P versus NP problem solution

Mikhail Kupchik

app. 183, building 2, Svobody avenue, Kiev, Ukraine

Abstract

Contains the solution of P versus NP compexity class relation problem. As expected, P $\neq$ NP. Idea of the proof is the following: Supposing P = NP we build a problem in class P of fixed time-complexity degree, with the following trait: every problem from P reduces to it, thus leading to contradiction with time hierarchy theorem.

Key words: Computational complexity, completixy class, nondeterministical computations, "easy to check" vs. "easy to find".

Email address: `mikhkup@mail.ru` (Mikhail Kupchik).

1 Introducion

Let's adjust terminology first.

(String) problem is any function $\omega : I \to S,\ \ I, S \subseteq \{0,1\}^+$. *Decision problem* is a problem for which $S = \{0,1\}$.

We use *single-tape Turing machines* as mathematical model of algorithm. Single tape contains input string, it is used as work tape and contains output string when TM halts. This is sufficient enough for this paper, as sublinear-space (-time) problems are not discussed separately from P-problems here. We consider only such Turing machines which halt on every input of any length. Tape alphabet is $\{0,1,\Lambda\}$. Tape is infinite in both directions, input string is placed between two infinite chains of Λ symbol (blank symbol) at left and right side. *Set of internal states* (which is not dependant on input string) is $Q = \{0, 1 \ldots q\}$. *Initial state* is state 0. There are three head movement cases per each step: keep position, move one cell left and move one cell right.

With respect to data representation. It's clear that, for example, representation of numbers in unary alphabet is superfluous and distorts the subject of investigation. On the other hand, there is no sense in looking for too "optimal" representation: the paper discusses the following question: can some quantity, like calculation time, space or some string length, be limited by a polynomial over problem's input string length. In other words, rather rough choice will be made regarding this quantity: either "some fixed-degree polynomial" or "greater than every fixed-degree polynomial". So coefficients and other details, which are determined by representation peculiarities still don't change the choice mentioned above.

A representation α will be considered *optimal*, if there is no representation β such that

$$\forall i\ \forall m \in N\ \ |\beta(i)|^m = o(|\alpha(i)|)$$

and

$$\forall i, j\ \ (\alpha(i) \neq \alpha(j)\ \Rightarrow\ \beta(i) \neq \beta(j))\,.$$

Complexity classes. Class $\mathcal{D}\mathrm{TIME}\,(T(\cdot))$ is a class of problems which are computable within $T(n)$ time limit on every input of length n. Class $\mathcal{P}$:

$$\mathcal{P} \stackrel{def}{=} \bigcup_{k=1}^{\infty} \mathcal{D}\mathrm{TIME}\left(O\left((\cdot)^k\right)\right).$$

Complexity class $\mathcal{NP}$ contains decision problems which solution can be verified within polynomial time. Decision problem $\omega : I \to \{0,1\}$ is in $\mathcal{NP}$ class,

iff there is a decision problem $\delta \in \mathcal{P}$ and polynomial $p(\cdot)$ such that

$$\forall i \in I \left(\omega(i) = 1 \Leftrightarrow \exists c \in \{0,1\}^\star \ (|c| < p(|i|) \wedge \delta\left(\langle i, c\rangle\right) = 1)\right),$$

c is denoted as *certificate*, or *witness*. Decision problem $\omega : I \to \{0,1\}$ is in co-$\mathcal{NP}$ class, iff there is a decision problem $\epsilon \in \mathcal{P}$ and polynomial $q(\cdot)$ such that

$$\forall i \in I \left(\omega(i) = 0 \Leftrightarrow \exists d \in \{0,1\}^\star \ (|d| < q(|i|) \wedge \epsilon\left(\langle i, d\rangle\right) = 1)\right).$$

2 Shrinks. Traits of shrinks of Turing machines of exponential-time complexity

2.1 Shrink definition

Let Ω be any TM with following properties:

(1) solves some decision problem;
(2) computable within $T(\cdot)$ time bound and $S(\cdot)$ space bound;
(3) number of internal states is not greater than 2^C;
(4) transition function is denoted by

$$\eta : \{0,1,\Lambda\} \times \{0,1\}^C \to \{0,1,\Lambda\} \times \{0,1\}^C \times$$

$$\times \{\text{moveleft} = 00, \text{moveright} = 01, \text{dontmove} = 10\};$$

(5) set of final states is F;
(6) when machine halts, it scans tape symbol which is an output.

Let's intoduce additional function

$$\text{convert} : \{0,1\}^n \to \{0,1\}^{2S(n)+C+1} :$$

$\text{convert}_{(i)}(e_1, \dots e_n)$ is calculated by the following rules:

(1) if $0 \le i < 2n$ and if i is even number, then $\text{convert}_i = e_{i/2}$;
(2) if $0 \le i < 2n$ and if i id odd number, then $\text{convert}_i = 0$;
(3) if $2n \le i < S(n)$ then $\text{convert}_i = 1$;
(4) if $S(n) \le i < S(n) + C + 1$ then $\text{convert}_i = 0$.

Definition 1 *Shrink of TM Ω (when input string length is fixed and equal to n) is a function*

$$F : \{0,1\}^{2S(n)+C+1} \to \{0,1\}^{2S(n)+C+1},$$

with a property:

$$\forall x \in \{0,1\}^+ \quad F_{(0)}^{(T(n))}\Big(\mathit{convert}(x)\Big) = \Omega(x), \tag{1}$$

where $A_{(i)}$ is an ith bit of A, $F^{(k)}$ is k-ary F composition.

Theorem 2 *For each TM Ω, which is computable within $\exp(O(\cdot))$ time bound, there exists a shrink. Calculation rule for one bit of this shrink, expressed by a recursive formula, has representation string of length $O(\cdot)$.*

PROOF. Let Ω to be a TM computable within 2^{Bn} time limit. Clearly, space limit is 2^{Bn} too. We define $A \stackrel{def}{=} 2B$ for future clarity between space and time measures.

Without loss of generality we can consider the following statement: when Ω halts, it scans tape cell which contains computation result. (If this is not true, Ω can be modified so that it performs additional steps of moving head to desired position after termination of calculations. This operation requires no more than 2^{Bn} additional steps, i.e. Ω will still belong to $\mathcal{D}\mathrm{TIME}\,(\exp(O(\cdot))$ class.)

Let's introduce some auxiliary denotions:

Function $\phi : \{0, 1, \Lambda\}^\star \to \{0, 1\}^\star$. Injective function from set of TM alphabet strings to set of boolean alphabet strings. One-symbol strings are converted in the following manner:

$$\phi(0) \stackrel{def}{=} 00,\ \phi(1) \stackrel{def}{=} 01,\ \phi(\Lambda) \stackrel{def}{=} 1x;$$

where x is any symbol, let it be 1 for distinctness.
Other strings are converted by rule:

$$\phi(ab) \stackrel{def}{=} \phi(a)\phi(b).$$

Function $\xi : \{0, 1\}^{C+2} \to \{0, 1\}^{C+4}$. As we can see, transition function η, which first argument is converted by ϕ^{-1} function, and first string of result is converted by ϕ function, is partially defined function $\{0, 1\}^{C+2} \to \{0, 1\}^{C+4}$. Let's remove the incompleteness of definition (arbitrarily), and denote the result as ξ.

Function $\chi : \{0, 1\}^{C} \to \{0, 1\}$. Its argument is $\phi(q)$, where q is TM state, $\chi = 1$ iff $q \in F$.

Function $\delta : \{0, 1\}^{3} \to \{0, 1\}^{3}$ is used as auxiliary, to avoid encumbering of fig. 1. It is a decoder of 2-bit binary code, combined with a circuit that blocks it, if stop bit is set.

δ is determined by the following table:

input			output		
1	2	3	1	2	3
0	0	0	1	0	0
0	1	0	0	0	1
1	any	0	0	1	0
any	any	1	0	1	0

With the aid of these functions the shrink of Ω TM can be represented as a set of boolean functions (finite for any fixed length of input), see fig. 1. Let's estimate the growth of representation of this circuit. If upper bound for length of the tape is N, then length of curcuit representation is no more than $O(N)$, clearly. But this is an estimation of *whole F's shrink* representation length.

Let Ω's input string to be fixed and equal to e, and suppose we want to know not all, but one bit of shrink, at any moment of time (step) t.

According to fig. 1,

$$\exists E : \{0,1\}^{(An+1+C)+Bn+2n} \to \{0,1\}^{(An+1+C)+Bn+2n} \quad \forall e \in \{0,1\ldots 2^n - 1\}$$

$$\forall x \in \left\{0,1\ldots 2^{An+1+C} - 1\right\} \quad \forall t \in \left\{0,1\ldots 2^{Bn} - 1\right\}$$

$$E(x;t;e) = \theta\Big(x;t;e;\ E(0;t-1;e), E(1;t-1;e),$$

$$\ldots E(2^{An+1+C} - 1; t-1; e)\Big). \tag{2}$$

E can be calculated in the following way:

(1) If $t = 0$ (initial state),
  (a) If $0 \le x < 2n$ (input string area hit),
    (i) If x is even then

$$E(x;t;e) \text{ is } \frac{x}{2}\text{th bit of string } e$$

    (ii) Else (x is odd) then

$$E(x;t;e) = 0$$

  (b) If $2n \le x < 2^{An}$ (empty tape area hit) then

$$E(x;t;e) = 1$$

  (c) If $2^{An} \le x < 2^{An} + 1 + C$ (hit of internal TM state or stop bit) then

$$E(x;t;e) = 0$$

(2) If $t > 0$ and $E(2^{An}; t-1; e) = 1$ (not initial state, TM Ω has already halted, any bit is requested),

$$E(x; t; e) = E(x; t-1; e)$$

(3) If $t > 0$ and $E(2^{An}; t-1; e) = 0$ (not initial state, TM Ω is not halted at step t), calculate

$$\begin{aligned}
\alpha_1 = \xi_{C+1}(\ & E(2^{An}+1; t-1; e) \dots E(2^{An}+1+C; t-1; e); \\
& E(0; t-1; e), E(1; t-1; e)) \qquad) \\
\alpha_2 = \xi_{C+2}(\ & E(2^{An}+1; t-1; e) \dots E(2^{An}+1+; t-1; e); \\
& E(0; t-1; e), E(1; t-1; e)) \qquad) \\
\alpha_3 = \xi_{C+3}(\ & E(2^{An}+1; t-1; e) \dots E(2^{An}+1+C; t-1; e); \\
& E(0; t-1; e), E(1; t-1; e)) \qquad) \\
\alpha_4 = \xi_{C+4}(\ & E(2^{An}+1; t-1; e) \dots E(2^{An}+1+; t-1; e); \\
& E(0; t-1; e), E(1; t-1; e)) \qquad) \\
\beta_1 = \xi_1(\ & E(2^{An}+1; t-1; e) \dots E(2^{An}+1+C; t-1; e); \\
& E(0; t-1; e), E(1; t-1; e)) \qquad) \\
& \dots \\
\beta_C = \xi_C(\ & E(2^{An}+1; t-1; e) \dots E(2^{An}+1+C; t-1; e); \\
& E(0; t-1; e), E(1; t-1; e)) \qquad)
\end{aligned}$$

(a) If $\alpha_1 = 0$, $\alpha_2 = 0$ (at step t TM Ω moves head left),

(i) If $x = 2$ or $x = 3$ (requested symbol, which is being written to tape at step t by TM Ω) then

$$E(2; t; e) = \alpha_3,\ E(3; t; e) = \alpha_4$$

(ii) If $x < 2$ (some other symbol requested, case 1) then

$$E(x; t; e) = E(An + x - 2; t-1; e)$$

(iii) If $4 \le x < 2^{An}$ (some other symbol requested, case 2) then

$$E(x; t; e) = E(x-2; t-1; e)$$

(b) If $\alpha_1 = 0$, $\alpha_2 = 1$ (at step t TM Ω moves head right),

(i) If $x = 2^{An} - 2$ or $x = 2^{An} - 1$ (requested symbol, which is being written to tape at step t by TM Ω) then

$$E(2^{An} - 2; t; e) = \alpha_3,\ E(2^{An} - 1; t; e) = \alpha_4$$

(ii) If $x < 2^{An} - 2$ (some other symbol requested) then

$$E(x; t; e) = E(x + 2; t - 1; e)$$

(c) If $\alpha_1 = 1$ (TM Ω doesn't move head at step t),

(i) If $x = 0$ or $x = 1$ (requested symbol, which is being written to tape at step t by TM Ω) then

$$E(0; t; e) = \alpha_3,\ E(1; t; e) = \alpha_4$$

(ii) If $2 \leq x < 2^{An}$ (some other symbol requested) then

$$E(x; t; e) = E(x; t - 1; e)$$

(d) If $x = 2^{An}$ (stop bit requested) then

$$E(2^{An}; t; e) = \chi(\beta_1 \ldots \beta_C)$$

(e) If $2^{An} + 1 \leq x < 2^{An} + 1 + C$ (internal state bit requested) then

$$E(x; t; e) = \beta_{x-2^{An}}$$

Conditional structure "if x then $a = y$ else $a = z$" can be represented by logical connectives in the form of

$$a = (x \wedge y) \vee (\neg x \wedge z). \tag{3}$$

More complex conditionals with more than one condition, and nested conditionals, can be expressed as a set of boolean formulas (3), with aid of auxillary variables.

Clearly, complete procedure of calculation of E can be represented as set of boolean formulas with cumulative length $O(n)$ (when $n \to \infty$ and Ω fixed), with aid of additional variables. $\square$

3 Some auxiliry theorems of predicate logic

Theorem 3 *Let* $x_1 \dots x_n \in D$, $f : D \to \{0,1\}, z \in \{0,1\}$, K *to be any predicate, such that for every function* f*:*

$$\forall x_1 \dots x_n \, \exists z \, K\left[f(x_1) \dots f(x_n), z\right] \tag{4}$$

Then expression

$$\exists f(\cdot) \, \forall x_1 \dots x_{n+1} \, \forall z \;\; \left(K\left[f(x_1) \dots f(x_n), z\right] \Rightarrow (z = f(x_{n+1}))\right) \tag{5}$$

is equivivalent to

$$\exists f_1(\cdot) \dots f_n(\cdot) \, \forall x_1 \dots x_{n+1} \, \forall z$$

$$\bigwedge_{i=1}^{n} \left(K\left[f_1(x_1) \dots f_n(x_n), z\right] \Rightarrow (z = f_i(x_{n+1}))\right). \tag{6}$$

PROOF. First of all, function f is renamed to f_1. Then we introduce new functions — $f_2 = f_3 = \dots = f_n = f_1$:

$$(5) \;\; \Leftrightarrow \;\; \exists f_1(\cdot) \dots f_n(\cdot) \;\; \forall x_1 \dots x_n \, \forall z \, \forall y$$

$$\Big(f_2(y) = f_1(y) \;\wedge\; f_3(y) = f_1(y) \;\wedge\; \dots \;\wedge\; f_n(y) = f_1(y)\Big) \wedge$$

$$\wedge \Big(K\left[f_1(x_1) \dots f_1(x_n), z\right] \Rightarrow (z = f_1(x_{n+1}))\Big). \tag{7}$$

Now we

- use n times idempotency of "$\wedge$" operation, repeating implication from expression (5) n times;
- use equality between function f_1 and $f_i, i > 1$ and change functional sumbol names in implications;

so we get:

$$(7) \;\; \Leftrightarrow \;\; \exists f_1(\cdot) \dots f_n(\cdot) \;\; \forall x_1 \dots x_n \, \forall z \, \forall y$$

$$\Big(f_2(y) = f_1(y) \;\wedge\; f_3(y) = f_1(y) \;\wedge\; \dots \;\wedge\; f_n(y) = f_1(y)\Big) \wedge$$

$$\bigwedge_{i=1}^{n} \Big(K\left[f_1(x_1) \dots f_n(x_n), z\right] \Rightarrow (z = f_i(x_{n+1}))\Big). \tag{8}$$

Then we exclude the variable y using the equation:

$$\forall u \, \forall v \, \forall w \; P(v) \wedge Q(u,w) \;\; \Leftrightarrow \;\; \forall u \, \forall v \; P(v) \wedge Q(u,v); \tag{9}$$

in such way:

$$(8) \;\Leftrightarrow\; \exists f_1(\cdot)\dots f_n(\cdot)\;\forall x_1\dots x_n\,\forall z$$
$$\bigwedge_{i=2}^{n}\Big(f_i(x_{n+1}) = f_1(x_{n+1})\Big)\wedge$$
$$\wedge\bigwedge_{i=1}^{n}\Big(K\,[f_1(x_1)\dots f_n(x_n), z] \Rightarrow (z = f_i(x_{n+1}))\Big). \tag{10}$$

Accoring to condition (4), in every case there exists z such that premise of implication in second conjunction of (10) is true (it does not depend on i). Then $z = f_1(x_{n+1}) = \dots = f_n(x_{n+1})$ and first conjunction of (10) is always true, so it can be excluded from expression:

$$(10) \;\Leftrightarrow\; (6),$$

which required to be proved. $\square$

Theorem 4 *Let $\{x_i\}$, z, f and K to comply to conditions from theorem 3.*

Then expression (5) is equivalent to

$$\exists f(\cdot)\, f^\star(\cdot)\,\forall x_1\dots x_{n+1}\,\forall z$$
$$\Big(K\,[f(x_1)\dots f(x_n), z] \Rightarrow (z = f^\star(x_{n+1}))\Big)\wedge$$
$$\wedge\Big(K\,[f^\star(x_1)\dots f^\star(x_n), z] \Rightarrow (z = f(x_{n+1}))\Big). \tag{11}$$

PROOF. We introduce a new function $f^\star = f$. Then we

- add to (5) the condition $\forall y\; f^\star(y) = f(y)$;
- use idempotency of "$\wedge$" operation, repeation implication of formula (5) two times;
- use equation $f = f^\star$ and change functional symbols in implications;
- exclude the variable y, using the equation (9);

so we get:

$$(5) \;\Leftrightarrow\; \exists f(\cdot)f^\star(\cdot)\;\forall x_1\dots x_n\,\forall z\,\forall y\;\Big(f(x_{n+1}) = f^\star(x_{n+1})\Big)\wedge$$
$$\Big(K\,[f(x_1)\dots f(x_n), z] \Rightarrow (z = f^\star(x_{n+1}))\Big)\wedge$$
$$\wedge\Big(K\,[f^\star(x_1)\dots f^\star(x_n), z] \Rightarrow (z = f(x_{n+1}))\Big). \tag{12}$$

According to condition (4), always there exists z such that premises of all implications in (12) are true. Then $z = f(x_{n+1}) = f^\star(x_{n+1})$ and first conjunction of (12) is always true, so it can be excluded from the expression:

$$(12) \;\Leftrightarrow\; (11),$$

which required to be proved. $\square$

Theorem 5 *Let $\{x_i\}$, f and predicate K to comply conditions from theorem 3.*

Then, expression (5) is equivalent to

$$\exists f_1(\cdot)\dots f_n(\cdot), \exists f_1^\star(\cdot)\dots f_n^\star(\cdot)$$

$$\forall \left\{x_i^j, x_i^{j\star}\right\}\Big|_{i=1\dots n, j=1\dots n}, \forall \left\{z_j, z_j^\star\right\}\Big|_{j=1\dots n}$$

$$\left(\bigwedge\nolimits_{i=1}^n \left(K\left[f_1(x_1^i)\dots f_n(x_n^i), z_i\right] \Rightarrow (z_i = f_i^\star(x_{n+1}^i))\right)\right) \wedge$$

$$\wedge \left(\bigwedge\nolimits_{i=1}^n \left(K\left[f_1^{i\star}(x_1)\dots f_n^{i\star}(x_n), z_i^\star\right] \Rightarrow (z_i^\star = f_i(x_{n+1}^i))\right)\right). \tag{13}$$

PROOF. We use theorem 4 and 3:

$$(5) \;\Leftrightarrow\; \exists f_1(\cdot)\dots f_n(\cdot)\, \exists f_1^\star(\cdot)\dots f_n^\star(\cdot)\, \forall x_1\dots x_{n+1}\, \forall z$$

$$\left(\bigwedge\nolimits_{i=1}^n \left(K\left[f_1(x_1)\dots f_n(x_n), z\right] \Rightarrow (z = f_i^\star(x_{n+1}))\right)\right) \wedge$$

$$\wedge \left(\bigwedge\nolimits_{i=1}^n \left(K\left[f_1^\star(x_1)\dots f_n^\star(x_n), z\right] \Rightarrow (z = f_i(x_{n+1}))\right)\right). \tag{14}$$

By formula $\forall u\, P(u) \wedge Q(u) \;\Leftrightarrow\; \forall u\, P(u) \;\wedge\; \forall v\, Q(v)$:

$$(14) \;\Leftrightarrow\; \exists f_1(\cdot)\dots f_n(\cdot)\, \exists f_1^\star(\cdot)\dots f_n^\star(\cdot)$$

$$\left(\forall x_1\dots x_{n+1}\, \forall z \bigwedge\nolimits_{i=1}^n \left(K\left[f_1(x_1)\dots f_n(x_n), z\right] \Rightarrow (z = f_i^\star(x_{n+1}))\right)\right) \wedge$$

$$\wedge \left(\forall x_1\dots x_{n+1}\, \forall z \bigwedge\nolimits_{i=1}^n \left(K\left[f_1^\star(x_1)\dots f_n^\star(x_n), z\right] \Rightarrow (z = f_i(x_{n+1}))\right)\right). \tag{15}$$

We rename the variables:

$$(15) \;\Leftrightarrow\; \exists f_1(\cdot)\dots f_n(\cdot)\, \exists f_1^\star(\cdot)\dots f_n^\star(\cdot)$$

$$\Big(\forall x_1 \dots x_n, x^\star_{n+1}\, \forall z^\star \bigwedge\nolimits_{i=1}^{n} \big(K\,[f_1(x_1)\dots f_n(x_n), z^\star] \Rightarrow (z^\star = f^\star_i(x^\star_{n+1}))\big)\Big) \wedge$$

$$\wedge \Big(\forall x^\star_1 \dots x^\star_n, x_{n+1}\, \forall z \bigwedge\nolimits_{i=1}^{n} (K\,[f^\star_1(x^\star_1)\dots f^\star_n(x^\star_n), z] \Rightarrow (z = f_i(x_{n+1})))\Big). \quad (16)$$

By formula $\forall u\, P(u) \;\wedge\; \forall v\, Q(v) \;\Leftrightarrow\; \forall u\, \forall v\; P(u) \wedge Q(v)$:

$$(16) \;\Leftrightarrow\; \exists f_1(\cdot)\dots f_n(\cdot)\, \exists f^\star_1(\cdot)\dots f^\star_n(\cdot)\, \forall x_1 \dots x_{n+1}, x^\star_1 \dots x^\star_{n+1}\, \forall z, z^\star$$

$$\Big(\bigwedge\nolimits_{i=1}^{n} \big(K\,[f_1(x_1)\dots f_n(x_n), z^\star] \Rightarrow (z^\star = f^\star_i(x^\star_{n+1}))\big)\Big) \wedge$$

$$\wedge \Big(\bigwedge\nolimits_{i=1}^{n} (K\,[f^\star_1(x^\star_1)\dots f^\star_n(x^\star_n), z] \Rightarrow (z = f_i(x_{n+1})))\Big). \quad (17)$$

Finally, by formula $\forall u\, P(u) \;\Leftrightarrow\; \forall u_1 \dots u_n\, P(u_1)\dots P(u_n)$:

$$(17) \;\Leftrightarrow\; (13).$$

□

4 AEA statements. AEASAT problem. Theorem of univeral reducibility to AEASAT problem

4.1 AEA statements and AEASAT problem

Definition 6 *AEA statement is a formula of predicate logic in the form of:*

$$\forall\chi_1\cdots\forall\chi_n\exists\chi_{n+1}\cdots\exists\chi_{n+m}\forall\chi_{n+m+1}\cdots\exists\chi_{n+m+k}\ K\left(\chi_1,\chi_2,\ldots\chi_{n+m+k}\right),$$

where K is a boolean formula which has no entries of any variables except $\chi_1,\chi_2,\cdots\chi_{n+m+k}$.

Table 1
Examples for definition 6

Expression	PLF[1]	AEA statement	Comment
0	yes	yes	$n=m=k=0$
$\forall\chi_1\exists\chi_2\forall\chi_3 P\left(\chi_1,\chi_2,\chi_3\right)$	yes	yes	$n=m=k=1$
χ_1	yes	no	has free entry of χ_1
$\forall\chi_1\left(\forall\chi_2\left(\chi_1\rightarrow\chi_2\right)\vee\chi_1\right)$	yes	no	2nd quntifier inside round parenthesis
$\forall\chi_1 a(\chi_1)\vee\forall\chi_1 b(\chi_1)$	yes	no	K is not distinguished
$\forall\chi_1\forall\chi_2\left(a(\chi_1)\vee b(\chi_2)\right)$	yes	yes	$K\left(\chi_1,\chi_2\right)=a(\chi_1)\vee b(\chi_2)$
$\exists\chi_1 0$	yes	yes	$K\left(\chi_1\right)=0$
$\exists\chi_1\exists\chi_1\left(\chi_1\right)$	yes	no	repeat of quatified variable
$\forall\chi_1\exists\chi_2 a(\chi_2)\ \vee\ \forall\chi_3\exists\chi_4 b(\chi_4)$	yes	no	K is not distinguished
$\forall\chi_1\exists\chi_2\exists\chi_3(a(\chi_2)\vee b(\chi_3))$	yes	yes	$K=a(\chi_2)\vee b(\chi_3)$
$\forall\chi_1\forall\chi_2 a(\chi_1,\chi_2)$	yes	yes	$n=m=0,\ k=2$

[1]predicate logic formula.

Definition 7 *AEASAT problem is a problem of decision is given AEA stamement true or false, provided by its representation.*

Theorem 8 *If $\mathcal{NP}=$ co-$\mathcal{NP}$ then AEASAT problem belongs to $\mathcal{NP}$ class (and therefore, certification algorithm time is limited by a fixed polynomial over input length).*

PROOF. Let's suppose $\mathcal{NP} = \text{co-}\mathcal{NP}$. This means, for every problem ω the following implication is true:

$$\exists \delta \in \mathcal{P}\, \exists p(\cdot)\ \forall i \in I \left(\omega(i) = 1 \Leftrightarrow \exists c \in \{0,1\}^\star\ (|c| < p(|i|) \wedge \delta\left(\langle i, c\rangle\right) = 1)\right) \Rightarrow$$

$$\Rightarrow \exists \epsilon \in \mathcal{P}\, \exists q(\cdot)\ \forall i \in I \left(\omega(i) = 0 \Leftrightarrow \exists d \in \{0,1\}^\star\ (|d| < q(|i|) \wedge \epsilon\left(\langle i, d\rangle\right) = 1)\right).$$

Consider the problem θ_1, which input is a set of values of boolean variables $\chi_1 \dots \chi_{n+m}$ and representation of boolean formula K, dependant on boolean variables $\chi_1 \dots \chi_{n+m+k}$; and output is "accepted" iff formula $K(\chi_1 \dots \chi_{n+m})$ is a tautology.

Clearly

$$\theta_1(\chi_1 \dots \chi_{n+m}, K) = 0$$

iff

$$\exists \chi_{n+m+1} \dots \chi_{n+m+k}\ (\neg K(\chi_1 \dots \chi_{n+m+k}) = 1).$$

Consequently $\theta_1 \in \text{co-}\mathcal{NP}$ by definition.

But by supposition this implies $\theta_1 \in \mathcal{NP}$. Then there exist a certifcation problem $\delta_1 \in \mathcal{P}$ such that

$$\theta_1(\chi_1 \dots \chi_{n+m}, K) = 1$$

iff

$$\exists \chi_{n+m+1} \dots \chi_{n+m+k}\ (\delta_1(K, \chi_1 \dots \chi_{n+m+k}) = 1).$$

Consider another problem θ_2, which input is a set of values of boolean variables $\chi_1 \dots \chi_n$ and representation of boolean formula K, dependant on boolean variables $\chi_1 \dots \chi_{n+m+k}$; and output is "accepted" iff

$$\exists \chi_{n+1} \dots \chi_{n+m}\, \forall \chi_{n+m+1} \dots \chi_{n+m+k}\ K(\chi_1 \dots \chi_n,\ \chi_{n+1} \dots \chi_{n+m+k}) = 1,$$

that is,

$$\exists \chi_{n+1} \dots \chi_{n+m}\, \exists \chi_{n+m+1} \dots \chi_{n+m+k}\ \delta_1(K, \chi_1 \dots \chi_{n+m+k}).$$

Clearly $\theta_2 \in \mathcal{NP}$ by definition, since $\delta_1 \in \mathcal{P}$. According to supposition, this implies $\theta_2 \in \text{co-}\mathcal{NP}$. Then there exist such cerfitication problem $\epsilon_2 \in \mathcal{P}$ such that

$$\theta_2(\chi_1 \dots \chi_{n+m}, K) = 0$$

iff

$$\exists \chi_{n+1} \dots \chi_{n+m+k}\ \epsilon_2(K, \chi_1 \dots \chi_{n+m+k}) = 1.$$

In other words,

$$\theta_2(\chi_1 \dots \chi_{n+m}, K) = 1$$

iff

$$\forall \chi_{n+1} \dots \chi_{n+m+k} \neg \epsilon_2(K, \chi_1 \dots \chi_{n+m+k}) = 1.$$

AEASAT problem input is a set of values of boolean variables $\chi_1 \dots \chi_n$ and representation of boolean formula K dependant on boolean variables $\chi_1 \dots \chi_{n+m+k}$; and output is "accepted" iff

$$\forall \chi_1 \dots \chi_n \, \exists \chi_{n+1} \dots \chi_{n+m} \, \forall \chi_{n+m+1} \dots \chi_{n+m+k} \, K(\chi_1 \dots \chi_{n+m+k}) = 1,$$

that is,

$$\forall \chi_1 \dots \chi_n \, \forall \chi_{n+1} \dots \chi_{n+m+k} \neg \epsilon_2(K, \chi_1 \dots \chi_{n+m+k}) = 1.$$

In other words,

$$\mathrm{AEASAT}(K) = 0$$

iff

$$\exists \chi_1 \dots \chi_{n+m+k} \, \epsilon_2(K, \chi_1 \dots \chi_{n+m+k}) = 1.$$

I.e. AEASAT $\in$ co-$\mathcal{NP}$ by definition of class co-$\mathcal{NP}$.

Then, according to supposition, AEASAT $\in \mathcal{NP}$. $\square$

Theorem 9 *If $\mathcal{NP} \subseteq \mathcal{P}$ or co-$\mathcal{NP} \subseteq \mathcal{P}$, then AEASAT problem is computable within polynomial time.*

PROOF. If $\mathcal{NP} \subseteq \mathcal{P}$ or co-$\mathcal{NP} \subseteq \mathcal{P}$, then $\mathcal{NP} =$ co-$\mathcal{NP}$ and by theorem 8 AEASAT $\in \mathcal{NP}$. According to supposition this implies AEASAT $\in \mathcal{P}$. $\square$

4.2 Universal reducibility of $\mathcal{D}TIME(\exp(O(\cdot)))$ to AEASAT problem

Theorem 10 *If problem AEASAT $\in \mathcal{P}$, then every decision problem from $\mathcal{D}TIME(\exp(O(\cdot))$ class is computable with polynomial time of fixed degree.*

PROOF. Let Ω be any TM, which solves some decision problem within 2^{Bn} time, where n is a length of its input string. Let's denote $A \stackrel{def}{=} 2B$. As before, without loss of generality we consider that Ω scans computation result when it halts.

We start reasoning from set of recursive formulas for computation of function $E(x; t; e)$, which is equal to xth bit of shrink at tth step of computation of $\Omega(e)$, taken from theorem 2 proof.

$E(x;t;e)$ is some function dependant on arguments:

$$x;t;e;E(2^{An};t-1;e),E(0;t-1;e),E(1;t-1;e),$$

$$E(2^{An}+1;t-1;e)\dots E(2^{An}+C;t-1;e),$$
$$E((2^{An}+x-2)\mathrm{mod}(2^{An});t-1;e),E(x+2;t-1;e),E(x;t-1;e).$$

Now we need to show that chain of formulas to calculate $E(x;t;e)$ is representable as AEA statement of effective (polynomial) length.

Each auxillary variable which is used to compute E, may belong to one of four types:

(1) variables which arrive when conditionals are replaced with constructs of form (3), there is a fixed number of them (we name it D);
(2) additional arguments (values of E funtion on other inputs), $C+6$ variables;
(3) bits of output string of ξ function (internal variables), $C+4$ variables;
(4) computation result for function χ, one variable;

in all $2C+D+11$. Considering this,

$$(2) \Leftrightarrow \exists E:\{0,1\}^{(A+B+2)n+C+1}\to\{0,1\}^{(A+B+2)n+C+1}$$

$$\forall e_1\dots e_n\ \forall x_1\dots x_{An+C+1}\ \forall t_1\dots t_{Bn}\ \forall a_1\dots a_{2C+D+11}$$
$$\Big\{\bigwedge\nolimits_{i=1}^{D}\Big(a_i=(a_j\wedge a_m)\vee(\neg a_j\wedge a_k),\ \ j,k,m\in\{1,2\dots D\}\Big)\ \wedge$$
$$\wedge\ \bigwedge\nolimits_{i=1}^{C}\Big(a_{D+i}=E(2^{An}+i;t-1;e)\Big)\ \wedge$$
$$\wedge\, a_{C+D+1}=E(0;t-1;e)\ \wedge\ a_{C+D+2}=E(1;t-1;e)\wedge$$
$$\wedge\, a_{C+D+3}=E(2^{An};t-1;e)\ \wedge\ a_{C+D+4}=E((x-2)\mathrm{mod}(2^{An});t-1;e)\wedge$$
$$\wedge\, a_{C+D+5}=E(x+2;t-1;e)\ \wedge\ a_{C+D+6}=E(x;t-1;e)\wedge$$
$$\wedge\ \bigwedge\nolimits_{i=1}^{C+4}\Big(a_{C+D+6+i}=\xi_i(a_{D+1},a_{D+2},\dots a_{C+D+2})\Big)\ \wedge$$

$$\wedge\, a_{2C+D+11}=\chi(a_{C+D+7},a_{C+D+8}\dots a_{2C+D+10})\ \Rightarrow\ E(x;t;e)=a_1\Big\}. \quad (18)$$

Let's designate for the sake of convenience

$$p\overset{def}{=}An+C+1,\ \ q\overset{def}{=}Bn,\ \ r\overset{def}{=}2C+D+11,\ \ s\overset{def}{=}C+6.$$

Core of expression (18) (i.e. unquantified internal part which will be denoted as $\mathrm{core}_{(18)}$ below), contains s entries of function E, not counting the last one, which includes function being determined. The idea is that function E should

have fixed collection of arguments, and had entries within the formula only with this collection of arguments. To archieve this, we introduce s equivalent functions $E^1 \dots E^s$, each of them has own collection of arguments$x_1^i \dots x_p^i$ and $t_1^i \dots t_q^i$, and also auxillary variables $a_1^i \dots a_r^i$; $i = 1 \dots s$.

Generality quantifiers of expression (18) are duplicated s times (except those which quantify variables $e_1 \dots e_n$). Core will be duplicated s times, components of new core will be named sub-cores.

Inner part of each subcore changes in the following manner: all variables x_k, t_k, a_k get upper index i (number of current sub-core), all constructs in form of

$$a_k^i = E(\alpha; \beta; e),$$

except the last one, are replaced by

$$a_k^i = E^j(\alpha; \beta; e),$$

where j iterates all numbers from 1 to s, and this is repeated for all sub-cores. Each sub-core contains exactly s entries of functional symbol E in the left side of implication.

Right side of implication

$$a_1^i = E(x; t; e)$$

contains the function being determined, therefore, it will change in other way: it will be replaced by

$$a_1^i = E^i(x; t; e),$$

where i is a number of current sub-core.

Thus,

$$\begin{gathered}
(18) \Leftrightarrow \forall e_1 \dots e_n \ \forall \left\{x_j^i\right\}\Big|_{i=1\dots s,\, j=1\dots p} \ \forall \left\{t_j^i\right\}\Big|_{i=1\dots s,\, j=1\dots q} \\
\exists \left\{E^i(x_1^i \dots x_p^i; t_1^i \dots t_q^i; e_1 \dots e_n)\right\}\Big|_{i=1\dots s} \ \forall \left\{a_j^i\right\}\Big|_{\substack{i=1\dots s \\ j=1\dots r}} \\
\bigwedge_{i=1}^{s} \mathrm{core}_{(18)}\Big(e_1 \dots e_n,\ x_1^i \dots x_p^i,\ t_1^i \dots t_q^i,\ a_1^i \dots a_r^i, \\
E^1(2^{An}+1; t^i-1; e) \dots E^{s-6}(2^{An}+C; t^i-1; e), \\
E^{s-5}(0; t^i-1; e), E^{s-4}(1; t^i-1; e), E^{s-3}(2^{An}; t^i-1; e), \\
E^{s-2}((x^i-2)\mathrm{mod}(2^{An}); t^i-1; e), E^{s-1}(x^i+2; t^i-1; e), E^s(x^i; t^i-1; e), \\
E^i(x; t; e)\Big).
\end{gathered} \tag{19}$$

Each subcore of expression (19) contains only one function E^i, which entries within expression has multiple set of arguments. To avoid this, let's introduce additional excess: all functions and variables are be duplicated, copies will be

denoted with an asterisk ($\star$) at superscript. Also cross-"calcation" will is used, i.e. no-asterisk function and variables determines valus of asterisk function, and vice versa:

$$(19) \Leftrightarrow \exists \Big\{ E^i(x^i_1 \dots x^i_p; t^i_1 \dots t^i_q; e_1 \dots e_n),$$

$$E^{i\star}(x^{i\star}_1 \dots x^{i\star}_p; t^{i\star}_1 \dots t^{i\star}_q; e_1 \dots e_n) \Big\}\Big|_{i=1\dots s}$$

$$\forall e_1 \dots e_n \, \forall \left\{ x^i_j, x^{i\star}_j \right\}\Big|_{i=1\dots s,\, j=1\dots p} \forall \left\{ t^i_j, t^{i\star}_j \right\}\Big|_{i=1\dots s,\, j=1\dots q} \forall \left\{ a^i_j, a^{i\star}_j \right\}\Big|_{i=1\dots s,\, j=1\dots r}$$

$$\bigwedge\nolimits_{i=1}^{s} \mathrm{core}_{(18)} \Big(e_1 \dots e_n,\ x^i_1 \dots x^i_p,\ t^i_1 \dots t^i_q,\ a^i_1 \dots a^i_r,$$

$$E^1(2^{An}+1; t^i-1; e) \dots E^{s-6}(2^{An}+C; t^i-1; e),$$

$$E^{s-5}(0; t^i-1; e), E^{s-4}(1; t^i-1; e), E^{s-3}(2^{An}; t^i-1; e),$$

$$E^{s-2}((x^i-2)\mathrm{mod}(2^{An}); t^i-1; e),$$

$$E^{s-1}(x^i+2; t^i-1; e), E^s(x^i; t^i-1; e),\ \ E^{i\star}(x^\star; t^\star; e) \Big) \wedge$$

$$\wedge \bigwedge\nolimits_{i=1}^{s} \mathrm{core}_{(18)} \Big(e_1 \dots e_n,\ x^{i\star}_1 \dots x^{i\star}_p,\ t^{i\star}_1 \dots t^{i\star}_q,\ a^{i\star}_1 \dots a^{i\star}_r,$$

$$E^{1\star}(2^{An}+1; t^{i\star}-1; e) \dots E^{s-6\star}(2^{An}+C; t^{i\star}-1; e),$$

$$E^{s-5\star}(0; t^{i\star}-1; e), E^{s-4\star}(1; t^{i\star}-1; e), E^{s-3\star}(2^{An}; t^{i\star}-1; e),$$

$$E^{s-2\star}((x^{i\star}-2)\mathrm{mod}(2^{An}); t^{i\star}-1; e),$$

$$E^{s-1\star}(x^{i\star}+2; t^{i\star}-1; e), E^{s\star}(x^{i\star}; t^{i\star}-1; e),\ \ E^i(x; t; e) \Big). \tag{20}$$

Correctness of conversion (18) $\Leftrightarrow$ 20 is proved in detail in theorem 5.

Two conjunctions in expression (20) can be united to one, and its internal part can be named sub-core of (20). It has the following property: for each functional symbol all its entries to sub-core has fixed collection of arguments.

Further, we use the rule

$$P(f(a)) \Leftrightarrow \forall x \Big((x = a) \Rightarrow P(f(x)) \Big) :$$

(for brevity instead of $E^i(x^i; t^i; e)$ ($E^{i\star}(x^{i\star}; t^{i\star}; e)$) we write E^i ($E^{i\star}$)):

$$(20) \Leftrightarrow \exists \Big\{ E^i(x^i_1 \dots x^i_p; t^i_1 \dots t^i_q; e_1 \dots e_n),$$

$$E^{i\star}(x^{i\star}_1 \dots x^{i\star}_p; t^{i\star}_1 \dots t^{i\star}_q; e_1 \dots e_n) \Big\}\Big|_{i=1\dots s}$$

$$\forall e_1 \dots e_n \, \forall \left\{ x^i_j, x^{i\star}_j \right\}\Big|_{i=1\dots s,\, j=1\dots p} \forall \left\{ t^i_j, t^{i\star}_j \right\}\Big|_{i=1\dots s,\, j=1\dots q} \forall \left\{ a^i_j, a^{i\star}_j \right\}\Big|_{i=1\dots s,\, j=1\dots r}$$

$$\bigwedge\nolimits_{i=1}^{s} \Big\{ \bigwedge\nolimits_{j=1}^{s-6} (x^j = 2^{An} + j) \wedge (t^j = t^i - 1) \wedge$$

$$\wedge\,(x^{s-5}=0)\wedge(t^{s-5}=t^i-1)\wedge(x^{s-4}=1)\wedge(t^{s-4}=t^i-1)\wedge$$
$$\wedge\,(x^{s-3}=2^{An})\wedge(t^{s-3}=t^i-1)\wedge$$
$$\wedge(x^{s-2}=(x^i-2)\mathrm{mod}(2^{An}))\wedge(t^{s-2}=t^i-1)\wedge$$
$$\wedge\,(x^{s-1}=x^i+2)\wedge(t^{s-1}=t^i-1)\wedge(x^s=x^i)\wedge(t^s=t^i-1)\Rightarrow$$
$$\Rightarrow \mathrm{core}_{(18)}\Big(\,e_1\dots e_n,\ x_1^i\dots x_p^i,\ t_1^i\dots t_q^i, a_1^i\dots a_r^i, E^1\dots E^s, E^{i\star}\Big)\ \wedge$$
$$\wedge\ \bigwedge_{j=1}^{s-6}(x^{j\star}=2^{An}+j)\wedge(t^{j\star}=t^{i\star}-1)\wedge$$
$$\wedge\,(x^{s-5\star}=0)\wedge(t^{s-5\star}=t^{i\star}-1)\wedge(x^{s-4\star}=1)\wedge(t^{s-4\star}=t^{i\star}-1)\wedge$$
$$\wedge\,(x^{s-3\star}=2^{An})\wedge(t^{s-3\star}=t^{i\star}-1)\wedge$$
$$\wedge(x^{s-2\star}=(x^{i\star}-2)\mathrm{mod}(2^{An}))\wedge(t^{s-2\star}=t^{i\star}-1)\wedge$$
$$\wedge\,(x^{s-1\star}=x^{i\star}+2))\wedge(t^{s-1\star}=t^{i\star}-1)\wedge(x^{s\star}=x^{i\star})\wedge(t^{s\star}=t^{i\star}-1)\Rightarrow$$
$$\Rightarrow \mathrm{core}_{(18)}\Big(\,e_1\dots e_n,\ x_1^{i\star}\dots x_p^{i\star},\ t_1^{i\star}\dots t_q^{i\star}, a_1^{i\star}\dots a_r^{i\star}, E^{1\star}\dots E^{s\star}, E^i\Big)\Big\}\quad(21)$$

Core of expression (21) will be denoted below as $\mathrm{core}_{(21)}$.

Now we need to find equivalent conversion of (21) to AEA statement. Note that condition of existence of function cannot be just replaced with existential quantifier, this way:

$$\forall e_1\dots e_n\ \forall\left\{x_j^i, x_j^{i\star}\right\}\Big|_{i=1\dots s,\,j=1\dots p}\ \forall\left\{t_j^i, t_j^{i\star}\right\}\Big|_{i=1\dots s,\,j=1\dots q}$$
$$\exists\left\{E^i, E^{i\star}\right\}\Big|_{i=1\dots s}\ \forall\left\{a_j^i, a_j^{i\star}\right\}_{i=1\dots s,\,j=1\dots r}$$
$$\mathrm{core}_{(21)}\Big(e_1\dots e_n, \left\{x_j^i, x_j^{i\star}\right\}, \left\{t_j^i, t_j^{i\star}\right\}, \left\{a_j^i, a_j^{i\star}\right\}, \left\{E^i, E^{i\star}\right\}\Big)$$

because functional symbol E^i ($E^{i\star}$) must be dependant *only* on variables $x_1^i\dots x_p^i; t_1^i\dots t_p^i$ ($x_1^{i\star}\dots x_p^{i\star}; t_1^{i\star}\dots t_p^{i\star}$), not on full set of variables $\left\{x_j^i, x_j^{i\star}\right\}\Big|_{i=1\dots s,\,j=1\dots p}\left\{t_j^i, t_j^{i\star}\right\}\Big|_{i=1\dots s,\,j=1\dots q}$.

Thus last conversion is made by rule

$$\exists f(\cdot)\,\forall x\forall y\,P(x,y,f(x))\Leftrightarrow$$
$$\Leftrightarrow\forall x\forall y\,\exists f\,\forall u\forall v\Big[\Big((y=u)\wedge f\Big)\Rightarrow\Big((y=v)\wedge f\Big)\Big]\wedge$$
$$\wedge\Big[\Big((y=u)\wedge\neg f\Big)\Rightarrow\Big((y=v)\wedge\neg f\Big)\Big]\wedge P(x,y,f),$$

so:

$$(21)\Leftrightarrow\forall e_1\dots e_n\ \forall\left\{x_j^i, x_j^{i\star}\right\}\Big|_{i=1\dots s,\,j=1\dots p}\ \forall\left\{t_j^i, t_j^{i\star}\right\}\Big|_{i=1\dots s,\,j=1\dots q}$$
$$\exists\left\{E^i, E^{i\star}\right\}\Big|_{i=1\dots s}\ \forall\left\{a_j^i, a_j^{i\star}\right\}_{i=1\dots s,\,j=1\dots r}$$

$$\forall \left\{u_j^i, u_j^{i\star}, v_j^i, v_j^{i\star}\right\}_{i=1\ldots s,\, j=1\ldots p} \quad \forall \left\{g_j^i, g_j^{i\star}, h_j^i, h_j^{i\star}\right\}_{i=1\ldots s,\, j=1\ldots q}$$

$$\bigwedge_{i=1}^{s}\bigwedge_{j=1}^{p}\bigwedge_{k=1}^{s}\bigwedge_{\substack{m=1\\(k,m)\neq(i,j)}}^{p}\Big((x_j^i = u_m^k)\wedge E^i \Rightarrow (x_j^i = v_m^k)\wedge E^i\Big)\wedge$$

$$\wedge\Big((x_j^i = u_m^k)\wedge \neg E^i \Rightarrow (x_j^i = v_m^k)\wedge \neg E^i\Big)\wedge\Big((x_j^{i\star} = u_m^{k\star})\wedge E^{i\star} \Rightarrow$$

$$\Rightarrow (x_j^{i\star} = v_m^{k\star})\wedge E^{i\star}\Big)\wedge\Big((x_j^{i\star} = u_m^{k\star})\wedge \neg E^{i\star} \Rightarrow (x_j^{i\star} = v_m^{k\star})\wedge \neg E^{i\star}\Big)\wedge$$

$$\wedge\bigwedge_{i=1}^{s}\bigwedge_{j=1}^{q}\bigwedge_{k=1}^{s}\bigwedge_{\substack{m=1\\(k,m)\neq(i,j)}}^{q}\Big((t_j^i = g_m^k)\wedge E^i \Rightarrow (t_j^i = h_m^k)\wedge E^i\Big)\wedge$$

$$\wedge\Big((t_j^i = g_m^k)\wedge \neg E^i \Rightarrow (t_j^i = h_m^k)\wedge \neg E^i\Big)\wedge\Big((t_j^{i\star} = g_m^{k\star})\wedge E^{i\star} \Rightarrow$$

$$\Rightarrow (t_j^{i\star} = h_m^{k\star})\wedge E^{i\star}\Big)\wedge\Big((t_j^{i\star} = g_m^{k\star})\wedge \neg E^{i\star} \Rightarrow (t_j^{i\star} = h_m^{k\star})\wedge \neg E^{i\star}\Big)\wedge$$

$$\mathrm{core}_{(21)}\Big(e_1\ldots e_n, \left\{x_j^i, x_j^{i\star}\right\}, \left\{t_j^i, t_j^{i\star}\right\}, \left\{a_j^i, a_j^{i\star}\right\}, \left\{E^i, E^{i\star}\right\}\Big) \tag{22}$$

Consequently, we got a chain of equivalent conversions:

$$(1) \Leftrightarrow (2) \Leftrightarrow (18) \Leftrightarrow (19) \Leftrightarrow (20) \Leftrightarrow (21) \Leftrightarrow (22);$$

then (1) $\Leftrightarrow$ (22).

Clearly, AEA statement (22) has representation length $O(n^W)$, $W = \mathrm{const}$.

The next algorithm reduces Ω to AEASAT.

(1) Add *additional conditions* to expression (22):
- condition of equality to 1 of the lowest bit of result at step $t = 2^{Bn}$.
- conditions that determine concrete values of variables $e_1 \ldots e_n$.

Acquired formula will is denoted as H.

(2) Calculate AEASAT(H). Output of this calculation ("accepted" or "declined") is an output of this algorithm.

H is true, i.e. AEASAT(H) = 1 iff (1), extended with additional conditions, is true. On other words, H is true, iff TM Ω accepts input e.

Let's suppose that AEASAT is computable within polynomial time $O(n^C)$. Then step 2 requires $O(n^{CW})$ time. Time of such degree is more than enough to perform step one.

Hence every problem from $\mathcal{D}\mathrm{TIME}\left(2^{Bn}\right)$ is computable within polynomial time, which degree is invariant to chosen problem. $\square$

5 The results

The next theorem is main result of this work.

Theorem 11 *There are problems in classes $\mathcal{NP}$, co-$\mathcal{NP}$ which are not computable within polynomial time. SAT problem is an example of such problem from $\mathcal{NP}$ class.*

PROOF.

Suppose that it is not true. Then for each decision problem from class $\mathcal{NP}$, there exists an algorithm which solves it within polynomial time.

According to theorem 9 this implies AEASAT $\in \mathcal{P}$.

Then by theorem 10, every problem from $\mathcal{D}\mathrm{TIME}\,(\exp(O(\cdot)))$, is computable within time $O(n^L)$, $L = \mathrm{const}$. Here L not determined by nature of the problem and it is an universal contant.

Calculability of such problem within $O(n^L)$ time implies that all problems from class $\mathcal{D}\mathrm{TIME}\left(O(n^{L+1})\right)$ are in class $\mathcal{D}\mathrm{TIME}\left(O(n^L)\right)$, that is not possible according to time hierarchy theorem (see [2]).

Consequenty, supposition was wrong and $\mathcal{NP} \nsubseteq \mathcal{P}$.

Similar argumentation is true for co-$\mathcal{NP}$ class also.

Because every problem from $\mathcal{NP}$ reduces to SAT (see [1]), supposition of its computability within polynomial time conflicts with result $\mathcal{NP} \nsubseteq \mathcal{P}$. $\square$

References

[1] S. C. Cook, *The complexity of theorem-proving procedures*, Third ACM Symposium on Theory of Computing (1971), ACM, New York, pp. 151-158.

[2] J. Hartmanis and R. E. Stearns. *On the computational complexity of algorithms.* Transactions of the American Mathematical Society (1965), 117(5), pp. 285-306.

[3] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations (R. E. Miller, ed.), Plenum Press, New York, (1972), pp. 85-104.

Fig. 1. Single step of turing machine as a shrink function