

PAWEŁ STACEWICZ

Informatyka jako formalna podstawa niektórych badań kognitywistycznych

ABSTRACT. Computer science as formal basis of some cognitive science research
The paper discusses the issue of interpreting computer science concepts (like Turing machine, computational complexity, computability) in the field of modeling cognitive processes. Such cognitive science interpretations are very interesting, because they lead to some important questions about the mind. Some examples of such questions are presented and analyzed in the text.

KEY WORDS: computer science, cognitive science, artificial intelligence, models of computations, computability, models of mind

1. Wstęp

Współczesne badania kognitywistyczne, określane na gruncie polskim jedną unifikującą nazwą **kognitywistyka** (ang. *cognitive science*), są w istocie przedsięwzięciem **interdyscyplinarnym**. Mając w centrum uwagi zagadnienie modelowania umysłu, sytuują się na pograniczu takich dyscyplin jak: neurobiologia, psychologia poznawcza, językoznawstwo, filozofia umysłu i informatyka. Sytuację tę trafnie odzwierciedla następująca charakterystyka Wodzisława Ducha [por. Duch, 1998, cytowane za Poczobut, 2016, s. 203]¹:

¹ Swoj pogląd na temat statusu metodologicznego kognitywistyki przedstawiłem w pracy Stacewicz, 2010, s. 107–112.

[...] kognitywistyka to nauka zmierzająca do zrozumienia ludzkiego i zwierzęcego umysłu poprzez integrację wyników różnych nauk. [...] Zadaniem kognitywistyki jest tworzenie modeli umysłu zgodnych ze wszystkimi gałęziami wiedzy, a więc wielka integracja wyników badań wielu niezależnych dziedzin.

W bogatej palecie badań kognitywistycznych informatyka zajmuje miejsce eksponowane². Nie pełni przy tym funkcji czysto pomocniczych, takich jak porządkowanie, przetwarzanie czy prezentacja danych, lecz realnie współtworzy główny badawczy nurt. Dzieje się, tak ponieważ formalne pojęcia informatyki (choćby pojęcie algorytmu) mogą być odnośzone wprost do aktywności poznawczej umysłu, czyli mogą być **interpretowane kognitywistycznie**. Mogą także pełnić funkcje **heurystyczne**, wyznaczając nowe zagadnienia, teorie i kierunki badań. O sile związków między informatyką i kognitywistyką świadczą dobitnie takie sformułowania jak „komputerowa metafora umysłu”, „obliczeniowa teoria umysłu” czy „paradygmat komputacyjny w kognitywistyce”³.

W niniejszej pracy zwrócę uwagę przede wszystkim na ceną dla kognitywistów heurystyczną funkcję informatyki. Po uzasadnieniu tezy, że znaczna część informatyki współczesnej stanowi naukę formalną, i wyjaśnieniu, na czym polega kognitywistyczna interpretacja jej formalizmów, przejdę do sformułowania i analizy pewnych inspirowanych informatycznie pytań o umysł. Podstawą postawienia tychże pytań będzie kognitywistyczna interpretacja takich pojęć, jak maszyna Turinga, metoda kodowania i nieobliczalność.

² Nie znaczy to, oczywiście, że inne dyscypliny (jak na przykład wspomniana wyżej neurobiologia) oraz różne od informatyczno-obliczeniowego paradygmaty badawcze (jak na przykład paradygmat poznania ucieleśnionego) nie odgrywają w kognitywistyce znaczącej roli. Choćby rolę taką odgrywają, to w niniejszym tekście skupiam się na relacji informatyka-kognitywistyka, eksponując tym samym paradygmat obliczeniowy.

³ Por. tytuły książek dwóch szczególnie aktywnych na polu kognitywistyki badaczy polskich, Marka Hetmańskiego i Marcina Miłkowskiego: *Umysł a maszyna. Krytyka obliczeniowej teorii umysłu* (M. Hetmański) oraz *Explaining the Computational Mind* (M. Miłkowski).

2. Informatyka jako nauka (częściowo) formalna

Metodologiczny status informatyki nie jest jednoznaczny. Zależnie od sposobu interpretowania i wykorzystywania jej głównych pojęć (jak algorytm czy dane) uznaje się informatykę za naukę: a) **formalną** – pokrewną matematyce, b) **inżyniersko-techniczną** – pokrewną elektronice, czy nawet c) **empiryczną** – bliską pod pewnymi względami naukom przyrodniczym (jak fizyka czy biologia)⁴. Trzeba przy tym pamiętać, że są to charakterystyki niewykluczające się wzajem, wskazujące bardziej na różne aspekty badań i zastosowań niż określające informatykę jako taką [por. Murawski, 2014; Knuth, 1974; Denning, 2005].

Mimo sygnalizowanej wyżej wieloaspektowości badań za cechę centralną informatyki – cechę, która umożliwiła zarówno zastosowania inżynierskie, jak i odniesienia do przyrody – należy uznać jej *formalizm*. Znaczy to, że obiekty informatyczne są w pierwszym rzędzie obiektami formalnymi, a dopiero w ramach konkretnych implementacji i zastosowań stają się obiektami **fizykalnymi** (jak na przykład sygnały odpowiadające takim czy innym typom danych) czy **technicznymi** (jak na przykład konkretne, w określony sposób skonstruowane, systemy komputerowe). Mówiąc jeszcze inaczej: aby mogły zaistnieć jakiegokolwiek wytwory informatyki stosowanej, niezbędna jest teoria, którą wypracowuje się w sposób formalny (quasi-matematyczny), w obrębie takich dyscyplin jak **algorytmika** czy **teoria obliczeń**⁵.

⁴ Ważnym współcześnie przejawem empiryczności informatyki są badania nad tzw. *obliczeniami naturalnymi* (ang. *natural computing*). Są to tego typu techniki informatyczne, które polegają bądź na wykorzystywaniu pewnych przyrodniczych procesów w celu zwiększenia efektywności obliczeń (jak to ma miejsce w przypadku obliczeń kwantowych), bądź na naśladowaniu w układach sztucznych (również cyfrowych) pewnych metod obserwowanych w przyrodzie (jak to ma miejsce w przypadku sztucznych sieci neuronowych). Por. Kari, Rozenberg, 2008.

⁵ Przy czym wymienione obszary badań należą do najbardziej podstawowych [por. Harrel, 2000]. Również poza nimi, właściwie na każdym poziomie informatyki, toczą się badania stricte formalne. Podajmy dla przykładu: metody przeszukiwania grafów (użyteczne na przykład w implementacjach gier komputerowych) czy schematy samoorganizacji sztucznych sieci neuronowych.

By wyjaśnić dokładniej, na czym polega formalny charakter informatyki, warto uwypuklić następujące kwestie:

- obiekty informatyczne, w tym algorytmy i struktury danych, są obiektami **abstrakcyjnymi**, które można: a) różnie implementować (kodować), b) różnie interpretować (w różnych dziedzinach problemowych);
- obiekty informatyczne mają pewne **własności formalne**, które pozostają niezależne od ich implementacji/interpretacji (jedną z nich jest złożoność czasowa algorytmów);
- języki programowania są **językami formalnymi**, przypominającymi języki różnych działów matematyki czy logiki;
- możliwe efekty działania programów komputerowych (także ich przebieg) można analizować w sposób **teoretyczny**, bez ich uruchamiania; wyniki te są „konsekwencjami” struktury programu⁶.

3. Kognitywistyczna interpretacja pojęć informatycznych

Formalny charakter informatycznych pojęć sprawia, że można je interpretować na różne sposoby i w różnych dziedzinach problemowych. W obecnym tekście interesuje nas **interpretacja kognitywistyczna**, która polega na odniesieniu czysto formalnych, znaczeniowo „pustych”, obiektów informatyki do aktywności poznawczej umysłu; mówiąc zaś inaczej, na próbie opisu pewnego wycinka tejże aktywności w kategoriach informatycznych.

Kwestię tę wyjaśnię szerzej na dwóch przykładach. Pierwszy z nich dotyczy podstawowych dla informatyki **struktur danych**, które wewnątrz programów komputerowych wiążą ze sobą takie obiekty, jak liczby, sym-

⁶ Pod tym względem program komputerowy przypomina system aksjomatyczny. Obowiązujące w danym systemie twierdzenia, wyprowadzane z aksjomatów w drodze systematycznego stosowania ściśle określonych reguł wywodu, można przyrównać do wyników programu.

bole czy rekordy⁷. Jeśli weźmiemy pod uwagę strukturę w postaci **drzewa binarnego**, to musimy stwierdzić, że dopóty, dopóki jest ona rozważana czysto matematycznie (na przykład ze względu na pytania typu „ile wynosi minimalna liczba poziomów drzewa kodującego uporządkowany ciąg n liczb?”), pozostaje ona obiektem czysto formalnym. W chwili jednak, gdy za pomocą tejże struktury zechcemy opisać sposób **kodowania wiedzy** w ludzkiej **pamięci**, zyskuje ona interpretacją kognitywistyczną [por. Nalepa, 2016]. Przykład drugi jest osadzony w bardzo zaawansowanym informatycznie obszarze badań nad sztuczną inteligencją, a dotyczy **sztucznych sieci neuronowych** [por. Tadeusiewicz, 1993]. W ich przypadku sytuacja przedstawia się podobnie jak wyżej. O ile dana sieć (na przykład perceptron wielowarstwowy) jest badana w oderwaniu od różnych możliwych zastosowań (na przykład ze względu na pytanie o najlepiej matematycznie uzasadniony algorytm uczenia się), pozostaje obiektem formalnym. Jeśli jednak sieć taką uczyni się modelem **percepcji** (wyjaśniającym, jak ludzki umysł rozpoznaje i kategoryzuje objekty z pewnej dziedziny), zyskuje ona sens kognitywistyczny.

Interpretacja **kognitywistyczna** nie jest, rzecz jasna, ani jedynym, ani szczególnie uprzywilejowanym sposobem „wypełniania treścią” formalnych obiektów informatyki. O wiele częściej objekty takie – począwszy od struktur danych, a skończywszy na zaawansowanych projektach systemów – interpretuje się praktycznie, na przykład w dziedzinie sterowania⁸. Innymi słowy, zastosowania techniczne zdecydowanie górują nad kognitywistycznymi. Bez tych ostatnich jednak współczesne badania nad umysłem byłyby niezwykle utrudnione [por. Miłkowski, 2013].

W niniejszym tekście za miarę efektywności kognitywistycznej interpretacji pojęć informatycznych uznaję jej funkcję heurystyczną, czyli zdolność do wzbudzania pewnych interesujących **pytań o umysł**. Zgodnie z takim nastawieniem dane pojęcie jest tym cenniejsze dla kognitywistyki,

⁷ Do typowych struktur danych należą: listy, stosy, tablice n-wymiarowe i drzewa [por. Harel, 2000].

⁸ Przykładowo: w realnych systemach sztuczne sieci neuronowe (zaimplementowane fizycznie) pełnią funkcję sterowników. Por. na przykład Antsaklis, 1990.

im bardziej pobudza badaczy do refleksji nad pewnymi strukturami czy funkcjami umysłu.

Szczególnie ważne w tym względzie wydają się pojęcia, bez których pewnych kwestii w ogóle nie dałoby się postawić. Na przykład: dopiero po uściśleniu koncepcji **algorytmu** (czego dokonał w roku 1936 Alan Turing⁹) można było pokusić się o pytania dotyczące poznawczych ograniczeń tego typu aktywności umysłowej człowieka, która ma charakter **algorytmiczny**. (Dodajmy w nawiasie, że nawet jeśli nie jest to jedyny typ aktywności wiedzotwórczej, to wydaje się, że jemu tylko przysługuje tak ważny dla nauki walor intersubiektywności¹⁰).

4. Informatyczne modele umysłu

Kognitywistyczne interpretacje informatycznych formalizmów mają swoją szerszą podstawę we wstępnym zabiegu przyrównania umysłu do pewnego systemu informatycznego. Mówiąc zaś inaczej, stanowią one niezbędny element typowej dla kognitywistyki procedury budowy **informatycznych modeli umysłu** (IMU)¹¹.

Modele IMU odwołują się do konkretnych informatycznych technik, które pozwalają wyjaśniać (a także: realizować sztucznie) pewne typy czynności poznawczych. Trzeba zauważyć przy tym, że różnym czynnościom odpowiadają w sposób „naturalny” określone techniki. Na przykład: sztucz-

⁹ Dokonał tego, formułując koncepcję maszyny uniwersalnej o stanach dyskretnych, zwanej dziś uniwersalną maszyną Turinga (UMT). Dopiero w świetle tej koncepcji można było określić algorytm precyzyjnie jako schemat operacji możliwych do wykonania przez UMT [por. Turing, 1936 oraz Stacewicz, 2016].

¹⁰ Intersubiektywność metody algorytmicznej wyraża się w tym, że każdy (konkretny) algorytm stosowany do tych samych danych, niezależnie od subiektywnych odczuć osób go wykonujących, musi być realizowany w tych samych krokach (etapach) i musi prowadzić do tego samego wyniku.

¹¹ Procedura taka ma swoje uzasadnienie w dwóch tezach dotyczących umysłu: 1) umysł zajmuje się przetwarzaniem odpowiednio zakodowanych informacji (co czyni go podobnym do przetwarzających dane komputerów); 2) umysł steruje informacyjnie ciałem (podobnie komputer może sterować podłączonymi doń urządzeniami fizycznymi).

ne sieci neuronowe sprawdzają się dobrze w modelowaniu percepcji i uczenia się, systemy eksperckie – w opisie wnioskowań (zawodnych i niezawodnych), zaś algorytmy genetyczne – pozwalają uchwycić niektóre aspekty zachowań twórczych. Choć dalekosiężnym celem kognitywistów jest konstrukcja modeli maksymalnie wszechstronnych, to aktualnie tworzone modele dotyczą wybranych tylko aspektów umysłowej aktywności (na przykład wspomnianych wyżej wnioskowań). Z tego powodu trzeba je nazwać **częstokowymi** [por. na przykład Stacewicz, 2010; Chuderski, 2016].

Z punktu widzenia informatyki modele IMU można charakteryzować na różnych **poziomach opisu**¹². Choć charakterystyki takie abstrahują od typów modelowanych czynności (na przykład od tego, czy chodzi o percepcję czy wnioskowanie), to pozwalają wyodrębnić pewne istotne pojęcia informatyczne, które w toku dalszej analizy mogą się okazać kognitywistycznie **inspirujące**. Mogą zatem prowadzić do ciekawych pytań o umysł.

4.1. Na najniższym poziomie informatycznego opisu, poziomie uwzględniającym elementarne metody kodowania i przekształcania danych, modele IMU można podzielić na: **turingowskie** i **pozaturingowskie**. Podział ten odnosi do wykorzystywanego w danym systemie informatycznym **modelu obliczeń**, abstrahuje natomiast od wysokopoziomowej architektury systemu. Przy czym, zgodnie z proponowanym nazewnictwem, funkcję podstawowego modelu odniesienia (modelu obliczeń) pełni zdefiniowany przez Alana Turinga matematyczny abstrakt, zwany uniwersalną maszyną Turinga (UMT)¹³.

Turingowskim modelem umysłu jest zatem każdy model IMU, który na odpowiednio niskim poziomie opisu pozostaje równoważny pewnej **maszynie Turinga**. Jeśli zatem model taki określa pewien schemat prze-

¹² Co ma swój odpowiednik po stronie programów komputerowych. Programy takie, będące typowym narzędziem formalnego opisu modeli IMU, mogą być zapisywane w językach programowania niskiego lub wysokiego poziomu. Do pierwszych należy Assembler, do drugich Pascal. Por. Wirth, 1989.

¹³ Por. Turing, 1936 oraz Harel, 2000. W gruncie rzeczy chodzi o model obliczeń **cyfrowych**, który jest realizowany w praktyce przez zdecydowaną większość użytkowanych obecnie komputerów.

tworzania danych (pamiętajmy, że umysł rozumiany obliczeniowo jest systemem przetwarzającym informacje), to schemat ten da się zapisać w postaci programu dla maszyny Turinga. A jeśli da się tak zapisać, to jest wykonywalny przez maszynę UMT¹⁴.

Ze względu na szczególne cechy obliczeń turingowskich wszystkie turingowskie modele umysłu (niezależnie od ich wysokopoziomowej struktury) muszą wykazywać trzy następujące właściwości: 1) **dyskretność** (cyfrowość) – rozumianą jako postulat przetwarzania dyskretnych symboli za pomocą dyskretnych stanów, 2) **skończoność** – rozumianą jako wymóg wykonywania skończonej liczby operacji w skończonym czasie, oraz 3) **determinizm** – czyli postulat niezmienności (w tym: nielosowości) schematu przetwarzania danych (określonego czasem na niższym poziomie poprzez schemat uczenia się).

Odnosząc powyższe wyjaśnienia do realnych programów – które są oczywiście skończonymi realizacjami wyidealizowanego modelu Turinga – można powiedzieć, że modele turingowskie mają postać programów dla maszyn **cyfrowych**¹⁵.

Zważywszy na praktykę badań kognitywistycznych (osadzonych w nurcie obliczeniowym), modele drugiego rodzaju, nazywane tutaj **pozaturingowskimi**, są raczej bytami hipotetycznymi. Dzieje się tak, ponieważ zdecydowana większość schematów opisujących (częstkowo) umysł jest przedstawiana w postaci programów dla komputerów cyfrowych. Mimo to, ponieważ w informatyce teoretycznej bada się także sys-

¹⁴ W niniejszym tekście przyjmuję, że używane wyżej pojęcia maszyny Turinga (konkretnej i uniwersalnej) jest dostatecznie dobrze znane. Z tego względu nie objaśniam go szczegółowo. Przystępną charakterystykę tych pojęć zawarłem w następujących opracowaniach: 1) wpis blogowy pt. „Co to znaczy, że umysł jest maszyną Turinga?” (<http://marciszewski.eu/?p=1189>), oraz 2) Marciszewski, Stacewicz, 2011, s. 120–122).

¹⁵ A ponieważ maszyny cyfrowe stanowią zdecydowaną większość obecnie konstruowanych i użytkowanych automatów, to modele turingowskie (choć zróżnicowane ze względu na przeznaczenie i strukturę) stanowią zdecydowaną większość modeli IMU. Z tego też względu uzasadniony jest proponowany podział dychotomiczny, modele turingowskie vs pozaturingowskie. (Systemów pozaturingowskich jest po prostu zbyt mało, by różne ich odmiany uwzględniać we wstępnym podziale).

temy niesprowadzalne do maszyn Turinga, warto rozważyć hipotetyczne modele pozaturingowskie.

Jeden ze sposobów (być może: jedyny) definiowania takich modeli polega na modyfikowaniu – a mówiąc ściślej, poszerzaniu – co najmniej jednej z kluczowych cech obliczeń turingowskich, tj. **dyskretności, skończoności i determinizmu** (zob. wyżej). I tak: rozszerzając cechę dyskretności przetwarzanych danych do cechy ciągłości, uzyskuje się możliwość konstrukcji modeli **analogowych**, które odwołują się do teorii obliczeń ciągłych [por. na przykład Shannon, 1941 i Pour-El, 1974]. Dopuszczając wykonywanie nieskończonej liczby operacji w skończonym czasie, zyskuje się możliwość badania modeli **infinitystycznych**, nawiązujących na przykład do teorii przyspieszających maszyn Turinga [por. na przykład Hogarth, 1994 i Shagrir, 2004]. Zastępując cechę determinizmu pewnego typu losowością, dopuszcza się do dyskusji **modele niedeterministyczne**, oparte na przykład na teorii algorytmów genetycznych [por. na przykład Michalewicz, 1992]¹⁶.

Choć wskazane możliwości są w niewielkim stopniu wykorzystywane przez kognitywistów, to moim zdaniem powinny być dla nich co najmniej inspirujące – do czego nawiązę w rozdziale czwartym.

4.2. Omówiony wyżej podział modeli IMU na turingowskie i pozaturingowskie abstrahuje od kwestii wyboru wysokopoziomowej **architektury** modelu. Dla kognitywistów tymczasem jest kwestią pierwszorzędną, czy architektura układu modelującego jest zgodna z rozpoznawaną na pewnym poziomie (na przykład neurobiologicznym lub psychologicznym) strukturą systemu poznawczego. Istotne jest także to, jak konkretna architektura wpływa na poznawczą moc systemu (czyli: zakres i szybkość rozwiązywanych problemów)¹⁷ [por. Newell, 1990; Chuderski, 2016].

¹⁶ Przynajmniej niektóre z typów obliczeń, które leżą u podstaw modeli pozaturingowskich, określa się mianem **hiperobliczeń** (ang. *hypercomputations*) – czyli tego typu technik przetwarzania danych, które pozwalają rozwiązywać więcej problemów niż obliczenia turingowskie. Z pewnością należą do nich obliczenia ciągłe. Por. Mycka, 2010.

¹⁷ Na podstawie teorii czysto informatycznych można stwierdzić, że odpowiednio dobrana architektura układu przetwarzającego pozwala istotnie zwiększyć szybkość obliczeń

Podział modeli IMU ze względu na typ (wysokopoziomowej) architektury odwołuje się przede wszystkim do badań nad **sztuczną inteligencją**¹⁸. Mówiąc zaś bardziej szczegółowo, człony podziału zależą w największym stopniu od zastosowanej w układzie modelującym **metody reprezentacji wiedzy**¹⁹. Metoda reprezentacji warunkuje strukturę systemu – jeśli nie sprzętową, to przynajmniej logiczną (określającą kształt programu uruchamianego na takim czy innym sprzęcie). Do najbardziej typowych sposobów reprezentowania wiedzy zaliczyć trzeba: struktury konekcyjne, predykatowe, regułowe i drzewiaste [por. Nalepa, 2016].

Na dostatecznie dużym poziomie ogólności wymienione przykładowo typy struktur dzielą się na: a) **symboliczno-regułowe** – wzorowane na znanych z logiki metodach zapisywania wiedzy w różnych rachunkach symbolicznych (na przykład w rachunku predykatów) oraz b) **subsymboliczno-konekcyjne** – odwołujące się do sposobu kodowania wiedzy w systemach nerwowych i mózgach organizmów żywych (w tym ludzi)²⁰. Te pierwsze są stosowane w systemach eksperckich (i opartych na nich logiczystycznych modelach IMU), te drugie – w sztucznych sieciach neuronowych (i bazujących na nich modelach neuropodobnych).

Wobec tak zarysowanego podziału zachodzi szereg pytań o kognitywistyczną wartość różnych architektur. Do pytań tych przejdę w rozdziale kolejnym.

(bez zmiany modelu obliczeń). Na przykład: architektury równoległe i współbieżne (komputerów cyfrowych) zapewniają szybsze rozwiązywanie niektórych problemów niż architektury sekwencyjne. Por. Harel, 2000.

¹⁸ Odwoływanie się do tychże badań jest jak najbardziej zasadne (a dla kognitywistyki typowe), ponieważ spośród wszystkich systemów informatycznych to właśnie sztuczne systemy inteligentne (realizujące zadania wymagające od ludzi inteligencji) są najlepszymi kandydatami na modele IMU. I to one właśnie wykazują największe zróżnicowanie, jeśli chodzi o wysokopoziomową architekturę układu.

¹⁹ Nie chodzi tu o wiedzę w sensie epistemologicznym, lecz o pewien zasób odpowiednio ustrukturyzowanych faktów i zależności, które system wykorzystuje w działaniu, na przykład w podejmowaniu decyzji, uczeniu się, kategoryzacji bodźców etc.

²⁰ Przy czym kwestią otwartą pozostaje, czy obydwa typy architektur (przede wszystkim chodzi tu o struktury konekcyjne) mogą być równie adekwatnie odwzorowane w ramach turingowskiego (a nie pozaturingowskiego) modelu obliczeń.

5. Pytania kognitywistyczne inspirowane przez informatykę

W ostatnim rozdziale pracy chciałbym wyeksponować niektóre z omawianych już wcześniej pojęć informatycznych i zbadać je pod kątem siły wzbudzania interesujących pytań o umysł. Będą to następujące pojęcia: kod i kodowanie, obliczalność i nieobliczalność, model obliczeń oraz architektura systemu przetwarzającego dane. Niektóre z tych pojęć (jak na przykład ostatnie) sytuują się w centrum badań kognitywistycznych, inne czekają dopiero na gruntowaną refleksję.

5.1. Formalne badania informatyków wskazują, że na najniższym poziomie **kodowania** danych doniosłą rolę odgrywa rozróżnienie między danymi **cyfrowymi** i **analogowymi** [por. Mycka, 2010]. Z matematycznego (i elementarnego zarazem) punktu widzenia tym pierwszym odpowiadają liczby wymierne (w praktyce: skończone i redukowalne do zapisów zero-jedynkowych), tym drugim zaś – liczby rzeczywiste (w tym niewymierne, a wśród nich nieobliczalne)²¹. W odniesieniu do konstrukcji inżynierskich możemy powiedzieć z kolei, że dane pierwszego typu są przetwarzane przez maszyny cyfrowe (o stanach dyskretnych), zaś dane typu drugiego przez układy analogowe, zdolne do operowania na sygnałach ciągłych (jak na przykład ciągle potencjały czy napięcia elektryczne).

Przenosząc owo doniosłe informatyczne rozróżnienie na płaszczyznę badań kognitywistycznych, możemy zadać fundamentalne pytanie o sposób kodowania informacji w umyśle/mózgu: **czy jest on analogowy, czy cyfrowy**²². Pytanie to wydaje się istotne również z tego względu, że z teorii obliczeń analogowych wynika, iż obliczenia takie mają większą moc niż cyfrowe, to znaczy pozwalają rozwiązywać więcej problemów niż te drugie (zob. uwagi w punkcie 3.1, nt. pozaturingowskich modeli umysłu, zob. też Mycka,

²¹ Liczby nieobliczalne są to takie „trudne” liczby niewymierne, dla których nie istnieje maszyna Turinga obliczająca je z dowolną zadaną dokładnością (czyli wyznaczająca dowolnie długie segmenty ich dziesiętnego rozwinięcia). Por. Turing, 1936.

²² Właściwie pytanie to należałoby postawić inaczej: *czy jest on analogowy, czy tylko cyfrowy?* Dziedzina sygnałów analogowych (opisywanych przez liczby rzeczywiste) jest bowiem rozszerzeniem dziedziny sygnałów cyfrowych (opisywanych przez liczby wymierne, będące podzbiorem rzeczywistych).

Piekarz, 2004). Gdyby zatem ludzki mózg był w istocie „przetwornikiem analogowym”, to niezależnie od innych swoich wysokopoziomowych właściwości, tylko dzięki własnościom elementarnym, wykazywałby większy **poznawczy potencjał** niż konstruowane przez informatyków automaty cyfrowe. Mówiąc krótko: swoim obliczeniowym potencjałem przewyższałby współczesne komputery. Informatyczne rozróżnienia i analizy nie przesądzą oczywiście, że tak właśnie jest. Pozwalają jednak sformułować konkretne źródłowe pytanie, które niczym „informatyczny drogowskaz” sugeruje kognitywistom kierunek badań przybliżających do takiej czy innej odpowiedzi.

Zauważmy nadto, że ewentualne pozytywne rozstrzygnięcie powyższego pytania (TAK: analogowy mózg przewyższa komputer cyfrowy) byłoby istotne również dla informatyki. Na zasadzie naukowego sprzężenia zwrotnego dałoby informatykom (śledzącym wyniki kognitywistów) silny impuls do poważnych badań nad niedocenianymi dziś technikami analogowymi.

5.2. Kolejne pojęcie informatyczne, które prowadzi do interesujących pytań z zakresu refleksji nad umysłem, to turingowska **nieobliczalność**. Jak wiadomo z teorii obliczeń cyfrowych – opisywanych ściśle za pomocą formalizmu maszyn Turinga – w dziedzinie obliczeń cyfrowych występują dwa rodzaje problemów nierozwiązywalnych, czyli **nieobliczalnych**. Po pierwsze, są to zagadnienia nieobliczalne **praktycznie** – a więc takie, dla których nie istnieją algorytmy o złożoności czasowej niższej niż wykładnicza. W ich przypadku nawet stosunkowo niewielkich rozmiarów dane powodują konieczność wykonania przez maszynę praktycznie nieosiągalnej liczby operacji elementarnych. Po drugie, są to zagadnienia nieobliczalne **zasadniczo** – czyli takie, dla których nie istnieją algorytmy/schematy pozwalające rozwiązać wszelkie możliwe instancje problemu. Każdy z takich problemów cechuje pewna nieusuwalna słabość. By go rozwiązać, trzeba szukać wyspecjalizowanych algorytmów lokalnych dla poszczególnych przypadków; wspomnianych przypadków (a zatem i odpowiednich algorytmów) jest jednak nieskończenie wiele²³.

²³ Kwestie te wyjaśniam szerzej w innym artykule pt. „O algorytmach i algorytmicznej dostępności wiedzy”, poświęconym przede wszystkim zagadnieniu nieobliczalności. Zob. Stacewicz, 2016.

Jeśli problemy nieobliczalne cyfrowo uznać za elitę zagadnień trudnych, to rodzi się w związku z nimi interesujące pytanie kognitywistyczne o **realne strategie**, jakie stosuje umysł, by problemom takim sprostać (niekoniecznie je rozwiązując). Uważam, że hipotezy ich dotyczące powinny brać pod uwagę techniki wskazywane, a niekiedy wręcz praktykowane, przez informatyków.

Pierwsza z nich postuluje takie działanie umysłu, które skutkuje **przeformulowaniem** oryginalnego problemu, bez osłabienia jednak jego praktycznej wartości. Choć nowa wersja problemu rozwiązuje się szybciej, to praktyczny zysk z rozwiązania pozostaje bez zmian. Tak właśnie postępują informatycy, gdy rozwiązują nieobliczalny praktycznie **problem komiwojażera**: zamiast poszukiwać najkrótszej drogi łączącej n miast (co wymaga ok. 2^n operacji elementarnych), zadawalają się znalezieniem drogi co najwyżej dwa razy dłuższej niż najkrótsza (co jest już algorytmicznie wykonalne, por. Harel, 2000). Problemy z drugiej kategorii trudnych, a więc nieobliczalne zasadniczo, są traktowane w sposób równie **pragmatyczny**: wydziela się z nich takie podproblemy, dla których istnieją wystarczająco wydajne algorytmy lokalne, inne podproblemy natomiast pomija się. Istotę obydwu zarysowanych wyżej metod dobrze oddaje hasło: „postępuj pragmatycznie, lecz mimo wszystko cyfrowo”.

Kolejna z hipotetycznych strategii wychodzi od negacji drugiej ze wskazówek zawartych w powyższym hasle. Polega ona na odejściu od cyfrowego modelu obliczeń, a mówiąc bardziej „kognitywistycznie”: na wprowadzeniu umysłu w stan, w którym można wykonywać obliczenia opisywane przez model **silniejszy od cyfrowego**.

Z informatycznej teorii wiadomo, że problemy o (cyfrowej) złożoności wykładniczej, a przynajmniej niektóre z nich, można rozwiązywać efektywnie za pomocą algorytmów **kwantowych**. Odwołanie się do tego typu obliczeń oznacza de facto zmianę turingowskiego modelu obliczeń na kwantowy [Deutsch, 1985]. Wiadomo również, że problemy nieobliczalne cyfrowo (takie jak problem stopu maszyny Turinga) stałyby się rozwiązywalne wówczas, gdyby zastosowało się do nich techniki **analogowe** [Mycika, 2010]. To z kolei jest równoznaczne z przejściem od modelu cyfrowego do analogowego (który dopuszcza operacje na fizycznych wielkościach

ciągłych). Fakty powyższe nie przesądzają oczywiście, że umysł ludzki jest zdolny do działań, których wewnętrzny, niskopoziomowy mechanizm opisują alternatywne modele obliczeń, ale stanowią one istotną **wskazówkę dla kognitywistów** (zob. też uwagi wieńczące punkt 5.1).

Ostatnia z inspirowanych informatycznie metod pokonywania nieobliczalności jest strategią najwyższego poziomu. Polega na stymulowaniu pewnych **aktów twórczych**, które mogą zaowocować powstaniem nowego typu algorytmu lub nawet nieznanego dotychczas modelu obliczeń. (Zauważmy w nawiasie rzecz oczywistą, że wszelkie opisywane wyżej algorytmy, metody i modele są dziełem twórczo myślących informatyków.)

Po stronie informatyki strategii takiej odpowiadają **zautomatyzowane techniki uczenia się** (ang. *machine learning*), które pozwalają maszynom generować nowe pojęcia i algorytmy, niezakładane wstępnie (i trudne do przewidzenia) przez ich twórców. Obecnie realizuje się je za pomocą różnych narzędzi sztucznej inteligencji, spośród których na szczególną uwagę zasługują sztuczne sieci neuronowe i algorytmy genetyczne [Mitchell, 1997]. Zwłaszcza w przypadku tych ostatnich, obejmujących pewne kroki/procedury losowe, końcowy „produkt” maszyny może istotnie wykroczyć poza wstępne założenia programisty.

Czy ludzka twórczość da się sprowadzić do fizycznych implementacji takich czy innych technik automatycznego uczenia się? Sprawa wydaje się trudna, zwłaszcza że twórczość najwyższego lotu, w tym matematyczna, opiera się na kreowaniu pewnych **abstrakcyjnych pojęć**, które pozwalają rozwiązywać problemy z wyższego jakby pułapu, z poziomu niezwanej wcześniej teorii abstrakcyjnej²⁴. Zdolność uchwytowania abstraktów zdaje się zaś czymś typowo ludzkim – a jeśli nawet możliwym do realizacji

²⁴ Licznych przykładów w tym względzie dostarcza historia matematyki. Na przykład abstrakcyjna teoria liczb zespolonych pozwoliła pokonać problem rozwiązywalności równań wielomianowych – problem dotyczący dziedziny liczbowej „niższego rzędu” niż liczby zespolone (to znaczy liczb rzeczywistych). W logice z kolei jest znany ważny dla informatyki wynik G. Boolosa [Boolos, 1987], który pokazuje, w jaki sposób w logice drugiego rzędu za pomocą pomysłowych abstrakcyjnych pojęć daje się dowodzić pewnych twierdzeń niedowodliwych (praktycznie) w logice rzędu pierwszego.

sztucznej, to w sposób bardzo wyrafinowany²⁵. Tak czy owak jednak, to do kognitywistów należy badanie tej zdolności, a także sprawdzanie, na ile efektywnie udaje się ją implementować za pomocą komputerowych algorytmów.

5.3. Kwestie omawiane wyżej – powiązane najsilniej z zagadnieniem **obliczalności** (niekoniecznie turingowskiej) – należy traktować jako interesujące dla kognitywistów wyzwanie, które podejmują obecnie w dość ograniczonym zakresie²⁶. Kwestia kolejna sytuuje się za to w samym centrum badań i jest znana dużo szerzej.

Chodzi o podstawowy dla kognitywistyki zabieg budowania modeli umysłu opartych na konkretnych informatycznych **architekturach**. Mówiąc zaś nieco inaczej: o wykorzystywanie tworzonych przez informatyków (z pobudek raczej praktycznych) wysokopoziomowych architektur do modelowania rozmaitych czynności poznawczych [Chuderski, 2016]. Jeśli zawęzić pole rozważań (i odniesień zarazem) do badań nad **sztuczną inteligencją**, to na pierwszy plan wybija się rozróżnienie między architekturaми: a) **regulowymi** – właściwymi na przykład systemom eksperckim, oraz b) **konekcyjnymi** – właściwymi sztucznym sieciom neuronowym²⁷.

²⁵ Niektórzy nazywają ją intuicją intelektualną i uznają za czynnik niealgorytmizowalny. Jakkolwiek jednak jest on niealgorytmizowalny, to dopiero w połączeniu z algorytmami staje się poznawczo efektywny. Por. dyskusję w blogu Cafe Aleph pt. „Czym jest intuicja intelektualna wg racjonalistycznej filozofii nauki?” (<http://marciszewski.eu/?p=90830>, a także pracę Marciszewski, 2013).

²⁶ Por. na przykład zawartość wydanego w roku 2016, *Przewodnika po kognitywistyce* [Bremer, 2016], w którym nie ma ani jednego rozdziału poświęconego tym kwestiom.

²⁷ Opozycję tę można rozumieć szerzej, jako rozróżnienie między architekturaми: a) **modularnymi** – w ramach których przewiduje się przejrzysty podział zadań systemu między wyraźnie odróżnialne moduły (powiązane ze sobą czytelnymi relacjami), oraz b) **rozproszonymi** – w ramach których poszczególne zadania są realizowane przez cały system, skonfigurowany tak, by rozproszona w nim „wiedza” – rozproszona w całym systemie, a nie skoncentrowana w odrębnych modułach – zapewniała efektywną realizację zadania. Prototypem architektur modularnych jest architektura von Neumanna (charakterystyczna dla pierwszych komputerów cyfrowych), zaś najpełniejszą realizacją struktur rozproszonych są sztuczne sieci neuronowe.

Z punktu widzenia kognitywistyki każda z wymienionych architektur ma swoje typowe (rzec by można: naturalne) zastosowania; każda z nich dostarcza także innego rodzaju wyjaśnień.

Architektury **koneksyjne** sprawdzają się najlepiej w modelowaniu prostych zadań poznawczych, takich jak rozpoznawanie obiektów, wstępna ich kategoryzacja czy proste reakcje na bodźce zewnętrzne. Ich moc **eksplanacyjna** jest minimalna. Nawet jeśli sieć rozwiązuje pewien problem, to nie dostarcza żadnej, czytelnej dla człowieka, wiedzy o logicznej strukturze rozwiązania – strukturze, która pozwoliłaby sprawdzić jego poprawność. Jedyna dostępna wiedza to rozproszony po całej sieci rozkład liczbowych wag połączeń między neuronami, które ustaliły się w trakcie pewnej procedury uczenia się. Wgląd w tę procedurę (procedurę reorganizacji wag) również nie daje informacji o strukturze rozwiązania.

Architektury **regulowe** z kolei pozwalają modelować złożone procesy rozumowań na podstawie danych symbolicznych. Ich niewątpliwą zaletą jest przejrzystość generowanych wyjaśnień. Jeśli system regulowy rozwiąże pewien problem, to niejako od razu ujawnia logiczną strukturę rozwiązania. Ową strukturą wyjaśniającą jest łańcuch przejrzystych znaczeniowo reguł, których konsekwentny wybór doprowadził od danych początkowych do wyniku. Dysponując takim przejrzystym wyjaśnieniem, każdy może sprawdzić, czy metoda rozwiązania faktycznie jest poprawna (jak również to, czy „gdzieś po drodze” system nie popełnił błędu).

Ponieważ obydwa typy architektur mają inne właściwości (i funkcjonalne, i eksplanacyjne), kognitywista staje przed poważnym pytaniem: „Jakiego typu kombinacja tych architektur, uzupełniona o pewne dodatkowe elementy²⁸, odzwierciedla najlepiej architekturę poznawczą człowieka?”. Podkreślmy na koniec, że pytanie to – podobnie jak pytania postawione wcześniej – jest inspirowane informatycznie.

²⁸ Jednym z elementów dodatkowych jest skojarzony z daną architekturą schemat rozwiązywania problemów, czy też adaptacji systemu o danej architekturze do potrzeb środowiska. Może to być na przykład schemat ewolucyjny.

Bibliografia

- Antsaklis P.J., (1990), "Neural networks in control systems", *IEEE Control Systems Magazine*, nr 3 (10), s. 3–5.
- Boolos G., (1987), "A curious inference", *Journal of Philosophical Logic*, nr 16, s. 1–12.
- Bremer J. SJ (red.), (2016), *Przewodnik po kognitywistyce*, Kraków, WAM.
- Cafe Aleph (<http://blog.marciszewski.eu/>), akademicki blog dyskusyjny W. Marciszewskiego i P. Stacewicza.
- Chuderski A., (2016), „Modelowanie procesów poznawczych”, [w:] *Przewodnik po kognitywistyce*, red. J. Bremer SJ, Kraków, WAM, s. 173–202.
- Denning P.J., (2005), "Is computer science science?", *Communications of the ACM*, t. 48, nr 4, s. 27–31.
- Deutsch D., (1985), "Quantum theory, the Church-Turing principle and the universal quantum computer", *Proc. Roy. Soc. Lond.*, A 400, s. 97–117.
- Duch W., (1998), „Czym jest kognitywistyka?”, *Kognitywistyka i Media w Edukacji*, nr 1, s. 9, 11.
- Harel D., (2000), *Rzecz o istocie informatyki. Algorytmika*, Warszawa, Wydawnictwa Naukowo-Techniczne.
- Hetmański M., (2000), *Umysł a maszyny. Krytyka obliczeniowej teorii umysłu*, Lublin, Wydawnictwo UMCS.
- Hogarth M., (1994), "Non-Turing computers and non-Turing computability", *PSA*, vol. 1, s. 126–138.
- Kari L, Rozenberg G., (2008), "The many facets of natural computing", *Communications of the ACM*, nr 10 (51), s. 72–83.
- Knuth D.E., (1974), "Computer science and its relation to mathematics", *American Mathematical Monthly*, nr 4 (81), s. 323–343.
- Marciszewski W., (2013), „Racjonalistyczny optymizm poznawczy w Gödlowskiej wizji dynamiki wiedzy”, [w:] *Przewodnik po epistemologii*, red. R. Ziemińska, Kraków, WAM.
- Marciszewski W., Stacewicz P., (2011), *Umysł – komputer – świat. O zagadce umysłu z informatycznego punktu widzenia*, Warszawa, Akademicka Oficyna Wydawnicza EXIT.
- Michalewicz Z., (1992), *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin, Springer.
- Miłkowski M., (2013), *Explaining the Computational Mind*, Cambridge, Mass., MIT Press.
- Mitchell T., (1997), *Machine Learning*, Singapore, McGraw Hill.
- Murawski R., (2014), *Filozofia informatyki. Antologia*, Poznań, Wydawnictwo Naukowe UAM.
- Mycka J., (2010), „Obliczenia dyskretne i ciągle jako realizacje antropomorficznej i fizycznej koncepcji efektywnej obliczalności”, [w:] *Światy matematyki. Tworzenie czy*

- odkrywanie, red. I. Bondecka-Krzykowska, J. Pogonowski, Poznań, Wydawnictwo Naukowe UAM, s. 247–260.
- Mycka J., Piekarczyk M., (2004), „Przegląd zagadnień obliczalności analogowej”, [w:] *Algorytmy, metody i programy naukowe*, red. S. Grzegórski, M. Miłoś, P. Murys, Lublin, Polskie Towarzystwo Informatyczne, s. 125–132.
- Nalepa G.J., (2016), „Sztuczne systemy oparte na wiedzy”, [w:] *Przewodnik po kognitywistyce*, red. J. Bremer SJ, Kraków, WAM, s. 122–142.
- Newell A., (1990), *Unified Theories of Cognition*, Cambridge, Mass., Harvard University Press.
- Poczobut R., (2016), „Status umysłu”, [w:] *Przewodnik po kognitywistyce*, red. J. Bremer SJ, Kraków, Wydawnictwo WAM, s. 204–238.
- Pour-El M.B., (1974), “Abstract computability and its relations to the general purpose analog computer”, *Transactions of the American Mathematical Society*, nr 199, s. 1–28.
- Rozenberg G., Back T., Kok J.N. (eds.), (2012), *Handbook of Natural Computing*, Berlin–Heidelberg, Springer.
- Shagrir O., (2004), “Super-tasks, accelerating Turing machines and uncomputability”, *Theoretical Computer Science*, nr 317, s. 105–114.
- Shannon C., (1941), “Mathematical theory of the differential analyzer”, *J. Math. Phys. MIT*, nr 20, s. 337–354.
- Stacewicz P., (2010), *Umysł a modele maszyn uczących się. Współczesne badania informatyczne w oczach filozofa*, Warszawa, Akademicka Oficyna Wydawnicza EXIT.
- Stacewicz P., (2016), „O algorytmach i algorytmicznej dostępności wiedzy”, *Studia Metodologiczne*, nr 36, s. 315–331.
- Tadeusiewicz R., (1993), *Sieci neuronowe*, Warszawa, Akademicka Oficyna Wydawnicza RM.
- Turing A.M., (1936), “On computable numbers, with an application to the Entscheidungsproblem”, *Proc. Lond. Math. Soc.*, nr 42, s. 230–265.
- Turing A.M., (1950), “Computing machinery and intelligence”, *Mind*, nr 49, s. 433–460.
- Wirth N., (1989), *Algorytmy + struktury danych = programy*, Warszawa, WNT.

Paweł Stacewicz
Zakład Filozofii Nauki, Socjologii i Podstaw Techniki
Wydział Administracji i Nauk Społecznych
Politechnika Warszawska
Plac Politechniki 1
00-661 Warszawa
e-mail: p.stacewicz@ans.pw.edu.pl