

IZABELA BONDECKA-KRZYKOWSKA

INFORMATYKA JAKO NAUKA

Informatyka jako dziedzina wiedzy jest dyscypliną stosunkowo nową. Wyrośla z matematyki oraz inżynierii i współcześnie również czerpie z obu tych dyscyplin. Zauważmy jednak, że trudno jednoznacznie określić, czym ona właściwie jest, jakie objekty bada i jakich używa metod. I dalej, choć może wydawać się, że wiemy, jaka to gałąź nauki, możemy nie orientować się, czym dokładnie zajmują się jej badacze.

Czy więc na pewno wiemy, jaka jest natura informatyki i jakie miejsce zajmuje ona wśród innych dyscyplin?

Angielska nazwa informatyki – *computer science* – sugeruje, że jest ona dziedziną nauki ścisłej¹, pojęcie to budzi jednak kontrowersje zarówno wśród informatyków, jak i filozofów. Wiele prac filozoficznych, w szczególności anglojęzycznych, poświęcono kwestii, czy informatyka jest nauką ścisłą (*science*), czy też gałęzią inżynierii.

Tradycyjnie w filozofii nauki przyjmuje się podział dyscyplin naukowych na formalne, jak matematyka i logika, oraz realne, a wśród nauk realnych wyróżnia się nauki przyrodnicze (biologia, chemia, fizyka), inżynieryjne, humanistyczne i społeczno-ekonomiczne. Gdzie więc, w takim podziale nauk, jest miejsce informatyki? Przyjrzyjmy się różnym stanowiskom badaczy rozważających to zagadnienie, poczynając od szukania odpowiedzi na pytanie, czy informatyka jest nauką formalną.

Dr IZABELA BONDECKA-KRZYKOWSKA – Zakład Logiki Matematycznej, Wydział Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu; adres do korespondencji: ul. Umultowska 87, 61-614 Poznań; e-mail: izab@amu.edu.pl

¹Angielskie słowo *science* oznacza nauki ścisłe (czyli nauki matematyczne i przyrodnicze) w odróżnieniu od nauk humanistycznych czy inżynierii, a więc nie „naukę” w ogólności.

INFORMATYKA JAKO DZIAŁ MATEMATYKI

Twierdzenie, że informatyka jest nauką formalną – lub po prostu działem matematyki – wynika najczęściej z czynionego *implicite* założenia, że informatycy zajmują się głównie tworzeniem programów i badaniem ich własności. Jest ona zatem nauką o programach i programowaniu. Programowanie należy przy tym rozumieć szeroko – jako proces rozpoczynający się tworzeniem specyfikacji, czyli definiowaniem wymagań stawianych programowi, a kończący się weryfikacją – sprawdzeniem jego poprawności. Na wszystkich etapach tego procesu korzysta się z metod formalnych. Wniosek: informatyka jest tylko kolejnym działem matematyki, matematyki stosowanej.

Jednym z prekursorów tego stanowiska jest Charles A.R. Hoare (HOARE 1969; HOARE 1989). Uważał on, że komputery są maszynami matematycznymi, których działanie można precyzyjnie przedstawić za pomocą programów komputerowych (wyrażeń matematycznych). Języki programowania, opisujące ich instrukcje, Hoare traktował jako teorie matematyczne zawierające definicje, aksjomaty i twierdzenia. Dzięki nim programista może stworzyć program zgodny ze specyfikacją i sprawdzać, czy rzeczywiście spełnia on stawiane przed nim założenia. Jest zatem programowanie aktywnością matematyczną, której uprawianie wymaga konsekwentnego stosowania tradycyjnych metod matematycznego rozumowania, obliczeń i dowodu. Hoare zdawał sobie jednak sprawę, że taki obraz informatyki to pewien ideał, do którego należy dążyć, i że współczesna mu informatyka jest odeń daleka.

Traktowanie informatyki jako działu matematyki znalazło wielu zwolenników. Jednym z nich jest Edsger Dijkstra (DIJKSTRA 1974, 1986). Uważa on mianowicie, że miejsce informatyki w przedstawionym powyżej podziale nauk znajduje się obok matematyki i logiki.

Dijkstra proponuje nawet nazywanie jej skrótem: VLSAL (*Very Large Scale Application of Logic*). Czy można by w jakiś sposób potwierdzić lub obalić taki pogląd?

Wielu filozofów postuluje odróżnienie nauk formalnych (matematyki i logiki) od realnych, które bazowałyby na badaniu natury zdań w ich obrębie. Twierdzenia matematyki i logiki są zdaniem analitycznymi, w przeciwieństwie do zdań syntetycznych pozostałych nauk. Zdania analityczne to te, które są prawdziwe na mocy znaczenia językowego, np. „Jeżeli Jan jest starszy od Piotra, to Piotr jest młodszy od Jana”. W zdaniach syntetycznych natomiast wartość logiczna nie może być ustalona bez odwoływania się do rzeczywistości innej niż językowa, np. „Poznań jest stolicą Polski”.

Jeśli uznać, że zdania nauk formalnych są analityczne, podczas gdy nauk empirycznych są syntetyczne, to dla stwierdzenia, czy informatyka jest nauką formalną czy realną, wystarczy analiza jej zdań.

Spróbujmy zatem odpowiedzieć na pytanie, czy zdania, z których zbudowane są programy komputerowe, są analityczne czy też syntetyczne.

Rozważmy typową instrukcję języków programowania: operację podstawienia – np. $A:=13*74$. Wyrażenia tego typu można rozumieć dwojako: jako rozkazy (zmiennej A przypisz iloczyn liczb 13 i 74) lub jako zdania oznajmujące. Oczywiście rozkazy nie są zdaniami w sensie logicznym, a więc nie mogą być ani prawdziwe, ani fałszywe – i co za tym idzie, nie mogą być rozumiane jako analityczne bądź syntetyczne. Rozważmy zatem powyższy zapis jako zdanie oznajmujące, stwierdzające, że zmienna oznaczana przez A otrzymuje wartość odpowiedniego iloczynu. A może być przy tym traktowane jako fizyczne miejsce w pamięci albo też jako oznaczenie bytu abstrakcyjnego. W pierwszym przypadku operację podstawienia rozumiemy jako stwierdzenie: fizyczne miejsce w pamięci A przyjmuje wartość fizycznego obliczenia $13*74$. Jest ono wówczas swego rodzaju przypuszczeniem co do tego, co stałoby się, gdyby operacja podstawienia została wykonana. Należy przy tym pamiętać, że najczęściej operacje podstawienia nie są wykonywane wprost (tak jak je zapisano w językach programowania) – są one kompilowane (tłumaczone za pomocą specjalnych programów) na podstawowe instrukcje maszynowe wykonywane podczas pracy programu. Zdanie reprezentujące operację podstawienia jest zatem tylko przypuszczeniem co do wyniku wykonania programu, i jest zdaniem syntetycznym, ponieważ jego prawdziwość zależy od rzeczywistości.

I dalej: jeśli A traktujemy jako abstrakcyjne miejsce w pamięci, to odwołujemy się tym samym do maszyn abstrakcyjnych. Należy zauważyć, że mówiąc o programach i ich działaniu, trzeba rozróżnić odwoływanie się do maszyn fizycznych i abstrakcyjnych. Gdy analizujemy program i jego *z a m i e r z o n e* działanie oraz zakładamy, że urządzenie, na którym program zostanie wykonany, nie zawiedzie, to odwołujemy się do maszyny abstrakcyjnej. Natomiast jeśli mówimy o programie i jego *a k t u a l n y m* działaniu – np. analizując jego efektywność – odnosimy się do urządzenia fizycznego. Nasuwa się przypuszczenie, że interpretując programy komputerowe jako działające na maszynach abstrakcyjnych przyjmujemy tym samym, że występujące w nich instrukcje są zdaniami analitycznymi. Czy tak jest rzeczywiście? Rozważmy, za Timothy Colburnem (COLBURN 2000), w jaki sposób możemy stwierdzić prawdziwość zdania postaci:

(*) Abstrakcyjne miejsce w pamięci A przyjmuje wartość abstrakcyjnego obliczenia $13*74$.

Stwierdzenie, jaką wartość przyjmie abstrakcyjne miejsce w pamięci komputera podczas wykonania programu, jest podobne do matematycznego założenia co do wartości zmiennej typu: niech $a = 965$. Takie zdanie nie jest ani analityczne, ani syntetyczne, i co więcej, trudno mówić o jego prawdziwości – po prostu założenie zostało zrobione. Tak więc wyrażenie (*) jest podobne do twierdzenia matematycznego – i to takiego, w którym występuje założenie, brak jednak tezy. Jest ono zatem mało ciekawe, podobnie jak matematyczne twierdzenie rozpoczynające się od „niech...”, ale bez następującego „...wtedy”. Każde wyrażenie dowolnego języka programowania można rozumieć abstrakcyjnie, a więc program komputerowy można traktować jako ciąg warunków w świecie abstrakcyjnym. Tak rozumiany program, podobnie jak ciąg matematycznych „niech”, nie jest jednak wyrażeniem matematycznym w żadnym interesującym nas sensie.

Wyrażenia języków programowania składające się na program komputerowy są więc albo zdaniami syntetycznymi, albo też fragmentarycznymi wyrażeniami matematycznymi, których nie można traktować jako zdania analityczne. Analiza zdań informatyki nie zezwala nam zatem na jednoznaczne zaklasyfikowanie informatyki do nauk formalnych².

Nie wszyscy jednak zgadzają się z poglądem, że informatyka jest nauką o programach i programowaniu. I tak, jeśli przyjmiemy, że obiektem badań informatyki nie są programy, lecz algorytmy³, twierdzimy tym samym że główny przedmiot jej zainteresowania jest rzeczywiście częścią matematyki. Algorytmy, badane obecnie przez informatyków, były i są rozważane przez matematyków, i to od początków istnienia matematyki jako nauki. Analogicznych argumentów można by, oczywiście, użyć na rzecz tezy, że to matematyka jest częścią informatyki i dojść do absurdałnego wniosku o równości obu dyscyplin.

Wzajemne związki matematyki i informatyki są niezaprzeczone⁴. Informatyka wpływa na matematykę na rozliczne sposoby: komputery wykorzystywane są do wykonywania szczególnie trudnych obliczeń numerycznych, badania własności obiektów matematyki, wspomaganie dowodzenia twierdzeń bądź też tworzenia dowodów w pełni automatycznych. Ponadto istnieje oczywisty związek między informatyką i matematyką na polu analizy numerycznej, logiki czy teorii liczb. Co więcej, także dzięki badaniu samych algorytmów powstało wiele płodnych i ciekawych problemów matematycznych; wprowadzają one nowe fascynujące

² Z twierdzeniem, że informatyka jest nauką formalną, ściśle związany jest paradygmat matematyczny. Więcej na temat paradygmatów w informatyce zob np. w EDEN 2007 oraz BONDECKA-KRZYKOWSKA 2010b.

³ Więcej na temat obiektu badań informatyki zob. BONDECKA-KRZYKOWSKA (w druku).

⁴ Por. BONDECKA-KRZYKOWSKA 2010a.

idee tam, gdzie ich brakowało (KNUTH 1974). Widzimy więc, jak znaczący wpływ wywiera informatyka na matematykę.

Wyniki matematyki klasycznej tak często okazują się użyteczne w informatyce, że może to wręcz zaskakiwać. Aparatura pojęciowa matematyki jest niezbędna informatykom do badania własności algorytmów, np. do dowodzenia ich poprawności, optymalności czy też efektywności. Okazuje się również potrzebna w procesie tworzenia specyfikacji programów.

Niezaprzeczalne wpływy i związki tych dwóch nauk nie dają jednak podstaw do stwierdzenia, że tworzą one tę samą gałąź wiedzy. W czasach nam współczesnych informatyka wykorzystywana jest przez wiele, jeśli nie przez większość, nauk przyrodniczych, np. przez biologię, ale nie twierdzimy przecież, że jest ona kolejnym jej działem.

Podsumowując, należy stwierdzić, że przedstawiona powyżej analiza zdań informatyki nie daje nam jednoznacznej odpowiedzi, czy są one analityczne, czy też nie. Wprawdzie niektóre twierdzenia mówiące o programach – w szczególności o ich poprawności – mogą być rozumiane jak twierdzenia matematyczne, a metody formalne wykorzystujemy w informatyce (np. w tworzeniu specyfikacji czy też weryfikacji programów), nie gwarantuje to jednak poprawności sądu, że informatyka jest gałęzią matematyki.

Wniosek: traktowanie informatyki jako kolejnej nauki formalnej – obok matematyki i logiki – jest nieuzasadnione. Wielu informatyków twierdzi, że w klasyfikacji nauk uprawiana przez nich dyscyplina nie powinna być umieszczana nie tylko w obrębie matematyki, ale nawet obok niej!

INFORMATYKA JAKO NAUKA EMPIRYCZNA

Wielu przeciwników traktowania informatyki jako nauki formalnej głosi pogląd, że jest ona nauką empiryczną. W naukach tych bowiem kluczową rolę odgrywa formułowanie i eksperymentalne testowanie hipotez – a te właśnie procesy znaleźć można również w informatyce. Najczęściej przyjmuje się, że tylko eksperymenty testują teorie i tylko one ujawniają nowe cechy badanych obiektów i zjawisk. Pozwalają one także formułować nowe teorie, lepiej wyjaśniające rzeczywistość. Przyjrzyjmy się zatem uważniej roli, jaką eksperymenty odgrywają w informatyce.

Zwolennicy twierdzenia, że informatyka jest nauką przyrodniczą, uważają często, że obiektem jej badań są programy komputerowe, których wykorzystanie nie ogranicza się tylko do rozwiązywania problemów. Służą one bowiem do

eksperymentalnego potwierdzania hipotez naukowych. Na przykład Timothy Colburn, wymieniając różne rozumienia testowania w informatyce, zwraca uwagę: „[...] można uruchomić program, by sprawdzić hipotezy w komputerowym modelu rzeczywistości” (COLBURN 2000, s. 171).

Uprawianie informatyki jako nauki eksperymentalnej widać w wielu zastosowaniach symulacji. Testowanie hipotez za pomocą programu przebiega w trzech etapach. Formuluje się hipotezę dotyczącą badanego zjawiska, następnie zaś, pisząc program komputerowy, modeluje się ją. Ostatnim etapem jest testowanie hipotezy poprzez uruchomienie programu. Wyniki wykonania programu decydują o tym, czy hipotezę uznaje się za potwierdzoną, czy też nie. Wiele zjawisk fizycznych, których nie można przetestować w rzeczywistości, bada się przeprowadzając ich komputerowe symulacje.

Testowanie hipotez ma kluczowe znaczenie dla całej metodologii nauk przyrodniczych. Możemy zatem stwierdzić, że programowanie – jako część informatyki – pod wieloma względami podobne jest do nauk przyrodniczych lub nawet, bardziej ogólnie, do nauk eksperymentalnych. Obiekty takie jak algorytmy czy specyfikacje można bowiem traktować jak hipotezy, które sprawdza się poprzez uruchomienie programów. Tego rodzaju testowanie pokazuje adekwatność wykorzystanego w programie modelu świata rzeczywistego. Programy są zatem tylko narzędziami w procesie sprawdzania – podobnie jak aparatura badawcza – i służą do potwierdzania lub obalania hipotez.

Niezwykle istotne, poza testowaniem już postawionych hipotez, jest w naukach eksperymentalnych formułowanie hipotez nowych – sprawdzanych później eksperymentalnie – zawierających przypuszczenia co do natury badanego zjawiska. Podobnie rzecz ma się w przypadku programów komputerowych. Na podstawie wyników symulacji wysuwa się nowe hipotezy dotyczące badanych zjawisk – np. geologicznych czy pogodowych – w celu poznania ich natury i konsekwencji. Symulacje służą zatem nie tylko wyjaśnianiu zjawisk, prowadzą one także do formułowania prognoz i przewidywań, które są nieodłącznym i oczywistym elementem uprawiania nauk eksperymentalnych.

Ekspertymentowanie z wykorzystaniem symulacji komputerowych jako metoda badawcza w wielu naukach nie czyni samej informatyki nauką eksperymentalną. Komputery, podobnie jak termometry czy mikroskopy, to instrumenty wykorzystywane w badaniach naukowych. Czy jednak można znaleźć metody eksperymentalne w samej informatyce?

Ekspertymentowanie za pomocą komputerów to nie tylko narzędzie w innych naukach, ale również w obrębie samej informatyki. W literaturze znaleźć można różnorodne poglądy dotyczące obiektu jej badań. Wśród nich jest taki, że bada

ona różne aspekty informacji, programy to modele pewnych procesów informacyjnych, a komputery ułatwiają ich sprawdzanie. Jedyna różnica między informatyką a innymi naukami eksperymentalnymi polega na tym, że informacja nie jest organizmem biologicznym ani też fizycznym – jak energia czy materia. Walter F. Tichy pisze:

Fakt, że – w dziedzinie informatyki – obiektem dociekań jest informacja, a nie energia lub materia, nie czyni różnicy w stosowalności tradycyjnej metody naukowej. Aby zrozumieć naturę procesów informacyjnych, informatycy muszą obserwować zjawiska, formułować wyjaśnienia i teorie oraz testować je⁵.

Przykładem eksperymentalnego falsyfikowania w informatyce jest eksperyment Knighta i Levesona (TICHY 1998), w którym analizowano prawdopodobieństwo awarii w programach wielowersyjnych (ang. *multiversion programs*). Teoria przewidywała, że prawdopodobieństwo awarii takiego programu jest iloczynem prawdopodobieństw awarii poszczególnych wersji. Badacze zaobserwowali jednak, że jest ono w rzeczywistych systemach znacznie większe. Eksperyment sfalsyfikował zatem podstawowe założenie tradycyjnej teorii.

Eksperymenty pomagają również informatykom wywodzić teorie z obserwacji. Dobrym przykładem takiego ich zastosowania jest udoskonalenie sztucznych sieci neuronowych. Po teoretycznym rozważeniu własności takich sieci zostały one, dzięki wynikom eksperymentów, poprawione i udoskonalone.

Nie wszyscy jednak zgadzają się ze stanowiskiem, że informatyka jest nauką przyrodniczą. Nauki przyrodnicze bowiem badają tylko zjawiska naturalne, podczas gdy komputery są wytworami pracy człowieka. Tedre twierdzi, że:

[...] trudno zrozumieć, w jakim sensie średni czas przeszukiwań algorytmu haszującego A (który jest konstrukcją człowieka) zaimplementowanego na komputerze marki B (który jest konstrukcją człowieka), gdzie zarówno A, jak i B opierają się na teoretyczno-technicznych konstrukcjach współczesnych obliczeń (które są dziełem człowieka), mogłyby być prawami natury w sensie, w którym powinny one odkrywać więcej na temat naturalnie pojawiających się zjawisk. Zamiast tego mówi nam o tym, jak dobrze poprzedni informatycy wykonali swoją pracę⁶.

W myśl tego stanowiska, niezależnie od tego, jakie zjawiska uznamy za obiekt badań informatyki – informację, komputery, obliczenia, procedury, programy czy algorytmy – są one wszystkie wytworami człowieka, a nie zjawiskami naturalnymi. Jeśli pojęcie „nauka przyrodnicza” odnosi się tylko do dziedzin badających

⁵ TICHY 1998, s. 33.

⁶ TEDRE 2011, s. 367.

zjawiska naturalne, to informatyka, podobnie jak matematyka, nie jest nauką przyrodniczą.

Można jednak twierdzić, że nawet jeśli uznamy informatykę za naukę o artefaktach, jakimi są komputery, to badania z nimi związane nie ograniczają się do ich zastosowań w innych naukach. Komputery wymagają głębszej, wielodyscyplinarnej analizy bądź też oddzielnej nauki zgłębiającej ich działanie i związane z nimi zjawiska (NEWEL, SIMON 1976). Wszak wszystkim wiadomo, że są one wielorakie i złożone. Zaprogramowana działająca maszyna (komputer) podobna jest do żywego organizmu i jako taka może być badana metodami eksperymentalnymi. Allen Newell i Herbert A. Simon piszą:

Każda nowo zbudowana maszyna jest eksperymentem. [...] Każdy nowy program jest eksperymentem⁷.

Istnieją również inne – poza powyżej przedstawionym – poglądy związane z ontologią informatyki. Można np. twierdzić, że informatyka jest nauką o algorytmach lub procedurach (SHAPIRO 2001). Takie jej rozumienie implikuje tezę, że komputer jest tylko maszyną ogólnego przeznaczenia, wykonującą procedury, a istotą informatyki jest badanie tych procedur. Skoro zaś algorytmy i procedury występują w świecie naturalnym, informatyka – podobnie jak inne nauki przyrodnicze – bada zjawiska naturalne. Przyjrzyjmy się zatem, jakimi metodami informatycy badają te zjawiska.

Podobnie jak wśród specjalistów innych nauk przyrodniczych, tak i wśród informatyków znaleźć można zarówno teoretyków, jak i eksperymentatorów. Zadaniem teoretyków jest przy tym dowodzenie pewnych własności obliczalności, np. poprzez analizę cech algorytmów. Ich wysiłki zmierzają do udowodnienia, że pewne obliczenia nie mogą być wykonane przy danych ograniczeniach zasobów⁸. W wyniku ich pracy powstają m.in. nowe teorie logiczne i modele semantyczne, pomocne w projektowaniu programów i umożliwiające prowadzenie rozumowań formalnych ich dotyczących (w tym weryfikację). Eksperymentatorzy natomiast badają algorytmy, obserwując ich wykonania na komputerach. Sprawdzają różnorodne wersje procedur, testując je i porównując (np. według kryterium czasu wykonania). Wiele miejsca w swej pracy poświęcają oni na tworzenie takich algorytmów, które byłyby optymalne dla rozwiązania danych klas zadań. Przez pojęcie optymalności rozumiemy tu, że algorytm wymaga takiej

⁷ NEWEL, SIMON 1976, s. 114

⁸ Najsłynniejszym problemem, z jakim zmagają się informatycy teoretycy, jest próba odpowiedzi na pytanie, czy $P = NP$.

ilości zasobów, jaką teoretycy uznali za minimalną. Reasumując, można powiedzieć, że informatycy zajmujący się zastosowaniami badają procedury użyteczne w praktyce, a efektem tych badań są programy i systemy komputerowe.

Informatyka zatem, podobnie jak inne nauki przyrodnicze, zawiera zarówno część teoretyczną, jak i praktyczną, przy czym celem tej pierwszej jest zrozumienie świata (tej jego części, którą stanowią procedury) bez względu na możliwe ich zastosowania. Teoria i praktyka odgrywają więc tę samą rolę w informatyce, co i w innych naukach eksperymentalnych.

Polemikę z powyższym poglądem przedstawia Juris Hartmanis (HARTMANIS 1993). Porównuje on rolę eksperymentu i teorii w informatyce oraz w fizyce. Argumentuje, że w naukach fizycznych istnieje ścisła interakcja między teorią i eksperymentem, że zaobserwowane podczas eksperymentów nowe, nieznanne badaczom zjawiska, niezgadające się z obowiązującą do tej pory teorią, próbuje się ująć w ramach teorii nowych. W informatyce takiej „niezgodności” między teorią a obserwacją nie ma. Teorie informatyczne nie konkurują pod względem lepszego opisu rzeczywistości i wyjaśniania natury jej obiektów. Nie powstają również nowe, zgodne z wynikami eksperymentów, bądź też ujawniające niewyjaśnione anomalie czy niespodziewane zjawiska. W historii informatyki daremnie by szukać jakiegoś kluczowego eksperymentu, decydującego o ważności teorii, tak jak miało to miejsce wielokrotnie w historii fizyki. Jeśli zatem informatyka nie jest nauką eksperymentalną, to gdzie jest jej miejsce w tradycyjnym podziale nauk?

INFORMATYKA JAKO INŻYNIERIA

Nie wszyscy informatycy zgadzają się z rozumieniem informatyki jako nauki formalnej bądź też eksperymentalnej. Niektórzy twierdzą, że uprawiana przez nich dyscyplina jest po prostu gałęzią inżynierii. Nauki ściśle określają możliwe rozwiązania problemów, inżynieria natomiast wybiera spośród nich te, które są możliwe do zrealizowania ze względów praktycznych i ekonomicznych.

Richard Hamming (HAMMING 1969) twierdzi nawet, że informatyki nie powinno nazywać się *computer science*, lecz *computer engineering*, co pozwoliłoby uniknąć wielu nieporozumień. Zakłada on, że zasadniczym zadaniem informatyków jest szukanie praktycznych rozwiązań, czyli tworzenie systemów komputerowych, a nie teoretyzowanie.

Znaleźć też można badaczy o bardziej radykalnych poglądach. I tak np. zdaniem Fredericka Brookesa traktowanie informatyki jako nauki formalnej jest dla

tej dyscypliny szkodliwe (BROOKES 1996). Prowadzi bowiem do uznania teorii za ważniejszą od praktyki i powoduje, że jej młodzi adepci skłaniają się ku rozważaniom teoretycznym, zapominając o najważniejszym – o użytkownikach i ich problemach. Informatyka bowiem jest dyscypliną inżynierską i główny jej cel stanowi komputerowe rozwiązywanie problemów.

Także Michael S. Mahoney stwierdza, że zadaniem informatyków jest praca nad praktycznym zastosowaniem komputerów:

O komputerach warto myśleć, warto mówić i warto się nimi zajmować tylko dlatego, że są użytecznymi urządzeniami, które robią coś dla kogoś⁹.

W myśl tego stanowiska informatykę uznać należy za gałąź inżynierii, gdyż jej badacze zajmują się tworzeniem systemów komputerowych służących potrzebom użytkowników.

Z tym poglądem wiąże się paradygmat technokratyczny, który postuluje porzucenie w informatyce metod formalnych na rzecz inżynierskich. Metody te są odpowiedniejsze nie tylko podczas tworzenia specyfikacji programów i samych systemów komputerowych, ale przede wszystkim w procesie ich weryfikacji.

Zwolennicy traktowania informatyki jako inżynierii twierdzą, że aby tworzyć niezawodne systemy komputerowe, należy dopuścić inne niż formalne metody badania ich poprawności, w tym testowanie. Uruchamiając program dla reprezentatywnego zestawu danych wejściowych, sprawdza się, czy działa on poprawnie. Wybór tych danych jest niezwykle istotny, gdyż większości programów nie można uruchomić dla wszystkich możliwych „wejść”, jest ich bowiem bardzo dużo, może nawet nieskończenie wiele. W wyniku takiego sprawdzenia nigdy nie osiągnie się absolutnej pewności co do poprawności działania programu, można natomiast znaleźć w nim błędy, które należy usunąć. Stanowi to jeden z najczęściej wysuwanych zarzutów wobec testowania jako metody weryfikacji programów. Czy jednak konieczne jest wymaganie od programu absolutnej poprawności?

W celu określenia poziomu niezawodności jakiegoś urządzenia bądź też produktu powstającego w procesie tworzenia oprogramowania inżynierowie używają metod statystycznych. Nie ma tu miejsca na metody formalne (DE MILLO 1979). Ważne jest tylko to, czy dany produkt odpowiada stawianym mu wymaganiom – czy np. jego awaryjność nie jest większa niż z góry zakładana.

Oczywiście możliwość formalnego udowodnienia poprawnego działania programów i urządzeń byłaby bardzo pożądana – w szczególności dotyczy to apli-

⁹ MAHONEY 2002, s. 25.

kacji, w których bardzo ważne jest zapewnienie bezpieczeństwa takim systemom, jak kontrola lotów, nadzór elektrowni atomowych, programy wspomagające pracę lekarzy (w tym diagnostyczne), zabezpieczanie operacji giełdowych czy też kontrola i ograniczanie dostępu w bankach. Niestety, obecnie żaden producent systemów informatycznych nie może przedstawić formalnego dowodu poprawnego działania swojego produktu. Co więcej, nie da się sprawdzić właściwego funkcjonowania dużych systemów komputerowych (złożonych z setek programów połączonych sieciami komputerowymi) empirycznie, gdyż nie jest możliwe chociażby przewidzenie wszystkich możliwych wejść (dla których poddajemy taki system sprawdzeniu), a co dopiero ich przetestowanie (MACKENZIE 1992). Z praktycznego punktu widzenia prace nad stworzeniem formalnych dowodów poprawności programów nie mają sensu (SAVITZKY 1989), gdyż bardzo wiele z nich (może nawet większość), działających i użytecznych w praktyce, jest formalnie niepoprawnych¹⁰. Co więcej, jeśli nawet uda się podać dowód poprawności programu jako takiego, nie gwarantuje to jego prawidłowego działania w połączeniu z innymi programami. A przecież współcześnie tworzone aplikacje składają się z wielu komponentów... Może więc rzeczywiście „[...] programowanie jest bardziej sztuką niż nauką”¹¹, a informatyka powinna znaleźć się poza tradycyjną klasyfikacją nauk?

INFORMATYKA JAKO NOWA DYSCYPLINA

Jak staraliśmy się pokazać, informatyka jako nauka jest blisko związana z matematyką, logiką i inżynierią, a informatycy wykorzystują w swojej pracy zarówno metody formalne, jak i eksperyment (DENNING 1985). Informatyka jednak nie jest ani nauką formalną, ani eksperymentalną, ani też dziedziną inżynierii.

Informatyka nie jest nauką formalną na wzór matematyki. Różni się od niej, ponieważ matematycy mają do czynienia z twierdzeniami, procesami nieskończonymi i relacjami statycznymi, informatycy natomiast badają algorytmy, konstrukcje skończone i relacje dynamiczne. Z warunkiem realizowalności nieodłącznie związana jest skończoność, podczas gdy matematycy operują w swych badaniach pojęciem nieskończoności. Informatycy potrzebują relacji dynamicznych do konstruowania modeli świata, matematycy na ogół ograniczają się do relacji statycz-

¹⁰ Wynika to chociażby z niedokładności reprezentacji liczb rzeczywistych w komputerze. Liczby rzeczywiste są reprezentowane w komputerach z pewnym przybliżeniem, a nie dokładnie. Wyniki zatem otrzymane w działaniach na takich liczbach mogą być w najlepszym razie prawidłowymi przybliżeniami wyników poprawnych.

¹¹ SAVITZKY 1989, s. 377.

nych. Dzięki algorytmom natomiast przesuwają się centrum zainteresowania z modeli statycznych (w matematyce) do analizy procesów i automatyzacji (w informatyce). Matematycy pracują zazwyczaj w oderwaniu od rzeczywistości fizycznej, podczas gdy praca informatyków jest z nią blisko związana: programy komputerowe prowadzą samoloty, kierują ruchem ulicznym, są częścią większości sprzętu AGD, z jakim obcujemy na co dzień w naszych domach.

Informatyka jednak, podobnie jak i matematyka, jest niezbędnym narzędziem w różnorodnych dyscyplinach nauki. Obejmuje ona zarówno teorię, jak i eksperyment, podobnie jak większość nauk eksperymentalnych. Metody naukowe (w tym eksperymenty) znaleźć można zarówno w zastosowaniach informatyki (symulacje komputerowe), jak i w niej samej. Eksperyment w informatyce wykorzystywany jest jednak inaczej niż np. w fizyce bądź biologii. Najczęściej nie służy bowiem do potwierdzania lub obalania hipotez teoretycznych, ale do ustalania możliwości realizacji nowych algorytmów i systemów (DENNING 1999). Testowanie algorytmów przez tworzenie realizujących je programów ma na celu sprawdzenie pewnych zagadnień praktycznych, które mogłyby zostać pominięte w formalnych dowodach własności algorytmów, a nie usterek w samych tych dowodach. Tak więc wyniki eksperymentów mogą prowadzić do stworzenia bardziej realistycznych modeli, a to z kolei sprzyja powstawaniu bardziej adekwatnych teorii.

Nie można także uznać informatyki za naukę przyrodniczą, ponieważ nauki przyrodnicze badają twory natury (atomy, komórki, gwiazdy, planety itp.), informatycy zaś sami tworzą badane przez siebie obiekty. Oczywiście, jeśli uznamy, że informatyka bada informację, również informację naturalną, powszechnie występującą w przyrodzie (zauważmy, że nie jest to pogląd powszechnie przyjęty), to można by pokusić się o twierdzenie, że jednak jest ona nauką przyrodniczą¹².

Rozważając natomiast powiązanie informatyki z inżynierią, musimy stwierdzić, że jest ono znacznie silniejsze niż między naukami ścisłymi a ich inżynierskimi odpowiednikami (przykładem może być chemia i inżynieria chemiczna). Wynika to zarówno z tego, że rozwój elektroniki ma istotny wpływ na wybór, które problemy informatyki są interesujące, jak i z tego, że wiele metod algorytmicznych zostało zaprojektowanych pierwotnie do rozwiązywania problemów inżynierskich. Co więcej, komputery stały się do tego celu niezastąpione – w systemach projektowania CAD, symulatorach, sterowaniu produkcją itd. (DENNING 1999). Dlatego też niektórzy twierdzą, że informatyka jest dziedziną inżynierii. Trzeba jednak podkreślić, że nie jest ona po prostu sumą zastosowań komputerów do różnych problemów.

¹² Por. paragraf „Informatyka jako nauka przyrodnicza”.

Istota informatyki jest niezależna od zastosowań, abstrakcyjna i bada się ją m.in. za pomocą technik formalnych. Informatycy zajmują się praktycznym zastosowaniem matematyki, szukając odpowiedzi na pytania: Jakie algorytmy można obecnie zastosować do obliczenia pierwiastków równania? Jakie zasoby są potrzebne do wykonania tych obliczeń? Jak można oszacować ich błędy? W jakim języku można opisać dany algorytm? Jakiego sprzętu on wymaga?

Jest zatem informatyka, podobnie jak matematyka, narzędziem abstrakcyjnym, różnorodnie stosowanym (GORN 1963), a nie tylko dziedziną inżynierii. A czym jest jako nauka? Uznać ją można za młodą naukę składającą się z części dedukcyjnej, eksperymentalnej oraz inżynierskiej. Różni się przy tym tak bardzo od innych nauk, że należy traktować ją jako nową dyscyplinę, która w tradycyjnym ich podziale nie pasuje do żadnej z grup. Aby uwypuklić ten fakt, niektórzy badacze postulują zmianę jej nazwy.

Juris Hartmanis i Herbert Lin proponują *computer science and engineering*, co ma wyraźnie wskazywać na to, że informatycy traktują badane obiekty zarówno w sposób naukowy, jak i inżynierski. Termin „inżynieria” w powyższej nazwie odnosi się do zastosowań praktycznych – budowania sprzętu i oprogramowania, natomiast „teoria” zwraca uwagę na wszystkie nieeksperymentalne jej działy, zmierzające do budowania podstaw matematycznych dla opisywania, wyjaśniania i rozumienia różnych aspektów obliczalności.

Jeszcze większy nacisk na związek informatyki z zagadnieniami obliczalności kładzie Paul Abrahams, proponując następującą definicję¹³:

Moja osobista definicja dyscypliny i jej nazwa to „computology”: nauka o procesach obliczalnych i sposobach, w jaki mogą być one realizowane¹⁴.

INFORMATYKA A OBLICZANIE

Wielu naukowców twierdzi, że trzecim paradygmatem nauki, po teorii i eksperymentacie, staje się obecnie obliczanie, a współczesna nauka to nauka obliczeniowa. Warto zatem zastanowić się, czym jest obliczanie i jaki ma związek z informatyką jako nauką, szczególnie, że twierdzi się często, iż informatyka jest nauką o obliczaniu lub obliczalności.

¹³ Również inni naukowcy proponowali zmianę nazwy dyscypliny z *computer science* (która sugeruje jej bliski związek z naukami ścisłymi) na np. *computics* (MCKEE 1995) lub *computing science* (DIJKSTRA 1999), podkreślające jej związek z obliczaniem.

¹⁴ ABRAHAMS 1987, s. 473.

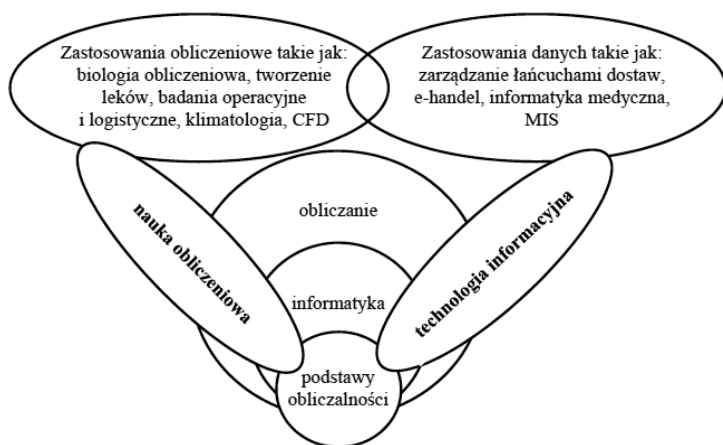
Jim Foley (FOLEY 2002) uważa, że obliczanie to informatyka oraz jej związek z innymi dyscyplinami naukowymi. Pojęcie obliczania pełni tu rozmaite funkcje:

- informuje o różnorodnych zastosowaniach komputerów w życiu człowieka – od poważnych obliczeń aż do rozrywki;
- pokazuje ważne problemy, które mogą zyskać rozwiązanie dzięki informatyce;
- stawia przed informatykami nowe problemy badawcze;
- wpływa na architekturę oraz własności fizycznych artefaktów, które wykonują obliczenia oraz przechowują i transmitują informację.

Informatyka zatem to nie jest obliczanie. Foley pisze:

[...] gdy mówię o obliczaniu i informatyce, mówię o dwóch różnych dyscyplinach a nie tylko o badaniach w obrębie dyscyplin¹⁵.

Obliczanie jako nauka zawiera informatykę, a więc łączy głębokie pytania teoretyczne dotyczące natury obliczania i informacji z dziedzinami, w których używa się komputerów (lub w których powinny one być używane). Trudno jest wskazać wyraźną granicę między informatyką a obliczaniem. Wielu informatyków prowadzi badania z zakresu obliczania. Ważnym aspektem takich badań jest ich interdyscyplinarność, powodująca, że ani obliczanie, ani inne dyscypliny nauki nie mogą niezależnie od siebie stworzyć nowej wiedzy. Dla zilustrowania związków między informatyką, obliczaniem, technologią informacyjną a nauką obliczeniową (ang. *computational science*) Foley przedstawia następujący schemat:



Ilustracja pokazuje, że informatyka jest tylko fragmentem obliczania jako nauki, zawierającym m.in. podstawy obliczalności.

¹⁵ FOLEY 2002, s. 6.

ZAKOŃCZENIE

Na podstawie przedstawionych rozważań można przyjąć, że informatycy wykorzystują w swej pracy zarówno eksperymenty, jak i rozważania teoretyczne. Oczywiście jest, że wyniki tych prac znacząco wpływają na osiągnięcia w wielu innych dziedzinach nauki, np. w ekonomii, naukach społecznych, naukach fizycznych, biologicznych czy w inżynierii.

Trudno jednak dać pełną odpowiedź na pytanie o miejsce informatyki w tradycyjnym podziale nauk. Nie jest ona bowiem całkowicie nauką formalną, choć podobnie jak matematyka jest użytecznym narzędziem uprawiania wielu innych dyscyplin. Nie jest ona także jedynie dyscypliną eksperymentalną bądź nauką przyrodniczą, ponieważ zarówno obiekt badań, jak i wykorzystywanie eksperymentu jako metody badawczej w informatyce znacznie odbiega od przedmiotu i metod badań stosowanych w innych naukach doświadczalnych. Podobnie też nie uznamy jej za inżynierię, albowiem istotą jej nie jest tylko rozwiązywanie problemów z wykorzystaniem komputerów.

Czym zatem jest informatyka? Jakie jest jej miejsce wśród innych nauk? To podstawowe i nadal otwarte pytanie o tożsamość informatyki jako nauki jest nie tylko problemem czysto filozoficznym, ma także swoje konsekwencje praktyczne. Pomaga bowiem odpowiedzieć, w jaki sposób i gdzie kształcić informatyków: na uniwersytetach obok wydziałów matematyki (lub nawet w ich obrębie) czy też na politechnikach i w innych szkołach technicznych.

Polska klasyfikacja dziedzin nauki i techniki z 2006 r. (znaleźć ją można m.in. w bazie „Nauka Polska”¹⁶) umieszcza informatykę (znajdującą się w grupie „1.3. Nauki o komputerach i informatyka”) wśród nauk przyrodniczych razem z matematyką (!), biologią i chemią, wszelako z zastrzeżeniem, że „opracowywanie sprzętu komputerowego” należy do: „2.2 Elektrotechnika, elektronika, inżyniera informatyczna” (do nauk inżynierskich i technicznych), a aspekty społeczne informatyki należą do: „5.8 Media i komunikowanie” (do nauk społecznych). Oczywiście nie rozwiązuje to filozoficznego sporu o tożsamość informatyki, daje jednak pewien wgląd w pojmowanie jej miejsca przez część naukowców.

Przyjąć można, że współcześnie przeważa interdyscyplinarne rozumienie informatyki, z którym wiąże się również specyficzne miejsce, jakie zajmuje ona pośród innych nauk. Jak sądzą niektórzy badacze, informatyka jest po części nauką przyrodniczą, ale także matematyczną czy techniczną, ponieważ zagadnienia, którymi się zajmuje, są kombinacją tradycji empirycznych, matematycznych i inżynierskich.

¹⁶ http://nauka-polska.pl/download/klasyfikacje_cdh/klasyfikacja_dziedzin_nauki_i_techniki.pdf

REFERENCJE

- ABRAHAMS Paul (1987), *What Is Computer Science?*, „Communications of the ACM” 30(6), s. 472-473.
- BONDECKA-KRZYKOWSKA Izabela (2010a), *O związkach informatyki z matematyką*, „Filozofia Nauki” 1, s. 77-89.
- BONDECKA-KRZYKOWSKA Izabela (2010b), *Paradygmaty informatyki*, [w:] *Światy matematyki. Tworzenie czy odkrywanie?*, red. Izabela Bondecka-Krzykowska, Jerzy Pogonowski, Poznań: Wydawnictwo Naukowe UAM 2010, s. 117-134.
- BONDECKA-KRZYKOWSKA Izabela (w druku), *O obiekcie badań informatyki*.
- BROOKS Frederick (1996), *The Computer Scientists as Toolsmith II*. „Communications of the ACM” 39(3), s. 61-68.
- COLBURN Timothy R. (2000), *Philosophy and Computer Science*, Armonk, NY: M.E. Sharpe.
- DE MILLO Richard A., LIPTON Richard J., PERLIS Alan J. (1979), *Social Processes and Proofs of Theorems and Programs*, „Communications of Association for Computing Machinery” 22, s. 271-280.
- DENNING Peter J. (1985), *What Is Computer Science?*, „American Scientists” 73, s. 16-19.
- DENNING Peter J. (1999), *Computer science: the discipline*, [w:] *Encyclopedia of Computer Science*, ed. Anthony Ralston, David Hemmendinger, John Wiley and Sons Ltd. (publikacja 2000).
- DIJKSTRA Edsger W. (1974), *Programming as a Discipline of Mathematical Nature*, „The American Mathematical Monthly” 81 (6), s. 608-612.
- DIJKSTRA Edsger W. (1986), *On a cultural gap*, „The Mathematical Intelligencer” 8 (1), s. 48-52.
- EDEN Amnon H. (2007), *Three Paradigms of Computer Science*, „Minds and Machines” 17, s. 135-167.
- FOLEY James (2002), *Computing > Computer Science*, „Computing Research News” 14 (4), s. 6.
- GORN Saul (1963), *The computer and information sciences: a new basic discipline*, „SIAM Review” 5 (2), s. 150-155.
- HALL Anthony (1990), *Seven myths of formal methods*, „IEEE Software” 7 (5), s. 11-19.
- HAMMING Richard W. (1969), *One man’s view of computer science*, „Journal of the ACM” 16 (1), s. 3-12.
- HARTMANIS Juris (1993), *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*, ed. Rudrapatna K. Shyamasundar, Lecture Notes in Computer Science 761, Berlin: Springer-Verlag, s. 1-12.
- HARTMANIS Juris, LIN Herbert (1992), *What Is Computer Science and Engineering?* [w:] *Computing the Future: A Broader Agenda for Computer Science and Engineering*, ed. Juris Hartmanis, Herbert Lin, Washington: National Academy Press, s. 163-216.
- HOARE C[harles] A[ntony] R[ichard] (1969), *An Axiomatic Basis for Computer Programming*, „Communications of the Association for Computing Machinery” 12 (10), s. 576-580.
- HOARE C[harles] A[ntony] R[ichard] (1989), *The Mathematics of Programming*, [w:] *Essays in Computing Science*” red. C[harles] A[ntony] R[ichard] Hoare, Prentice-Hall, Inc. Upper Saddle River, NY.
- KNUTH Donald E. (1974), *Computer science and its relation to mathematics*, „The American Mathematical Monthly” 81 (4), s. 323-343.
- MACKENZIE Donald (1992), *Computers, Formal Proofs and the Law Courts*, „Notices of the American Mathematical Society” 39, s. 1066-1069.
- MAHONEY Michael S. (2002), *Software as Science—Science as Software*, [w:] *History of Computing: Software Issues*, red. Ulf Hashagen, Reinhard Keil-Slawik, Arthur L. Norberg, Berlin: Springer Verlag.
- MCKEE George (1995), *Computer science or simply ‘computics’? The open channel*, „Computer” 28 (12), s. 136.

- NEWELL Allen, SIMON Herbert A. (1976), *Computer science as empirical inquiry*, „Communications of the Association for Computing Machinery” 19 (3), s. 113-126.
- PIERCE John R. (1968), *Keynote address. Conference on Academic and Related Research programs in Computing Science (5-8 June 1967)*.
- SAVITZKY Steve (1989), Letter in the Technical Correspondence section. „Communications of Association for Computing Machinery” 32 (3), s. 375-376.
- SHAPIRO Stuart C. (2001), *Computer Science: The Study of Procedures* (mps).
- TEDRE Matti (2011), *Computing as a Science: A Survey of Competing Viewpoints*, „Minds and Machines” 21, s. 361-387.
- TICHY Walter F. (1998), *Should Computer Scientists Experiment More?* „Computer” 31 (5), s. 32-40.

INFORMATYKA JAKO NAUKA

Streszczenie

Artykuł ten jest próbą odpowiedzi na pytanie o status informatyki jako dyscypliny wiedzy oraz o miejsce, jakie zajmuje ona pośród innych nauk.

Wśród informatyków i filozofów dominują cztery poglądy dotyczące statusu informatyki jako nauki. Pierwszym z nich jest twierdzenie, że informatyka to gałąź matematyki. Jeżeli podstawową działalnością informatyków jest pisanie programów – działalność matematyczna – to informatyka jest po prostu działem matematyki, czyli nauką formalną. Nie wszyscy jednak badacze zgadzają się z tym poglądem, twierząc, że pisanie programów to rodzaj eksperymentu, a informatyka jest nauką przyrodniczą. Eksperymentowanie za pomocą komputerów jest czymś więcej niż tylko metodą pomocniczą w innych naukach, odgrywa bowiem istotną rolę również w obrębie samej informatyki. Wielu informatyków uważa jednak, że traktowanie uprawianej przez nich dyscypliny jako nauki formalnej lub przyrodniczej jest niewłaściwe, ponieważ zaniedbuje podstawowe stawiane przed nią zadanie – rozwiązywanie problemów z wykorzystaniem komputerów. Informatyka jest zatem gałęzią inżynierii. Innym dominującym obecnie poglądem jest traktowanie informatyki jako nowej dyscypliny wiedzy, która wykorzystuje zarówno osiągnięcia inżynierii, jak i metody formalne i eksperymentalne. Sama nie jest jednak ani gałęzią matematyki, ani nauką przyrodniczą, ani też dziedziną inżynierii.

COMPUTER SCIENCE AS A BRANCH OF KNOWLEDGE

Summary

The article is an attempt to answer the question of the status of computer science as a branch of knowledge and its place among other disciplines. Four views on the status of computer science as a branch of knowledge dominate nowadays among computer scientists and philosophers. The first of them is a claim that science is a branch of mathematics. Because writing programs is a basic activity of computer scientists, which is in fact a mathematical activity, so a computer science is a formal science, simply a party of mathematics. But not all agree with this view—some specialists claim that writing computer programs is a kind of experiment and that computer science is a natural science. Experimenting using computers is not only a supporting method in other sciences but it plays an important role within the very computer science. However, many computer scientists believe that treating their discipline as a formal or natural science neglects the basic objective of computer science, i.e. problem solving using computers. So computer science is not a science at all, it is an engineering discipline. The fourth, currently dominating view claims that computer science

is a new discipline of knowledge that uses engineering, experimental and formal methods. But it is neither a branch of mathematics, nor a natural science, nor an engineering discipline.

Summarised by Izabela Bondecka-Krzykowska

Słowa kluczowe: informatyka, dziedzina nauki, obliczanie.

Key words: computer science, branch of science, computing.

Information about Author: IZABELA BONDECKA-KRZYKOWSKA, PhD—Department of Mathematical Logic at the Faculty of Mathematics and Computer Science at the Adam Mickiewicz University; address for correspondence: ul. Umultowska 87, PL 61-614 Poznań; e-mail: izab@amu.edu.pl