



ISSN 1897 0311

WIOSNA
2007

O NAS

KONTAKT

SPIS TREŚCI

ARCHIWUM

ANALIZA TEZY KOMPUTACJONIZMU

STRONA

1

2

3

4

5

6

Marcin Miłkowski

1. Co głosi komputacjonizm?

Zwykle uznaje się, że **komputacjonizm*** głosi, iż procesy umysłowe mają charakter obliczeniowy, czyli algorytmiczny, przy czym chodzi zwykle o swoście zawężoną klasę procesów umysłowych: o procesy poznawcze (np. Shapiro [1995], Rapaport [1998]). Istnieją wprawdzie inne interpretacje tej tezy (np. Bringsjord [1998]), według których komputacjonizm głosi też, że komputacjoniści wytworzą sztuczne podmioty poznające. Takie rozumienie jednak uznaję za niecharakterystyczne i będę zajmować się tezą komputacjonizmu w sensie klasycznym, będącą założeniem roboczym przyjmowanym w naukach empirycznych noszących nazwę **kognitywistyki** czy **nauk o poznaniu**.

Warto od razu odróżnić dwie tezy komputacjonizmu klasycznego:

(a) Procesy poznawcze są obliczeniowe.

(b) Procesy umysłowe są obliczeniowe.

Można uznawać tylko tezę (a), odrzucając jednocześnie tezę (b), gdy sądzi się, że istnieją procesy umysłowe P nie będące poznawczymi, i że procesy P nie są obliczeniowe (algorytmiczne). Narzucającym się intuicyjnie przykładem takich procesów są reakcje emocjonalne, nastroje itd., w których trudno wyróżnić choćby aspekt poznawczy. Zdarzają się jednak wypowiedzi przedstawicieli kognitywistyki, którzy uważają, że procesy emocjonalne również mają naturę obliczeniową. Pamiętajmy więc o słabszej tezie (a) i o jej mocniejszym wariantcie (b).

Cóż jednak oznacza twierdzenie, że jakieś procesy - umysłowe czy poznawcze - są obliczeniowe? Zamiennie ze słowem **obliczeniowe**

używa się też słowa **algorytmiczne**, co sugeruje, że są to synonimy. Należy więc dokonać analizy sensu wyrażenia **algorytmiczny**, orzekanego o procesach umysłowych (czy wężziej: o procesach poznawczych). W tym celu odwołam się do przedstawionej gdzie indziej (Miłkowski [2003]) próby wyróżnienia odpowiednio dwóch sensów „algorytmiczności”; przy czym jeden z rodzajów algorytmiczności wiąże się z trzema rodzajami relacji między abstrakcyjnymi ciągami algorytmicznymi a konkretnymi procesami umysłowymi.

Gdyby w świecie w ogóle nie występowały procesy algorytmiczne, wówczas teza komputacjonizmu byłaby fałszywa. Jeśli wszystkie procesy są algorytmiczne, to teza ta byłaby po prostu banalna (albo nie dotyczyłaby prostej algorytmiczności wszystkich procesów fizycznych). Interesująca, niebanalna teza, że umysł jest maszyną algorytmiczną, wymaga, aby (1) tylko niektóre procesy w świecie były algorytmiczne lub (2) procesy algorytmiczne niższego rzędu implementowały procesy algorytmiczne wyższego rzędu, stanowiące dopiero algorytmy umysłu. Tu opieram się na uzyskanych wcześniej wynikach analizy pojęcia procesu algorytmicznego, z których wynika, że przy dosyć skromnych i intuicyjnych założeniach wszystkie procesy są rzeczywiście algorytmiczne (wystarczy założyć, że wszystko, co istnieje, ma przyczynę i skutek, oraz że świat jest skończonym zbiorem powiązanych przyczynowo obiektów). W związku z tym warunek (1) jest fałszywy. Trzeba więc dokładniej powiedzieć, na czym polega warunek (2) - czyli implementowanie procesów algorytmicznych wyższego poziomu przez procesy algorytmiczne niższego poziomu. W przeciwnym razie

*Jestem świadom, że nazwa tej tezy nie jest do końca szczęśliwa, gdyż może sprawiać wrażenie, że pojęcie obliczania mylnie w filozofii nazywa się też - jak w pracach psychologów - „komputacją”. Obliczania nie należy utożsamiać z pojęciem procedur numerycznych (operujących tylko na argumentach liczbowych). Jeśli procesy psychiczne polegają na przetwarzaniu danych, a przyjmujemy, że są to dane cyfrowe - to takie przetwarzanie będzie zawsze już obliczaniem. Niestety, „obliczeniowizm” to nazwa pokraczna, utworzona niezgodnie z zasadami tworzenia terminów technicznych, dlatego też posługuję się mianem „komputacjonizm”.



ISSN 1897 0311

WIOSNA
2007

O NAS

KONTAKT

SPIS TREŚCI

ARCHIWUM

wszelkie odmiany komputacjonizmu okażą się nieinteresujące.

2. Sensy „algorytmiczności”

Przez pojęcie „algorytmiczność” można rozumieć cechę polegającą na:

- (a) Obliczeniowej naturze procesu, czyli jego rekurencyjności.
- (b) Kompresowalności informacji o procesie, czyli jego algorytmicznej upraszczalności.

Pierwsze pojęcie algorytmiczności wiąże się z klasycznym pojęciem algorytmu jako funkcji przejścia maszyny Turinga o określonym alfabetcie, mającej przynajmniej jeden argument wyjściowy i zero lub więcej argumentów wejściowych. Nie chodzi w tym wypadku o ewentualne algorytmy nierekurencyjne czy też algorytmy maszyn analogowych, lecz o najbardziej klasyczne funkcje rekurencyjne maszyn cyfrowych. Stan badań nad maszynami analogowymi nie pozwala, moim zdaniem, wnioskować, że miałyby one gruntownie większą moc obliczeniową od cyfrowych lub też w zasadniczo inny sposób byłyby algorytmiczne. Tezy, iż istnieją analogowe maszyny o większej mocy obliczeniowej od każdego możliwego komputera cyfrowego, wkładam między bajki. Podobnie traktuję tezy dotyczące komputerów kwantowych. Swoją drogą, komputacjonizm analogowy lub komputacjonizm kwantowy są możliwymi stanowiskami w filozofii umysłu. Tu jednak chodzi mi o trudniejszy - z powodu angażowania sporu w uniwersalia - problem cyfrowej natury obliczeniowej umysłu. Przez algorytmiczność rozumie się więc efektywną obliczalność, czyli rekurencyjność. Często stosuje się też odróżnienie między algorytmami a procedurami heurystycznymi, gdzie algorytm rozwiązujący problem P różni się tym od procedury heurystycznej, że zawsze dostarcza poprawnego wyniku (por. np. Newell, Shaw, Simon [1972], s. 123). Niemniej, zarówno procedury heurystyczne, jak algorytmy są efektywnie obliczalne przez maszyny Turinga: to, co jest algorytmem dla problemu P' , może być procedurą heurystyczną dla problemu P i na odwrót. Zdaje się, że w komputacjonizmie nikt nie sądzi, aby jednoznacznie była wyznaczona klasa problemów rozwiązywanych przez algorytmy umysłowe, i że owe rozwiązania nie mają natury heurystycznej. Dlatego też, od tego znaczenia wyrażenia

algorytm będę w tym artykule całkowicie abstrahować.

Drugi sens związany jest z teorią złożoności algorytmicznej Kołmogorowa-Chaitina (np. Chaitin [1975]). Charakteryzuje się w niej niestochastyczność (nielosowość) ciągów. Ciąg danych jest stochastyczny, wtedy kiedy danego ciągu nie można skompresować (bezstratnie)* w taki sposób, żeby ciąg wynikowy łącznie z algorytmem dekompresującym był krótszy od ciągu wejściowego. Ciągi niestochastyczne można nazywać tu **ciągami algorytmicznymi**, choć nie jest to równoznaczne z tym, że ciąg ten jest rekurencyjny.

Ciągi algorytmiczne w sensie pierwszym określam mianem **ciągów algorytmicznych obliczeniowo**, zaś drugi rodzaj ciągów - jako **ciągi algorytmiczne niestochastycznie**** Relacje między tymi pojęciami algorytmiczności stanowią ciekawy temat tak dla matematyka, jak i filozofa matematyki; tu jednak będę od nich abstrahować.

Jaki sens ma orzekanie algorytmiczności o procesach fizycznych? Wydaje się na pierwszy rzut oka, że ciągi algorytmiczne w obu sensach są abstraktami, zaś zwykle procesom fizycznym przypisuje się konkretność (moje rozważania chcę jednak prowadzić w oderwaniu od zajmowania stanowiska w sporze o uniwersalia, tj. nie muszę zakładać żadnego z rozwiązań, aby przypisywać procesom fizycznym naturę algorytmiczną. Uważam, że równie dobrze skrajny nominalista, jak i platonik może posłużyć się proponowanymi przeze mnie terminami). Odpowiedź jest prosta - stany fizyczne tworzą pewne ciągi, a te ciągi są algorytmiczne - nie uwzględnia wielu bardziej skomplikowanych relacji między stanami zdarzeń fizycznych wchodzących w skład procesów a abstrakcyjnymi stanami idealnych maszyn Turinga (czy innego, równoważnego modelu obliczeń). Warto wspomnieć o kilku relacjach fundujących algorytmiczność - obliczeniową - procesów fizycznych (poniższy przegląd nie rości sobie prawa do kompletności!). Są to:

- kodowanie (opisywanie);
- generowanie (powodowanie);
- realizowanie (implementacja).

O procesach fizycznych **algorytmiczność** można orzec zawsze i tylko wtedy, gdy między danym procesem fizycznym a ciągiem wartości dyskretnej algorytmicznym obliczeniowo zachodzi odpowiednia

* Kompresja bezstratna polega na zapisaniu informacji przy użyciu mniejszej ilości pamięci maszyny. Kosztem kompresji jest czas wykonywania algorytmu kompresującego i dekompresującego przy odczycie informacji skompresowanych.

**Będę swobodnie posługiwał się pojęciami „ciąg” i „proces” czy też w ogóle je pomijał - i tak na przykład „algorytmiczność obliczeniowa”, zgodnie z moją konwencją terminologiczną, to cecha procesów algorytmicznych w sensie podanym w punkcie (a).



ISSN 1897 0311

WIOSNA
2007

O NAS

KONTAKT

SPIS TREŚCI

ARCHIWUM

relacja.

Jest oczywiste, że pewne procesy fizyczne mogą zostać opisane za pomocą ciągów dyskretnych. Wystarczy dokonać prostej operacji pomiaru, nie dającej nam nigdy w wyniku wartości ciągłych (dokładność urządzeń pomiarowych jest skończona). Te ciągi dyskretnie są (a) kompresowalne (algorytmiczne niestochastycznie) albo są (b) tekstami programów, tj. kodami funkcji rekurencyjnych odpowiedniej maszyny (algorytmiczne obliczeniowo). Takimi procesami będą na przykład procesy opisywane przez prawa nauki (gdyż prawa naukowe można potraktować jako odpowiednie algorytmy kompresji informacji o świecie), a z drugiej strony - procesy obliczeniowe w komputerach osobistych. Nawiasem mówiąc, jeśli niestatystyczne prawa nauki są uniwersalne dla całej rzeczywistości - tj. jeśli nie ma w rzeczywistości procesów stochastycznych, to wszystkie procesy fizyczne są algorytmiczne w sensie Chaitina. Procesy mające kodowanie algorytmiczne nazwę „procesami algorytmicznymi deskrypcyjnie”. Pozostałe dwie relacje - generowania i realizowania - nie zachodzą między ciągami nielosowymi, czyli algorytmicznymi w sensie Chaitina, a procesami fizycznymi.

Za algorytmiczne można także uznać procesy wygenerowane przez obliczeniowe procesy algorytmiczne - tj. procesy mające kodowanie *obliczeniowe* (rekurencyjne). Warto zauważyć, że takie procesy będą również algorytmiczne obliczeniowo przy dwóch założeniach wymienionych na początku tego tekstu („wszystko, co istnieje, ma przyczynę i skutek” oraz „świat jest skończonym zbiorem”). Wydawać by się zatem mogło, że wyróżnianie tej relacji ma niewielki sens - jednak nie chciałbym uzależniać analizy znaczenia pojęć od przyjętych założeń ontologicznych.

Procesy realizujące obliczenia nie pokrywają się natomiast z procesami opisywanymi algorytmicznie. Rozważmy przykład. Pewne ciągi fizyczne kodujące programy (a zatem rekurencyjne ciągi algorytmiczne deskrypcyjnie) są wprowadzane do układów elektronicznych urządzenia (a zatem wywołują skutki algorytmicznie pochodne), te zaś generują pewne ciągi (znowu algorytmicznie pochodne). Kiedy dane urządzenie realizuje jakiś program? Musi mianowicie istnieć taki opis tego urządzenia, przy którym stany

dyskretne tego urządzenia zgodnie z pewną funkcją rekurencyjną kodują ciągi dyskretnie, które nazwiemy „programem” tego urządzenia. Nadto działające urządzenie musi generować pewne dalsze ciągi dyskretnie, czyli generować zdarzenia składające się na procesy algorytmiczne pochodne. Gdyby uznać, że dane urządzenie realizuje algorytm zawsze i tylko wtedy, gdy istnieje taki opis urządzenia, który spełnia kryteria algorytmiczności obliczeniowej deskrypcyjnej, to wówczas także niezmienną się ciągi liczb (np. wydrukowana na papierze tabliczka mnożenia) realizowałyby algorytmy. Wydaje się jednak, że mówiąc o realizacji, mamy na myśli urzeczywistnianie, czyli wykonywanie pewnych działań. Muszą więc zachodzić pewne zmiany w świecie, polegające - po prostu - na wprowadzaniu danych wejściowych i wyprowadzaniu wyjściowych. Tabliczka mnożenia jest wydrukowana i nie powoduje sama z siebie żadnych zmian w świecie (choć może być takich zmian warunkiem koniecznym). Można byłoby wprawdzie obstawać, że tabliczka mnożenia jednocześnie mnoży wszystkie możliwe dane wejściowe i podaje wszystkie wyniki. W tym sensie miałyby dane wejściowe - lecz takie ujęcie wydaje mi się bardzo wątpliwe. Każda tabela wartości liczbowych stanowiłaby wówczas komputer cyfrowy. A skoro tabelki nie są komputerami, to tym bardziej nie stwierdzimy tego o wydrukowanym w podręczniku dla programistów algorytmie sortowania, dla którego nie wypisano żadnego zbioru danych wejściowych.

Podsumujmy: realizacja algorytmu polega na tym, że pewien ciąg algorytmiczny deskrypcyjnie A (zakodowany na nośniku fizycznym) w połączeniu z urządzeniem fizycznym U powoduje pewne procesy algorytmiczne pochodne B , które z kolei generują ciągi algorytmiczne deskrypcyjnie C . Te procesy, które realizują algorytmy (w tym wypadku są to łącznie procesy A , B , C i procesy zachodzące w U), nazwę „procesami algorytmicznymi realizacyjnie” (choć brzmi to może dosyć sztucznie).

Niektórzy komputacjoniści nadają wyrażeniu **obliczenie** sens węższy, wiążący się z bardzo szczególnym rodzajem relacji realizacji. I tak na przykład Jerry Fodor ([2003], s.174) napisał, że **obliczanie jest relacją przyczynową pomiędzy symbolami, która zachowuje treść**. W przeciwnym razie ma się okazać, że każdy proces przyczynowy



ISSN 1897 0311

WIOSNA
2007

O NAS

KONTAKT

SPIS TREŚCI

ARCHIWUM

jest rodzajem obliczania (jak wskazałem we wstępie). Dodatkowy warunek Fodora polega na tym, że algorytmy pojmujemy też w kategoriach semantycznych - wszak Fodor przyjmuje istnienie języka myśli. Myślenie polega na przeprowadzaniu dowodów w logicznym kodzie mózgu - stąd każdy krok dowodu musi zachowywać właściwość prawdziwości: jeśli interpretacja symboli, które uruchamiają maszynę, jest prawdziwym twierdzeniem, to interpretacja symboli stworzonych przez maszynę również jest prawdziwa (Pinker [2002], s. 88). Pinker i Fodor zakładają, że wszelkie obliczenia są realizowane za pośrednictwem kodu mającego interpretację semantyczną. To jednak stanowisko jest kontestowane przez innych kognitywistów, wskazujących, że programy nie muszą zawierać żadnych elementów deklaracyjnych (twierdzeń w klasycznym rachunku zdań) - ani mózg nie musi przetwarzać języka myśli. Dlatego też definicja Fodora jest twórcza - tak reguluje zakres pojęcia obliczania, że wyklucza z niego pewnych komputacjonistów, nawet konstruujących roboty. Jeśli zaś każdą konfigurację symboli wejściowych i wyjściowych maszyny można zinterpretować jako zdanie (a więc mówić o obliczaniu jako o relacji zachowującej treść zdań) - to wówczas stanowisko Fodora będzie prowadzić do uznania, że każdy proces przyczynowy jest rodzajem obliczania.

Istnieją więc odpowiednio cztery możliwe sensy algorytmiczności:

- (1) Algorytmiczność niestochastyczna deskrypcyjna;
- (2) Algorytmiczność obliczeniowa deskrypcyjna;
- (3) Algorytmiczność obliczeniowa pochodna;
- (4) Algorytmiczność obliczeniowa realizacyjna.

3. Teza komputacjonizmu a sensy algorytmiczności

Mamy przed sobą cztery możliwości zgodnie z wyłuszczonej powyżej sensami wyrażenia **algorytmiczny** oraz dwie odmiany komputacjonizmu - słabszą (dotyczącą wyłącznie charakteru procesów poznawczych) oraz mocniejszą (dotyczącą wszelkich procesów umysłowych). Zdaje się, że nie chodzi bynajmniej o niestochastyczny charakter procesów mózgowych - czy to powodowanych przez procesy umysłowe, czy też realizowanych przez nie algorytmów, czy też ich opisów - komputacjoniści wszak chętnie korzystają z sieci neuro-

nowych, którym przypisują stochastyczność. Tylko „symboliczna” frakcja komputacjonistów (patrz: Wróbel [2003]) mogłaby ewentualnie negować stochastyczność, lecz nie jest to chyba wyznacznik samego komputacjonizmu. Zostawmy więc stochastyczność i Chaitina na boku. Zresztą, jak już wspomniałem, raczej cała rzeczywistość, a nie tylko procesy poznawcze czy umysłowe, byłaby algorytmiczne niestochastycznie, gdyby niestatystyczne prawa przyrody były uniwersalne (a nie jest sprawą filozofa rozstrzyganie *a priori* czy jest tak w istocie). Nie wydaje się więc, żeby o ten sens algorytmiczności chodziło.

Chodzi więc o algorytmiczność obliczeniową. Rozważmy trzy warianty: deskrypcyjny, pochodny i realizacyjny.

- (1) **Komputacjonizm symulacyjny** byłby tezą głoszącą, że procesy umysłowe mają pewien opis algorytmiczny w sensie obliczeniowym; np. procesy umysłowe można symulować przez algorytmy cyfrowe (jest to, jak się zdaje, odpowiednik tzw. słabej sztucznej inteligencji);
- (2) **Komputacjonizm realizacyjny** byłby tezą głoszącą, że mózgi realizują algorytmy cyfrowe (teza mocnej sztucznej inteligencji);
- (3) **Komputacjonizm totalny** - o pochodnej algorytmiczności mózgów mówiliby zwolennicy tezy o algorytmicznym charakterze rzeczywistości fizycznej (Fredkin, Wolfram). Nie są to jednak komputacjoniści *sensu stricto*.

Innymi słowy, komputacjonizm realizacyjny i symulacyjny odpowiada dwóm sensom algorytmiczności obliczeniowej i dwóm programom uprawiania teorii sztucznej inteligencji.

Warunkiem sensowności komputacjonizmu jest możliwość realizowania algorytmów przez procesy fizyczne, czyli takiej cechy rzeczywistości fizycznej, iż jest ona w stanie zawierać procesy poddające się detekcji cyfrowej i opisowi przez funkcje rekurencyjne. Zdaje się, że gdyby świat fizyczny miał całkowicie indeterministyczny i chaotyczny charakter, to byłoby to raczej niemożliwe (pamiętajmy, że niestochastyczność ze stałym współczynnikiem kompresji wiąże się z rekurencyjnością).

Na początku wspomniałem, że teza komputacjonizmu byłaby pod pewnymi warunkami niebanalna, nawet gdyby cała rzeczywistość fizyczna miała charakter algorytmiczny. Chodzi również o obliczeniowy sens algorytmiczności. Otóż teza *komputacjonizmu wielopozio-*



ISSN 1897 0311

WIOSNA
2007

O NAS

KONTAKT

SPIS TREŚCI

ARCHIWUM

...mowego głosiłaby, że bazyowy poziom obliczeniowy rzeczywistości realizuje algorytmy wyższego poziomu; wśród tych algorytmów znajdują się algorytmy poznawcze (czy też szerzej: w ogóle algorytmy umysłowe). Można się spotkać z programami, które stanowią interpretatory innych programów, które z kolei mogą być maszynami realizującymi kolejne jeszcze programy (interesujące przykłady z tzw. gry w życie podaje Dennett. Tak więc można byłoby twierdzić, że w przeciwieństwie np. do kijeków, które nie realizują algorytmów umysłowych, mózgi wykonują takie algorytmy, albo działanie mózgow jest symulowalne przy użyciu algorytmów - w zależności od wersji komputacjonizmu), mimo że cała rzeczywistość realizuje pewne bazowe algorytmy (jak głoszą np. Wolfram czy Fredkin). Innymi słowy, żeby zachować niebanalność tezy komputacjonizmu w wypadku obliczeniowo-cyfrowej natury fizycznego świata, należałoby zakładać istnienie wielu poziomów programów (co jest raczej mało kontrowersyjne, chociaż samo pojęcie „poziomu” należy do nieostrych). Po prostu jeden poziom obliczeniowy stanowiłby podstawę implementacji następnego, np. dzięki temu, że na owym poziomie byłby realizowany program będący tzw. maszyną wirtualną.

Wydaje się, że najczęściej w kognitywistyce mamy do czynienia albo ze słabym komputacjonizmem symulacyjnym, albo z mocnym komputacjonizmem realizacyjnym (i wielopoziomowym), choć możliwe też są, oczywiście, inne stanowiska.

4. Status tezy komputacjonizmu

Podkreśla się najczęściej, że komputacjonizm jest tezą empiryczną, a nie aprioryczną (Rapaport [1998]). Próby apriorycznego obalenia komputacjonizmu przyjmuję ze zdziwieniem, sądząc, że programy badawcze falsyfikuje raczej ich niepowodzenie niż wymyślne eksperymenty myślowe. Największe zarzuty komputacjonizmowi stawiał, jak wiadomo, John Searle w swoim argumente z *Chińskiego Pokoju*, lecz w naukach kognitywnych panuje przekonanie, że jest to argument wadliwy (interesującą argumentację przeciwko Searlowi podaje m.in. Gokieli [2002]).

Aprioryczne argumenty są często przez kognitywistów odrzucane, dlatego że opierają się zwykle na sensie wyrażen języka potocznego,

a wyrażenia te są nie tylko chwiejne, ale i dlatego, że ich sens można w praktyce naukowej regulować czy wręcz całkowicie zmieniać. To, że w potocznym sensie maszyny nie myślą, nie oznacza, iż przy odpowiednio technicznej definicji „maszyny” nie jesteśmy wszyscy pewnego rodzaju maszynami (a że myślimy, tego chyba nikt nie neguje). Mówiąc krótko, tzw. intuicje pojęciowe traktowane są z pewną podejrzliwością - cóż z tego, że użytkownikom danego języka (np. angielskiego) brakowało wiedzy lub wyobraźni, aby utworzyć odpowiednie wyrażenia? Język można zmienić, nie jest on żadną instancją rozstrzygającą wszelkie spory. Podobnie rzecz się ma z naszymi przeświadczeniami - dopóki „nie okażą hartu” w obliczu danych doświadczenia, dopóty są niesprawdzone. Nie można więc polegać na intuicjach językowych czy potocznych przeświadczeniach - jest to wyraz pewnej naiwności. Z punktu widzenia naiwnego obrazu świata wiele zjawisk jest niezrozumiałych, więc podobnie może się dziać w odniesieniu do zjawisk umysłowych, których mechanizmy działania są dla nas nadal pewną zagadką. Dotyczące ich eksperymenty myślowe mogą wytwarzać pewne złudzenia - nawet złudzenia konceptualne. Innymi słowy, zwolennicy komputacjonizmu przesuwają *onus probandi*, iż potoczne intuicje są uprawnione, na rzeczników potocznego obrazu świata. A uprawnienie tych intuicji polega raczej na zderzeniu ich ze świadectwami empirycznymi niż z wątpliwymi analizami sensu wyrażen takiego lub innego języka.

Jeśli komputacjonizm jest tezą empiryczną to czy jest ona poparta świadectwami empirycznymi? Otóż, ściśle rzecz biorąc, jest to nie tyle teza, co *hipoteza*, założenie robocze przyjmowane w ramach programu badawczego zespołu nauk kognitywnych. Przy najostrożniejszym odczytaniu komputacjonistyczny program uprawiania nauk kognitywnych polega na obliczeniowym modelowaniu procesów poznawczych, bez rozstrzygnięcia zawczasu, czy być może nie istnieją procesy poznawcze, które nie są w rzeczywistości realizowane algorytmicznie (lecz które, być może, mogłyby być symulowane algorytmicznie). Podparta świadectwami empirycznymi teza, że procesy poznawcze nie dawałyby się nawet symulować algorytmicznie, stanowiłaby falsyfikację komputacjonizmu w jego najsłabszym wariancie (czyli również w wariantach mocniejszych).



ISSN 1897 0311

WIOSNA
2007

O NAS

KONTAKT

SPIS TREŚCI

ARCHIWUM

5. Wnioski

Zaprezentowałem cztery główne, jak mi się zdaje, znaczenia wyrażenia „algorytmiczny”, a następnie odniosłem je do dwóch podstawowych sensów komputacjonizmu.

Procesy umysłowe, którym często przypisuje się algorytmiczność, lepiej dają się rozumieć, gdy tzw. metaforę komputerową zastąpimy dosłownym sformułowaniem też komputacjonizmu, z wykorzystaniem zdefiniowanych wyżej terminów **algorytmiczności realizacyjnej** i **algorytmiczności de-skrypcyjnej**. Okazuje się wówczas, że ogólną etykietą **komputacjonizm** opatruje się grupę pokrewnych, lecz z pewnością nie identycznych stanowisk. Istnieją mocniejsze i słabsze odmiany. Podsumujmy krótko:

- (1) Komputacjonizm **słaby symulacyjny** głosi, że procesy poznawcze można symulować za pomocą obliczeń maszyny Turinga;
- (2) Komputacjonizm **słaby realizacyjny** głosi, że procesy poznawcze można realizować za pomocą obliczeń maszyny Turinga;
- (3) Komputacjonizm **mocny symulacyjny** głosi, że procesy umysłowe można symulować za pomocą obliczeń maszyny Turinga;
- (4) Komputacjonizm **mocny realizacyjny** głosi, że procesy umysłowe można realizować za pomocą obliczeń maszyny Turinga;
- (5) Komputacjonizm **wielopoziomowy** głosi, że obliczenia procesów (umysłowych/poznawczych) są realizowane przez maszynę wirtualną, zaimplementowaną nie bezpośrednio na niealgorytmicznych procesach fizycznych, lecz na innych procesach obliczeniowych.

Marcin Miłkowski

BIBLIOGRAFIA

- Chaitin, Gregory J., 1975, *Randomness and Mathematical Proof*, „Scientific American” 232, Nr 5 (maj 1975), ss. 47-52.
- Dennett, Daniel, „Rzeczywiste wzorce”, przeł. Marcin Miłkowski w: Tadeusz Szubka, Robert Poczobut (red.), *Analityczna metafizyka umysłu. Najnowsze kontrowersje*.
- Fodor, Jerry, 2003, *Reprezentacje i pojęcia: podstawy teoretyczne*, przeł. M. Gokieli, *Przegląd Filozoficzno-Literacki* 4 (6)/2003.
- Fredkin, Edward, witryna internetowa, <http://www.digitalphilosophy.org> (artykuły i książki Fredkina).
- Gokieli, Marcin, *Searle i jego gwoździe*, w: *Filozofia i nauki szczegółowe*, pod red. T. Ciecierskiego, L. M. Nijakowskiego, J. Szymanika, Warszawa 2002.
- Minsky, Marvin (w przygotowaniu), *The Emotion Machine*, tekst dostępny pod adresem <http://web.media.mit.edu/~minsky/>.
- Miłkowski, Marcin, *Czy wszystkie procesy są algorytmiczne*, w: *Kognitywistyka*, Szymanik, Jakub i Zajenkowski, Marcin (red.), Warszawa 2003.
- Newell, Allen; Shaw, J. C., Simon, H. A., 1972, *Badania empiryczne przeprowadzone na maszynie do teorii logiki: Studium heurystyczne* w: Edward A. Feigenbaum, Julian Feldman (red.), *Maszyny matematyczne i myślenie*, PWN, Warszawa.
- Pinker, Steven, 2002, *Jak działa umysł*, przeł. M. Koraszewska, Książka i Wiedza, Warszawa.
- Rapaport, William J., 1998, *How minds can be computational systems*, *Journal of Experimental and Theoretical Artificial Intelligence*, 10 (1998), s. 403-419.
- Shapiro, Stuart C., 1995 *Computationalism, Minds and Machines*, tom 5, nr 4 (listopad 1995), s. 517-524.
- Wolfram, Stephen, 2002, *A New Kind of Science*, Wolfram Media, Champaign (Illinois).
- Wróbel, Szymon, *Spór inżynierów: koneksjonizm a klasyczna architektura umysłu*, *Przegląd Filozoficzno-Literacki* 4 (6)/2003.

