff \exists x_1 \ldots x_m [P(a, x_1, x_2, \ldots, x_m) = 0]$$

where $P$ is a polynomial?

Open Problem (reformulation) Are unambiguous NDDMs are as powerful as (deterministic) Turing machines?
Diophantine Complexity: Time and Space vs Size

\[ P(a, x_1, \ldots, x_m) = 0 \]

- **NDDM**
  - **input** \( a \)
  - \( P(a, x_1, \ldots, x_m) \)
  - **YES**
  - **NO**
  - **accept** \( a \)
  - **reject**

**SIZE\( (a) \)**: the minimal possible value of
\[ |x_1| + \cdots + |x_m| \]
where \( |x| \) denotes the binary length of \( x \).
Diophantine Complexity: Time and Space vs Size

\[ P(a, x_1, \ldots, x_m) \neq 0 \]

- **YES**
- **NO**

**SIZE(a)** = the minimal possible value of \(|x_1| + \cdots + |x_m|\) where \(|x|\) denotes the binary length of \(x\).
Diophantine Complexity: D vs NP

Leonard Adleman and Kenneth Manders [1975] introduced the class \( \mathbf{D} \) consisting of all sets \( \mathbf{M} \) having representations of the form

\[
a \in \mathbf{M} \iff \exists x_1 \ldots x_m \left[ P(a, x_1, \ldots, x_m) = 0 \& |x_1| + \cdots + |x_m| \leq |a|^k \right]
\]

where \( |x| \) denotes the binary length of \( x \).
Diophantine Complexity: D vs NP

Leonard Adleman and Kenneth Manders [1975] introduced the class $D$ consisting of all sets $M$ having representations of the form

$$a \in M \iff \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m) = 0 \& |x_1| + \cdots + |x_m| \leq |a|^k]$$

where $|x|$ denotes the binary length of $x$.

Open question. $D \supseteq NP$. 
Diophantine Complexity: D vs NP

Leonard Adleman and Kenneth Manders [1975] introduced the class $D$ consisting of all sets $M$ having representations of the form

$$a \in M \iff \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m) = 0 \& |x_1| + \cdots + |x_m| \leq |a|^k]$$

where $|x|$ denotes the binary length of $x$.

Open question. $D \ ? \ NP$.

**Theorem (Chris Pollett [2003]).**

$$D \subseteq \text{co-NLOGTIME} \implies D = NP$$
Diophantine Complexity: D vs NP

Leonard Adleman and Kenneth Manders [1975] introduced the class $D$ consisting of all sets $M$ having representations of the form

$$a \in M \iff \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m) = 0 \& |x_1| + \cdots + |x_m| \leq |a|^k]$$

where $|x|$ denotes the binary length of $x$.

**Open question.** $D \nsubseteq NP$.

**Theorem (Chris Pollett [2003]).**

$$D \subseteq \text{co-NLOGTIME} \implies D = NP$$

Helger Lipmaa [2003] introduced $PD$, the “deterministic part” of the class $D$. 
Above Hilbert’s Tenth Problem: Computational Chaos

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m) = 0] \]
Above Hilbert’s Tenth Problem: Computational Chaos

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m[P(a, x_1, \ldots, x_m) = 0] \]

\[ \mathcal{M}_n = \mathcal{M} \cap \{a \mid a \leq n\} \]
Above Hilbert’s Tenth Problem: Computational Chaos

\[ a \in \mathcal{M} \iff \exists x_1 \ldots x_m [P(a, x_1, \ldots, x_m) = 0] \]

\[ \mathcal{M}_n = \mathcal{M} \cap \{ a \mid a \leq n \} \]

The set \( \mathcal{M}_n \) can be effectively calculated from \( \|\mathcal{M}_n\| \), the cardinality of \( \mathcal{M}_n \).
Above Hilbert’s Tenth Problem: Computational Chaos

\[ a \in M \iff \exists x_1 \ldots x_m[P(a, x_1, \ldots, x_m) = 0] \]

\[ M_n = M \cap \{a \mid a \leq n\} \]

The set \( M_n \) can be effectively calculated from \( \|M_n\| \), the cardinality of \( M_n \).

The descriptive or Kolmogorov complexity of \( M_n \) is at most \( \log(n) \).
Above Hilbert’s Tenth Problem: Computational Chaos

Gregory Chaitin [1987] constructed a particular one-parameter exponential Diophantine equation and considered the set of all values of the parameter for which the equation has infinitely many solutions:

\[ a \in M \iff \exists \infty x_1 \ldots x_m [E(a, x_1, x_2, \ldots, x_m) = 0] \]

He proved that so called prefix-free Kolmogorov complexity of the initial segment

\[ M_n = M \cap \{a \mid a \leq n\} \]

is equal to \( n \) (up to an additive constant).
Above Hilbert’s Tenth Problem: Computational Chaos

Gregory Chaitin [1987] constructed a particular one-parameter exponential Diophantine equation and considered the set of all values of the parameter for which the equation has infinitely many solutions:

\[ a \in \mathcal{M} \iff \exists \infty x_1 \ldots x_m [E(a, x_1, x_2, \ldots, x_m) = 0] \]

He proved that so called \textit{prefix-free} Kolmogorov complexity of the initial segment

\[ \mathcal{M}_n = \mathcal{M} \cap \{a | a \leq n\} \]

is equal to \( n \) (up to an additive constant).

Informally, one can say that the set \( \mathcal{M} \) is completely chaotic.
Toby Ord and Tien D. Kieu [2003] constructed another particular one-parameter exponential Diophantine equation which for every value of the parameter has only finitely many solutions and considered the set of all values of the parameter for which the equation has even number of solutions:

\[ a \in \mathcal{M} \iff \exists \text{even } x_1 \ldots x_m [E(a, x_1, x_2, \ldots, x_m) = 0] \]

They proved that the prefix-free Kolmogorov complexity of the initial segment

\[ \mathcal{M}_n = \mathcal{M} \cap \{a \mid a \leq n \} \]

is also equal to \( n \) (up to an additive constant).
A Generalization of Hilbert’s Tenth Problem

**Theorem (M. Davis [1972]).** Let $\mathcal{U}$ be a proper subset of the set \{0, 1, 2, \ldots, $\infty$\}. There is no algorithm do decide, for a given Diophantine equation, whether the number of solutions of this equation belongs to $\mathcal{U}$. 
Theorem (M. Davis [1972]). Let $\mathcal{U}$ be a proper subset of the set $\{0, 1, 2, \ldots, \infty\}$. There is no algorithm to decide, for a given Diophantine equation, whether the number of solutions of this equation belongs to $\mathcal{U}$.

Hilbert’s 10th problem is the case $\mathcal{U} = \{0\}$. 
Theorem (Matiyasevich [2006]). Let $\mathcal{U}$ be a decidable infinite set with infinite complement. One can construct an exponential Diophantine equation which for every value of the parameter has only finitely many solutions and such that for the set

$$a \in \mathcal{M} \iff \exists^{\mathcal{U}} x_1 \ldots x_m \left[ E(a, x_1, x_2, \ldots, x_m) = 0 \right]$$

the prefix-free Kolmogorov complexity of its initial segment

$$\mathcal{M}_n = \mathcal{M} \cap \{a \mid a \leq n\}$$

is equal to $n$ (up to an additive constant).
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced Diophantine games.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

Peter selects the values of the parameters \( a_1, \ldots, a_m \)
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

*Peter selects the values of the parameters* \( a_1, \ldots, a_m \)

*Ursula selects the values of the unknowns* \( x_1, \ldots, x_m \)

Ursula is the winner if and only if the value of the polynomial turns out to equal to 0.
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

Peter selects the values of the parameters \( a_1, \ldots, a_m \)

Ursula selects the values of the unknowns \( x_1, \ldots, x_m \)

- Peter selects \( a_1 \)
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced \textit{Diophantine games}.

\begin{equation}
P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0
\end{equation}

Peter selects the values of the \textit{parameters} $a_1, \ldots, a_m$  
Ursula selects the values of the \textit{unknowns} $x_1, \ldots, x_m$

- Peter selects $a_1$
- Ursula selects $x_1$
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

Peter selects the values of the parameters \(a_1, \ldots, a_m\)

Ursula selects the values of the unknowns \(x_1, \ldots, x_m\)

- Peter selects \(a_1\)
- Ursula selects \(x_1\)
- Peter selects \(a_2\)

Ursula is the winner if and only if the value of the polynomial turns out to equal to 0.
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

*Peter selects the values of the parameters* \(a_1, \ldots, a_m\)

*Ursula selects the values of the unknowns* \(x_1, \ldots, x_m\)

- Peter selects \(a_1\)
- Ursula selects \(x_1\)
- Peter selects \(a_2\)
- Ursula selects \(x_2\)

Ursula is the winner if and only if the value of the polynomial turns out to equal to 0.
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

*Peter selects the values of the parameters* \( a_1, \ldots, a_m \)  
*Ursula selects the values of the unknowns* \( x_1, \ldots, x_m \)

- Peter selects \( a_1 \)
- Ursula selects \( x_1 \)
- Peter selects \( a_2 \)
- Ursula selects \( x_2 \)
- .........................................................
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games.*

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

Peter selects the values of the parameters \( a_1, \ldots, a_m \)

Ursula selects the values of the unknowns \( x_1, \ldots, x_m \)

- Peter selects \( a_1 \)
- Ursula selects \( x_1 \)
- Peter selects \( a_2 \)
- Ursula selects \( x_2 \)
- ............................................................
- Peter selects \( a_m \)

Ursula is the winner if and only if the value of the polynomial turns out to equal to 0.
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

Peter selects the values of the *parameters* \( a_1, \ldots, a_m \)

Ursula selects the values of the *unknowns* \( x_1, \ldots, x_m \)

- Peter selects \( a_1 \)
- Ursula selects \( x_1 \)
- Peter selects \( a_2 \)
- Ursula selects \( x_2 \)
- ..............................................................
- Peter selects \( a_m \)
- Ursula selects \( x_m \)

Ursula is the winner if and only if the value of the polynomial turns out to equal to 0.
Diophantine Games

James Jones [1974], based on ideas of Michael O. Rabin [1957], introduced *Diophantine games*.

\[ P(a_1, \ldots, a_m, x_1, \ldots, x_m) = 0 \]

Peter selects the values of the parameters \( a_1, \ldots, a_m \)

Ursula selects the values of the unknowns \( x_1, \ldots, x_m \)

- Peter selects \( a_1 \)
- Ursula selects \( x_1 \)
- Peter selects \( a_2 \)
- Ursula selects \( x_2 \)
- .............................................................
- Peter selects \( a_m \)
- Ursula selects \( x_m \)

Ursula is the winner if and only if the value of the polynomial turns out to equal to 0.
Simple Sets and Difficult Games

Question: Who has the winning strategy in the following game:

\[(x_1 + a_2)^2 + 1 - (x_2 + 2)(x_3 + 3) = 0\]
Simple Sets and Difficult Games

**Question:** Who has the winning strategy in the following game:

\[(x_1 + a_2)^2 + 1 - (x_2 + 2)(x_3 + 3) = 0\]

**Hint:** Peter is the winner if and only if there infinitely many primes of the form \(n^2 + 1\).
Theorem (Jones[1982]) In the game:

\[
\left\{ \{a_1 + a_6 + 1 - x_4\}^2 \cdot \{ \langle (a_6 + a_7)^2 + 3a_7 + a_6 - 2x_4 \rangle^2 \\
+ \langle [(x_9 - a_7)^2 + (x_10 - a_9)^2] [(x_9 - a_6)^2 + (x_10 - a_8)^2 ((x_4 - a_1)^2 \\
+ (x_10 - a_9 - x_1)^2)] [(x_9 - 3x_4)^2 + (x_10 - a_8 - a_9)^2] [(x_9 - 3x_4 - 1)^2 \\
+ (x_10 - a_8 a_9)^2] - a_{12} - 1 \rangle^2 + \langle [x_10 + a_{12} + a_{12}x_9 a_4 - a_3]^2 \\
+ [x_5 + a_{13} - x_9 a_4]^2 \rangle \right\} - x_{13} - 1 \right\} \{a_1 + x_5 + 1 - a_5\} \left\{ \langle (x_5 - x_6)^2 \\
+ 3x_6 + x_5 - 2a_5 \rangle^2 + \langle [(a_{10} - x_6)^2 + (a_{11} - x_8)^2] [(a_{10} - x_5)^2 \\
+ (a_{11} - x_7)^2 ((a_5 - a_1)^2 + (a_{11} - x_8 - a_2)^2)] [(a_{10} - 3a_5)^2 \\
+ (a_{11} - x_7 - x_8)^2] [(a_{10} - 3a_5 - 1)^2 + (a_{11} - x_7 x_8)^2] - x_{11} - 1 \rangle^2 \\
+ \langle [a_{11} + x_{11} + x_{11} a_{10} x_3 - x_2]^2 + [a_{11} + x_{12} - a_{10} x_3]^2 \rangle \right\} = 0
\]

Ursula has a winning strategy but no computable winning strategy.
Other Applications of Hilbert’s 10th problem to Games

A. H. Lachlan [1970] introduced another kind of game as a possible tool to establish results about the lattice of effectively enumerable sets. He conjectured that for these games it can be decided which of the two players has the winning strategy. He obtained partial results in this direction but recently M. Kummer [2006] proved many results about undecidability of Lachlan’s games using the undecidability of Hilbert’s tenth problem.
Other Applications of Hilbert’s 10th problem to Games

K. Prasad [1991] proved for “traditional” multi-person non-cooperative games with polynomial \textit{payoff functions} that there is no algorithm to decide whether a game has a \textit{Nash equilibrium in pure strategies}; for a similar result for \textit{mixed strategies} one would need single-fold representations and thus at present undecidability is established only for the case when the payoff functions are exponential polynomials.
Other Applications of Hilbert’s 10th problem to Games

K. Prasad [1991] proved for “traditional” multi-person non-cooperative games with polynomial *payoff functions* that there is no algorithm to decide whether a game has a *Nash equilibrium in pure strategies*; for a similar result for *mixed strategies* one would need single-fold representations and thus at present undecidability is established only for the case when the payoff functions are exponential polynomials.

K. Prasad [1991] also translated Chaitin’s result from the question about the infinitude of the number of solutions of an exponential Diopantine equation to the question of the infinitude of the number of Nash equilibria in multi-person non-cooperative games.
Models of computation motivated by biology: Recombination

Splicing Rule:

\[ U_1 \# U_2 \& U_3 \# U_4 : \langle x_1 U_1 U_2 x_2, y_1 U_3 U_4 y_2 \rangle \mapsto \langle x_1 U_1 U_4 x_2, y_1 U_3 U_2 y_2 \rangle \]
Models of computation motivated by biology: Recombination

Splicing Rule:

\[ U_1 \# U_2 \& U_3 \# U_4 : \langle x_1 U_1 U_2 x_2, y_1 U_3 U_4 y_2 \rangle \mapsto \langle x_1 U_1 U_4 x_2, y_1 U_3 U_2 y_2 \rangle \]

Splicing rules by themselves can generate only regular languages; however, with some additional controls they are as powerful as, say, Turing machines.

P. Frisco [2001] used the power of Diophantine equations in a (new) proof of this fact.
Membrane computing, motivated by metabolism in living cells, was introduced by George Paun [1998].
Models of computation motivated by biology: Metabolism

Membrane computing, motivated by metabolism in living cells, was introduced by George Paun [1998]