

17

Jak umysł rozmawia ze światem w kodzie binarnym

Rozdział autorstwa Witolda Marciszewskiego z książki: Witold Marciszewski i Paweł Stacewicz, „Umysł – Komputer – Świat. O zagadce umysłu z informatycznego punktu widzenia”, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2011.

Wprowadzenie. Oto jedno z doświadczeń rozmowy umysłu z fragmentem świata, jakim jest system informacyjny zmaterializowany w pewnych obiektach fizycznych. Jest to doświadczenie komunikacyjne zachodzące po wielokroć każdego dnia w życiu obywateli społeczności informatycznej.

Gdy stoję przed bankomatem dla pobrania gotówki, rysunki cyfr na klawiszach i wyświetlane instrukcje wysyłają do moich oczu falę impulsów świetlnych, które system percepcyjny przetwarza na ciągi impulsów elektrycznych w postaci binarnej. Te, przetworzone z kolei w mózgu na impulsy idące do palców, sprawiają, że naciskam kolejno cyfry komunikujące bankowi mój kod identyfikacyjny. Naciśnięcie klawisza uruchamia mechanizm elektryczny, który wysyła do serwera w banku taki ciąg zer i jedynek, jaki w kodzie binarnym wyraża daną liczbę, np. 101 gdy wybiorę liczbę pięć. Ciągami zer i jedynek wyraża się także kwota, którą życzę sobie otrzymać. Gdy bankomat nada instrukcję „wyjmij kartę”, wyświetla to literami dla mojej wygody, ale w jego języku są to wciąż ciągi zer i jedynek w takim kodzie, który każdej literze alfabetu przyporządkowuje określoną liczbę.

Od czego zależy sukces takiej komunikacji? Od tego, jak długie ciągi zer i jedynek zdoła pomieścić nośnik pamięci oraz jak szybko ciąg wyjściowy, powiedzmy, cyfry identyfikatora i poleceń, zostanie przetwarzany na ciąg wyjściowy, jak polecenie dla mechanizmu podającego banknoty. Gdyby miało to trwać godzinami, bankomaty nie miałyby racji bytu; i tak, wielkie usprawnienie gospodarki i naszej codzienności nie doszłoby do skutku.

To zaś, czy nośniki pamięci pomieszczą tyle danych, ile trzeba, i czy dane będą przetwarzane z konieczną w danym działaniu prędkością, zależy od pracy fizyków i matematyków oraz inżynierów stosujących wyniki obu tych nauk – fizyki w zakresie sprzętu oraz matematyki w zakresie oprogramowania czyli algorytmów. I tak, mamy wniosek światopoglądowy, że przyszłość świata, z gwałtownie narastającą złożonością jego problemów, zależy od tego, czy skali

tej złożoności podolają systemy przetwarzania informacji. Wymagają one bowiem coraz pojemniejszych nośników pamięci, coraz szybszych procesorów i coraz bardziej wyrafinowanych, a więc i bardziej złożonych, algorytmów.

Ten punkt światopoglądu informatycznego powinni trwale zapisać w swych głowach i sercach nauczyciele i publicyści odpowiedzialni za edukację społeczeństwa oraz politycy odpowiedzialni za politykę naukową i oświatową. Tylko ten kraj bowiem ma szansę na sukces w wyścigu cywilizacyjnym, którego gospodarka, nauka i oświata nadążą za wciąż rosnącymi wymogami informatyzacji.

Tak więc, temat kodowania zajmuje poczesne miejsce w światopoglądzie informatycznym i zasługuje, żeby poświęcić mu w tych esejach niemało uwagi. Wypełnia on tekst obecny i w różnych aspektach jest kontynuowany w następnych esejach. Wychodzimy (§1) od myśli Galileusza, że zjawiska przyrody odsłaniają się nam w swych prawidłowościach dopiero wtedy, gdy je opiszemy językiem matematyki. Jego podobieństwo do kodu rysuje się wtedy, gdy spojrzymy na dane doświadczenia zmysłowego jak na język złożony ze znaków mających za przedmiot pewne elementy rzeczywistości. Właśnie te znaki są kodowane w języku matematyki. Język kodujący może sam stać się kodowanym, i tak się dzieje np. wtedy, gdy wszystkie wyrażenia matematyczne zapisujemy w kodzie binarnym. Tym dwóm krokom poświęcone są dwa kolejne ustępy pierwszego odcinka. O kodzie binarnym mówi się w §1 tyle tylko, ile konieczne, by go wkomponować w Galileuszową wizję świata. Jego dokładniejsze omówienie w §3 jest poprzedzone przygotowaniem natury terminologicznej w §2. Odcinek §3, poświęcony relacji kodów binarnego i analogowego stanowi zarazem wprowadzenie do następnego eseju, mającego za przedmiot dualizm cyfrowo-analogowy.

§1. Porozumienie umysłu ze światem przez kod matematyczny

1.1. W poprzednim eseju oddaliśmy sprawiedliwość zasługom atomizmu, doceniając jego inspirację w myśleniu o złożoności świata. Ale z samą tą inspiracją nie daleko byśmy zaszli, gdyby u progu nowożytnej nauki nie przemówiła wielkim głosem myśl Galileusza (1564-1642). Myśl zamknięta w takiej oto sentencji:

§ *Księga natury jest pisana w języku matematycznym. I w tej drugiej: matematyka jest alfabetem, w którym Bóg opisał wszechświat.* Tak więc,
§ Natura dostarcza nam wiedzy o sobie samej pod warunkiem, że rozszyfrujemy jej kod, a jest to kod matematyczny.

Nauka nowożytna to niezmierny dziś ocean wiedzy, a myśl i dorobek Galileusza, w połączeniu z pionierskim dziełem Kopernika, jest jak ujście doń rzeki toczącej swe wody wcześniej przez dwa tysiąclecia. Jeśli zapytać, skąd ten nurt wypływa, gdzie i jakie są jego źródła, to nasza rzeczna metafora nadal się sprawdza. Okazuje się bowiem, że jest to tak, jak w przyrodzie. Dające się rozpoznać źródło stanowi uchwytny tej rzeki początek, ale on z kolei ma swe początki w sączących się niewidzialnie, pochowanych w murawie strumyczkach, bez których by nie zaistniało nasze oznaczone na mapie źródło.

Rozpoznawalne źródło tego nurtu, który wieńczy myśl Galileusza stanowi wizja Pitagorasa (ok. 570-495 p.n.e.) wyrażona w jego słynnej sentencji: *Liczba jest zasadą, źródłem i korzeniem wszystkich rzeczy*.¹ Nie zależy jednak zapominać, że wielka idea nie bierze się z niczego. Pracują na nią pokolenia anonimowych umysłów, aż ów dorobek złożony z mikroskopijnych części utworzy masę krytyczną, z której się wyłoni odnotowane przez historię wielkie odkrycie.

W naszej historii nie mały mają udział kupcy fenicy. Wszak dynamikę cywilizacji w pokaźnej mierze tworzy pęd do zysku (to myśl z Adama Smitha, oświeceniowego pioniera teorii cywilizacji). Obrotni Fenicjanie zarabiający kupiectwem gdzie tylko się dało w śródziemnomorskim rojowisku ludów, potrzebowali jakiegoś sposobu, żeby wyrażać, porównywać i kalkulować swe krociowe zyski. Wynaleźli więc cyfry, za czym się kryła idea liczby, choć akurat na to drugie kupcy mniej zwracali uwagi. Ale idea powstała i zaprzętała ona usposobionych bardziej metafizycznie kapłanów Babilonu i Egiptu. Do nich po studia i wtajemniczenie udał się młody Grek Pitagoras, a po powrocie do Krotony w Wielkiej Grecji stworzył szkołę, w której idee matematyczne zrosły się w nierozdzielny całość z wizją świata jako rządzonego liczbą. Stały się więc światopoglądem, co czyni Pitagorasa protoplastą światopoglądu informatycznego.

Inną wielką postacią w prehistorii informatyzmu był Leibniz (1646-1716) ze swą słynną maksymą: *cum Deus calculat fit mundus* – gdy Bóg rachuje, staje się świat. Przemawia ona i dziś do niektórych osób, nie stroniących od orientacji teologicznej. Inni natomiast wolą wersję bardziej świecką, którą można oddać słowami „gdy świat rachuje, staje się świat”. Pojawia się ona u autorów, którzy uważają wszechświat za gigantyczny komputer obliczający swą własną ewolucję według programu, jakim są prawa fizyki.

Skoro wszechświat jest taki na wskroś matematyczny, to nie tylko on do nas przemawia językiem liczb, ale też odbiera ów język, gdy w nim ślemy nasze do niego sygnały. Zachodzi jakby porozumiewanie się dwukierunkowe. W drugim kierunku również obowiązuje kod matematyczny. O tyle jednak nie ma tu symetrii, że świat przemawia w formie zdań oznajmujących, in-

¹ Przytoczone wg przekładu angielskiego: *Number is the principle, the source and the root of all things* w książce: Kenneth Sylvan Guthrie and David Fideler, *The Pythagorean Sourcebook*, s.21, cytowanej na stronie www.skyscript.co.uk/pythagoras.html.

formując nas, że jest taki a taki. Od nas natomiast nie oczekuje informacji, jacy jesteśmy (to on nas wytworzył i nie jesteśmy dlań zagadką). Jeśli przemawiamy do świata, to po to, żeby go przekształcać stosownie do naszych potrzeb; można porównać takie komunikaty do tych, jakie program komputerowy śle do swego fizycznego otoczenia, np. drukarki. A konieczność uzyskania wpływu na nasze otoczenie we wszechświecie może być nagląca. Nie życzymy sobie np. zderzenia naszej planety z kolosalną asteroidą, toteż gdy zacznie się do nas zbliżać, trzeba będzie wysłać precyzyjnie obliczone sygnały, które by ją odwiodły z trajektorii tak dla naszych losów katastrofalnej.

Inżynieria kosmiczna, inżynieria genetyczna, inżynieria mózgu – to działy nauki i techniki, które zmieniają świat fizyczny, operując zrozumiałym dlań kodem formułowanych matematycznie praw przyrody. Uwzględniając należycie ów kod, ślemy do świata sygnały polecające zmianę jakiegoś stanu dotychczasowego na stan pożądany; sygnały takie, będąc sterowane przez komputer, funkcjonują w kodzie cyfrowym.

Tak więc, w światopoglądzie informatycznym kluczowa jest idea sprawczej roli kodu cyfrowego. Obecne zaś pokolenie jest pierwszym w dziejach, które na każdym kroku, po wielokroć każdego dnia, korzysta z owej sprawczej siły. Gdy płacę kartą bankową lub legitymuję się nią w bankomacie, gdy aktywizuję telefon PIN-em lub wybieram klawiszami numer adresata, gdy loguję się do sieci podając swą nazwę i hasło itd., za każdym razem posługuję się kodem cyfrowym. Taka codzienna praktyka podnosi ogólny poziom świadomości informatycznej. Sprzyja to przyswojeniu sobie światopoglądu informatycznego, ale żeby od tej świadomości dojść do światopoglądu, trzeba wykonać ileś kroków, i to niebanalnych.

1.2. Powiązać kodowanie zerojedynekowe ze światopoglądem, to zadanie na ileś godzin wykładowych lub ileś rozdziałów tekstu, a wobec niedostępności tych zasobów, posłużę się skrótem biograficznym – z życia Leibniza. Jest on unikalnym pionierem i prekursorem światopoglądu informatycznego. A mógł się stać kimś takim tylko dlatego, że był wynalazcą arytmetycznej notacji binarnej czyli dwójkowej. Rozmyślał nawet o jej zastosowaniach w maszynach liczących, a że nic jeszcze nie wiedziano o elektryczności, próbował projektować binarne urządzenia mechaniczne, co nie mogło mieć praktycznie przyszłości. Ale jego fascynacja binarnością pomoże uchwycić związek między światopoglądem i kodem binarnym.

Co jak co, ale sprawa stworzenia świata należy niezaprzeczalnie do kwestii światopoglądowych. Leibniz doszedł w tej sprawie do przekonania, że wszechświat można stworzyć z samych zer i jedynek. Był z tego odkrycia tak dumny, że planował upamiętnić je medalem na cześć własną jako odkrywcy, gdzie byłyby dwa napisy. Jeden miał brzmieć: [matematyczny] *model stworzenia świata odkryty przez G.W.L.* [inicjały Leibniza]. Drugi napis wyjaśniałby istotę odkrycia: *liczba jeden wystarcza, żeby z NIC wyprowadzić WSZYSTKO*; liczby zero nie wymienił osobno, uważając, że mieści się już ona w pojęciu

„nie”. Kiedy indziej wyraził tę myśl żartobliwym nieco rysunkiem, na którym Stwórca jako materiałem do stworzenia świata dysponuje wyłącznie sterą zer i jedynek.

Jest to myśl zaskakująco podobna do tej, którą współczesny fizyk John Wheeler wyraził formułą *It from Bit*, gdzie „It” oznacza coś, co jest, czyli rzeczywistość, a „Bit” symbolizuje informację kodowaną zerojedynekowo. Ten skrajnie zwięzły aforyzm szeroko się wśród fizyków i filozofów upowszechnił, i tak idea Leibniza żyje nowym życiem jako fundament światopoglądowy, a pełniejsze jeszcze rozwinięcie, nawiązujące do uniwersalnej maszyny Turinga, znajdziemy dziś u fizyka Davida Deutscha – pioniera i guru obliczeń kwantowych (por. \triangleright esej 18. §4.1).

Nim zapuścimy się głębiej w tajniki kodu binarnego, trzeba usunąć pewną przeszkodę terminologiczną. Przyjęło się używać zamiennie terminów „kod cyfrowy” i „kod binarny” (inaczej, „dwójkowy”). Jest to zrozumiałe praktycznie, bo najczęściej mamy do czynienia z kodem cyfrowym, który jest binarny; w tym sensie mówimy o cyfrowej platformie multimedialnej. Ale pojęciowo jest to nie w porządku, bo istnieją kody cyfrowe inne niż binarny, choćby stosowany w informatyce kod szesnastkowy. Ma też rację bytu stary dobry kod dziesiętny. Oczywiście, są to wszystko kody cyfrowe, a binarny jest jednym z nich. Będę więc w obecnym kontekście przestrzegał tego, że pojęcie kodu cyfrowego jest nadrzędne względem binarnego, a rozłączne względem kodów stosujących inne symbole, nie cyfrowe lecz np. literowe.

§2. Rodzaje kodów i notacji

2.1. Termin „kod”, zrazu techniczny, gdy przeniknął do języka potocznego, nabrał, by tak rzec, nie najlepszych manier: chwiejności znaczeniowej, która czyni, że słowo raz oznacza to, raz tamto, od kontekstu do kontekstu. Nie trzeba za to mowy potocznej ganić, ma bowiem ta sytuacja również dobrą stronę, mianowicie elastyczność i ekonomiczność. Gdy jeden i ten sam wyraz obsługuje różne potrzeby, zasób językowy się nie rozrasta do jakichś kolosalnych rozmiarów, co musiałoby nastąpić, gdyby zamiast jednego wieloznacznego słowa mieć wiele jednoznacznych. W §1 posługiwałem się słowem „kod” trochę jak scyzorykiem, raz korzystając z jednego raz z innego z kilku odcieni znaczeniowych. Pora jednak rzecz uporządkować, czyniąc z tego terminu narzędzie bardziej wyspecjalizowane, lepiej wyostrzone.

Pomocna w tym będzie pewna definicja zaczerpnięta ze źródła należycie autorytatywnego (przycaczam ją też w oryginalnym brzmieniu niemieckim, żeby zainteresowanym umożliwić ocenę trafności proponowanej dalej interpretacji). Dobrze się ona nadaje na punkt wyjścia, który w miarę potrzeby będzie modyfikowany pod kątem obecnych rozważań. Oto jej tekst.

Code ist eine Vorschrift zum Umsetzen der Zeichen einer Menge A in Zeichen (Codezeichen) oder Zeichenfolgen (Codewörter) einer Menge B. Diese Abbildung muss eindeutig und umkehrbar sein. *dtv-Atlas zur Informatik* 1995.

Kod jest to reguła (przepis, instrukcja) zastępowania znaków z jakiegoś zbioru A znakami lub ich ciągami z jakiegoś zbioru B. To odwzorowanie musi być jednoznaczne i odwracalne.

Powyższa definicja posługuje się dobrze zdefiniowanymi w logice pojęciami zbioru, ciągu i odwzorowania. Pomimo tych zalet, nada się do celu tych rozważań dopiero po pewnej adaptacji. Mianowicie, po takim rozszerzeniu pojęcia kodu, żeby jego zakresem objąć również sytuacje, w których zbiór znaków nie jest odwzorowywany w inny zbiór znaków, lecz w zbiór zdarzeń składających się na pewien proces. Żeby odróżnić te dwa przypadki – odwzorowania ciągu znaków na inny ciąg znaków, oraz odwzorowania ciągu znaków na proces pozajęzykowy – pierwszy przypadek określam jako *kod syntaktyczny*, a drugi jako *kod sterujący*.

To drugie określenie jest pewnym skrótem, bo tym co bezpośrednio steruje procesem jest program utworzony ze znaków wziętych do tego celu z repertuaru znaków z danego języka kodowania. Np. w kodzie binarnym ów repertuar (zbiór znaków) składa się z symboli „0” i „1”, w których formułowane są programy (ciągi poleceń) w języku wewnętrznym maszyny. To program steruje działaniem maszyny, ale że czyni to za pomocą określonych znaków kodowych, zawierający je kod, jako przewidziany do tego zadania, zaproponowałem, dla celów tych rozważań, nazywać kodem sterującym.

W pewnych kontekstach dogodnie będzie ów przepis odwzorowania nazywać *kluczem kodowym*. Nawiązuje to do powszechnego w literaturze sposobu mówienia. Znajdujemy go np. w takich kontekstach, jak komunikat: „w 1932 roku trzem polskim matematykom udało się złamać klucz kodowy Enigmy”. Tym co rozpoznano („złamano”) był przepis zamieniania tekstu niemieckiego na szyfr czyli ciągu liter niezrozumiałe dla tego, kto nie zna przepisu zamiany czyli klucza kodowania. Odczytanie szyfrowanego tekstu wymaga odkrycia zastosowanego w danym przypadku klucza, co właśnie udawało się wywiadowcom alianckim w akcji „Enigma”.

Skupimy się w tych rozważaniach na kodach sterujących. Do tej bowiem kategorii należy kod binarny rozważany tu, w szczególności, ze względu na to, co wiedza o nim wnosi do światopoglądu informatycznego. Trzeba jednak dla pełności obrazu wspomnieć o funkcjonowaniu kodu syntaktycznego. Oto przykład. Niech ciągami znaków z dwóch zbiorów będą:

A: MARCISZEWSKI

B: nbsdjtafxtij

Co jest w tym przypadku kluczem kodowym? Jest to następujący przepis czyli, inaczej, reguła: „Zastąp każdą literę ze zbioru dużych liter (A) przez literę ze

zbioru małych liter (B) następującą w alfabecie po tej zastępowanej, a w przypadku ostatniej litery alfabetu zastąp ją pierwszą.” I tak dostajemy zmienione nie do poznania nazwisko autora tego eseju. Jak widać, przyporządkowanie elementów w obu sekwencjach jest jednoznaczne i odwracalne: znając kod, potrafimy z wiersza pierwszego uzyskać drugi, jak i odwrotnie. Cała ta operacja zachodzi w obrębie dwóch zbiorów znaków i ustala między nimi odwzorowanie wzajemnie jednoznaczne. A że relacje międzyznakowe określamy mianem syntaktycznych, tłumaczy to. Dlaczego właściwe jest w tym przypadku określenie „kod syntaktyczny”.

Ale nazywamy kodami, i to w sensie technicznym, a nie jakimś metaforycznym, również takie układy, w których mamy nie dwa lecz tylko jeden zbiór znaków. Takimi są wymienione wyżej kody: maszynowy, neuronowy, genetyczny. Są one tej natury, że gdy A jest zbiorem znaków, to B jest zbiorem zdarzeń powodowanych przez znaki; ciągi takich zdarzeń z B są odwzorowaniami ciągów znaków z A. Co do fizycznej natury znaków, to nie koniecznie muszą to być nieruchome znaki na papierze. Mogą to być znaki dynamiczne, jak nadawane przez kogoś czy przez coś sygnały; są one impulsami powodującymi jakieś reakcje.

Dla tego rodzaju kodu zaproponowałem wyżej określenie „kod sterujący” (z grecka można by powiedzieć – cybernetyczny). Sprawdza się ono w odniesieniu do różnych odmian kodu: maszynowy steruje procesami w komputerach, neuronowy procesami w niektórych organizmach, a genetyczny procesami we wszystkich organizmach.

2.2. Pojęcie kodu tak rozszerzone, że nie zamyka się w podanej wyżej definicji z *Atlasu*, zdolne zatem objąć kody sterujące, dobrze się sprawdza w odniesieniu do kodu maszynowego czyli wewnętrznego języka komputera. Wyobraźmy sobie taką oto scenkę z czasów, gdy nie było jeszcze systemów operacyjnych, kompilatorów, aplikacji itd., a programowanie odbywało się za pomocą ręcznego sterowania kablami. Konstruktor maszyny sprawił np. tak, że seria impulsów 1010 stanowi polecenie „dodaj”. Operator, mając przed sobą gniazdko z czterema wejściami, programuje dodawanie w ten sposób, że wtyka kabel w pierwsze i trzecie wejście, ale nie w drugie i czwarte, a konstrukcja maszyny sprawia, że ta konfiguracja impulsów aktywizuje drukarkę do pobierania danych. Reakcja drukarki jest drugim członem stosunku odwzorowania, którego pierwszym członem jest konfiguracja impulsów niesionych przez kable. Gdzie w tej opowieści mamy ciąg znaków, jakiego wymaga definicja kodu przyjęta tu w punkcie wyjścia?

Łatwo wskazać, że ciągiem znaków kodu sterującego jest tu konfiguracja kabli utworzona przez operatora. A dlaczego mówimy o znakach? Jest to trochę tak, jak w powiedzeniu „na znak dowódcy oddział dał ognia”. Żeby podniesienie ręki było sygnałem do takiej akcji, trzeba było uprzednio relację tę ustanowić, co czyni regulamin walki. Żeby konfiguracja kabli 1010 była znakiem aktywizującym drukarkę, trzeba było tę relację ustanowić, co

uczynił konstruktor danej maszyny. Nazwiemy przeto *kodelem maszynowym* kompletny zbiór instrukcji, które – pod kątem pewnego planu funkcjonowania – wprowadzają odwzorowanie między każdą z tych instrukcji i wywołanym przez nie stanem maszyny.

Żeby pod pojęcie kodu móc podciągnąć sekwencje sygnałów sterujących zaplanowanym ciągiem zdarzeń, trzeba było to pojęcie zliberalizować (w stosunku do definicji z *Atlasu*) na dwa naraz sposoby: (1) dopuścić, by w relacji odwzorowania wystąpił tylko jeden, a nie konieczne dwa, ciąg znaków, podczas gdy drugim członem relacji jest ciąg zdarzeń nie będących znakami, (2) dopuścić pojmowanie znaków jako sygnałów sterujących sekwencją zdarzeń według pewnego ustalonego planu.

2.3. Pojęcie planu wprowadza w nasze rozważania odniesienie do celu: planuje się ciąg kroków, żeby osiągnąć określony cel. Czy sprawdza się to w odniesieniu do kodu genetycznego? Jeśli nie, to trzeba uznać, że terminu „kod” używamy tu metaforycznie, albo jeszcze bardziej pojęcie kodu poluzować. Przeanalizujmy następującą definicję (z wikipedii).

Kod genetyczny jest to reguła, według której informacja genetyczna, zawarta w sekwencji nukleotydów kwasu nukleinowego (DNA lub RNA), w komórkach wszystkich organizmów może ulegać „tłumaczeniu” na kolejność (sekwencję) aminokwasów w ich białkach (w procesie biosyntezy białek, czyli translacji).

Początek zgadza się z naszą wyjściową definicją w §2.1 o tyle, że mowa jest o pewnej regule. Zamiast o znakach mówi się o informacji, ale te dwa sposoby mówienia można uznać za równoważne. Mamy też jednoznaczne odwzorowanie, mianowicie sekwencji (tj. ciągu) nukleotydów w sekwencji aminokwasów, co nazywa się translacją czyli przekładem i tym przypomina kodowanie znaków z jednego zbioru przez znaki z innego zbioru. Są tu więc niewątpliwe analogie natury strukturalnej.

Pozostaje jednak pytanie, na jakiej zasadzie można by nukleotydy nazwać znakami. Czego byłyby one znakami? Zgodnie z pojęciem kodu sterującego, do roli znaków kandydują elementy sterujące procesem, a więc nukleotydy, do roli zaś obiektów przez nie oznaczanych i sterowanych – aminokwasy. Podobnie jak w przypadku kodu maszynowego, to rozszerzenie pojęcia znaku da się obronić, o ile cecha bycia znakiem zaistniała w wyniku jakiegoś planu, który skutkuje realizowaniem danego zbioru instrukcji czyli wykonywaniem pewnego programu.

Wolno więc powiedzieć, że reguła odwzorowująca konfigurację nukleotydów na konfigurację aminokwasów jest kodelem (z kategorii sterujących), jeśli ta pierwsza stanowi zapis jakiegoś programu. Czy tak jest istotnie? Wraz z tym pytaniem okazuje się, że nasze próby uściślenia pojęcia kodu prowadzą na próg problematyki światopoglądowej. Jeśli się zgodzić, że istotnie mamy tu do czynienia z programem, to narzuca się pytanie, skąd on się wziął. Wiemy

w przypadku programów tworzonych w kodzie maszynowym, że czynnikiem sprawczym jest twórczość programisty. Programistę tworzącego w kodzie genetycznym trudno wypatrzyć. Ale jeśli istotnie funkcjonuje tu pewien program, to z jego pochodzenia trzeba jakoś zdać sprawę. Skąd się on bierze? Różnie będą na to odpowiadać ewolucjoniści, materialiści, teiści, panteiści itd., ale mimo ostrych nawet różnic będą oni członkami tej samej wspólnoty: tych, co uznają obiektywne istnienie informacji i za kluczowe uważają kwestie dotyczące jej wzajemnych oddziaływań z materią i z energią, a więc podzielają światopogląd informatyczny.

§3. Sukces kodu binarnego, jego relacja do analogowego

3.1. Pora omówić kod binarny bardziej systematycznie. Prowadzą do tego dwa kroki. Z jednej strony trzeba dokładniej przyjrzeć się funkcjonowaniu kodów sterujących, z drugiej zaś – zasadom pozycyjnej notacji arytmetycznej.

Jako możliwie najprostsza ilustracja *kodu sterującego* dobrze się nadaje pozytywka muzyczna. Zdejmijmy z niej pokrywę, żeby zobaczyć, co znajduje się w środku i jak to działa. Następnie porównamy jej funkcjonowanie z grą żywego muzyka korzystającego z kodu partytury czyli z notacji muzycznej.

Pozytywka wydaje dźwięki w wyniku zaczepiania metalowymi wypustkami, umieszczonymi na obrotowym wałku, o rozmieszczone odpowiednio struny. Układ strun przypomina harfę, a wypustki są ekwiwalentem uderzających w struny palców muzyka. Pozytywka działa pod pewnym względem na zasadzie analogowej. Wprawdzie odbiega od niej tym, że wykonywany proces nie jest ciągły, lecz skwantowany (dyskretny), ale podobna jest zasada jednoznaczego odwzorowania zjawiska mechanicznego w zjawisko akustyczne. Ruch wałka i wchodzące w grę kształty, a więc jakości mechaniczne, odwzorowują się w jakościach akustycznych. Mianowicie, trzy parametry dźwiękowe wykonywanego utworu są dokonującym się, w miarę obrotów wałka, odwzorowaniem trzech jakości mechanicznych: (1) wysokość dźwięku odpowiada długości struny, (2) długość dźwięku odpowiada wielkości trącej ją wypustki, (3) to zaś w jakiej kolejności poruszana jest taka czy inna struna jest odpowiednikiem rozmieszczenia wypustek na wałku. Mamy tu więc do czynienia ze sprawstwem czysto fizycznym, z którego – dzięki konstrukcji mechanizmu – rodzi się powyższe odwzorowanie.

Spełniony jest więc warunek odwzorowania wymieniony w definicji kodu. Sterowanym procesem jest wykonanie utworu muzycznego, podczas gdy składającymi się nań zdarzeniami są kolejno wytwarzane dźwięki. A co stanowi w pozytywce ów sterujący ciąg znaków? Nie są to znaki konwencjonalne, czyli wprowadzone umownie, jak litery alfabetu, cyfry, czy ciągi jednych i drugich. Są to obiekty fizyczne zdolne na zasadzie fizycznej przyczynowości sterować procesem. Takim obiektem jest w pozytywce konfiguracja wypustek

z ich określonymi wielkościami i rozmieszczeniem na wałku. To ona steruje układem wydawanych dźwięków; przy innej konfiguracji byłby to inny układ, a stąd – inny utwór muzyczny.

Czy można określić tę fizyczną konfigurację jako ciąg znaków, co byłoby warunkiem koniecznym, żeby uznać, że coś ona koduje? Układ fizyczny wytwarzający swe odwzorowanie w innym układzie fizycznym może być ciąg znaków, o ile został skonstruowany według jakiegoś planu. Za planem bowiem kryje się jakaś myśl jego autora komunikowana przez dzieło. A co jest środkiem komunikacji, zasługuje na miano znaku. Gdy nakręcam pozytywkę i docierają do mnie jej dźwięki, realizuje się intencja zakomunikowania utworu każdemu, kto zechce go posłuchać, intencja tak kompozytora utworu, jak producenta pozytywki.

3.2. Porównując komunikat nadawany przez pozytywkę z tym, który zawarty jest w partyturze odtwarzanego przez nią utworu, dokonujemy pewnej analizy jednego i drugiego kodu, która dobrze naprowadza na uchwycenie swoistości kodu binarnego. Partyturę różni od pozytywki to przede wszystkim, że jest zapisana w pewnej notacji.

Notacja jest językiem graficznym, który nie jest tak wszechstronny ani powstający tak żywiołowo, jak język naturalny. Jest to język wąsko wyspecjalizowany do określonego celu, w jakim został umyślnie stworzony. W każdym z tych punktów partytura muzyczna wzorowo spełnia pojęcie notacji.

Czy mieści się ona w kategorii kodów syntaktycznych? Tak, ponieważ zbiór symboli składających się na notację muzyczną stanowi jednoznaczne odwzorowanie pewnego zbioru terminów technicznych z teorii muzyki. Między tymi zbiorami wyrażen zachodzi wzajemne odwzorowanie. Gdy weźmiemy pod uwagę takie jakości muzyczne, jak wysokość dźwięku, jego głośność i czas trwania, to każdy poziom na skali którejkolwiek z tych jakości określamy zarówno przez pewien termin z teorii muzyki, jak i przez odpowiadający mu symbol z notacji muzycznej zapisywany na pięciolinii. Np. nuta zapisana między drugą a trzecią (od dołu) linią, zapisana w ciągu symboli poprzedzonym przez rysunek zwany kluczem wiolinowym oznacza ten sam dźwięk, który w języku teorii muzyki ma nazwę „*a* razkreślne”, zaś w akustyce jest definiowany jako dźwięk o częstotliwości 440 Hz. Taka reguła przekładu należy do klucza kodowego koordynującego język teorii muzyki z językiem zapisu nutowego.

Mamy tu do czynienia z kodem syntaktycznym, istnieje bowiem klucz kodowy, który w sposób jednoznaczny przyporządkowuje ciągi wyrażen z teorii muzyki i z zapisu nutowego. Pojawia się pewien kłopot, gdy trzeba określić, co jest językiem oryginału, a co językiem kodowania. Inaczej niż, powiedzmy, w sytuacji szyfrowania komunikatów przez wywiady, nie jest tak, że najpierw mamy język zastany (jak niemiecki dla szyfrantów Enigmy), w którym się formułuje tekst oryginalny, a następnie powstaje język kodowy, na który się

ten oryginał przekłada. W przypadku muzyki oba języki rozwijały się mniej więcej równolegle w czasie i oba są podobnie wymagające wtajemniczenia.

Dopiero, gdy podejmiemy do zagadnienia z pojęciem kodu sterującego, a przy tym skorzystamy z modelowego przypadku pozytywki, rzecz się rozjaśnia. Do pewnego stopnia konfiguracja znaków w partyturze jest odpowiednikiem konfiguracji wypustek na walcu pozytywki. Jedna i druga steruje procesem wykonywania określonego utworu. Zachodzi zarazem ta ważna różnica, że konfiguracja elementów pozytywki sama jedna wystarcza do wytworzenia danego utworu, byleby dostarczyć energii do obrotów (co czyni sprężyna), podczas gdy zapis nutowy jest tylko częścią kodu sterującego. Drugą część stanowi wiedza muzyka, o której można domniemywać, że jest zapisana w jakimś kodzie neuronowym. Dzięki niej partytura jako program wykonania utworu jest interpretowana przez program neuronowy, który impulsami słanymi do mięśni steruje procesem wykonywanym przez układ fizyczny złożony z rąk muzyka i z instrumentu. Partytura okazuje się więc kodem sterującym.

Zmierzając do rozważań nad kodem binarnym, trzeba zauważyć to podobieństwo, że notacja muzyczna na równi z kodem binarnym należy do kodów cyfrowych. Opisuje ona wielkości mierzalne: każdej z nich jest przyporządkowana określona liczba, jak częstotliwość dźwięku, jego trwanie, odległości czasowe między dźwiękami, rytm itd. Jest więc partytura, w swej strukturze głębokiej, zapisem ciągu liczb czyli cyfrowym, ale że byłby on dla wykonawcy nieczytelny, wprowadzono system jakby skrótów definicyjnych w stylu ikonycznym, co umożliwiła szybką percepcję.

Np. umieszczenie znaku nutowego na określonej wysokości w pięciolinii, wraz z rodzajem otwierającego daną pięciolinię klucza, wskazuje na wysokość dźwięku czyli częstotliwość fali akustycznej. Wyrażenia uformowane z tego alfabetu wskazują na sposób wytworzenia sygnału mającego wchodzić w skład utworu, np. na to, w który uderzyć klawisz lub którą strunę poruszyć smyczkiem. Są to więc procesy fizyczne. Ten przeto kod, w połączeniu z energią impulsów emitowanych przez osobę muzyka, jest sprawcą zdarzeń fizycznych, jakimi są wytwarzane dźwięki. To zatem, że jest ona kodem syntaktycznym ze względu na odwzorowanie w słownictwie teorii muzycznej, nie przeszkadza jej być zarazem kodem sterującym w stosunku do wykonywanego wedle niej utworu.

Co się tyczy kodu binarnego w komputerze, należy on, podobnie jak partytura, do cyfrowych kodów sterujących, ale z gruntu inny jest alfabet i składnia. Alfabet składa się z dwóch tylko symboli, zera i jedynki, składnia jest taka, jak w pozycyjnym zapisie arytmetycznym, impulsy są elektryczne, a powodowane przez nie zdarzenia są to stany procesora i pamięci. Efektywność kodu binarnego, gdy go zastosować w aparaturze elektronicznej, okazuje się rewelacyjna. To zaś stwarza perspektywę światopoglądową, w której cywilizacja przyszłości, uzbrojona w zawrotne moce obliczeniowe, jawi się jako potężna moc stwórcza. Przyjrzyjmy się, skąd się bierze taka skala efektywności.

3.3. Trzy są źródła sukcesu kodu binarnego. Jedno z nich to bycie notacją arytmetyczną pozycyjną, co stwarza możliwość tworzenia dowolnie wielu dowolnie długich ciągów cyfr, a więc niczym nie ograniczone bogactwo konfiguracji sygnałów. Drugim są techniki arytmetyzacji języka, także naturalnego, co sprawia, że opisy procesów (np. obrazy i dźwięki tworzące akcję filmu) oraz polecenia sterujące procesami (np. polecenie „skopiuj film na mój dysk”) dają się wyrażać w notacji złożonej wyłącznie z cyfr. Trzecim źródłem jest uprzywielejoniana więź łącząca notację binarną z fizyką, mianowicie fakt że alfabetowi arytmetycznemu, jako złożonemu z dwóch tylko symboli, da się jednoznacznie przyporządkować dwa rodzaje impulsów fizycznych zdolnych sterować procesami, jak impuls (elektryczny, świetlny, mechaniczny etc.) oraz jego brak. Dzięki temu jest to język obdarzony sprawczością fizyczną.

§ Wynalazek *pozycyjnego zapisu liczb*, mający za podstawę pojęcie zera, należy do fundamentalnych zdobyczy cywilizacji. W jego narodzinach miały udział trzy wielkie kultury średniowiecza: hinduska, arabska i europejska. Początek dał jakiś Hindus, który dostrzegł coś, co umknęło wcześniej uwadze mądrych Greków i Rzymian: że istnieje liczba całkowita mniejsza od jeden, mianowicie NIC (przedmiotów); zasługuje więc ona na własny oznaczający ją symbol.

Więść o istnieniu zera, transportowana szlakiem handlowym łączącym Indie z Bliskim Wschodem, dotarła przez kupców do arabskiego Bagdadu, gdzie w dziewiątym wieku kwitł ośrodek nauk, szczególnie matematyki. Pracował w nim wielki Al-Chwarizmi; to od jego imienia pochodzi łacińskie „algorithmus”, zaś „algebra” to skrót arabskiego tytułu jednego z jego dzieł. Drugim wielkim ośrodkiem nauki i postępu technicznego w świecie arabskim była Kordoba, gdzie szybko przyswojono sobie nową zdobycz.

Islam był naonczas w Kordobie tak tolerancyjny, że pozwalano Chrześcijanom spokojnie tam studiować, z czego skorzystał pewien kleryk, późniejszy papież Sylwester drugi, wszechstronnie jak na owe czasy oświecony (m.in. patronował tworzeniu się Polski za Chrobrego jako forpoczty chrześcijańskiej Europy na wschodzie). Tę dobrą nowinę o istnieniu zera i jego roli dla zapisu liczb Sylwester obwieścił Europie, a choć od tego oznajmienia do wejścia w praktykę arytmetyczną upłynęło sporo czasu, początek został zrobiony. Później, już w Europie, wprowadzono zero w innych niż dziesiętna notacjach, zastosowano w liczeniu na ułamkach dziesiętnych, powiązано je z algebrą i z teorią zbiorów.

Żeby zdać sobie sprawę z kolosalnej roli symboliki pozycyjnej, a w tym cyfry zero dla teorii i praktyki obliczeń, trzeba porównać tę notację z jej starszą siostrą – notacją rzymską. W notacji dziesiętnej cały słownik do zapisywania liczb ma tylko dziesięć symboli cyfrowych i za ich pomocą można utworzyć nazwę dowolnie wielkiej liczby; obojętnie, ile miliardów znaków musi mieć zapis jakiejś liczby, nigdy nie zabraknie dlań sposobów zapisu. Rzymianom zasób

zaczynał się wyczerpywać już przy liczbie tysiąc (litera „M”); bo ile można wymyślać nowych liter na kolejne rzędy wielkości? Ponadto, notacja rzymska cierpi na dotkliwy brak reguł, które określałyby sposób tworzenia oznaczeń dla coraz to większych liczb. Przydzielanie tego zadania wymyślanym ad hoc literom, kojarzącym się już to z wielkością oznaczanego zbioru (jak trzy pierwsze cyfry), już to z łacińską danej liczby nazwą (jak np. „C” od „centum”) czyni ten język nieznośnie przypadkowym, mało stosownym do opisywania świata arytmetyki, który jest sam w sobie tak doskonale uporządkowany. Potrafi natomiast temu sprostać notacja pozycyjna.

3.4. Pozycyjność polega na tym, że pozycja cyfry w ciągu symboli wskazuje, czego (tj. jakiego zbioru) jest to liczba: pojedyncze „5” oznacza pięć jednostki. Ten sam symbol na pierwszej pozycji w ciągu dwucyfrowym („50”, „51 etc.) oznacza pięć dziesiątek; na pierwszej pozycji w ciągu trzycyfrowym – pięć setek itd., podczas gdy zero oznacza brak tego rodzaju zbiorów. Zapewnia tę zaletę możliwość wypełniania następujących dalej pozycji zerami. Nigdy więc nie zabraknie cyfr do reprezentowania dowolnie wielkich liczb. Tym większą bowiem liczbę dana cyfra potrafi reprezentować, w im dłuższy się ją wyposażą orszak ciągnących się za nią cyfr. A że taki ciąg da się wydłużać w nieskończoność, nigdy nie zabraknie środków językowych dla określania coraz to większych liczb. Ta cnota systemu pozycyjnego błyszczy w pełni, gdy trzeba operować nieskończonymi ciągami liczb całkowitych; jesteśmy spokojni, że istnieje określenie dla dowolnej, jakiej byśmy tylko potrzebowali, liczby w tym ciągu. Na gruncie informatyki pojawiła się taka potrzeba np. wtedy, gdy dla celów dowodu, że istnieją liczby nieobliczalne, czyli że istnieją problemy nierozstrzygalne dla komputera, trzeba było uprzednio ponumerować wszystkie elementy nieskończonego zbioru maszyn Turinga.

Systemów pozycyjnych możemy tworzyć, ile dusza zapagnie, ale nie każdy służyłby równie dobrze naszym potrzebom rachunkowym. System taki powstaje w wyniku decyzji, po której liczbie ma się zaczynać drugi rząd wielkości; drugi, to znaczy taki, w którym wielkość, czyli liczbę, oddajemy już nie jednym lecz dwoma symbolami cyfrowymi. W systemie dziesiętnym (który preferujemy jako istoty dziesięciopalcowe) zaczyna się on po liczbie 9. I tak, pierwszy rząd wielkości obejmuje dziesięć liczb jednocyfrowych (od 0 do 9), poczem zaczyna się drugi, inicjowany liczbą 10 jako pierwszą wśród mających oznaczenia dwucyfrowe. Kończy się on liczbą 99; po niej następuje rząd trzeci, inicjowany najmniejszą liczbą oznaczaną trzycyfrowo. Te zasady tworzenia rzędów wielkości znamy od przedszkola. Warto jednak je odświeżyć jako pomocne w charakteryzowaniu *kodu binarnego*, zwanego też *kodem dwójkowym*.

Dla wyjaśnienia kodu binarnego dobrze jest posłużyć się porównaniem z czymś tak swojskim, jak system dziesiętny. W dziesiętnym określają pozycję kolejne potęgi liczby 10, w binarnym – kolejne potęgi liczby 2. Popatrzmy na system dziesiętny: Tym, od czego się w nim zaczyna pierwszy rząd wielkości

jest liczba 1, drugi zaczyna się od 10, następny od 100 itd. Liczby te, zauważmy, są kolejnymi potęgami liczby 10. I tak:

$$1 = 10^0, 10 = 10^1, 100 = 10^2, 1000 = 10^3, 10000 = 10^4...$$

Jak widać, wykładnik potęgowy możemy rozpoznać po liczbie zer następujących po jedynce. Zauważmy też, kiedy się zwiększa liczba zer: następuje to wtedy, gdy wyczerpuje się możliwość oznaczania następnych liczb w ramach dotychczasowego rzędu wielkości. Ponieważ w systemie dziesiętnym brakuje nam już pojedynczej cyfry, która oznaczałaby liczbę następującą po 9, problem ten rozwiązujemy przechodząc do następnego rzędu wielkości czyli napisem „10”. Gdy wraz z liczbą 99 wyczerpuje się możliwość zapisywania dwucyfrowo liczb od niej większych, przychodzi pora na ciąg trzycyfrowy – jedynkę z dwoma zerami, co oznacza liczbę 10^2 .

Dokładnie ten sam schemat uzyskiwania kolejnych rzędów wielkości stosujemy w kodzie binarnym. Dysponując tylko cyframi „0” i „1”, już dla oddania liczby dwa musimy przejść do następnego rzędu wielkości, a więc wykorzystać zestawienie „10”. Liczbę trzy oznaczmy jako większą o jeden od 10, co da napis „11”, a że jest to największa liczba wyrażalna dwucyfrowo, jej następnik (cztery) trzeba nazwać trzycyfrowo napisem „100”. Dodawszy jeden (by oznaczyć pięć), mamy nazwę „101”, potem „110” (sześć), potem „111” (siedem). I tak dochodzimy do największej liczby dającej się oznaczyć trzy cyfrowo, stąd jej następnik znajdzie się w nowym, czterocyfrowym, rzędzie, jako desygnat nazwy „1000”. Umieścimy to w ramach sekwencji kolejnych potęg liczby dwa, w której znowu (jak w każdym systemie pozycyjnym) liczba zer po jedynce wskazuje na wykładnik potęgowy podstawy systemu.

„1” oznacza jeden czyli 2^0
 „10” oznacza dwa czyli 2^1
 „100” oznacza cztery czyli 2^2
 „1000” oznacza osiem czyli 2^3
 „10000” oznacza szesnaście czyli 2^4
 „100000” oznacza trzydzieści dwa czyli 2^5
 itd.

Pozycyjna notacja arytmetyczna nabiera cech kodu, gdy posługujemy się arytmetyką w celu przełożenia na jej język wyrażeń innego języka. Ponieważ ten inny, z którego tłumaczymy, jest powszechnie zrozumiały, a ten nowy wprowadzamy umownie i trzeba się go dopiero uczyć, sytuacja jest trochę taka, jak w szyfrowaniu depech wojennych czy szpiegowskich. Tylko trochę, bo nie ma tu intencji utajnienia, jak w walce wywiadów, ale o tyle podobna, o ile język zastany i zrozumiały zastępujemy tworzonym w pewnym celu kodem. Tego rodzaju zabieg nazywamy *arytmetyzacją języka*, a gdy chodzi o jakieś poszczególne w tym języku wyrażenia, mówimy o arytmetyzacji zapisu. Aryt-

metyzacja języka należy do kluczowych w informatyce i logice procedur, toteż zasługuje na osobną uwagę, którą jej poświęcimy w eseju 19, §4.

3.5. Jeśliby problematykę kodowania przedstawić w postaci drzewa, którego rozgałęzienia prowadzą do kluczowych w tej problematyce zagadnień, to jeden konar reprezentowałby procedurę arytmetyzacji, a drugi kwestię kodowania analogowego. Ta druga jest nie mniej ważna, lecz bardziej, by tak rzec, delikatna, stąd wymaga osobnego rozważenia.

Kwestia ta została poruszona wyżej w §3.1, gdzie mowa jest o pozytywce, że jest ona mechanizmem o cechach analogowości, dzielając z urządzeniami analogowymi to, że funkcjonuje na zasadzie odwzorowania rządzonego prawami fizyki. Strzegłem się w tamtym kontekście terminu „kod analogowy”, ponieważ trzeba się nieco potrudzić nad uzgodnieniem takiego określenia z definicją kodu jako rodzaju instrukcji (▷ wyżej §2.1).

Może nie każdy czytelnik tekstów naukowych jest w pełni świadom, że terminologia specjalistyczna to nie zawsze jest coś takiego, jak budowla mocno osadzona na fundamencie i trwająca w wyznaczonym jej miejscu, lecz bywa też czymś na kształt statku zmieniającego swe położenie stosownie do różnych okoliczności. Największa stabilność panuje może w fizyce, ale daleko jest do niej w zagadnieniach dotyczących języka i umysłu. Posuwając o krok dalej metaforę statku, można powiedzieć, że jest to żaglowiec, który musi się liczyć z kierunkiem i siłą wiatru. A wiatry w tej sferze bywają zmienne i kapryśne. Inaczej się w niej porusza ten, kto poddaje się np. wiatrom behawioryzmu, a inaczej ktoś zorientowany na logiczne podstawy informatyki. Inaczej zdeklarowany wyznawca materializmu, a inaczej ktoś mający rezerwę do tego sposobu myślenia.

Przypowieść o wiatrach służy wyjaśnieniu, dlaczego w propozycji, którą tu wysuwam nie mogę się powołać na ogólnie akceptowany stan nauki ani na ustaloną definitywnie terminologię, lecz staję przed wyborem, z którego kierunku brać wiatr w żagle. Wybór dotyczy tego, czy mówić o kodowaniu analogowym zachodzącym według praw przyrody (pełniących rolę jakby klucza kodowego), czy poniechać takiej praktyki, trzymając się wąskiego pojęcia kodowania jako operacji czysto syntaktycznej (takiej np. jak arytmetyzacja języka).

Ponieważ śledzimy tu podobieństwa i różnice między komputerem a układem umysł/mózg, trzeba się zainteresować, jakim językiem przemawia w tej materii nauka o mózgu. Autorem szczególnie kompetentnym w tej materii jest John Eccles (1903-1997) jako laureat nagrody Nobla w dziedzinie fizjologii i medycyny za odkrycie mechanizmów jonowych pobudzenia i hamowania na zewnątrz i wewnątrz błony komórkowej neuronów. Chodzi więc o procesy pokrewne zachodzącym w komputerze z powodu tej samej elektrycznej natury. Eccles zajmował się w szczególności percepcją wzrokową, zwłaszcza postrzeganiem rysów twarzy, a więc procesami poznawczymi, co

do których toczy się żywa debata na temat szans zrównania się inteligencji maszynowej z ludzką.

Warto sięgnąć w tej kwestii do książki napisanej przez Ecclesa wspólnie z Karlem Popperem (1902-1994), jednym z najwybitniejszych myślicieli 20-go wieku. Popper wywarł nieprzeciętny wpływ na obecny kształt metodologii i filozofii nauki, liberalnej filozofii społecznej, ontologii, epistemologii, a także matematyki (teoria prawdopodobieństwa) i fizyki (teoria kwantów, indeterminizm). Książka nosi tytuł *The Self and Its Brain. An Argument for Interactionism* (Routledge, London 1977), tytuł z rozmysłem prowokujący do polemiki. Wbrew bowiem redukcjonistycznej tendencji do utożsamienia umysłu (*self*) z mózgiem, Eccles i Popper ujmują umysł i mózg jako dwa odrębne układy będące ze sobą w interakcji. Prawami tej interakcji autorzy wyjaśniają analizowane przez nich pod kątem poznawczym zjawiska psychologiczne i fizjologiczne.

W tym kontekście pojawia się pojęcie kodowania i dekodowania w procesach postrzegania wzrokowego. Autorzy używają tych terminów w odniesieniu do procesów typowo analogowych. Kluczy kodowych, czyli zasad odwzorowania dostarczają wtedy nie konwencje językowe lecz prawa fizyki i fizjologii. Oko koduje na siatkówce impulsy świetlne analogowo w ten sposób, że je odwzorowuje w reakcjach chemicznych; te są z kolei kodowane jako impulsy elektryczne idące do ośrodków mózgowych, a tam, tworząc niewyobrażalnie złożone konfiguracje, podlegają one dekodowaniu przez umysł (*mind*). Niech przykład języka, którym autorzy opisują te procesy, pogłębi nasz wgląd w ideę kodowania analogowego.

There is the projected image on the retina, but for transmission and handling it all has to be made into a mosaic arrangement coded in firing frequencies of the milion or so optic nerve fibres [s. 479].

Żeby uzmysłwić, na czym polegają tego rodzaju procesy, autorzy porównują je do tego, co zachodzi w analogowym odbiorniku telewizyjnym.

Odnotowując tu i przejmując na własny użytek to pojęcie kodu, które obejmuje zarówno kodowanie symboliczne rządzone jakimś umownym kluczem, jak i kodowanie analogowe, rządzone kluczem praw przyrody, dalece to pojęcie rozszerzamy. Nie jest to jednak jakaś anomalia lecz raczej proces typowy dla języka naukowego, który jest elastycznie przystosowany do modyfikacji drogą uogólnień czy abstrakcji.

Rozbudowanie w tym kierunku aparatury pojęciowej ma żywotne znaczenie dla zagadnień z kręgu umysł-komputer-świat. Uzyskujemy możliwość opisów rozwiązywania problemów, a więc pewnego przetwarzania informacji, czynionego w jednolitym języku. Możemy teraz swobodnie porównywać, co do różnic i podobieństw, rozwiązywanie problemów przez maszynę cyfrową i przez mózg, czyniąc to w kategoriach kodu cyfrowego i kodu analogowego. Możemy też stawiać pytania co do sposobu interakcji tych kodów (gdy taka za-

chodzi) w sieciach neuronowych. Wyjaśnia się też to, co dla pewnych badaczy, np. behawiorystów, wydawało się zagadkowe, mianowicie: jak są możliwe rozumowania pozawerbalne? Odpowiadamy: są możliwe na tych samych prawach, co kod analogowy. Zachodzą one w tym kodzie, a nie w symbolicznym. To znaczy, świat odwzorowuje się w takich rozumowaniach bezpośrednio, a nie za pośrednictwem symboli.

Dalej zaś, gdy przejdziemy do zagadnień mocy obliczeniowej, będzie już wiadomo, że mamy na myśli moc, która cechuje nie tylko urządzenia cyfrowe lecz także analogowe. Bez ustaleń terminologicznych podjętych w obecnym eseju, język niezbędny w naszych rozważaniach byłby nie dość giętki. Pora obecnie na podjęcie kolejnego zagadnienia, które podobnie jak kwestia kodów jest dla informatyzmu zasadnicze – do problemu mocy obliczeniowej w powiązaniu z dualizmem: *cyfrowe versus analogowe*.

