

10

Czy umysł jest liczbą?

Dawno, dawno temu... gdy nie istniały jeszcze komputery i cała ich obliczeniowa moc... daleko, na południu Europy... w krainie zwanej niegdyś Helladą... żyło sobie starożytne bractwo Pitagorejczyków.

Pitagorejczycy wyznawali kult liczby. Wszystko z czymkolwiek się zetknęli usiłowali opisać za pomocą liczb, a były to i przyroda, i muzyka, i gwiazdy, i... wiele, wiele innych rzeczy. Na frontonie ich pradaawnej siedziby widniał dumny napis: „Wszystko jest liczbą”...

Nasz kolejny esej zaczynamy bajkowo, ponieważ zupełnie fantastycznie zdaje się przedstawiać pogląd, zgodnie z którym umysł mógłby być liczbą. Cóż to za pogląd? Czy głosi on, że umysł danego człowieka ma jakiś rozmiar (na podobieństwo stopy) i rozmiar ten określa, jak sprawnie ów człowiek myśli? A może stwierdza on po prostu tyle, że ludzi – żyjących i nieżyjących – daje się ponumerować, a następnie przyporządkować im umysłom zgodne z tą numeracją liczby? Oczywiście NIE, nie o takie sprawy idzie.

Chodzi o coś, co wiąże umysł z matematyką nie tak bezpośrednio, lecz równie ściśle: o *wewnątrz-umysłowy kod*, który miałby wypełniać umysł, a jednocześnie sterować jego pracą. Skąd taki pomysł? Otóż wiemy z biologii, że wewnątrz wszystkich komórek organizmu tkwi specjalny mikro-zapis, tzw. kod DNA, który warunkuje w dużym stopniu cechy tegoż organizmu. Być może kod ten da się ująć liczbowo, być może też podobny zapis, odpowiedzialny jednak za nasze zdolności poznawcze, wbudowała natura w ludzki mózg. Wiemy dalej, tym razem z informatyki, że sterujące pracą automatów programy są w istocie kodami liczbowymi (przeważnie binarnymi), które opisują krok po kroku, w sposób dostosowany do fizycznej konstrukcji automatu, co i kiedy automat ma robić. Wiedza ta nasuwa z kolei przypuszczenie, że myśleniem człowieka kieruje jakiś hipotetyczny super-program, który nie dość że ma postać sekwencji odpowiednio dobranych liczb (kodujących różne operacje myślowe), to da się zapisać nadto jako jedna gigantyczna super-liczba, mieszcząca w sobie pełną informację o kodujących myślowe operacje liczbowych sekwencjach.

Jeśli podjąć tak fantastyczną linię rozumowania, to wyłania się natychmiast kluczowe pytanie o typ liczby i typ „rozumiejącej” tę liczbę maszyny, które miałyby reprezentować wspólnie (maszyna + liczba) czyjś umysł. Zastanawiając się nad możliwymi odpowiedziami – możliwymi, bo nikt jeszcze nie udzielił odpowiedzi ostatecznej – przyjrzymy się na początek niezmiernie interesującej historii pojęcia liczby.

§1. Jak dojrzewało pojęcie liczby?

1.1. Matematyczna historia liczb zaczyna się niewątpliwie od liczb *naturalnych*, oznaczanych dziś zbiorczą literką N , a obejmujących kolejno jedynkę, dwójkę, trójkę itd., zawsze o jeden więcej. Liczby takie służą najlepiej do zliczania przedmiotów, a swoje miano naturalnych zawdzięczają prawdopodobnie temu, że ludzie posługują się nimi spontanicznie, czyli naturalnie. Zapewne też do ich zaistnienia wystarczyło naturalne wyposażenie człowieka, a więc zdolne do wskazywania przedmiotów i przydatne do prostych obliczeń palce. Już tym liczbom jednak przysługuje pewna nienaturalna, tj. niemożliwa do zaobserwowania w przyrodzie, własność, a mianowicie to, że biegną one w nieskończoność.

Znając liczby naturalne, mając nadto potrzebę zapisywania wyników pomiarów, zdołali nasi matematyczni przodkowie wymyślić liczby inne, postaci $\frac{m}{n}$, gdzie zarówno m , jak i n muszą należeć do zbioru N -ów. Dziś zwie się je *wymiernymi*, która to nazwa oddaje dobrze fakt, że stosuje się je do mierzenia (a nie do prostego zliczania przedmiotów). Za pomocą liczby wymiernej $\frac{m}{n}$ można bowiem określić precyzyjnie, że w danej wielkości w (np. długości belki) mieści się dokładnie m razy pewna jednostka, będąca wynikiem podziału danego wzorca na n równych części. Dla przykładu: zapis $\frac{7}{10}$ oznacza, że mierzona wielkość zawiera dokładnie 7 dziesiątych części przyjętego wzorca. Operacje na liczbach wymiernych nie okazały się już tak proste jak uchwytnie intuicyjnie działania na liczbach naturalnych, np. zwykłe dodanie $\frac{1}{3}$ i $\frac{1}{4}$ wymagało przeprowadzenia dość zawilej procedury, która w dzisiejszej notacji przyjmuje postać $(1 \cdot 4 + 1 \cdot 3)/(3 \cdot 4)$.

Co najważniejsze jednak, analiza pomiarów opisywanych za pomocą liczb typu $\frac{m}{n}$ doprowadziła do nie lada kłopotu – kłopotu, z którym trudno było się uporać podtrzymując wspomnianą koncepcję liczby. Chodzi o fakt następujący: mając kwadrat o boku 1, chcemy przedstawić za pomocą liczby wymiernej, czyli dokładnie zmierzyć, jego przekątną. Okazuje się – czego dowiedli po raz pierwszy wspomniani wyżej Pitagorejczycy – że zadania tego wykonać się nie da. Innymi słowy, nie istnieje liczba (liczba wymierna), która wyrażałaby ostatecznie długość przekątnej. Dziś wiemy, że poszukiwaną wartością jest jedna z nieskończenie wielu liczb *niewymiernych*, a więc $\sqrt{2}$; dla ówczesnych uczonych był to jednak poznawczy szok.

Starożytni Grecy poznali bardzo wiele liczb niewymiernych, spośród których dwie najślynniejsze to π – liczba wyrażająca stały stosunek między obwodem okręgu a jego średnicą (każdego okręgu), oraz φ – liczba tzw. złotego podziału, czyli takiego podziału odcinka na części x i y , że stosunek całości $(x+y)$ do większej części x jest identyczny jak stosunek większej części x do mniejszej y (miarą tego podziału jest właśnie φ czyli $(\sqrt{5} + 1)/2$). Przy okazji tych dwóch przykładów warto wyjaśnić, że Grecy traktowali liczby inaczej niż my – to znaczy nie algebraicznie, lecz geometrycznie. Ich twierdzenia o liczbach (w tym różne wzory na pola i objętości) dotyczyły tak naprawdę stosunków między obiektami geometrycznymi, takimi jak linie, figury, bryły i kąty. Również sposób ich dowodzenia odwoływał się do geometrycznych właściwości różnych kształtów.

Szukające odkrycie wielkości (geometrycznych), których nie dało się ująć precyzyjnie za pomocą dotychczas wypracowanej teorii liczb (wymiernych), wywołało *poznawczy kryzys*, a wraz z nim naturalną reakcję obronną. W jej wyniku nastąpiły zmiany dwojakiego rodzaju. Z jednej strony utrwalił się obraz liczby jako obiektu niedoskonałego, który nie wszystko pozwala opisać – stąd też skoncentrowano się jeszcze bardziej na samej geometrii, nie wymagającej tak naprawdę zapisów liczbowych¹. Z drugiej strony wypracowano nowe (z dzisiejszego punktu widzenia prowizoryczne) ujęcie liczby, które pozwoliło operować nowo-odkrytymi wielkościami niewymiernymi, np. porównywać je, dodawać, a nawet (tak, tak!) całkować; od nazwiska pomysłodawcy ujęcie to nazywa się niekiedy teorią *proporcji Eudoksosa*. Podkreślić jednak trzeba, że istota wielkości niewymiernych pozostała nadal tajemnicą i na w pełni zadowalające rozwiązanie trzeba było poczekać ponad tysiąc lat.

1.2. Co się działo jednak pomiędzy? Czy na matematycznym firmamencie rozblęły jakieś nowe, teorio-liczbowe gwiazdy? Rzecz jasna rozblęły, lecz działo się to bardzo powoli.

Po pierwsze pojawiło się *zero*. Owa „pusta, nic-nie-znacząca” liczba przeniknęła do kultury europejskiej z dalekich Indii, gdzie zajmował się nią niejaki Brahmagupta (żyjący w VII wieku n.e.). Dzięki niej mógł powstać w pełni rozwinięty system dziesiętny, a także właściwe temu systemowi mechaniczne reguły rachowania w słupkach. Dzięki zeru także mogła zaistnieć idea wielkości nieskończenie małych, zbiegających granicznie do zera, a więc podstawa rachunku różniczkowego (który wynaleziono, rzecz jasna, wiele stuleci później, bo dopiero w wieku XVII).

Po drugie wprowadzono *liczby ujemne*. Ich europejskie początki (bo dużo wcześniej używał ich m.in. wspomniany wyżej Brahmagupta) przypadają na wiek XV, kiedy to przyjęto do wiadomości, że w praktyce kupieckiej występują nader często straty (także długi) i straty te najlepiej zapisywać za pomocą specjalnych liczb poprzedzonych znakiem minus. Z punktu widzenia matema-

¹ Językowa pozostałość takiego nastawienia przetrwała właściwie do wieku XIX, do którego matematyków nazywano geometrami.

tyki „czystszej” ważniejsze jednak, że dopuszczając do użycia liczby ujemne zgodzono się na następujący, wysoce abstrakcyjny, pogląd: liczby nie muszą oznaczać niczego istniejącego, mogą oznaczać coś nieistniejącego lub być tylko i wyłącznie pewnym pomocniczym zapisem, który pozwala efektywnie przekształcać złożone, matematyczne wyrażenia (np. równania).

Ostatnie spostrzeżenie prowadzi wprost do kolejnego elementu naszej skróconej historii liczb, a mianowicie *algebry*. Za sprawą algebry bowiem liczby przeistoczyły się w *symbole* – symbole, którymi można manipulować mechanicznie, w częściowym przynajmniej oderwaniu od ich znaczeń (czyli np. wyników pomiarów). Typowym rozwinięciem nowego podejścia było rozwiązywanie równań, w których występowały symbole danych, niewiadomych i współczynników. Chcąc wyznaczyć wyniki równania, należało postępować zgodnie z określonym przez algebrę schematem, np. żeby rozwiązać równanie kwadratowe z niewiadomą x , trzeba było przenieść wszystkie symbole oprócz zera na lewą stronę, obliczyć wyróżnik zwany deltą i zależnie od jego wartości zastosować gotowy wzór na obliczenie jednej lub dwóch wartości $x - a$. Innym sukcesem podejścia algebraicznego stało się opracowanie szeregu schematów mechanicznych działań na liczbach zapisanych dziesiętnie (np. poznawanego dziś w szkole podstawowej schematu dodawania w słupkach).

Dziesiętny system pozycyjny trafił do Europy w wieku XIII za pośrednictwem lubujących się w algebrze Arabów, którzy w wieku IX udoskonalili i rozpowszechnili znacznie wcześniejsze pomysły Hindusów (kluczową rolę odegrała tu praca uczonego arabskiego al-Chwarizmiego „O liczbach indyjskich”).

Był to wynalazek rewolucyjny. Za jego sprawą zapisy liczbowe stały się bardzo przejrzyste (nie musimy tego tłumaczyć, bo dziś niemal każdy myśli o liczbach „dziesiętnie”), co ważniejsze jednak, system ów pozwolił zalgorytmizować obliczenia i to w sposób zrozumiały dla ogółu. Było to tym ważniejsze, że używany wcześniej zapis rzymski (jego próbkę widzieliśmy wyżej w nazwach stuleci: wiek IX i XIII) cechowała skrajna rachunkowa niewydolność.

Powróćmy jednak do wieńczącej liczbową historię Starożytności wynalazku Eudoksosa. Stwierdziliśmy, że pomysł ten pozwolił Grekom wyjść z impasu wywołanego odkryciem wielkości niewymiernych, był jednak prowizoryczny i nie dość ścisły. Choć tak było w istocie, to czas – czas ponad tysiącletni – musiał zrobić swoje.

W wieku XIX powolna kumulacja nowych matematycznych idei doprowadziła do ścisłej definicji *liczby rzeczywistej* za pomocą tzw. przekrojów Dadekinda. Owe nowe-stare liczby objęły swoim zasięgiem wszystko to, co wcześniej określano mianem liczb: liczby naturalne, wymierne, niewymierne, zero i liczby ujemne. Zaproponowana nazwa miała wyrażać fakt, że określono precyzyjnie coś, co miało odpowiadać rzeczywistym wynikom pomiarów (wszak przy odpowiednim doborze skali mogły one być nie tylko dodatnie, lecz również zerowe i ujemne – jak np. przy pomiarze temperatury).

Nazwa ta mówiła jednak coś więcej, a mianowicie przeciwstawiała liczby rzeczywiste obiektom innym, które znano już od dawna, a opatrywano niezbyt elegancką etykietką liczb urojonych (później zaś bardziej elegancką: *zespolonych*). Owe inne liczby nie służyły (przynajmniej początkowo) ani do liczenia, ani do mierzenia, lecz miały znaczenie czysto algebraiczne. Za ich pomocą udawało się po prostu rozwiązywać pewne równania, które w dziedzinie liczb rzeczywistych bądź rozwiązywały się bardzo trudno, bądź nie rozwiązywały się wcale (początkowo chodziło o równania trzeciego stopnia).

Liczbę zespoloną definiuje się jako parę liczb rzeczywistych (x, y) ; traktując x jako rzeczywisty składnik pary, a y jako składnik urojony. w ten sposób liczba staje się czymś więcej niż pojedynczą wartością rzeczywistą – ujmując rzecz graficznie, odpowiada jej nie punkt na osi, lecz punkt na płaszczyźnie (o współrzędnych x i y). Owe nowe liczby – zespolone jak widać z dwóch rzeczywistych składników – mają nadto tę ciekawą własność, że niektóre z nich po podniesieniu do kwadratu dają rzeczywiste liczby ujemne, np. $(0,1)^2$ daje $(-1,0)$. Istnieją więc zespolone pierwiastki liczb ujemnych (np. $(\sqrt{-1} = (0,1))$). To właśnie ta cecha zdecydowała o algebraicznej przydatności nowych liczb w rozwiązywaniu równań.

1.3. Wprowadzając na scenę liczby zespolone, zakończyliśmy pierwszą odsłonę wrywkowej historii liczb, która to historia ma nam dopomóc w zamierzonym dalej zestawieniu pojęcia umysłu z pojęciem liczby. Pora zatem na tymczasowe podsumowanie, którego dokonamy w tabelce.

Liczby...	...i ich przeznaczenie
naturalne	<i>zliczanie przedmiotów</i>
wymierne	<i>mierzenie długości, pól, proporcji itp.</i>
niewymierne	<i>j.w.</i>
proporcje Eudoksosa	<i>j.w. (próba ścisłego ujęcia istoty liczb niewymiernych)</i>
zero	<i>umożliwienie pozycyjnego zapisu liczb dziesiętnych</i>
ujemne	<i>zapisywanie strat, długów itp.</i>
rzeczywiste	<i>pomiary wszelkich wartości</i>
zespolone	<i>ułatwienia algebraiczne, dziś: obliczenia inżynierskie</i>

Czy domknięcie tabelki po pozycji „liczby zespolone” należy traktować jako ostateczne i czy w ogóle miałyby sens jakiegokolwiek domknięcie? Oczywiście NIE. Historia matematyki poucza, że pojęcie liczby się zmienia, a przyczyną owej nieustającej zmienności są zarówno pewne kryzysy poznawcze (choćby ten, który nastąpił po odkryciu wielkości niewymiernych), jak i nowe potrzeby praktyczne (choćby ta, która legła u podstaw wprowadzenia liczb

ujemnych jako dobrego narzędzia opisu kupieckich strat). Jednym słowem – pojęcie liczby nie było i *nie jest pojęciem zamkniętym*.

Druga ważna uwaga do zawartości tabelki wiąże się ze wskazaniem jej kluczowego elementu – kluczowego z punktu widzenia dalszych rozważań. Elementem tym są liczby *rzeczywiste niewymierne*. Liczby te, mimo istnienia ich ścisłej matematycznej definicji, otacza pewna aura tajemniczości, która komponuje się dobrze z kluczową dla nas i równie zagadkową tematyką umysłu. Co to za aura? Otóż liczby niewymierne mają nieskończone i nieregularne rozwinięcia dziesiętne, co sprawia, że na dobrą sprawę nigdy nie będziemy w stanie poznać ich dokładnych wartości (definiujemy je wprowadzając granice ciągów, ale dokładnych wartości tychże granic nigdy nie poznamy).

W pewnym sensie zatem liczby niewymierne są nieobliczalne. Nie jest to jednak ten sens, o którym mówią informatycy, gdy wskazują na problemy nierozwiązywalne za pomocą pewnych typów maszyn liczących. Typ, który będzie nas interesował w kolejnym punkcie, to komputery cyfrowe i oddająca ich istotę maszyna Turinga.

§2. O maszynach Turinga i liczbach nieobliczalnych

W latach 30-tych XX wieku, a więc w okresie wykraczającym znacznie poza naszą szkicową rekonstrukcję historii liczb, świat matematyczny elektryzowały dwa pytania: (P1) *Czy wszystkie dobrze określone problemy matematyczne mają rozwiązania algorytmiczne?*, a jeśli tak nie jest, to (P2) *Czy istnieje algorytm rozstrzygający, że dany problem ma rozwiązanie algorytmiczne?* Pytania te dotyczyły w istocie naturalnych ograniczeń ówczesnej algebry². Stawiając je, nie spodziewano się, rzecz jasna, odkrycia odpowiednich *algorytmów* – z uwagi na ogrom otwartych po dziś dzień zagadnień matematycznych byłoby to oczekiwanie groteskowe. Chciano po prostu wiedzieć, czy algorytmy takie mogą istnieć.

Ścisła odpowiedź na (P1) i (P2) wymagała wstępnego określenia, co należy rozumieć przez algorytm czy też, jak wówczas mówiono, procedurę mechaniczną. Zadanie to wykonał i to wykonał perfekcyjnie nasz dobry znajomy ze stronic tej książki: Alan Turing. Zdefiniował mianowicie pewną *abstrakcyjną maszynę*, której zasady działania wyrażały ściśle, a przy tym niezwykle sugestywnie, ideę postępowania zalgorytmizowanego (zob. [Turing 1936]). Co to za maszyna?

2.1. Wyobraźmy sobie mechanizm złożony ze skończonego rejestru stanów, tablicy zmian stanów, głowicy odczytująco-zapisującej oraz nieskończonej, po-

² Mamy na myśli wczesną postać algebry (w odróżnieniu od współczesnej algebry abstrakcyjnej), która koncentrowała się na mechanicznych rachunkach symbolicznych.

kratkowanej taśmy, na której można zapisywać symbole pewnego alfabetu. Wyobraźmy sobie dalej, że głowica porusza się po taśmie, zmieniając jej zawartość, a cała maszyna przechodzi skokowo, w sposób odgórnie zaplanowany, od jednego stanu do drugiego. Gdy w bieżącej kratce „widzi” symbol x , wpisuje w niej inny symbol y , przesuwa głowicę o jedną kratkę w lewo lub w prawo, po czym zmienia swój bieżący stan p na inny stan q . Można powiedzieć zatem, że „mechanizm głowicowo-taśmowy” działa na podstawie sekwencji instrukcji typu „JEŚLI (stan= p), (symbol= x), TO zmień stan na q , zmień symbol na y , przesuń głowicę w prawo lub w lewo”. Celem takiej przeplatanki stanów, ruchów głowicy i operacji odczytu/zapisu jest zastąpienie wejściowego zbioru symboli na taśmie zbiorem wynikowym. Przeplatanka kończy się, gdy maszyna znajdzie się w wyróżnionym stanie końcowym k – wówczas programista może zinterpretować otrzymany wynik, czyli ciąg symboli, które pozostały na taśmie maszyny.

W prostym języku stanów i symboli alfabetu daje się układać wiele programów, mniej lub bardziej skomplikowanych. Na przykład, gdyby ktoś chciał napisać programik do mnożenia dwóch liczb, musiałby wybrać alfabet, w którym liczby byłyby kodowane (np. symbole 0 i 1), zdecydować się na liczbę stanów maszyny (np. 5) oraz wymyślić odpowiedni ciąg instrukcji zawierających symbole alfabetu i stany maszyny. Następnie, wystarczyłoby umieścić na taśmie automatu dwie odpowiednio zakodowane liczby, a ten wykonawszy program, pozostawiłby na niej odpowiednio zakodowany wynik.

Taki mniej więcej opis maszyny realizującej algorytmy dał w roku 1936 wspomniany wyżej Alan Turing³. Znacznie później udowodniono, że maszyna taka ma możliwości identyczne z dowolnie zaawansowanym *komputerem cyfrowym* – niezależnie od użytej w nim technologii przetwarzania danych i stopnia złożoności oprogramowania. Innymi słowy: dowolnie skomplikowany program komputerowy można zakodować w języku zrozumiałym dla maszyny tak prostej jak opisana wyżej. Oczywiście, trzeba przy tym liczyć się ze skrajną nieprzejrzyistością owego programu oraz żółtym tempem jego wykonywania.

Warto w tym miejscu wskazać pewną analogię między językiem „rozumianym” przez maszynę Turinga a językiem wewnętrznym komputera. Podobnie jak wszelkie programy komputerowe (niezależnie od języka programowania, w którym je zakodowano) są ostatecznie kompilowane do postaci ciągów binarnych i dopiero w takiej postaci realizowane przez komputer, tak wszelkie algorytmy (niezależnie od stopnia ich złożoności) daje się zapisać w postaci umożliwiającej ich wykonanie przez maszynę Turinga (choć w takim wypadku należy się liczyć, rzecz jasna, z ogromnym stopniem złożoności i nieprzejrzyistości kodu).

³ Co ciekawe, inspirację dla swojej matematycznej konstrukcji czerpał Turing z psychologii. Mianowicie, starał się zanalizować drobiazgowo czynności rachmistrza wykonującego obliczenia przy użyciu ołówka i kartki papieru. Usiłował odnaleźć wśród działań rachmistrza takie, których nie dałoby się już podzielić na mniejsze. To co odnalazł, zaszył właśnie w strukturze i zasadach działania swojej maszyny.

Gwoli ścisłości trzeba wyjaśnić, że Turing wymyślił tak naprawdę dwie maszyny (a dokładniej: dwa typy maszyn). O pierwszej pisaliśmy wyżej, o drugiej trzeba powiedzieć tyle, że jest to pewnego rodzaju nad-maszyna, zbudowana identycznie jak maszyna zwykła, lecz zdolna symulować pracę dowolnej maszyny zwykłej (a to dzięki programowi interpretującym część symboli na taśmie jako pełen opis maszyny symulowanej). Ów drugi pomysł wszedł do historii nauki pod dobrze dziś znaną nazwą *uniwersalnej maszyny Turinga* (w skrócie UMT).

Objaśnijmy tę ideę dokładniej. Gdybyśmy umieścili na taśmie UMT kod konkretnej maszyny M_i (czyli kod opisujący jej alfabet, jej możliwe stany i tablicę zmian stanów) oraz pewne dane początkowe dla M_i , to UMT działałaby następująco. Czytałaby z taśmy raz dane wejściowe, raz program maszyny M_i (m.in. tym steruje jej uniwersalny program), a poza tym wykonywałaby dokładnie to samo, co przewiduje zakodowany na jej taśmie program maszyny M_i . Znaczy to, że UMT wypisywałaby na swojej taśmie (w polach następujących po opisie maszyny M_i) dokładnie te same symbole, które wypisywałaby na swojej taśmie maszyna M_i .

Gdybyśmy teraz, znając już różnicę między obydwoma pomysłami, chcieli uściślić powiedzenie, że „maszyna Turinga jest równoważna dowolnemu komputerowi cyfrowemu”, to musielibyśmy stwierdzić, co następuje. Po pierwsze: że konkretna maszyna M_i jest równoważna komputerowi realizującemu konkretny program (np. program do sumowania liczb). Po drugie zaś: że uniwersalna maszyna UMT jest równoważna nie komputerowi wyspecjalizowanemu, lecz przygotowanemu do realizacji różnych możliwych programów.

2.2. Po szkicowej prezentacji maszynowych idei Turinga możemy powrócić do porzuconego wątku i kontynuować przerwana na kilka stron historię liczb. Kontynuacja w tym właśnie miejscu jest jak najbardziej zasadna, ponieważ to właśnie pojęcie maszyny (w domyśle: liczącej) umożliwiło odkrycie nowej klasy liczb, zwanych *nieobliczalnymi*.

Spróbujmy odtworzyć rozumowanie, które doprowadziło Turinga do tegoż odkrycia. Początek już znamy: opisana wyżej abstrakcyjna maszyna do przetwarzania symboli. Podawszy jej definicję, mógł Turing przyjąć, że zarówno dane na wejściu maszyny, jak i generowane przez nią wyniki wolno rozumieć jako liczby. Choć interpretacja taka wydaje się sztuczna – wszak nie każdy program komputerowy służy celom czysto rachunkowym – to jest jak najbardziej możliwa. To zaś dlatego, że istnieją w matematyce jednoznaczne metody przekształcania ciągów symboli, np. generowanych przez maszyny, w liczby (i to naturalne). Pierwsza z nich została opracowana przez matematyka niemieckiego Kurta Gödla (dla potrzeb logiki) i z niej właśnie skorzystał w swoim rozumowaniu Turing (zob. esej 18, §3).

Mając do dyspozycji wspomnianą metodę, mógł Turing przyjąć dalej, że nie tylko dane maszyny i nie tylko jej symboliczny wynik, lecz także samą maszynę wolno potraktować jako liczbę. Dzieje się tak, ponieważ maszynę defi-

niuje jej program (tj. tablica zmian stanów), a ten stanowi ciąg symboli, który znowu (podobnie jak dane i wynik) można zakodować na sposób gödłowski, jako liczbę naturalną.

Podsumowując zatem: zarówno danym, jak i maszynom, jak i wynikom maszyn nadał Turing jednoznaczny *interpretację liczbową* (naturalno-liczbową). Stwierdził dalej, że wszystkie możliwe wyniki wszelkich możliwych maszyn dla wszelkich możliwych danych tworzą *zbiór liczb obliczalnych*, czyli dających się uzyskać mechanicznie (inaczej algorytmicznie). W tym momencie mógł zadać kluczowe pytanie: *czy istnieje coś jeszcze?*

Chcąc na nie odpowiedzieć, zastosował tzw. *rozumowanie przekątniowe*, zaproponowane po raz pierwszy w roku 1890 przez George'a Cantora przy okazji dowodu, że zbiór liczb rzeczywistych R zawiera istotnie więcej liczb niż zbiór liczb naturalnych N (to znaczy jest nieprzeliczalny).

Rozumowanie Cantora – dowodzące nieprzeliczalności zbioru R – można przedstawić w odniesieniu do liczb z przedziału $(0,1)$, a więc pewnego wycinka liczb rzeczywistych. Załóżmy wbrew dowodzonej tezie, że liczby z tego przedziału są przeliczalne, to znaczy można je ustawić w ciąg (wypisać jedna za drugą). Jeśli tak jest, to możemy wyobrazić sobie tabelę, w której kolejnych wierszach stoją rozwinięcia dziesiętne kolejnych liczb, np. 0,23423? Jeśli przyjmiemy dalej, że w kolumnach tabeli stoją kolejne cyfry rozwinięć, to możemy zapytać, czy któryś z wierszy zawiera specjalnie dobraną liczbę L – taką liczbę, której kolejne składniki rozwinięcia różnią się (np. o jedność) od cyfr stojących na *przekątnej* tabeli. Liczba L nie może stać w żadnym z wierszy, ponieważ różni się ona (z definicji) od każdej liczby z tabeli na co najmniej jednym miejscu po przecinku (właśnie tym miejscu, które wskazaliśmy odwołując się do przekątnej). A jeśli liczby L (poprawnie zdefiniowanej przecież) nie ma w tabeli, to musimy się zgodzić, że istnieje choćby jedna liczba rzeczywista (choćby liczba L), której nie da się „wtłoczyć” w nasz założony (na początku) ciąg. Otrzymujemy sprzeczność z początkowym założeniem. A zatem liczb rzeczywistych z przedziału $(0,1)$ nie da się ustawić w ciąg; są one *nieprzeliczalne*.

Podstawą swojej wersji rozumowania przekątniowego uczynił Turing nieskończoną, dwuwymiarową tabelę liczb – tabelę obejmującą numery maszyn M_i , kody danych D_j , oraz zapisane liczbowo wyniki przetworzenia danych D_j przez maszynę M_i , które to wyniki możemy oznaczyć jako W_{ij} (wynik i -tej maszyny dla j -tych danych).

W pierwszej kolumnie tabeli umieścił liczby reprezentujące maszyny M_i (tzn. ich numery). W pierwszym wierszu natomiast wstawił zakodowane liczbowo dane D_j , które będą podawane każdej z maszyn do przetwarzania; daje to ten sam, co w pierwszej kolumnie, ciąg kolejnych liczb naturalnych (gdy dane stanowią parę, trójkę etc. liczb, to odpowiednia metoda kodowania redukuje je do jednej). Na przecięciach kolejnych wierszy i kolumn, czyli w wewnętrznych polach tabeli, przewidział miejsce na wyniki przetworzenia (czyli obliczenia) określonych danych przez określoną maszynę (czyli liczby W_{ij}). Oto schemat stosownej tabeli.

$[M_i/D_j/W_{ij}]$	D_1	D_2	D_3	...
M_1	W_{11}	W_{12}	W_{13}	...
M_2	W_{21}	W_{22}	W_{23}	...
M_3	W_{31}	W_{32}	W_{33}	...
\vdots	\vdots	\vdots	\vdots	\ddots

Znając już punkt wyjścia, powtórzmy teraz rozumowanie przekątniowe Turinga. Weźmy pod uwagę wszystkie wyniki W_{ij} ulokowane na przekątnej tabeli. Po wypisaniu ich w jednym wierszu tworzą one pewien ciąg nieskończony, przeliczalny. Zmieńmy teraz wszystkie wyrazy tego ciągu w systematyczny sposób, np. dodając do każdego jedność. Tak powstaje nowy ciąg, który się różni od wszystkich zapisanych w kolejnych wierszach tabeli. Różni się on od ciągu z pierwszego wiersza, bo w tamtym jest na pierwszym miejscu (tj. w pierwszej kolumnie) jakaś liczba n , a tu będzie $n + 1$. Na drugim miejscu różni się od drugiej liczby z wiersza drugiego, na trzecim od trzeciej liczby z wiersza trzeciego itd. Mamy więc nowy zbiór liczb, różny od każdego z figurujących w wierszach tabeli. Ale przecież w tabeli udało się zmieścić wszystkie możliwe maszyny służące do obliczeń! Nowy ciąg zatem, spisany z przekątnej, nie może pochodzić od żadnej z zarejestrowanych na liście maszyn, czyli maszyn produkujących liczby obliczalne. A zatem liczba reprezentowana przez ten ciąg *nie należy do obliczalnych*⁴.

2.3. Spoglądając krytycznie na przytoczone rozumowanie, nie sposób oprzeć się (pierwszemu) wrażeniu, że Turing odkrył coś, co ponad dwa tysiące lat wstecz odkryli Pitagorejczycy, a mianowicie... *liczby niewymierne*. Uczynił to wprawdzie w innym kontekście matematycznym i za pomocą innych pojęć – ale jednak...

Turingowskie liczby obliczalne bowiem to nic innego jak wyniki systematycznych i skończonych przekształceń liczb naturalnych, które to wyniki muszą utworzyć zbiór przeliczalny, a więc sprowadzalny ostatecznie do zbioru liczb naturalnych (takim jest na przykład zbiór liczb wymiernych, ▶por. §1.1). Wielkości nieobliczalne z kolei mają być czymś, co wykracza – na mocy przedstawionego rozumowania – poza różne klasy liczb przeliczalnych, a czymś takim są właśnie liczby nazywane ogólnie niewymiernymi.

⁴ Por. artykuł Witolda Marciszewskiego p.t. „Dlaczego nie każde rozumowanie da się zmechanizować”, dostępny w Internecie, pod adresem <http://www.calculamus.org/forum/5/wol-marc.html>, a także w tomie „Filozofia i logika. W stronę Jana Woleńskiego” pod red. J.Hartmana, Aureus, Kraków 2000.

Pierwsze wrażenie może być jednak mylące, a to za sprawą swojej nieprecyzyjności. O liczbach niewymiernych można bowiem myśleć w kategoriach obliczeniowych, to znaczy można wskazywać pewne procedury algorytmiczne, które pozwalają wyznaczać pewne wielkości niewymierne (choć nie wszystkie) z dowolną dokładnością. Podkreślmy tę frazę: *z dowolną dokładnością*. Na przykład: dla słynnej liczby Eulera e istnieje prosty wzór, ($e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$), który dla coraz większych n daje coraz lepsze przybliżenie liczby e (zbiegające granicznie do jednej jedynej wartości; na co istnieje stosowny dowód). Sprawę tę wolno ująć i tak, że dla pewnych liczb niewymiernych da się wskazać (przynajmniej teoretycznie) pewien ściśle określony algorytm znajdowania kolejnych cyfr ich przedstawień dziesiętnych – zarówno przed, jak i po przecinku (np. dla liczby e cztery kolejne cyfry wyglądają tak: 2,718).

Mając to wszystko na uwadze, możemy turingowskie rozumowanie przekątniowe zinterpretować nieco inaczej niż wyżej. Możemy mianowicie przyjąć, że w tabeli T ujęto wszystkie maszyny zdolne wykonywać wszystkie procedury obliczeniowe prowadzące do wszystkich tych liczb niewymiernych, których przedstawienia dziesiętne dają się wyznaczyć krok po kroku, z dowolną zadaną dokładnością. Przy takim rozumieniu nadal pozostaną pewne wielkości nie uwzględnione wewnątrz tabeli. O ich istnieniu upewni nas ponownie systematyczna przeróbka wyników leżących na przekątnej. Owe hipotetyczne wielkości będą *nieobliczalnymi liczbami niewymiernymi*, czyli tymi spośród liczb niewymiernych, do których nie może nas doprowadzić żadna maszyna (w domyśle: maszyna równoważna automatom Turinga).

Możemy powiedzieć zatem, że Alan Turing odkrył coś więcej niż Pitagorejczycy. Wyodrębnił z liczb niewymiernych ważną klasę liczb osiągalnych dla maszyn i sterujących nimi algorytmów (tj. klasę liczb obliczalnych; w skrócie KLO). Co ciekawe, do klasy KLO należą wszystkie bodaj liczby niewymierne znane Pitagorejczykom (w tym π i φ). Co jeszcze ciekawsze, klasa KLO jest przeliczalna, a więc równoliczna (w matematycznym sensie) ze zbiorem liczb naturalnych. Ponieważ zaś wszystkich liczb niewymiernych, mieszczących w sobie przeliczalną klasę KLO, jest nieprzeliczalnie wiele, to liczb pozostałych, a więc nieobliczalnych, musi być *nieprzeliczalnie wiele*. Jest ich więc nieskończoność i to nieskończoność większej mocy niż ta, która cechuje liczby naturalne.

Klasa niewymiernych liczb obliczalnych, czyli dość wyjątkowych liczb niewymiernych, jest mocno zróżnicowana. Należą do niej przede wszystkim niewymierne liczby algebraiczne, czyli takie, które są pierwiastkami wielomianów o wymiernych współczynnikach (np. liczba $\sqrt{2}$, która jest pierwiastkiem wielomianu $x^2 - 2$, czy staro-grecka liczba złotego podziału φ , która jest pierwiastkiem wielomianu $x^2 - x + 1$), ale także pewne ważne liczby nie-algebraiczne (tzw. liczby przestępne, jak π i e).

2.4. Napotykając pojęcie liczby nieobliczalnej, wyobraźnia ludzka ma nie lada kłopot. Kłopot ten wiąże się z silnym przyzwyczajeniem ludzi do *symbolicznego zapisywania* liczb, najlepiej w postaci dziesiętnej (przykładowo: $3/4$ to $0,75$, a liczba e to $2,1728\dots$ i dalej, zgodnie z regułą określoną przez pewien wzór; zob. wyżej). Dzięki zapisywaniu ludzie potrafią uchwycić liczby jako coś skończonego i zamkniętego w jednym napisie (jak w przypadku $0,75$), albo jako coś nieskończonego wprawdzie, lecz opisanego skończonym wzorem, pozwalającym nadto wyrażać liczbę coraz dokładniej (jak w przypadku liczby e).

Jeśli teraz, mając na uwadze wskazany kontekst psychologiczny, spojrzymy na liczby nieobliczalne, to muszą się one wydać niezwykle dziwne. W ich przypadku bowiem nie mają zastosowania *żadne reguły zapisu* – gdyby reguły takie obowiązywały, to moglibyśmy sami lub za pomocą jakiejś maszyny podawać kolejne cyfry danej liczby, a to znaczyłoby, że jest ona obliczalna. Weźmy dla przykładu pewną liczbę nieobliczalną L , która jest „pokrewna i stosunkowo bliska” liczbie Eulera e . Jej dziwny, bo nieobliczalny, charakter moglibyśmy wyrazić jakoś tak: „jest to liczba $2,718$, a po ósemce nie wiadomo co, bez żadnej reguły, która pozwoliłaby nam kontynuować ten zapis⁵”.

Ze względu na użytą wyżej frazę „*nie wiadomo co*” wydawać by się mogło, że liczby nieobliczalne są zupełnie niedefiniowalne i nie ma żadnego sposobu na to, by uchwycić ich matematyczny sens. Tymczasem istnieje pewna droga pośrednia, która nie dość, że przemawia do wyobraźni, to pozwala jeszcze kojarzyć nasze dziwne liczby z pewnymi dobrze określonymi problemami matematycznymi. Mówiąc obrazowo i wstępnie: jest to droga okrężna, a jej bieg znamy coraz lepiej dzięki *informatyce*.

Wyjaśnijmy to dokładniej. Otóż znamy z informatyki pewne dobrze określone, a więc ściśle zdefiniowane, problemy, które sami informatycy zwą *nieobliczalnymi*. A zwą je tak, ponieważ udowodniono ponad wszelką wątpliwość, że choć istnieją rozwiązania tych problemów, to nie istnieją algorytmy (w sensie procedur realizowalnych przez maszyny Turinga), które mogłyby do tych rozwiązań doprowadzić. Co bardzo ważne, niektóre ze wspomnianych problemów są bardzo łatwo uchwytnie, a więc dostępne dla wyobraźni przeciętnego człowieka, niekoniecznie matematyka.

O sprawach tych pisaliśmy szerzej w eseju 2, p.t. „*Czy komputery mogą być nieobliczalne?*”. Przypomnijmy w tym miejscu jeden ze szczególnie sugestywnych i prostych do wyrażenia problemów nieobliczalnych. Jego punkt wyjścia to zbiór precyzyjnie określonych języków programowania z ustaloną symboliką i ustalonymi regułami składni (czyli budowy wyrażeń złożonych z wyrażeń prostych). Sam problem zaś wyraża się równie ścisłym (i prostym w konstrukcji) pytaniem: czy istnieje algorytm, który dla dwóch ta-

⁵ Mogłaby to być co najwyżej jakaś reguła losowa, określająca prawdopodobieństwa, z jakimi mogą pojawiać się kolejne cyfry naszej liczby.

kich, zupełnie dowolnych języków programowania, pozwalałby sprawdzić bezbłędnie i w skończonym czasie, czy języki te są *równoważne składniowo*, a więc, czy z ich instrukcji podstawowych daje się zbudować takie same zbiory instrukcji złożonych? Wiadomo, że w każdym przypadku problem ma rozwiązanie – każde dwa języki bowiem są albo równoważne składniowo, albo nie. Mimo to poszukiwany algorytm sprawdzania (prowadzący do prostych odpowiedzi „tak lub „nie”) nie istnieje. Problem więc, choć dobrze zdefiniowany, zalicza się do nieobliczalnych.

Informatycy (do spółki z matematykami) zidentyfikowali wiele innych problemów nieobliczalnych. Najbardziej podstawowy z nich to tzw. *problem stopu* – sformułowany po raz pierwszy przez Alana Turinga. Wyraża się on następującym pytaniem: „Czy istnieje taki uniwersalny program komputerowy (maszyna Turinga), który dla dowolnego innego programu i dowolnych jego danych wejściowych, jest w stanie rozstrzygnąć, czy dla tych właśnie danych program ów zakończy pracę, czyli zatrzyma się”. Okazuje się, że programu takiego być nie może, a to przesądza o nieobliczalności podanego problemu. Problem stopu można uznać za kanoniczny, bo wykorzystuje się go przy dowodzeniu, że inne problemy są również nieobliczalne (z grubsza rzecz biorąc, dowód polega na redukcji rozpatrywanego problemu do problemu stopu). Por. [Harel 2000].

Co jednak łączy nierozwiązywalne problemy algorytmiczne z liczbami nieobliczalnymi? Czy ich hipotetyczny związek odwołuje się do turingowskich pojęć maszyny i jej kodu? Oczywiście TAK, a oto stosowne wyjaśnienie.

Skoro pewien problem P jest nieobliczalny, czyli nie ma algorytmicznego rozwiązania, to nie istnieje rozwiązująca go maszyna Turinga. Nie istnieje zatem jej *obliczalny kod*, który moglibyśmy zapisać (gdyby istniał) jako pewną, ściśle określoną, liczbę obliczalną. Mówiąc zaś inaczej: hipotetycznej metodzie rozwiązywania problemu P odpowiada pewna liczba nieobliczalna L_P , która za sprawą nieobliczalności problemu P nie może być kodem żadnej maszyny Turinga.

Wypada na koniec podzielić się pewnym arcyciekawym spostrzeżeniem. Odwoływaliśmy się wyżej do pojęcia maszyny Turinga uznając, że niektóre problemy należą do nieobliczalnych, bo nie istnieje rozwiązująca je maszyna Turinga. Niewykluczone jednak, że istnieje maszyna alternatywna (np. probabilistyczna lub jakaś inna), która część takich problemów potrafiłaby rozwiązywać. Gdyby tak właśnie było, to w klasie liczb nieobliczalnych (odpowiadających problemom nierozwiązywalnym za pomocą maszyn Turinga) należałoby wskazać pewną nową podklasę, która zawierałaby (wbrew dotychczasowej nazwie klasy nadrzędnej) liczby obliczalne – obliczalne za pomocą innego rodzaju maszyny niż ta, którą zdefiniował Turing. Byłoby to równie doniosłe odkrycie jak to, którego dokonał sam Turing, wyodrębniając z liczb niewymiernych nieznaną wcześniej klasę liczb obliczalnych KLO (obliczalnych za pomocą jego maszyny, która – przypomnijmy to – jest równoważna współczesnym komputerom cyfrowym).