

Dlaczego problem $P \stackrel{?}{=} NP$ jest tak trudny?

Wojciech CZERWIŃSKI

24 maja 2000 roku Instytut Matematyczny Claya ogłosił listę siedmiu Problemów Milenijnych, czyli zagadnień, które zostały uznane za najważniejsze otwarte problemy matematyczne opierające się rozwiązaniom od lat. Wśród nich był jeden problem zaliczany do informatyki teoretycznej, o którym wielu Czytelników zapewne słyszało. Chodzi oczywiście o tytułowy problem: „Czy $P = NP$ ”? Jest on powszechnie uznawany za najważniejsze pytanie informatyki teoretycznej.

Problem nazywamy *decyzyjnym*, jeśli odpowiedź to „tak” lub „nie” (a nie np. liczba). Klasa P (od *Polynomial time*) zawiera te problemy, które dadzą się rozwiązać w czasie wielomianowym. Definicja klasy NP (od *Nondeterministic Polynomial time*) jest nieco bardziej złożona. Powiemy, że problem L należy do NP , o ile dla każdego możliwego wejścia w , na które odpowiedź jest „tak”, istnieje *świadek* tej pozytywnej odpowiedzi v , co najwyżej wielomianowo większy niż w , taki, że gdy dostaniemy parę (w, v) , to możemy w czasie wielomianowym potwierdzić, że istotnie odpowiedź na w to „tak”. Intuicyjnie więc $P \stackrel{?}{=} NP$ pyta mniej więcej, czy każdy problem, dla którego można sprawdzić poprawność odpowiedzi szybko (w czasie wielomianowym), można również rozwiązać szybko (w czasie wielomianowym). Już w latach 50. problem ten pojawiał się w dyskusjach, ale precyzyjnie zdefiniował go w 1971 roku Stephen Cook. Od tej pory jest otwarty, mimo że naprawdę wiele osób próbowało go rozwiązać.

Dlaczego jednak uważamy ten problem za aż tak ważny i ciekawy? Przecież jest wiele starych problemów, których nikt jeszcze nie rozwiązał. Można na to pytanie odpowiedzieć na kilka sposobów. Po pierwsze, jest bardzo wiele praktycznych problemów, o których wiemy, że należą do klasy NP , jednak nie znamy dla nich żadnego algorytmu wielomianowego. Często rozważanym przykładem jest tzw. *problem komiwojażera*, w którym mamy dany graf o n wierzchołkach (reprezentujących miasta), a na każdej krawędzi liczbę, która oznacza, ile czasu potrzebuje komiwojażer, aby przejechać między tymi właśnie miastami. Pytanie brzmi, czy może on objechać wszystkie miasta i wrócić do domu w zadanym czasie (lub szybciej). Łatwo zauważyć, że istotnie należy on do NP , świadkiem jest tutaj samo rozwiązanie, czyli cykl o wystarczająco małym czasie przejazdu. Innym przykładem, ważnym w dalszej części artykułu, jest problem *spełnialności formuł* (znany jako *SAT* od angielskiego *satisfiability* – spełnialność). Rozważamy w tym przypadku formuły logiczne zawierające n zmiennych: x_1, \dots, x_n . Możemy założyć, że formuła jest zawsze koniunkcją klauzul C_i , czyli postaci $C_1 \wedge \dots \wedge C_k$. Każda klauzula natomiast jest alternatywą zmiennych x_i lub ich negacji $\neg x_i$. Przykładowa formuła wygląda więc tak: $(\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3 \vee x_4)$. Pytanie brzmi, czy istnieją takie wartości $x_1, \dots, x_n \in \{0, 1\}$, dla których formuła jest prawdziwa. W naszym przykładzie istnieją: są to, na przykład, $x_1 = x_3 = 1$ oraz $x_2 = x_4 = 0$. *SAT* należy do NP , bo świadkiem spełnialności jest tu po prostu takie wartościowanie zmiennych, które spełnia formułę. A więc, tak jak i wcześniej, świadkiem jest obiekt, o którego istnieniu pytamy. Jest to częste zjawisko, ale wcale nie zawsze świadek musi być takiej postaci. Wiele praktycznych problemów da się przepisać jako problem spełnialności formuły powyższej postaci, więc bardzo przydatne byłoby umieć rozwiązywać je szybko, najlepiej w czasie wielomianowym. Niestety, nie znamy takich metod, a najlepsze znane algorytmy działają w czasie wykładniczym. Znaczący to, że nie są istotnie lepsze od bezmyślnego podstawiania wszystkich możliwych 2^n wartości na zmienne x_i . Co ciekawe, oba powyższe problemy są *NP-zupełne*, co oznacza, że każdy inny problem z klasy NP da się przeformułować na *SAT* albo na problem komiwojażera niewiele (co najwyżej wielomianowo) większej wielkości. Oznacza to, że gdybyśmy umieli rozwiązać jakiś problem *NP-zupełny* (np. *SAT*), to wszystkie inne problemy z klasy NP rozwiązalibyśmy również

Równoważnie definiuje się P jako zbiór problemów rozpoznawalnych przez deterministyczne maszyny Turinga w czasie wielomianowym, a NP jako zbiór problemów rozpoznawalnych przez niedeterministyczne maszyny Turinga w czasie wielomianowym.



w czasie wielomianowym (poprzez sformułowanie w postaci SATa, a potem rozwiązanie go). A to oznaczałoby, że $P = NP$. Ta właśnie cecha problemów NP-zupełnych świadczy o ich znaczeniu: wystarczy rozwiązać jeden z nich, a wiemy, że $P = NP$, z drugiej strony, oczywiście, jeśli tylko jeden z nich nie należy do P, to $P \neq NP$.

Gdybyśmy umieli rozwiązywać SAT w czasie wielomianowym, to wiele problemów rozwiązalibyśmy dużo szybciej. To dobrze. Jednak byłyby też złe strony tej sytuacji. Większość szyfrów opiera się na tym, że pewne obliczenia wymagają dużo czasu i nie są znane metody istotnie lepsze od sprawdzania wszystkich możliwych wartości (jak wyżej dla wartości zmiennych x_i). Gdyby jednak $P = NP$, to te obliczenia już nie potrzebowałyby tyle czasu i większość szyfrów przestałaby być bezpieczna. To złe. To kolejna motywacja do zrozumienia, czy $P = NP$, czy też nie. I to trochę innego rodzaju, bo w tym przypadku, jeśli zrozumiemy, że $P \neq NP$, to też mamy zysk: wiemy, że niektóre szyfry są chociaż trochę bezpieczne. Te dwie motywacje nie są jedynymi powodami, dla których chcemy znać odpowiedź. To tylko poszlaki, że problem jest naprawdę ważny i fundamentalny. Być może najważniejszą przyczyną jest przekonanie, które stoi u źródeł nauk podstawowych, że zrozumienie tak istotnego problemu, niezależnie od wyniku, może skutkować nieprzewidywalnymi wręcz korzyściami. Tak jak to było z odkryciem przez Watsona i Cricka, że DNA ma strukturę podwójnej helisy.

Dochodzimy teraz do pytania, o którym tak naprawdę chcę opowiedzieć, czyli: skoro uważamy ten problem za tak ważny i tylu ludzi bardzo się stara, to dlaczego wciąż jest nierozwiązany? Można by uznać, że jest to pytanie głupie: jak to dlaczego – jest po prostu trudny. Nie pozostawiamy jednak na tym poziomie odpowiedzi, bo historia jest dużo bardziej subtelna. Już przez ponad 45 lat ludzie próbowali rozwiązać problem, pewne metody rozwijały się, ale nie udawało się nimi rozwiązać $P \stackrel{?}{=} NP$. Więc zadawali sobie pytanie: „dlaczego nam się nie udaje?”. I często tym sposobem dochodzili do wniosku, że pewne metody po prostu nie mogą działać. Historia tych badań jest fascynująca i może nam bardzo wiele powiedzieć o teorii złożoności, a także chociaż trochę wytłumaczyć, dlaczego $P \stackrel{?}{=} NP$ jest aż tak trudny i jakie bariery stoją na drodze.

Wszystko zaczęło się w latach 70. Po tym, jak Cook zadał pytanie, matematycy próbowali je rozwiązać. Naturalną techniką była wtedy tak zwana metoda *diagonalizacji*. Wiele pierwszych rezultatów z teorii złożoności (czyli dziedziny zajmującej się klasyfikacją problemów na bardziej złożone i mniej złożone) używa właśnie tej metody. Użyta jest np. w dowodach nierozstrzygalności problemu stopu, twierdzenia (dużo starszego) o tym, że liczb rzeczywistych jest więcej niż naturalnych, albo mniej znanych twierdzeń o hierarchii czasowej i pamięciowej. Niech *język* będzie zbiorem słów nad pewnym ustalonym alfabetem, czyli ciągów liter z tego alfabetu. Rozważmy maszyny, które dostają słowo, wykonują jakieś obliczenie i odpowiadają „tak” lub „nie”, typowy przykład to maszyny Turinga, być może z jakimiś ograniczeniami (jak działanie w czasie wielomianowym). Zbiór słów, dla których maszyna M odpowie „tak”, nazywamy językiem tej maszyny, oznaczamy go $L(M)$. Rozważmy teraz dwie klasy maszyn \mathcal{C}_1 i \mathcal{C}_2 . Aby wykazać, że maszyny z \mathcal{C}_2 opisują pewne języki, które nie są opisywane przez maszyny z \mathcal{C}_1 , wystarczy wskazać taką maszynę $M \in \mathcal{C}_2$, że dla dowolnej maszyny $M' \in \mathcal{C}_1$ istnieje takie słowo w , iż maszyny M i M' potraktują je inaczej (jedna z nich powie „tak”, a druga „nie”). Tę właśnie metodę nazywamy diagonalizacją. Okazuje się, że większość dowodów, które używają tej metody, podlega *relatywizacji*. Aby powiedzieć, co to oznacza, opowiemy najpierw o wyroczni. Rozważmy jakąś wyrocznię O , która odpowiada „tak” lub „nie” na pewne pytania poprawnie i za darmo. Możemy dodać maszynie możliwość zadawania pytań do takiej wyroczni. Przykładowo wyrocznia mogłaby odpowiadać na pytanie, czy dana liczba jest potęgą liczby pierwszej. Widać wtedy, że maszyna, która miałaby za zadanie stwierdzić, czy dana liczba jest pierwsza, miałaby, być może, ułatwione zadanie, gdyby mogła

Więcej o związkach problemu $P = NP$ z kryptologią można przeczytać w artykule Tomasza Kazany, w poprzednim numerze *Delty*.



Rozwiązanie zadania M 1546.

Udowodnimy, że taka liczba n nie istnieje.

Przyjmijmy oznaczenie $f(x) = x^2 + x + 1$. Bezpośrednio sprawdzamy, że dla dowolnego x zachodzi równość

$$f(x)f(x-1) = f(x^2).$$

Ponadto, jeżeli x jest liczbą całkowitą różną od 0, to

$$x^2 < f(x) < (x+1)^2,$$

więc $f(x)$ nie jest kwadratem liczby całkowitej.

Udowodnimy, że dla każdego $n \geq 1$ zachodzi równość

$$(*) \quad \prod_{k=1}^n f(a_k) = f(a_n^2),$$

co w połączeniu z obserwacją z poprzedniego akapitu zakończy dowód.

Przeprowadzimy dowód indukcyjny. Dla $n = 1$ równość (*) jest spełniona.

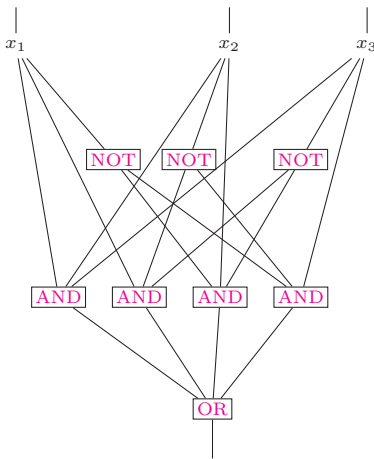
Przypuśćmy, że zachodzi ona dla pewnego $n \geq 1$; wówczas

$$\begin{aligned} \prod_{k=1}^{n+1} f(a_k) &= f(a_{n+1}) \prod_{k=1}^n f(a_k) = \\ &= f(a_{n+1})f(a_n^2) = \\ &= f(a_{n+1})f(a_{n+1} - 1) = \\ &= f(a_{n+1}^2), \end{aligned}$$

co oznacza, że (*) zachodzi również dla $n + 1$. To kończy rozwiązanie.

pytać taką wyrocznię. Klasę maszyn \mathcal{C} z możliwością korzystania z wyroczni O oznaczmy przez \mathcal{C}^O . Powiemy, że dowód, iż maszyny z \mathcal{C}_2 rozpoznają więcej języków niż maszyny z \mathcal{C}_1 , się relatywizuje, jeśli dla dowolnej wyroczni O ten dowód pokazuje również, że maszyny z \mathcal{C}_2^O rozpoznają więcej języków niż maszyny z \mathcal{C}_1^O . Jak powiedzieliśmy, zasadniczo dowody używające diagonalizacji się relatywizują. I dlatego przełomem było twierdzenie, które w roku 1975 udowodnili Baker, Gill i Solovay. Pokazali oni, że istnieje taka wyrocznia A , że $P^A = NP^A$ oraz taka wyrocznia B , że $P^B \neq NP^B$. Oznacza to, że żaden dowód, który się relatywizuje, na pewno nie pokaże, że $P \neq NP$ (bo jest wyrocznia A) ani że $P = NP$ (bo jest wyrocznia B). Jeśli chcemy rozwiązać $P \stackrel{?}{=} NP$, to musimy iść inną drogą.

Ten wynik spowodował, że społeczność badaczy teorii złożoności zarzuciła metody diagonalizacji przy rozwiązywaniu problemu $P \stackrel{?}{=} NP$. W zasadzie słusznie, ale ruch ten był, być może, zbyt radykalny. Można bowiem wyobrazić sobie, że diagonalizacja jest tylko jednym z argumentów w dowodzie i wówczas cały dowód się nie relatywizuje. W każdym razie mniej więcej w tamtym czasie popularne zaczęły być obwody logiczne, to z nimi wiązano teraz nadzieję na



postęp w kwestii problemu $P \stackrel{?}{=} NP$. Obwód logiczny o n wejściach i jednym wyjściu oblicza funkcję $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Wszystko można zakodować jako ciąg zer i jedynek, natomiast jedynek na wyjściu traktować jako odpowiedź „tak”, a zero jako odpowiedź „nie”. A zatem obwód logiczny jest dobrą alternatywą dla maszyn i można też używać go do opisu języków. Pomiedzy n wejściami a wyjściem obwód ma bramki jednego z trzech typów: AND, OR oraz NOT. Bramki AND i OR mogą mieć wiele wejść, a wyjście mają jedno, na którym zwracają koniunkcję (AND) lub alternatywę (OR) wejść. Bramka NOT ma jedno wejście i jedno wyjście, na którym wypuszcza zanegowane wejście. Można składać z tych bramek dowolnie skomplikowane funkcje, na rysunku przedstawiono obwód dla funkcji $\text{xor} : \{0, 1\}^3 \rightarrow \{0, 1\}$, która zwraca sumę wejść modulo dwa. Co prawda, każdą funkcję można opisać pewnym obwodem (Ambitnego Czytelnika zachęcamy do wykazania tego faktu), ale już gdy ograniczymy pewne parametry obwodów, to nie wszystkie funkcje dadzą się opisać i sytuacja staje się znacznie ciekawsza. Przykładowo jeden z ważniejszych wyników tamtego okresu (1983 rok) to dowód, że ciąg funkcji $\text{xor} : \{0, 1\}^n \rightarrow \{0, 1\}$, które zwracają sumę wejść modulo dwa, nie może być opisany ciągiem obwodów (dla coraz większej liczby zmiennych) o stałej głębokości, czyli długości najdłuższej ścieżki od wejścia do wyjścia (czyli xor nie należy do klasy AC^0). Tego typu rezultatów niemożliwości dowodzone wtedy więcej i wierzone, że postęp w tych technikach może doprowadzić do wykazania, że $P \neq NP$. Oczekiwany dowód miałby podążać mniej więcej po następującej linii: 1) definiujemy jakąś sprytną miarę skomplikowania funkcji boolowskiej, 2) wykazujemy, że miara funkcji SAT (czyli przypisującej kodowaniom zero-jedynkowym spełnialnych formuł jedynek, a reszcie zero) lub jakiejś innej funkcji z klasy NP jest duża, 3) pokazujemy, że wszystkie obwody wielomianowej wielkości opisują funkcje o małej mierze. Ponieważ wiemy, że każdą funkcję z P można zapisać wielomianowym obwodem (w takim obwodzie można zakodować algorytm wielomianowy, który jest wykonywany przez bramki), to punkty 1)–3) pokazywałyby, że $P \neq NP$.

Niestety, (a może raczej na szczęście) i tu pojawiła się nieoczekiwana trudność. W roku 1994 Razborov i Rudich przedstawili swoją pracę na temat „dowodów naturalnych”, w której pokazywali, że takiej miary nie można zdefiniować, o ile tylko prawdziwa jest powszechnie uznawana za prawdziwą hipoteza na temat istnienia pseudolosowych funkcji. Główne idee ich dowodu, mimo, że przełomowe i bardzo pomysłowe, dadzą się pokrótce naszkicować. Autorzy pokazali też, że wszystkie wcześniejsze dowody różności klas złożoności korzystały z pewnych własności funkcji, które to własności nazwali naturalnymi. Powiemy, że własność ϕ funkcji jest *naturalna*, jeśli spełnia trzy warunki: i) co najmniej $1/2^n$ funkcji $f : \{0, 1\}^n \rightarrow \{0, 1\}$ spełnia ϕ , ii) jeśli f spełnia ϕ , to nie ma obwodu wielomianowej wielkości, który ją opisuje, iii) spełnianie ϕ



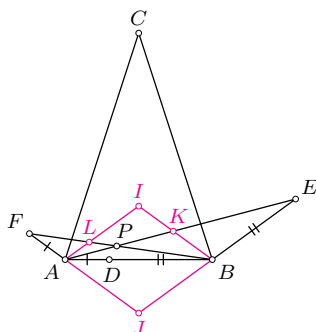
Rozwiązanie zadania M 1547.

Oznaczmy środek okręgu wpisanego w trójkąt ABC przez I , a punkt symetryczny do niego względem prostej AB przez J – wówczas czworokąt $AIBJ$ jest rombem. Niech ponadto $K = AE \cap BI$ oraz $L = BF \cap AI$.

Zauważmy, że

$$\sphericalangle JBD + \sphericalangle DBE = \sphericalangle ABI + 2 \cdot \sphericalangle ABC = 36^\circ + 2 \cdot 72^\circ = 180^\circ,$$

wobec czego punkty J, B, E są współliniowe. Analogicznie dochodzimy do wniosku, że punkty J, A, F są współliniowe.



Z równoległości par prostych $AI \parallel BE$ oraz $BI \parallel AF$ wynika, że

$$\frac{IL}{AL} = \frac{IB}{AF} \quad \text{oraz} \quad \frac{BK}{IK} = \frac{BE}{IA}.$$

W połączeniu z $AD = AF$, $BD = BE$ oraz $IA = IB$ otrzymujemy więc

$$\frac{AD}{BD} \cdot \frac{BK}{IK} \cdot \frac{IL}{AL} = 1,$$

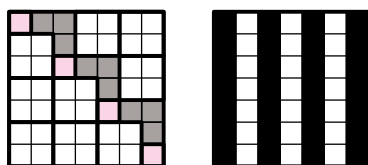
a zatem z twierdzenia odwrotnego do twierdzenia Cevy dla trójkąta ABI wynika, że proste AK, BL, ID przecinają się w jednym punkcie. Tym samym punkt P leży na prostej ID , co jest równoważne tezie zadania.



Rozwiązanie zadania M 1548.

Odpowiedź: $n = 29$.

Przypuśćmy, że pola tablicy pokolorowano tak, że każdy kwadrat 2×2 zawiera co najwyżej dwa czarne pola. Wówczas każdy z 9 białych kwadratów 2×2 oraz każda z 3 szarych figur o polu 3 na rysunku z lewej zawiera co najwyżej 2 czarne pola, a ponadto każdy z 4 kolorowych kwadratów jednostkowych może być czarny. To oznacza, że łącznie w tablicy jest co najwyżej $9 \cdot 2 + 3 \cdot 2 + 4 = 28$ czarnych pól.



Z kolei kolorowanie 28 pól przedstawione na drugim rysunku ma tę własność, że każdy kwadrat 2×2 zawiera dokładnie dwa czarne pola. To oznacza, że szukana najmniejsza wartość n jest równa 29.

daje się obliczyć w czasie $2^{O(n)}$. Gwoli ścisłości, definicja naturalności zależy jeszcze od tego, jakie klasy złożoności chcemy rozróżniać, ale my tu skupimy się na powyższej definicji. Gdybyśmy chcieli realizować plan naszkicowany w poprzednim akapicie, to funkcja spełniałaby ϕ , gdyby miała dużą miarę skomplikowania – autorzy przedstawili przekonujące argumenty, że dla każdej rozsądnej miary rzeczywiście spełnione byłyby punkty i), ii), iii). Niemniej jednak, co było głównym wynikiem pracy, o ile tylko istnieją funkcje pseudolosowe, to nie istnieje żadna naturalna własność ϕ (a więc nie użyjemy jej w dowodzie). Funkcje pseudolosowe to takie, które wyglądają jak losowe z punktu widzenia obserwatora o ograniczonej mocy obliczeniowej. Idea dowodu jest następująca. Weźmy dowolną funkcję, kandydata na funkcję pseudolosową. Losowa funkcja spełnia własność ϕ z prawdopodobieństwem co najwyżej $1/2^n$, co wynika z i). Natomiast funkcja pseudolosowa musi być prosta w opisie, więc ma obwód wielomianowej wielkości (z ii)), czyli spełnia ϕ z prawdopodobieństwem 0. A zatem z prawdopodobieństwem co najmniej $1/2^n$ własność ϕ rozróżnia funkcję losową od naszej, a to, czy funkcja spełnia ϕ , można obliczyć w czasie $2^{O(n)}$ (z iii)). Więc nasza funkcja nie jest pseudolosowa, bo można ją odróżnić od losowej z sensownym prawdopodobieństwem i w odpowiednio ograniczonym czasie. Nie ma więc żadnej takiej – sprzeczność z powszechnie uznawaną hipotezą. Razborov i Rudich w 2007 roku otrzymali nagrodę Gödla za ten rezultat, przez wielu uznawany za najważniejszy wynik dotyczący problemu $P \stackrel{?}{=} NP$.

Również i ten wynik istotnie zmienił społeczność informatyków. Ludzie odwrócili się od obwodów logicznych, przestali dowodzić dolne ograniczenia dla nich, bo jak pokazuje powyższy wynik – nie tędy droga. Być może znów zrobiono to zbyt pochopnie, a barierę należy raczej traktować jako drogowskaz, którego nie iść, tzn. żeby nie definiować naturalnych własności w swoich dowodach. Niedawno zaczęto stosować pewne techniki omijające powyższe bariery i używające algebry. Żeby zrozumieć ideę, przyjrzyjmy się szkicowi dowodu wspomnianego wyżej faktu, że xor nie należy do klasy AC^0 . Dowód bada, jak dobrze można przybliżyć funkcję $f : \{0, 1\}^n \rightarrow \{0, 1\}$ wielomianem wielu zmiennych nad ciałem F_3 o stopniu co najwyżej \sqrt{n} . Przez przybliżenie mamy na myśli zwrócenie takiego samego wyniku w większości przypadków. Okazuje się, że każdą funkcję należącą do klasy AC^0 można przybliżyć całkiem niezłe, natomiast funkcji xor nie da się przybliżyć dobrze. Akurat ten dowód nie omija bariery dowodów naturalnych, ale stosując podobne techniki używające wielomianów niskich stopni nad skończonymi ciałami lub pierścieniami, można ominąć obie wspomniane bariery. Ale i ta technika ma ograniczenia; niedawna praca Aaronsona i Wigdersona z roku 2009 wprowadza nową barierę zwaną algebraizacją. Jest to uogólnienie relatywizacji. Inkluzja $C_2 \subseteq C_1$ się algebraizuje, jeśli $C_2^A \subseteq C_1^A$, dla pewnej wyroczni A oraz wyroczni \bar{A} , która powstała z A poprzez rozszerzenie z funkcji boolowskich do wielomianów niskich stopni nad skończonym ciałem (dla pierścieni to też działa). Jeśli się algebraizuje, to pewnymi technikami bazującymi na wielomianach niskich stopni (tzw. algebraizującymi się) nie da się jej obalić, podobnie jak fakt, iż $NP^A = P^A$ pokazuje, że technikami, które się relatywizują, nie da się obalić inkluzji $NP \subseteq P$. Autorzy wykazali też, że inkluzja $NP \subseteq P$ się algebraizuje, czyli techniki, które się algebraizują, nie pozwolą na jej obalenie. Oznacza to, że piłeczka poszukiwaczy dowodu $P \neq NP$ po raz kolejny została odbita.

Jaka jest przyszłość problemu $P \stackrel{?}{=} NP$? Nie wiadomo. Większość badaczy twierdzi, że $P \neq NP$, ale są też poważani naukowcy, którzy twierdzą, że wręcz przeciwnie, $P = NP$. Najprawdopodobniej nieprędko się to wyjaśni, bo, jak widać, nie bez przyczyny problem ten trzyma się długo, kolejne bariery rzucają nam kłody pod nogi. Musimy uzbroić się w cierpliwość. Albo sami ruszyć do starcia z niedostępnym do tej pory problemem. Tylko pamiętajmy, że jeśli zdecydujemy się na to ostatnie, to należy wcześniej mądrze i gruntownie przygotować się do ataku!