Komputery, Paradoksy i Podstawy Matematyki

[POLISH TRANSLATION BY: s.ramiramzes@gmail.com]

Niektórzy wielcy myśliciele XX wieku pokazali, że nawet w sterylnym świecie matematyki niezupełność i losowość są powszechne

Gregory J. Chaitin

Każdy wie, że komputer to bardzo praktyczna rzecz. Właściwie to komputery są niezbędne we współczesnym społeczeństwie. Ale czego nie pamiętają nawet eksperci od komputerów to że -- tylko trochę przesadzam -- komputer został wynaleziony po to, by pomóc wyjaśnić pewną filozoficzną kwestię co do podstaw matematyki. Zaskakujące? Bynajmniej.

Ta niezwykła historia rozpoczyna się z Davidem Hilbertem, znanym niemieckim matematykiem, który na początku XX wieku zaproponował, by całkowicie sformalizować wszelkie rozumowanie matematyczne. Okazało się, że nie da się sformalizować rozumowania matematycznego, więc w pewnym sensie jego pomysł był wielką porażką. Ale w innym sensie pomysł Hilberta był sukcesem, bo formalizm jest jednym z największych darów XX wieku -- nie dla rozumowania matematycznego lub dedukcji -- lecz dla programowania, dla obliczania. Oto zapomniany fragment intelektualnej historii.

Opowiem tutaj tą historię bez zagłebiania się w matematyczne szczegóły. Będzie więc niemożliwe całkowicie wyjaśnić ważną pracę zasłużonych ludzi, takich jak Bertrand Russell, Kurt Gödel i Alan Turing. Wciąż, cierpliwy czytelnik powinien móc uchwycić esencję ich argumentów i dostrzec, co zainspirowało niektóre z moich własnych pomysłów co do losowości znajdującej się w matematyce.

Paradoksy Logiczne Russella

Pozwólcie mi zacząć z Bertrandem Russellem, matematykiem, który później stał się filozofem, a w końcu humanistą. Russell jest kluczowy, bo odkrył pewne denerwujące paradoksy w samej logice. Tzn. znalazł przypadki, w których rozumowanie wydające się poprawne prowadzi do sprzeczności. Russell miał ogromny udział w rozpowszechnieniu rozumienia, że te sprzeczności stanowią poważny kryzys, i że trzeba je jakoś rozwiązać.

Paradoksy odkryte przez Russella przyciągnęły ogromną uwagę w kręgach matematycznych, ale co dziwne tylko jeden został nazwany jego nazwiskiem.
By zrozumieć paradoks Russella, weźmy pod uwagę **zbiór wszystkich zbiorów, które nie są swoimi elementami**. Teraz zapytajmy: Czy ten zbiór jest swoim elementem? Jeśli jest swoim elementem, to nie powinien być i vice versa.

Zbiór wszystkich zbiorów w paradoksie Russella jest jak ten fryzjer w małym miasteczku, który strzyże wszystkich tych, którzy sami się nie strzygą. Opis ten wydaje się dość rozsądny, dopóki nie zapytacie: "Czy fryzjer ten strzyże się sam?". Strzyże się sam wtedy i tylko wtedy, gdy nie strzyże się sam. Możecie teraz powiedzieć: "Kogo obchodzi ten hipotetyczny fryzjer? To tylko głupia gra słów!". Ale jeśli zajmujesz się matematycznym pojęciem zbioru, nie jest tak łatwo odrzucić ten problem logiczny.

Paradoks Russella to teoriomnogościowe echo wcześniejszego paradoksu, znanego już starożytnym Grekom. Jest on często nazywany paradoksem Epimenidesa lub paradoksem kłamcy. Istota problemu jest taka: Epimenides podobno wykrzyknął: "To zdanie jest fałszywe!" Czy jest fałszywe? Jeśli to zdanie jest fałszywe, znaczy to, że musi być prawdziwe. Ale jeśli jest prawdziwe, to jest fałszywe. Więc cokolwiek założysz co do jego prawdziwości, jesteś w tarapatach. Dwuzdaniowa wersja tego pradoksu brzmi tak: "Następne zdanie jest prawdziwe. Poprzednie zdanie jest fałszywe.". Każde z tych zdań samo jest OK, ale połączone nie mają sensu. Możecie to odrzucać jako nic nie znaczące gry słowne, ale niektóre z wielkich umysłów XX wieku brały je bardzo poważnie.

Jedną z reakcji na kryzys w logice była próba Hilberta uczieczki w formalizm. Jeśli wchodzi się w tarapaty z rozumowaniem, które wydaje się OK, rozwiązaniem jest używać logiki symbolicznej do stworzenia sztucznego języka i być bardzo ostrożnym, by tak określić reguły, by nie wyskakiwały sprzeczności. Przecież język potoczny jest mętny -- nigdy nie wiesz, do czego odnosi się zaimek.

Plan ratunkowy Hilberta

Pomysł Hilberta polegał na stworzeniu doskonałego sztucznego języka do rozumowania, do uprawiania matematyki, do dedukcji. Podkreślił on więc znaczenie metody aksjomatycznej, w której pracuje się w zbiorze podstawowych postulatów (aksjomatów) i dobrze określonych reguł dedukcji do wyprowadzania prawdziwych twierdzeń. Ten sposób uprawiania matematyki idzie wstecz do starożytnych Greków, a konkretnie do Euklidesa i jego geometrii, która jest pięknie jasnym systemem matematycznym.

Innymi słowy intencją Hilberta było być zupełnie precyzyjnym co do reguł gry -- co do definicji, podstawowych pojęć, gramatyki języka -- tak, by każdy mógł się zgodzić co do tego, jak matematykę powinno się uprawiać.

W praktyce byłoby to zbyt pracochłonne używać takiego formalnego systemu aksjomatycznego do rozwijania nowej matematyki, ale miałoby to filozoficzne znaczenie.

Propozycja Hilberta wydawała się dość prostolinijna. W końcu on tylko szedł drogą wyznaczoną przez formalną tradycyjną matematykę, opierając się na długiej historii prac Leibniza, Boole'a, Fregego i Peano. Ale chciał pójść na całość do samego końca i sformalizować CAŁĄ matematykę. Wielką niespodzienką jest to, że nie da się tego zrobić.

Hilbert pomylił się -- ale pomylił się na bardzo owocny sposób, bo zadał bardzo dobre pytanie. Właściwie to zadając to pytanie, stworzył zupełnie nową dyscyplinę zwaną METAMATEMATYKĄ, introspektywną dziedzinę matematyki, w której bada się co matematyka może, a czego nie może osiągnąć.

Zasadnicza idea jest taka: Jak tylko zamkniesz matematykę w sztucznym języku a la Hilbert, jak tylko ustalisz całkowicie formalny system aksjomatyczny, to możesz zapomnieć, że ma to jakiekolwiek znaczenie, i po prostu patrzeć na to jako na grę rozgrywaną przy pomocy znaków na papierze, które pozwalają ci dedukować twierdzenia z aksjomatów. Oczywiście matematyką zajmuje się dlatego, że ma ona znaczenie. Ale jeśli chcesz badać matematykę przy użyciu metod matematycznych, musisz wykrystalizować znaczenie i po prostu badać sztuczny język z całkowicie ścisłymi regułami.

Jakiego rodzaju pytania można zadawać? Cóż, jedno z pytań to czy można udowodnić np. że 0=1. (Mamy nadzieję, że nie.) Rzeczywiście, dla każdego zdania, nazwijmy je A, można zapytać, czy jest możliwe udowodnić A lub zaprzeczenie A. Formalny system aksjomatyczny uważany jest za zupełny, jeśli można udowodnić, że A jest prawdziwe lub, że jest fałszywe.

Hilbert miał wizję stworzenia reguł tak ścisłych, że każdy dowód można by wysłać do niestronniczego sędziego, mechanicznej procedury, która powiedziałaby: "Ten dowód stosuje się do reguł", lub być może: "W 4. linijce jest błąd ortograficzny", lub "To w 4. linijce, co niby wynika z 3. linijki, tak naprawdę nie wynika." I to by był koniec; żadnej obrony.

Jego pomysł nie polegał na tym, że matematykę powinno się uprawiać w ten sposób, ale raczej że jeśli można by tak zrobić z matematyką, to można by wtedy użyć matematyki do badania mocy matematyki. A Hilbert myślał, że naprawdę potrafiłby tego dokonać. Możecie sobie więc wyobrazić jak bardzo, bardzo szokujące było, gdy w 1931 r. matematyk austriacki Kurt Gödel pokazał, że plan ratunkowy Hilberta wcale nie był rozsądny. Nigdy nie można by go zrealizować, nawet teoretycznie.

Niezupełność Gödlowska

Gödel rozsadził wizję Hilberta w 1931 r., będąc na wydziale Uniwersytetu Wiedeńskiego, choć przybył z dzisiejszej Republiki Czeskiej, z Brna. (Wtedy była to część imperium Austro-Węgierskiego.) Później Gödel przyłączył się do Einsteina w Instytucie Badań Zaawansowanych w Princeton.

Niezwykłym odkryciem Gödla było to, że Hilbert śmiertelnie się mylił: Tak naprawdę nie ma sposobu, by mieć formalny system aksjomatyczny dla całej matematyki, w którym byłoby kryształowo przejrzyste, czy coś jest prawidłowe czy nie. Dokładniej, Gödel odkrył, że plan nie udaje się już nawet dla elementarnej arytmetyki, z liczbami 0, 1, 2, 3,... i dodawaniem i mnożeniem.

Każdy system formalny, który próbuje zawrzeć w sobie całą prawdę i tylko prawdę o dodawaniu, mnożeniu i liczbach 0, 1, 2, 3,... musi być niezupełny. Właściwie to będzie albo sprzeczny albo niezupełny. Więc jeśli założysz, że mówi on tylko prawdę, to wtedy, nie powie on całej prawdy. W szczególności jeśli założysz, że aksjomaty i reguły dedukcji nie pozwalają na dowodzenie fałszywych twierdzeń, to będą twierdzenia, których nie da się udowodnić.

Dowód niezupełności Gödla jest bardzo sprytny. Jest bardzo paradoksalny. Wydaje się prawie zwariowany. Gödel zaczyna z paradoksem kłamcy: zdaniem "Jestem fałszywe!", które nie jest ani prawdziwe ani fałszywe. Tak naprawdę Gödel kostruuje zdanie, które mówi o sobie: "Nie da się mnie udowodnić!" Teraz jeśli możesz skonstruować takie zdanie w elementarnej teorii liczb, w arytmetyce, zdanie matematyczne, które samo się opisuje, musisz być bardzo sprytny -- ale jeśli MOŻESZ to zrobić, łatwo zobaczyć, dlaczego jesteś w tarapatach. Dlaczego? Bo jeśli to zdanie można udowodnić, jest ono koniecznie fałszywe, i dowodzisz fałszywego wyniku. Jeśli nie da się go udowodnić, jak samo o sobie mówi, to jest ono prawdziewe, i matematyka jest niezupełna.

Dowód Gödla zawiera wiele skomplikowanych technicznych szczegółów. Ale jeśli popatrzycie na jego oryginalną pracę, znajdziecie coś, co bardzo przypomina programowanie w LISP. Jest tak, bo dowód Gödla zawiera definiowanie wielu funkcji rekurencyjnie, funkcji zajmujących się listami -- dokładnie tym, czego właśnie dotyczy LISP. Więc chociaż nie było żadnych komputerów ani języków programowania w 1931 r., z pomocą spojrzenia wstecz, można wyraźnie zobaczyć język programowania w istocie oryginalnej pracy Gödla.

Inny sławny matematyk tamtej ery, John von Neumann (który, zbiegiem okoliczności, miał istotny wpływ na stworzenie technologii komputerowej w Stanach Zjednoczonych), docenił wgląd Gödla natychmiastowo. Nigdy nie dotarło do von Neumanna, że plan Hilberta był nierozsądny.

Więc Gödel był nie tylko niezwykle sprytny, miał też odwagę wyobrazić sobie, że Hilbert może się mylić.

Wielu ludzi postrzegało wniosek Gödla jako absolutnie niszczycielski: Cała tradycyjna filozofia matematyki skończyła jako kupa na podłodze. Jednak w 1931 r. było także kilka innych zmartwień w Europie. Miał miejsce wielki kryzys i wojna się szykowała.

Maszyna Turinga

Następny wielki krok naprzód nadszedł pięć lat później, w Anglii, gdy Alan Turing odkrył nieobliczalność. Przypomnijcie sobie, że Hilbert powiedział, że powinna istnieć "mechaniczna procedura" decydująca, czy dowód stosuje się do reguł czy nie. Hilbert nigdy nie sprecyzował, co ma na myśli, mówiąc o mechanicznej procedurze. Turing zasadniczo powiedział: "O co ci naprawdę chodzi, to o maszynę" (maszynę, którą teraz nazywamy maszyną Turinga).

Oryginalna praca Turinga zawiera język programowania, tak jak praca Gödla, lub coś, co teraz nazwalibyśmy językiem programowania. Ale te dwa języki programowania są bardzo różne. Turinga nie jest takim wysokopoziomowym językiem jak LISP; jest to bardziej jak język maszynowy, surowy kod jedynek i zer przezyłanych do głównego procesora komputera. Wynalazek Turinga z 1936 r. jest tak naprawdę okropnym językiem maszynowym, którego nikt nie użyłby dzisiaj, bo jest zbyt szczątkowy.

Choć hipotetyczne maszyny liczące Turinga są bardzo proste, a ich język maszynowy dość prymitywny, są bardzo elastyczne. W jego pracy z 1936 r. Turing twierdzi, że taka maszyna może przeprowadzić dowolne obliczenia, jakie człowiek jest w stanie zrobić.

To myślenia Turinga bierze teraz dramatyczny obrót. Co, pyta, jest NIEMOŻLIWE dla takiej maszyny? Czego nie może zrobić?I od razu znajduje problem, którego żadna maszyna Turinga nie potrafi rozwiązać: problem stopu. Jest to problem polegający na stwierdzeniu z góry, czy dana maszyna Turinga (lub program komputerowy) znajdzie poszukiwane rozwiązanie i zatrzyma się.

Jeśli dopuścić ograniczenie czasowe, bardzo łatwo jest rozwiązać ten problem. Powiedzmy, że chcesz wiedzieć, czy program zatrzyma się w rok. Wtedy po prostu uruchamiasz go na rok i albo zatrzymuje się albo nie. Co Turing pokazał, to że wpadasz w straszne tarapaty, jeśli nie określisz limitu czasowego, jeśli próbujesz wydedukować, czy program się zatrzyma bez uruchamiania go.

Pozwólcie mi zarysować rozumowanie Turinga: Przypuśćmy, że MOŻESZ

napisać program, który sprawdza, czy dowolny dany program komputerowy w końcu się zatrzyma. Nazwijmy go testerem zakończenie. Teoretycznie możnaby dać mu program, a on wyplułby odpowiedź: "Tak, ten program się zakończy" lub "Nie, będzie się kręcił na swoich kołach w jakiejś nieskończonej pętli i nigdy się nie zatrzyma".
Teraz stwórzmy drugi program, który używa testera zakończenia, by ewaluować pewien program. Jeśli badany program się zakończy, niech ten nowy program będzie tak zrobiony, że wchodzi w nieskończoną pętlę. A teraz subtelna część: Daj swojemu nowemu programowi kopię siebie. Co robi?

Pamiętaj, napisałeś ten nowy program tak, że wejdzie w nieskończoną pętlę, jeśli testowany program się zakończy. Ale tutaj on SAM jest tym testowanym programem. Więc jeśli się zakończy, to wchodzi w nieskończoną pętlę, czyli się nie zakończy -- sprzeczność. Przeciwne założenie nie pomaga: Jeśli się nie zakończy, tester zakończenia wskaże na to i program nie wejdzie w nieskończoną pętlę, więc zakończy się. Ten paradoks doprowadził Turinga do wniosku, że ogólnego testera zakończenia nie da się zrobić.

Interesujące jest to, że Turing od razu wyciągnął wniosek: Jeśli nie ma sposobu, by określić z góry jakimś obliczeniem, czy program się zatrzyma czy nie, to nie może się tego też dać stwierdzić przy pomocy rozumowania. Żaden formalny system aksjomatyczny nie pozwoli ci wydedukować, czy program się w końcu zatrzyma. Dlaczego? Bo gdybyś mógł użyć formalnego systemu aksjomatycznego w ten sposób, to dałoby ci to sposób na obliczenie z góry, czy program się zatrzyma czy nie. A to jest niemożliwe, bo dochodzisz do paradoksu jak: "To zdanie jest fałszywe!" Możesz stworzyć program, który zatrzymuje się wtedy i tylko wtedy, gdy nie zatrzymuje się. Paradoks ten jest podobny do tego, który Gödel odkrył w swoich badaniach nad teorią liczb. (Przypomnijmy, że patrzył na nic bardziej skomplikowanego, niż 0, 1, 2, 3,... i dodawanie i mnożenie.) Wyczyn Turinga polegał na tym, że pokazał on, że ŻADEN formalny system aksjomatyczny nie może być zupełny.

Po rozpoczęciu II Wojny Światowej Turing zajął się krytpografią, von Neumann zaczął pracować nad tym, jak obliczać wybuchy bomb atomowych, i ludzie zapomnieli o niezupełności formalnych systemów aksjomatycznych na jakiś czas.

Losowość w Matematyce

Pokolenie matematyków, którzy zajmowali się tymi głębokimi filozoficznymi kwestiami zasadniczo zniknęło wraz z II Wojną Światową. Wtedy ja pokazałem się na scenie.

W późnych latach 50-tych, gdy byłem młody, przeczytałem artykuł o Gödlu i niezupełności w Scientific American. Wynik Gödla fascynował mnie, ale nie potrafiłem go naprawdę zrozumieć; myślałem, że coś mi tu śmierdzi. Co do podejścia Turinga, podobało mi się, że wchodziło ono znacznie głębiej, ale wciąż nie byłem usatysfakcjonowany.
To wtedy miałem śmieszny pomysł dotyczący losowości.

Gdy byłem dzieckiem, czytałem także wiele o innym słynnym zagadnieniu intelektualnym, nie podstawach matematyki, ale podstawach fizyki -- o teori względności i kosmologii, a nawet częściej o mechanice kwantowej.
Dowiedziałem się, że gdy rzeczy stają się bardzo małe, świat fizyczny zachowuje się w zupełnie zwariowany sposób. Tak naprawdę rzeczy są losowe -- z natury nieprzewidywalne. Czytałem o tym wszystkim i zacząłem się zastanawiać, czy istnieje

też losowość w czystej matematyce.
Zacząłem podejrzewać, że może to jest prawdziwa przyczyna niezupełności.

Dobry przypadek to elementarna teoria liczb, w której istnieją bardzo trudne pytania. Weźmy pod uwagę liczby pierwsze. Poszczególne liczby pierwsze zachowują się bardzo nieprzewidywalnie, jeśli interesuje was ich dokładna struktura.
To prawda, że istnieją statystyczne prawidłowości. Istnieją takie coś jak twierdzenie o liczbach pierwszych, które dość dokładnie przewiduje ogólne rozłożenie liczb pierwszych. Ale jeśli chodzi o dokładne rozmieszczenie poszczególnych liczb pierwszych, wygląda to dość losowo.

Zacząłem więc myśleć, że losowość obecna w matematyce stanowi głębszą przyczynę całej tej niezupełności. W połowie lat 60-tych ja, i niezależnie A. N. Kołmogorow w ZSRR, wymyśliliśmy coś nowego, co lubię nazywać algorytmiczną teorią informacji. Brzmi to bardzo imponująco, ale zasadnicza idea jest bardzo prosta: Jest to po prostu sposób na mierzenia złożoności obliczeniowej.

Jednym z pierwszych miejsc, z których dowiedziałem się o złożoności obliczeniowej, to od von Neumanna. Turing rozważał komputer jako pojęcie matematyczne -- doskonały komputer, który nigdy nie popełnia błędów, kóry ma tak wiele czasu miejsca, jak tylko potrzebuje, żeby pracować. Po tym, jak Turing wpadł na ten pomysł, następnym logicznym krokiem było badanie czasu potrzebnego do przeprowadzenia obliczeń -- miary ich złożoności. Około roku 1950 von Neumann podkreślił znaczenie złożoności czasowej obliczeń, i jest to teraz dobrze rozwinięta dziedzina.

Mój pomysł polegał na tym, żeby nie patrzeć na czas, choć z praktycznego punktu widzenia czas jest bardzo ważny. Mój pomysł był taki, żeby patrzeć na ROZMIAR programów komputerowych, na ilość informacji, jaką trzeba dać komputerowi, by wykonał pewne zadanie. Dlaczego jest to interesujące? Bo złożoność rozmiarowa programów łączy się z pojęciem entropii w fizyce.

Przypomnijmy, że entropia grała szczególnie istotną rolę w pracy słynnego fizyka Ludwiga Boltzmanna i pojawia się w mechanice statystycznej i termodynamice. Entropia mierzy stopień nieporządku, chaos, losowości w systemie fizycznym. Kryształ ma małą entropię, a gaz (powiedzmy w temperaturze pokojowej) ma dużą entropię.

Entropia jest powiązana z fundamentalnym pytaniem filozoficznym: Dlaczego czas biegnie tylko w jednym kierunku? W życiu codziennym istnieje oczywiście wielka różnica pomiędzy poruszaniem się do przodu i do tyłu w czasie. Szklanki tłuką się, ale nie sklejają się nagle na nowo. Podobnie w teorii Boltzmanna entropia musi wzrastać -- system musi stawać się bardziej i bardziej nieuporządkowany. Oto znane Drugie Prawo Termodynamiki.

Współcześni Boltzmannowi nie widzieli, jak wywnioskować ten wynik z fizyki Newtonowskiej. Mimo wszystko w gazie, gdzie atomy odbijają się jak kule bilardowe, każda interakcja jest odrwacalna. Gdybyś mógł jakoś nagrać małą część gazu przez krótki czas, nie potrafiłbyś odróżnić, czy film jest puszczony do przodu czy do tyłu. Ale teoria gazu Boltzmanna mówi, że ISTNIEJE strzałka czasu -- system rozpocznie w uporządkowanym stanie, a skończy w bardzo pomieszanym nieuporządkowanym stanie. Jest nawet takie straszne określenie na ten ostateczny stan: "śmierć cieplna".

Powiązanie pomiędzy moimi pomysłami a teorią Boltzmanna pojawia się dlatego, że rozmiar programu komputerowego jest analogiczny do stopnia nieuporządkowania systemu fizycznego. Gaz może wymagać dużego programu, by stwierdzić, gdzie znajdują się jego atomy, podczas gdy kryształ nie wymaga dużego programu z powodu swojej regularnej struktury. Entropia i złożoność rozmiarowa programów są więc blisko powiązane.

To pojęcie złożoności rozmiarowej jest także powiązane z filozofią metody naukowej. Ray Solomonoff (komputerowiec pracujący wtedy w Zator Company w Cambridge, Massachusetts) zaproponował ten pomysł na konferencji w 1960 r., choć ja się dowiedziałem o jego pracy dopiero po tym, jak wpadłem na podobny pomysły kilka lat później.

Pomyślcie tylko o brzytwie Occama, idei, że najprostsza teoria jest najlepsza. Cóż, czym jest teoria? To program komputerowy do przewidywania obserwacji. A stwierdzenie, że najprostsza teoria jest najlepsza, tłumaczy się na stwierdzenie, że krótki program komputerowy stanowi najlepszą teorię.

Co jeśli nie ma najbardziej zwięzłej teorii? Co jeśli najbardziej zwięzły program do odtworzenia danego zbioru danych eksperymentalnych jest tego samego rozmiaru jak te dane? Wtedy teoria jest niedobra -- jest ugotowana -- i dane są niekompresowalne, losowe.

Teoria jest dobra tylko wtedy, gdy kompresuje dane do znacznie mniejszego zbioru teoretycznych założeń i reguł wnioskowania.

Można więc zdefiniować losowość jako coś, czego w ogóle nie można skompresować. Jedyny sposób, by opisać zupełnie losowy obiekt lub liczbę komuś, to pokazać go i powiedzieć: "To jest to". Ponieważ nie ma żadnej struktury lub prawidłowości, nie ma żadnego krótszego opisu. Drugą skrajnością jest obiekt lub liczba, który ma bardzo regularną strukturę. Być może można by go opisać, mówiąc, że jest to milion powtórzeń 01, na przykład. Jest to bardzo duży obiekt o bardzo krótkim opisie.

Mój pomysł był taki, żeby użyć złożoności rozmiarowo-programowej do zdefiniowania losowości. A gdy zaczniesz przyglądać się rozmiarowi programów komputerowych -- kiedy zaczniesz myśleć o pojęciu złożoności rozmiarowej lub informacyjnej zamiast złożoności czasowej -- wtedy coś ciekawego się dzieje: Gdziekolwiek się nie zwrócisz, znajdziesz niezupełność. Dlaczego? Bo pierwsze pytanie, jakie zadajesz w mojej teorii, prowadzi cię w kłopoty. Mierzysz złożoność czegoś poprzez rozmiar najmniejszego programu obliczającego to coś. Ale jak możesz być pewnym, że twój program to najmniejszy możliwy? Odpowiedź jest taka, że nie możesz. To zadanie, dość zadziwiająco, przekracza możliwości rozumowania matematycznego. Pokazanie, że tak jest, wymaga trochę zaangażowania, więc zacytuję tylko sam wynik, który jest jednym z moich ulubionych twierdzeń o niezupełności: Jeśli masz n bitów aksjomatów, nigdy nie udowodnisz, że program jest najmniejszym możliwym, jeśli ma on więcej niż n bitów długości. To jest, wchodzisz w tarapaty z programem, jeśli jest on większy od skomputeryzowanej wersji aksjomatów -- lub dokładniej, jeśli ma większy rozmiar od programu sprawdzającego dowody dla aksjomatów i powiązanych z nimi regułami wnioskowania.

Więc okazuje się, że nie można w ogólnym przypadku obliczyć złożoności rozmiarowo-programowej, bo określić złożoność rozmiarowo-programową czegoś oznacza znać rozmiar najmniejszego programu, który to coś oblicza. Nie można tego zrobić, jeśli program jest większy od aksjomatów.

Pozwólcie, że wyjaśnię, dlaczego tak twierdzę. Zbiory aksjomatów, których matematycy zazwyczaj używają, są dość zwięzłe, w przeciwnym wypadku nikt by w nie wierzył. W praktyce istnieje ten ogromny świat prawdy matematycznej -- nieskończona ilość informacji -- ale dowolny zbiór aksjomatów wyłapuje tylko malutką, skończoną ilość informacji.
Oto w zasadzie dlaczego Gödlowska niezupełność jest naturalna i nieunikniona, raczej niż tajemnicza i skomplikowana.

Dokąd Teraz?

Wniosek ten jest bardzo dramatyczny. W tylko trzech krokach idziemy najpierw od Gödla, u którego wydawało się szokujące, że istnieją granice rozumu, do Turinga, u którego wydaje się to znacznie bardziej rozsądne, aż do rozważania złożoności rozmiarowo-programowej, w której niezupełność, granice matematyki, po prostu uderzają w twarz.

Ludzie często mi mówią: "Cóż, to wszystko bardzo fajne. Algorytmiczna teoria informacji to niezła teoria, ale daj mi przykład czegoś konkretnego, co uważasz, że przekracza rozumowanie matematyczne". Przez lata jedną z moich ulubionych odpowiedzi było: "Może wielkie twierdzenie Fermata". Ale coś śmiesznego się stało: W 1993 r. Andrew Wiles nadszedł z dowodem. Był błąd, ale teraz wszyscy są przekonani, że dowód jest poprawny. Więc mamy problem.
Algorytmiczna teoria informacji pokazuje, że istnieje wiele rzeczy, których nie da się udowodnić, ale nie potrafi tego rozstrzygnąć dla konkretnych problemów matematycznych.

Jak więc, pomimo niezupełności, matematycy robią tak duże postępy? Te wyniki niezupełności z pewnością mają w sobie coś pesymistycznego. Jeśli zmierzyć się z nimi wprost, wydawałoby się, że nie da się robić postępów, że matematyka jest niemożliwa. Na szczęście dla tych z nas, którzy zajmują się matematyką, nie wydaje się tak być. Być może jacyś młodzi matematycy następnego pokolenia udowodnią, dlaczego tak być musi.