

Izabela Bondecka-Krzykowska

O przedmiocie badań informatyki

Studia Philosophiae Christianae 50/2, 5-24

2014

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej bazhum.muzhp.pl, gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

IZABELA BONDECKA-KRZYKOWSKA

O PRZEDMIOCIE BADAŃ INFORMATYKI

Streszczenie. Jednym z podstawowych zagadnień ontologicznych filozofii informatyki jest pytanie o przedmiot badań informatyków. W pracy dokonano analizy wybranych poglądów dotyczących obiektów informatyki: informacji, komputerów, programów komputerowych oraz algorytmów.

Uznanie informatyki za naukę o informacji prowadzi do wielu problemów filozoficznych związanych nie tylko z definicją wszechobecną w nauce informacji, ale również z jej naturą. Podobne kwestie związane są z przyjęciem, że przedmiotem badań informatyki są komputery. W pracy przeanalizowano między innymi związek pomiędzy komputerem, rozumianym jako byt abstrakcyjny, a rzeczywistymi maszynami, będącymi jego urzeczywistnieniami, oraz problem rozróżnienia *hardware'u* od *software'u*. Z kolei stwierdzenie, że podstawową działalnością informatyków jest tworzenie programów komputerowych prowadzi do poglądu, że informatyka jest nauką o programach. W pracy dokonano zestawienia poglądów na naturę programów, w tym na rozróżnienie programów jako kodów od programów wykonywanych na komputerach. Praca zawiera również analizę pojęcia algorytmu oraz procedury jako obiektów badań informatyki, w tym omówienie cech algorytmu.

Słowa kluczowe: filozofia informatyki, ontologia informatyki, informacja, komputer, program komputerowy, algorytm

1. Wstęp. 2. Informacja. 3. Komputer. 4. Program. 5. Algorytm. 6. Zakończenie.

1. WSTĘP

Współcześnie komputery są nieodłączną częścią życia większości ludzi. Pełnią one istotne funkcje już nie tylko w pracy człowieka, ale również poza nią. Ludzie wykorzystują komputery w rozmaity sposób: do robienia zakupów, planowania podróży czy jako sposób spędzania wolnego czasu. Coraz częściej kontakty *on-line* zastępują tradycyjne rozmowy i spotkania z przyjaciółmi. Komputery stały się integralną częścią naszej rzeczywistości. Ich wszechobecność powoduje, nie tylko wśród naukowców, wzrost zainteresowania kwestiami filozoficznymi związanymi z komputerami, a tym samym z informatyką. Coraz więcej pisze się, i to nie tylko w czasopiśmie naukowych, na temat zagadnień filozoficznych związanych z informatyką. Próbuje się nawet definiować filozofię informatyki jako kolejną gałąź filozofii. Można zatem zaryzykować stwierdzenie, że informatyka doczekała się refleksji filozoficznej lub przynajmniej taka refleksja została zapoczątkowana.

Czy jednak mówiąc o informatyce, jesteśmy zgodni co do tego, o czym mówimy? Czy wiemy, czym zajmują się informatycy, jakie przedmioty badają?

Polskie słowo *informatyka* pochodzi od słowa *informacja* i sugeruje bliski związek informatyki z informacją. Podobne pochodzenie mają wyrazy określające tę dyscyplinę nauki w języku francuskim (*informatique*) i niemieckim (*Informatik*). Z kolei angielską nazwę *computer science* można rozumieć jako wskazanie na to, że informatyka jest po prostu nauką o komputerach. Zatem samo słowo oznaczające dyscyplinę wiedzy nazywaną w Polsce informatyką sugeruje różne rozumienie jej ontologii. Czy informatyka zajmuje się badaniem informacji (*informatique*, *Informatik*), czy też komputerów (*computer science*)? Wśród informatyków i filozofów znaleźć można różne odpowiedzi na to pytanie. Jedną z najczęstszych jest stwierdzenie, że informatyka bada informację.

2. INFORMACJA

W pojawiających się obecnie definicjach informatyki powszechnie spotyka się twierdzenie, że informatyka jest nauką zajmującą się badaniem różnych aspektów informacji. Na przykład w Encyklopedii PWN pod hasłem informatyka znajdziemy definicję „dyscyplina naukowa zajmująca się przetwarzaniem informacji z użyciem komputerów”¹. Twierdzi się, że informatycy zajmują się badaniem sposobów reprezentowania i przetwarzania informacji w systemach informatycznych², efektywnością tych procesów, możliwościami ich zastosowań, ale również naturą samej informacji (jej filozofią). Dlatego też niektórzy informatycy uważają, że ich dyscyplina to nie tylko nauka, ale również sztuka: „Informatyka w ogólności jest sztuką i nauką reprezentowania i przetwarzania informacji a w szczególności przetwarzania informacji za pomocą maszyn logicznych nazywanych automatycznymi komputerami cyfrowymi”³.

Jeśli zatem przedmiotem badań informatyki jest informacja, to nasuwa się podstawowe pytanie: co to jest informacja? Jest to jedno z podstawowych i współcześnie najczęściej zadawanych pytań w filozofii nauki, ponieważ pojęcie informacji pojawia się w wielu, jeśli nie

¹ <http://encyklopedia.pwn.pl/haslo/3914698/informatyka.html>

² Por. np. S. Gorn, *The computer and information sciences: a new basic discipline*, SIAM Review 5(1963)2, 150–155; G. Forsythe G., *A university's educational program in computer science*, Communications of the ACM 10(1967)1, 3–11; W.F. Atchison, S.D. Conte, J.W. Hamblen, T.E. Hull, T.A. Keenan, W.B. Kehl, E.J. McCluskey, S.O. Navarro, W.C. Rheinboldt, E.J. Schweppe, W. Viavant, D.M. Young, *Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science*, Communications of the ACM 11(1968)3, 151–197; P.J. Denning, *What Is Computer Science?*, American Scientists 73(1985)(January-February), 16–19; J. Hartmanis, H. Lin, *What is Computer Science And Engineering?*, w: *Computing the Future: A Broader Agenda for Computer Science and Engineering*, red. J. Hartmanis, H. Lin, National Academy Press, Washington 1992, 163–216; J. Hartmanis, *Some Observations about the Nature of Computer Science*, w: *Foundations of Software Technology and Theoretical Computer Science: 13th Conference, Bombay, India, December 15–17, Proceedings*, red. R.K. Shyamasundar, Springer-Verlag Lecture Notes in Computer Science 761, Berlin 1993, 1–12.

³ G. Forsythe, art. cyt., 4.

w większości, dyscyplin naukowych, na przykład w biologii, fizyce, naukach społecznych i ekonomicznych. Czy „informacja” w kontekstach różnych nauk to zawsze to samo pojęcie? Od odpowiedzi na to pytanie zależy sposób jej definiowania.

Definicje informacji można podzielić na dwie grupy. Pierwsza z nich, to definicje lokalne, specyficzne dla danej dyscypliny wiedzy, które zazwyczaj nie są odpowiednie dla innych nauk. Druga grupa to definicje globalne, tak zwane ogólne definicje informacji, w których próbuje się uchwycić jej cechy wspólne dla wszystkich dyscyplin naukowych używających tego pojęcia. Taką ogólną, zdaniem jej autora odpowiednią dla wszystkich nauk, definicję podał między innymi George Ifrah: „Informacja stanowi zespolenie elementu formującego i elementu formowanego, istotę tego zespolenia określa zmiana stanu polegająca na tym, że pojawienie się elementu formującego pobudza element formowany zgodnie z regułami ustalonego z góry kodu”⁴.

Z kolei przykładem podejścia lokalnego do definiowania informacji jest jej rozumienie powszechne w informatyce jako „sygnałów lub symboli, które niosą znaczenie dla ludzkiego obserwatora”⁵. Naturę tak zdefiniowanego przedmiotu badań informatyki próbuje opisać stworzona przez Claude’a Shannona matematyczna teoria komunikacji. Teoria ta traktuje informację jak zjawisko fizyczne, a przedmiotem jej zainteresowania jest sposób kodowania i przesyłania informacji a nie jej znaczenie. Informacja związana jest z wybieraniem jednego symbolu ze zbioru możliwych symboli, a największą ilość informacji niesie tekst, w którym prawdopodobieństwo wystąpienia każdej litery jest takie samo, to znaczy ciąg losowy. Zatem matematyczna teoria komunikacji bada informację na poziomie syntaktycznym. Teoria Shannona nie jest zatem teorią informacji w potocznym rozumieniu tego słowa. Informacja ma w niej bowiem wyłącznie znaczenie techniczne, całkowicie pomija się nie tylko jej znaczenie, ale również kontekst,

⁴ G. Ifrah, *Historia powszechna cyfr*, tłum. z fr. K. Marczevska, K. Szeżyńska-Mačkowiak, Wydawnictwo W.A.B. Warszawa 2006, 868.

⁵ P. J. Denning, *Computer science: the discipline*, w: *Encyclopedia of Computer Science*, red. A. Ralston, E.D. Reily, D. Hemmendinger, Nature Pub. Group 2000, 25.

w którym się pojawia (odpowiedź „tak” zawiera tę samą ilość informacji zarówno jako odpowiedź na pytanie „Czy chcesz herbaty?”, jak i jako odpowiedź na pytanie „Czy wyjdiesz za mnie?”). Shannon i jeden z kontynuatorów jego prac Warren Weaver stworzyli zatem czysto ilościową teorię informacji⁶.

Zupełnie inne od ilościowego podejście do informacji opisują teorie semantyczne, w których charakteryzuje się informację w terminach: dane plus znaczenie. Luciano Floridi⁷ zaproponował ogólną definicję informacji semantycznej, która głosi, że informacja to zawartość semantyczna poprawnie sformułowanych, znaczących danych; przy czym nie istnieje informacja bez danych, dane te są bytami relacyjnymi różnego typu oraz znaczenie informacji jest niezależne od informowanego. Przez „poprawne sformułowanie” rozumie on zgodność ze składnią systemu, języka lub kodu, który analizujemy⁸. Z kolei termin „znaczące” podkreśla, że dane muszą mieć pewne znaczenie w używanym systemie czy języku. Takie rozumienie terminu ‘informacja’ staje się obecnie standardem również w informatyce, szczególnie w tych jej dyscyplinach, w których informacja ma kluczowe znaczenie, takich jak: teoria systemów informacyjnych, zarządzanie systemami informacyjnymi czy projektowanie baz danych.

Jednak jak dotąd nie powstała żadna powszechnie akceptowana definicja informacji, a co za tym idzie nie ma też ogólnej przyjętej teorii informacji opisującej jej naturę. W tej sytuacji dla wielu obserwatorów wydaje się skrajnie nienaukowe twierdzenie, że informatyka bazuje na

⁶ Obaj twórcy matematycznej teorii komunikacji zaprzeczali związkowi ich pojęcia informacji z pojęciem tym używanym w codziennym życiu (por. P.J. Denning, *Computer science: the discipline*, art. cyt.), a Shannon przeciwstawiał się nawet używaniu w odniesieniu do niej terminu „teoria informacji”.

⁷ L. Floridi, *Information*, w: *The Blackwell Guide to the Philosophy of Computing and Information*, red. L. Floridi, Blackwell Publishing 2004, 40–61; Tenże, *Semantic Conceptions of Information*, w: *Stanford Encyclopedia of Philosophy 2005* [<http://plato.stanford.edu/entries/information-semantic/>].

⁸ Składnię interpretuje się bardzo szeroko, nie ograniczając się do jej znaczenia w lingwistyce. Informacja semantyczna nie musi być językowa. Obrazy są często bardziej komunikatywne niż ich słowne opisy np. w instrukcjach obsługi czy montażu.

pojęciu, które nie jest obiektywnie zdefiniowane. Może zatem angielska nazwa informatyki jako *computer science* lepiej oddaje jej naturę niż nazwa polska? Być może informatycy nie zajmują się badaniem informacji jako takiej, lecz urządzeniami ją przetwarzającymi, komputerami?

3. KOMPUTER

Wielu informatyków uważa, że centralne miejsce w ich pracy zajmuje komputer, a co za tym idzie, że informatyka jest nauką o komputerach i związanych z nimi zjawiskach⁹. Czym zatem są komputery i jaka jest ich natura?

Komputer, jako przedmiot badań informatyki, można rozumieć na dwa sposoby: jako urządzenie fizyczne i jako obiekt abstrakcyjny. Wydaje się, że pierwsze z nich nie budzi żadnych wątpliwości. Komputery są bowiem urządzeniami, z którymi stykamy się na co dzień. Jednak owa powszechność komputerów i różnorodność ich form utrudnia podanie jednoznacznej definicji urządzenia, jakim jest komputer. Czy *smartfon* to jeszcze telefon, czy już komputer? Czy procesory wbudowane w większość nowoczesnych maszyn to komputery, czy też tylko części komputerów? Jeśli tylko części, to czy wbudowanie w urządzenie, na przykład w sprzęt AGD, procesora powoduje, że urządzenie to staje się komputerem? Wydaje się, że nie ma powszechnej zgody co do cech definiujących komputery i odróżniających komputery od innych maszyn. Co więcej, rozumienie terminu komputer ulegało (i nadal ulega) zmianom: od osób wykonujących obliczenia¹⁰, poprzez urządzenia służące usprawnieniu obliczeń, takie jak starożytne abaki i liczydła czy wynalezione w XVII wieku maszyny mechaniczne służące do wykonywania podstawowych działań arytmetycznych, aż po maszyny uniwersalne wykonujące szeroką klasę zadań. Dlatego też bardzo trudno

⁹ R.W. Hamming, *One man's view of computer science*, Journal of the ACM 16(1969)1, 3–12; A. Newell, A.J. Perlis., H.A. Simon, *Computer science*, Science 157(1967), 1373–1374.

¹⁰ W pierwszej połowie XX wieku komputerami nazywano ludzi wykonujących obliczenia z użyciem specjalnych algorytmów.

jest odpowiedzieć na, z pozoru proste, pytanie, jak wyglądał pierwszy komputer i kto był jego wynalazcą¹¹.

Wydaje się, że dobrze oddaje współczesne rozumienie komputera definicja podana przez G. Ifraha: „Komputer jest automatem złożonym z urządzenia wejściowego i wyjściowego, pamięci, jednostki centralnej, dokonującej wszelkiego rodzaju przekształceń danych wyrażonych w postaci ciągu znaków (stanowiących materialne przedstawienie zakodowanych informacji), który pozwala – w granicach fizycznych możliwości urządzenia – wykonywać wszelkie obliczenia typu symbolicznego (a zatem rozwiązywać wszelkie zadania, których rozwiązania można wyrazić w postaci algorytmu) pod kierunkiem jednostki sterującej, działającej zgodnie z programami zapisanymi w pamięci (i wobec tego traktującej instrukcje wykonawcze w ten sam sposób co dane przeznaczone do przetworzenia)”¹². Zwraca się w niej uwagę nie tylko na elementy komputera i sposób jego działania, ale również wskazuje podstawowe jego cechy, takie jak uniwersalność. Za twórcę koncepcji maszyny uniwersalnej uważa się Charlesa Babbage’a, który od roku 1833 prowadził prace nad projektem urządzenia, nazwanego przez niego *maszyną analityczną*¹³. W planach tej maszyny Babbage zaprojektował nie tylko podstawowe elementy konstrukcyjne, których funkcjonalne odpowiedniki można znaleźć we współczesnych komputerach (jednostkę liczącą, centralną jednostkę sterującą, pamięć, jednostkę wprowadzania danych i poleceń oraz jednostkę wyprowadzania

¹¹ J.P. Eckert i J.W. Mauchly, twórcy zbudowanej w 1946 roku maszyny ENIAC uważanej przez niektórych badaczy za pierwszy w historii komputer, złożyli wniosek patentowy, chcąc opatentować „komputer”, a tym samym zapoczątkowali wieloletni spór o to, kto jest wynalazcą komputera. Dopiero w roku 1973 r. sąd w Minneapolis rozstrzygnął ten spór, uznając ich prawa do konstrukcji ENIAC-a, ale odmawiając im autorstwa wynalazku komputera. Za wynalazcę komputera sąd uznał J.V. Atanasoffa, który w latach 30. XX wieku zbudował komputer ABC, służący do rozwiązywania równań różniczkowych.

¹² G. Ifrah, dz. cyt., 846.

¹³ Dokładny opis maszyny analitycznej (oraz wcześniejszych urządzeń służących do obliczeń) znaleźć można np. w: I. Bondecka-Krzykowska, *Historia obliczeń. Od rachunku na palcach do maszyny analitycznej*, Wydawnictwo Naukowe UAM, Poznań 2012.

wyników), ale również możliwość jej programowania¹⁴. Projektując swoją maszynę, Babbage określił główne części i funkcje analitycznych maszyn obliczeniowych przeznaczonych do rozwiązywania obszernej klasy problemów a nie tylko jednego zadania. Tak rozumiana uniwersalność, znajdująca odzwierciedlenie również w podanej powyżej definicji Ifraha, jest traktowana współcześnie jako cecha odróżniająca komputery od innych urządzeń.

W drugim, powszechnym obecnie, rozumieniu komputera odchodzi się od jego opisu jako urządzenia fizycznego na rzecz podejścia abstrakcyjnego. Komputer można bowiem traktować jako byt abstrakcyjny opisywany w sposób formalny – jako maszynę matematyczną. Jedną z takich ścisłych, matematycznych definicji podał Alan Turing, a abstrakcyjny byt przez niego opisany nazywa się dzisiaj maszyną Turinga.

Maszyna Turinga składa się z nieskończonej taśmy, złożonej z klatek, w których maszyna może zapisywać lub wymazywać symbole, oraz przesuwanej się po niej głowicy wykonującej instrukcje (które stanowią program maszyny). W każdym momencie maszyna znajduje się w jednym z możliwych stanów, a głowica „obserwuje” jedną z klatek na taśmie. Działanie maszyny jest bardzo proste. Głowica czyta symbol znajdujący się na taśmie i zgodnie z instrukcjami programu może go pozostawić bez zmian lub zamienić na inny symbol oraz przesunąć się o jedną klatkę w lewo lub w prawo na taśmie. Po czym maszyna może przejść w inny stan. Wykonując kolejno instrukcje programu, maszyna Turinga „oblicza”.

Opisy działania komputera koncentrujące się (podobnie jak w przypadku maszyny Turinga) na obliczaniu można określić jako opisy z punktu widzenia procesora (jednostki liczącej). Patric Hayes¹⁵ zapro-

¹⁴ Maszynę analityczną nazywa się „dziadkiem współczesnych komputerów”. Por. R. Ligonnère, *Prehistoria i historia komputerów od początków rachowania do pierwszych kalkulatorów elektronicznych*, tłum. z fr. R. Dulnicz, Zakład Narodowy im. Ossolińskich, Kraków 1992, 92.

¹⁵ P.J. Hayes, *What is a computer? An electronic discussion*, *Monist* 80(1997)3, 389–404.

ponował podejście alternatywne koncentrujące się na innym istotnym elemencie komputera – na pamięci.

Hayes twierdzi, że komputer jest maszyną, która operuje na wzorcach „wejściowych” i która zamienia je, zgodnie z instrukcjami, na inne wzorce „wynikowe”. Ulegające zmianie wzorce znajdują się w pamięć komputera, porównywanej przez autora do „magicznego papieru”, na którym zapis może się spontanicznie zmieniać lub na którym mogą pojawiać się nowe napisy. Różne rodzaje komputerów można wtedy definiować abstrakcyjnie w terminach składni ich języka wzorców, rodzaju zmian syntaktycznych, których mogą dokonywać, oraz teorii wiążącej wzorce z tymi zmianami.

Jeśli komputery można rozumieć zarówno jako urządzenia fizyczne, jak i jako byty abstrakcyjne, to pojawia się pytanie, jaki jest związek pomiędzy tymi obiektami. Odpowiedź na to pytanie nie jest łatwa. Jeśli traktować urządzenia nazywane komputerami jako realizacje pewnych bytów abstrakcyjnych, to wszystkie systemy fizyczne (i to nie tylko maszyny!) można nazwać komputerami. Każdy system można bowiem rozumieć jako realizację jakichś abstrakcyjnych bytów formalnych, takich jak zbiory czy równania różniczkowe, jednak nie możemy traktować każdego takiego systemu jako komputera. Jest to tak zwany problem uniwersalnej realizowalności: wszystko może być komputerem, ponieważ dla każdego obiektu można znaleźć taki jego opis, zgodnie z którym obiekt ten jest cyfrowym komputerem¹⁶. Co więcej, opis taki jest zawsze zależny od opisującego, a więc subiektywny. Można zatem wysunąć przypuszczenie, że „bycie komputerem” jest pierwotne w stosunku do pojęcia maszyny Turinga lub innej maszyny abstrakcyjnej¹⁷.

Rzeczywiste urządzenia nazywane komputerami istnieją i trudno zaprzeczyć temu, że nie wszystko jest komputerem. Próby zdefiniowania komputerów jako przedmiotu informatyki, pomimo że związane z wieloma trudnościami, nie są zatem bezsensowne, prowadzą one do

¹⁶ Por. J.R. Searle, *Is the Brain a Digital Computer?*, Proceedings and Addresses of the American Philosophical Association 64(1990), 21–37.

¹⁷ Por. P.J. Hayes, art. cyt.

głębszego rozumienia natury komputerów i stawiają przed filozofami nowe problemy, chociaż, jak twierdzą niektórzy, „nazwać tę dziedzinę [informatykę] »nauką o komputerach« – to tak, jak nazwać chirurgię »nauką o nożu«”¹⁸. Co jest zatem przedmiotem badań informatyków? Czym zajmują się informatycy?

4. PROGRAM

Wśród osób niezwiązanych zawodowo z informatyką powszechne jest przekonanie, że informatycy zajmują się przede wszystkim pisaniem programów komputerowych. Również sami informatycy często zgadzają się z tym poglądem, choć rozumieją programowanie szerzej niż tylko jako pisanie programów rozwiązujących problemy. Informatycy tworzą nowe algorytmy, nowe języki programowania, prowadzą badania teoretyczne związane na przykład z własnościami języków programowania, poprawnością programów czy też z optymalnością zaimplementowanych w tych programach algorytmów. Można zatem stwierdzić, że zajmują się oni teorią i praktyką programowania komputerów¹⁹, a przedmiotem badań informatyki są programy. Bornat pisze: „Budowanie programów, opisywanie programów, katalogowanie własności języków programowania, obserwowanie zachowania programów: to dokładnie to, czym jest aktywność nazywana »informatyką«”²⁰. Jeśli informatyka jest nauką o programach, to należy odpowiedzieć na pytanie o definicję i cechy programu. Czym jest przedmiot badań informatyków, nazywany programem komputerowym?

W podręcznikach informatyki znaleźć można definicję programu komputerowego jako ciągu instrukcji zapisanego w odpowiednim języku programowania. Na przykład M. Ben-Ari definiuje program jako „sekwencję symboli opisującą obliczenia zgodnie z pewnymi regułami

¹⁸ D. Harel, *Rzecz o istocie informatyki – Algorytmika*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001, 19.

¹⁹ Por. H. Khalil, L.S. Levy, *The academic image of computer science*, ACM SIG-SE Bulletin 10(1978)2, 31–33.

²⁰ R. Bornat, *Is 'Computer Science' science?*, 2006, maszynopis, 4.

zwanymi językiem programowania”²¹. Tak rozumiany program jest zatem bytem abstrakcyjnym. Napisy tworzące program można traktować jak wyrażenia matematyczne i argumentować, że opisują one z niespotykaną dokładnością zachowanie, zamierzone lub niezamierzone, komputerów, na których są wykonywane²². Jednak istnieje jeszcze inne rozumienie programu komputerowego.

James Fetzer²³ zwrócił uwagę na rozróżnienie programu jako napisów (ciągów instrukcji zapisanych w języku programowania) od programu jako procesu wykonywanego przez konkretne maszyny. Programy-napisy to byty abstrakcyjne (nieistniejące w czasie i przestrzeni, a charakteryzowane tylko za pomocą relacji logicznych), wykonywane na maszynie abstrakcyjnej. Z kolei programy-procesy są nieodłącznie związane z fizycznymi maszynami, na których są wykonywane. Takie rozróżnienie nazywane jest w literaturze dwuznacznością Fetzera.

Można twierdzić, że programy rozumiane jako procesy (p) wykonywane na komputerze są obiektami matematycznymi, ponieważ są one w pełni opisywane przez wyrażenia matematyczne (programy rozumiane jako napisy s_p). Eden pisze: „Programy-napisy s_p są wyrażeniami matematycznymi. Wyrażenia matematyczne reprezentują obiekty matematyczne. Program p jest w pełni i dokładnie charakteryzowany przez programy-napisy s_p . Zatem program jest obiektem matematycznym”²⁴.

Programy-napisy można traktować na przykład jako wzorce²⁵, rozumiane jako tablice komórek (lub miejsc) zawierających symbole, lub też jako funkcje ze zbioru stanów początkowych (lub „wejść”) w zbiór

²¹ M. Ben-Ari, *Understanding Programming Languages*, John Wiley & Sons, Chichester 2006, w: *Wikipedia* [http://pl.wikipedia.org/wiki/Program_komputerowy].

²² Por. C.A.R. Hoare, *An Axiomatic Basic for Computer Programming*, Communications of the Association for Computing Machinery 12(1969)10, 576–580.

²³ J. Fetzer, *Program Verification: The Very Idea*, Communications of Association for Computing Machinery 31(1988)9, 271–280.

²⁴ A.H. Eden, *Three Paradigms of Computer Science*, Minds and Machines 17(2007), 145.

²⁵ Por. P. Suber, *What is Software?*, Journal of Speculative Philosophy 2(1998), 89–119.

stanów końcowych (lub „wyjść”)²⁶, czyli jako obiekty matematyczne. Wtedy język programowania, w którym zapisano program, można rozumieć jako system formalny, dla którego program jest jedynie modelem. Takie podejście prowadzi nas do poglądu, że programy komputerowe pod niektórymi względami przypominają teorie naukowe.

Teorie naukowe oraz programy mają cechę wspólną: pomimo tego, że są obiektami o charakterze syntaktycznym, to jednak posiadają znaczenie semantyczne (ang. *semantic significance*). Istnieją jednak liczne różnice. Teorie naukowe, w przeciwieństwie do programów komputerowych, nie posiadają zdolności przyczynowych (ang. *causal capability*). Programy mogą bowiem wpływać na zachowanie się maszyn, na których są wykonywane, natomiast teorie nie wpływają bezpośrednio na opisywaną przez nie rzeczywistość, a więc programy nie są teoriami naukowymi. Czym zatem jest program komputerowy, jeśli nie jest on ani obiektem matematycznym, ani teorią naukową? Moore twierdzi: „Program komputerowy jest zbiorem instrukcji, które komputer może wykonać (lub co najmniej istnieje efektywna procedura przekształcenia ich do postaci, którą może wykonać komputer), by wykonać zadanie”²⁷.

Definicja ta uzależnia pojęcie programu od pojęcia komputera, ponieważ to, co uważamy za instrukcje dla komputera, i w konsekwencji to, co uważamy za program, jest wyznaczone przez operacje, które może wykonać maszyna. Przy takim podejściu status obiektu jako programu może ulegać zmianie w czasie. Pewien ciąg instrukcji mógł nie być programem wcześniej, przed skonstruowaniem odpowiednich komputerów, a jest programem komputerowym obecnie. Żeby uniknąć tej paradoksalnej własności, próbuje się definiować programy niezależnie od pojęcia komputera.

Pojęcie programu można związać na przykład z pojęciem algorytmu. Programy komputerowe wyrażają bowiem algorytmy, używa-

²⁶ J. Fetzer, *Philosophical Aspects of Program Verification*, Minds and Machines 1(1991), 197–216.

²⁷ J.H. Moore, *Three myths of computer science*, The British Journal for the Philosophy of Science 29(1978), 214.

jąc języków programowania, a wykonanie programu przez komputer można traktować jak ożywienie algorytmu²⁸. Programy komputerowe są zatem swego rodzaju pomostem pomiędzy algorytmami (bytami abstrakcyjnymi) a komputerami (bytami konkretnymi). James Fetzer²⁹ zwraca uwagę na różne rozumienie terminu „program” używanego w informatyce. Programem nazywa się: (1) algorytm, (2) zakodowany algorytm, (3) zakodowany algorytm, który może zostać skompilowany, lub też (4) zakodowany algorytm, który może zostać skompilowany i wykonany przez maszynę. Wszystkie te określenia bezpośrednio nawiązują do pojęcia algorytmu.

5. ALGORYTM

Wielu informatyków uważa, że pojęcie algorytmu jest centralnym pojęciem ich dyscypliny i że inne przedmioty takie, jak informacja, program czy komputer pełnią jedynie funkcję pomocniczą. Programy są bowiem tylko implementacją algorytmów, a komputery umożliwiają ich wykonanie. Przy takim podejściu to właśnie algorytm jest wspólnym mianownikiem i pojęciem unifikującym wszystkie, coraz liczniejsze, dyscypliny informatyki. Przedmiotem badań informatyków są zatem algorytmy. David Harel zatytułował swoją książkę, powszechnie używaną jako podręcznik, *Rzecz o istocie informatyki – algorytmika*, co dobrze oddaje omawiany pogląd³⁰.

Czym zatem jest przedmiot badań informatyków? Jaka jest natura algorytmu? Donald Knuth³¹ podaje następujące jego określenie: „Al-

²⁸ Por. J. Hartmanis, H. Lin, art. cyt.

²⁹ J. Fetzer, *Program Verification: The Very Idea*, art. cyt.

³⁰ Przyjmując taki pogląd, musimy uznać, że informatyka istniała na długo przed wynalezieniem pierwszych komputerów. Już starożytni Egipcjanie i Babilończycy byli informatykami, ponieważ sprawnie posługiwali się algorytmami do rozwiązywania zadań i to zarówno tych sprowadzających się do działań na liczbach, jak i zadań geometrycznych (potrafili obliczać między innymi pola powierzchni figur i objętości niektórych brył).

³¹ D.E. Knuth, *Computer science and its relation to mathematics*, *The American Mathematical Monthly* 81(1974)4, 323–343.

gorytm to dokładnie określony ciąg reguł mówiących o tym, w jaki sposób otrzymać konkretną informację wyjściową z danej informacji wejściowej w skończonej liczbie kroków. Szczególna reprezentacja algorytmów nazywa się programem³². W definicji tej znajdujemy podstawowe cechy algorytmu: określoność (rozumianą jako jednoznaczny opis każdego kroku) i skończoność (wykonanie zadania w skończonej liczbie kroków). Współcześnie wśród cech algorytmów wyróżnia się również: poprawność (algorytm daje poprawne wyniki), prostotę (operacje powinny być jak najprostsze) i efektywność. Prowadzi nas to do szeregu definicji zbliżonej do następującego określenia: „Algorytm jest »efektywną metodą«, przepisem pokazującym sposób wykonania zadania lub rozwiązania problemu, który wykonywany w sposób ścisły zakończy zadanie lub da poprawną odpowiedź po skończonej liczbie kroków³³. Efektywność w tej definicji jest pojęciem rozumianym intuicyjnie, blisko związanym z obliczalnością, które może być formalizowane na różne sposoby. Powstało wiele matematycznych precyzacji pojęcia efektywnej obliczalności, nie tylko wspomniane wcześniej maszyny Turinga, ale również: funkcje rekurencyjne, rachunek lambda (Churcha i Kleene’go), maszyny Posta czy algorytmy Markowa. Ciekawym z filozoficznego punktu widzenia wynikiem matematycznym jest stwierdzenie, że wszystkie te formalizmy opisują ten sam zbiór funkcji. Jest to argument na rzecz tezy, że matematyczne precyzacje dobrze oddają intuicyjny sens pojęcia metody efektywnej (jest to tzw. teza Churcha-Turinga).

Za jedną z najważniejszych cech algorytmu, oprócz efektywności, praktycznie zawsze wymienia się jego skończoność. Jednak w informatyce znaleźć można liczne przykłady procedur nieskończonych, od systemów operacyjnych aż po algorytmy matematyczne (takie jak algorytmy wyznaczania kolejnych cyfr rozwinięcia liczby π). Trzeba zatem albo rozszerzyć klasyczne pojęcie algorytmu i dopasować je do informatycznego użycia, albo też zrezygnować z definiowania informatyki jako nauki o algorytmach, na przykład na rzecz tezy, że przedmiotem

³² Tamże, 323.

³³ P. Suber, art. cyt., 19–20.

badania informatyki są procedury³⁴. Wtedy oczywiście pojawia się szereg innych problemów, na przykład kwestia odróżnienia algorytmu od procedury, problem zdefiniowania procedur oraz określenia ich cech. Wydaje się, że takie podejście nie rozwiązuje problemu ontologicznego dotyczącego określenia przedmiotu badań informatyki, lecz tylko przenosi go z mówienia o algorytmach do mówienia o procedurach.

Podsumowując, można stwierdzić, że pojęcie algorytmu, choć często używane i, zdawać by się mogło, ściśle zdefiniowane, wymaga refleksji filozoficznej, którą wymusza jego współczesne używanie w informatyce. Obecnie dalecy jesteśmy od zgody, co do natury algorytmu jako przedmiotu badań informatyki.

6. ZAKOŃCZENIE

Pokazane powyżej próby określenia przedmiotu badań informatyki wiążą się zawsze z problemami związanymi z naturą tych przedmiotów. Niezależnie od tego, czy za główny przedmiot zainteresowań informatyków uznamy informację, komputery, programy, czy też algorytmy, zawsze dochodzimy do wniosku, że brakuje pogłębionej refleksji filozoficznej co do ich definicji i własności.

Uznanie informatyki za naukę o informacji prowadzi do wielu ciekawych problemów filozoficznych związanych nie tylko z definicją wszechobecną w nauce informacji, ale również z jej naturą. Czy informacja to tylko fizyczne dane, czy też istotne jest ich znaczenie? Czy powinniśmy szukać ilościowej teorii informacji (w stylu teorii komunikacji Shannona), czy też skoncentrować się na poszukiwaniu teorii semantycznej i próbować odpowiedzieć na pytania w rodzaju: jaki jest związek informacji z wiedzą, prawdziwością, fałszem lub błędem?, kiedy informacja jest użyteczna?

Przed podobnymi trudnościami stajemy, gdy uznamy, że przedmiotem badań informatyki są komputery. Komputer można bowiem rozumieć dwojako: jako urządzenie fizyczne i jako obiekt abstrakcyjny, będący modelem takiego urządzenia. Jednak nawet, zdawałoby się,

³⁴ S. Shapiro, *Computer Science: The Study of Procedures*, 2001, maszynopis.

prostsze podejście do komputera jako do urządzenia zmusza nas do zastanowienia się, które maszyny (lub części maszyn) należy uznać za komputery, a które nie. Jakie cechy powinno mieć urządzenie, by móc nazywać je komputerem? Z kolei charakterystyka komputerów w terminach maszyn abstrakcyjnych jest zbyt szeroka (prowadzi do uznania, że wszystko jest komputerem cyfrowym). Powstaje również pytanie o związek pomiędzy komputerem rozumianym jako byt abstrakcyjny a rzeczywistymi maszynami będącymi jego urzeczywistnieniami. Otwarty pozostaje też problem rozróżnienia *hardware'u* od *software'u*³⁵.

Jedną z podstawowych działalności informatyków jest tworzenie programów komputerowych. Stąd pomysł, by traktować informatykę jako naukę o programach. Jednak program komputerowy można rozumieć i interpretować na wiele sposobów. Najistotniejsze z filozoficznego punktu widzenia jest rozróżnienie programów jako kodów (ciągów instrukcji) od programów wykonywanych na komputerach. Powstaje zatem pytanie o naturę programu. Czym jest program w sensie abstrakcyjnym? Próby odpowiedzi na to pytanie są bardzo różne: mówi się, że program jest obiektem matematycznym lub teorią naukową. Z kolei program rozumiany jako wykonywany na komputerach ściśle wiąże pojęcie programu z pojęciem komputera, które również nie jest wolne od problemów.

Pojawienie się bytów, jakimi są programy komputerowe, zmusza filozofów do nowego spojrzenia na pewne kwestie filozoficzne. Przykładem może być klasyczne rozróżnienie w filozofii obiektów abstrakcyjnych i konkretnych. Zakłada się najczęściej, że każdy obiekt jest albo abstrakcyjny, albo konkretny. Na przykład algorytmy, podobnie jak przestrzenie topologiczne czy liczby, są obiektami abstrakcyjnymi, podczas gdy dyski twarde i komputery są obiektami konkretnymi (fizycznymi). Jednak w takim podziale obiektów programy nie znajdują

³⁵ Pojęcia *hardware* i *software* używane są w języku polskim w wersji oryginalnej lub też tłumaczone są odpowiednio jako sprzęt i oprogramowanie, co z filozoficznego punktu widzenia może być mylące. Więcej na ten temat można znaleźć w: P. Suber, art. cyt.

swojego miejsca. Są one bowiem zarówno abstrakcyjne (programy-napisy), jak i konkretne (procesy wykonywane na konkretnych maszynach). Czy mają one zatem szczególny (dualny) charakter? Jeśli tak, to czy są jedynymi obiektami tego rodzaju?

Chcąc uniknąć wymienionych powyżej trudności związanych z określeniem programu jako przedmiotu badań informatyki można twierdzić, że informatycy badają algorytmy lub procedury. Wtedy jednak należy zapytać, czym są algorytmy i jaka jest ich natura. Okazuje się, że powszechnie przyjmowaną definicję algorytmu trzeba albo zmienić, albo porzucić, ponieważ nie pasuje ona do wszystkich zastosowań tego terminu w informatyce. Co więcej, cechy algorytmu, takie jak na przykład efektywność, należy doprecyzować i wyjaśnić. Powstaje również pytanie o wzajemne relacje takich pojęć, jak: przepis, algorytm i procedura.

Wszystkie próby definiowania informatyki poprzez przedmiot jej zainteresowań wiążą się z licznymi problemami. Można zatem porzucić takie próby na rzecz stwierdzenia, że informatyka to to, czym zajmują się informatycy. Co więcej, można twierdzić, że szukanie przedmiotu badań tej dyscypliny to tylko poszukiwanie etykietek, które nie są istotne; istotne jest tylko to, że informatycy wiedzą, czym się zajmują. Wydaje się jednak, że nawet etykiety są z wielu powodów ważne. Z jednej strony wyznaczają status akademicki informatyki, oznaczają zawód, określają pośrednio standardy kształcenia oraz metody pracy informatyków. Informatyka jako dyscyplina akademicka (i nie tylko) powinna być odróżnialna od innych dyscyplin, takich jak: matematyka, fizyka czy chemia, ale również od robienia na drutach i budowania domów. Z drugiej strony, samozrozumienie i jasność conceptualna są ważną częścią dojrzałej nauki.

Dick Hamming podczas Turing Lecture w roku 1968 ostrzegał, że obraz danej dyscypliny może w znaczący sposób wpływać na jej późniejszy rozwój³⁶. Pomimo tego, że nie możemy w sposób definitywny odpowiedzieć na pytanie, co jest przedmiotem badań informatyków,

³⁶ R.W. Hamming, art. cyt.

to musimy co jakiś czas sprawdzać i odświeżać nasze poglądy na to, czym zajmuje się i czym powinna się zajmować ta dyscyplina.

BIBLIOGRAFIA

- Atchison W.F., Conte S.D., Hamblen J.W., Hull T.E., Keenan T.A., Kehl W.B., McCluskey E.J., Navarro S.O., Rheinboldt W.C, Schweppe E.J., Viavant W., Young D.M., *Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science*, Communications of the ACM 11(1968)3, 151–197.
- Ben-Ari M., *Understanding Programming Languages*, John Wiley & Sons, Chichester 2006, w: Wikipedia [http://pl.wikipedia.org/wiki/Program_komputerowy].
- Bondecka-Krzykowska I., *Historia obliczeń. Od rachunku na palcach do maszyny analitycznej*, Wydawnictwo Naukowe UAM, Poznań 2012.
- Bornat R., *Is 'Computer Science' science?*, 2006, maszynopis.
- Denning P.J., *What Is Computer Science?*, American Scientists 73(1985) (January-February), 16–19.
- Denning P.J., *Computer science: the discipline*, w: *Encyclopedia of Computer Science*, red. A. Ralston, E.D. Reily, D. Hemmendinger, Nature Pub. Group 2000–2034.
- Eden A.H., *Three Paradigms of Computer Science*, Minds and Machines 17(2007), 135–167.
- Fetzer J., *Philosophical Aspects of Program Verification*, Minds and Machines 1(1991), 197–216.
- Fetzer J., *Program Verification: The Very Idea*, Communications of Association for Computing Machinery 31(1988)9, 271–280.
- Floridi L., *Information*, w: *The Blackwell Guide to the Philosophy of Computing and Information*, red. L. Floridi, Blackell Publishing 2004, 40–61.
- Floridi L., *Semantic Conceptions of Information*, w: *Stanford Encyclopedia of Philosophy* 2005 [<http://plato.stanford.edu/entries/information-semantic/>].

- Forsythe G., *A university's educational program in computer science*, Communications of the ACM 10(1967)1, 3–11.
- Gorn S., *The computer and information sciences: a new basic discipline*, SIAM Review 5(1963)2, 150–155.
- Hamming R. W., *One man's view of computer science*, Journal of the ACM 16(1969)1, 3–12.
- Harel D., *Rzecz o istocie informatyki – Algorytmika*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.
- Hartmanis J., *Some Observations about the Nature of Computer Science*, w: *Foundations of Software Technology and Theoretical Computer Science: 13th Conference, Bombay, India, December 15–17, Proceedings*, red. R.K. Shyamasundar, Springer-Verlag Lecture Notes in Computer Science 761, Berlin 1993, 1–12.
- Hartmanis J., Lin H., *What is Computer Science And Engineering?*, w: *Computing the Future: A Broader Agenda for Computer Science and Engineering*, red. J. Hartmanis, H. Lin, National Academy Press, Washington 1992, 163–216.
- Hayes P. J., *What is a computer? An electronic discussion*, Monist 80(1997)3, 389–404.
- Hoare C.A.R., *An Axiomatic Basic for Computer Programming*, Communications of the Association for Computing Machinery 12(1969)10, 576–580.
- Ifrah G., *Historia powszechna cyfr*, tłum. z franc. K. Marczevska, K. Szeżyńska-Mačkowiak, Wydawnictwo W.A.B., Warszawa 2006.
- Khalil H., Levy L.S., *The academic image of computer science*, ACM SIGCSE Bulletin 10(1978)2, 31–33.
- Knuth D.E., *Computer science and its relation to mathematics*, The American Mathematical Monthly 81(1974)4, 323–343.
- Ligonnière R., *Prehistoria i historia komputerów od początków rachowania do pierwszych kalkulatorów elektronicznych*, tłum. z fr. R. Dulinicz, Zakład Narodowy im. Ossolińskich, Kraków 1992.
- Moore J.H., *Three myths of computer science*, The British Journal for the Philosophy of Science 29(1978), 213–222.

- Newell A., Perlis A.J., Simon H.A., *Computer science*, Science 157(1967), 1373–1374.
- Searle J.R., *Is the Brain a Digital Computer?*, Proceedings and Addresses of the American Philosophical Association 64(1990), 21–37.
- Shapiro S., *Computer Science: The Study of Procedures*, 2001, maszynopis.
- Suber P., *What is Software?*, Journal of Speculative Philosophy 2(1998), 89–119.

ABOUT THE OBJECT OF COMPUTER SCIENCE'S STUDIES

Abstract. One of the fundamental ontological issues of the philosophy of computer science is the question about the object of computer scientists' research. In the paper some views concerning the objects of computer science are analyzed. They include information, computers, computer programs and algorithms.

Recognition of computer science as a science of information leads to many philosophical problems related not only to the definition of information, but also of its nature. Similar issues are associated with the adoption of computers as the subject of computer science's research. In the paper the relationship between the computer as an abstract object and its implementation – real machines is examined. Also the problem of distinguishing hardware from software is considered. Finding that the main computer scientist's activity is software creation, leads to the view that computer science is the science of programs. The main views on the nature of programs, including the distinction between program as code and program as an executed process are compared and discussed. The paper also contains an analysis of algorithms and procedures as objects of computer science's research, including a discussion of the main features of the algorithm.

Keywords: philosophy of computer science, ontology of computer science, information, computer, software, algorithm