ROCZNIKI POLSKIEGO TOWARZYSTWA MATEMATYCZNEGO
Seria II: WIADOMOŚCI MATEMATYCZNE XV (1972)

STANISŁAW WALIGÓRSKI (Warszawa)

O zakresie informatycznego kształcenia na uniwersytetach

Rozszerzona wersja referatu wygłoszonego na konferencji w Sopocie w maju 1971 r.

Jeśli zastanawiamy się nad uniwersyteckim kształceniem w zakresie szeroko pojętej wiedzy o maszynach matematycznych, którą tu będziemy nazywać informatyką, to — jak się wydaje — dobrym punktem wyjścia dla tych rozważań może być tzw. „Curriculum 68". Są to zalecenia dotyczące programów akademickich w zakresie informatyki, opracowane przez specjalnie do tego celu powołany Komitet Programu Studiów Informatycznych ACM (*Curriculum Committee on Computer Science of the Association for Computing Machinery*). Chociaż członkami i konsultantami Komitetu byli przede wszystkim profesorowie uczelni amerykańskich, jednak „Curriculum" zostało uznane za dobre wytyczne również przez środowiska europejskie. Mówiąc o wykorzystaniu informacji i zaleceń zawartych w „Curriculum" na terenie polskim, warto jednak mieć na uwadze przynajmniej trzy kwestie: 1° Metody i tradycje uniwersyteckiego kształcenia w matematyce są u nas odmienne od amerykańskich i to rzutuje na związki informatyki z matematyką na naszych uniwersytetach. 2° Zapotrzebowanie na specjalistów z tej dziedziny, aktualne i przewidywane, oraz sposób zatrudniania absolwentów tej specjalności i wykorzystywania ich kwalifikacji są z natury rzeczy inne u nas niż w USA i Europie zachodniej. 3° Możliwości sprzętowe i kadrowe. Należy pamiętać, że kształcenie studentów i przygotowanie do tego kadry naukowej wymaga — dla większości działów informatyki — odpowiedniego sprzętu i oprogramowania. Bez tego kwalifikacje pracowników uczelni będą wielce problematyczne, bo w tej dziedzinie obok szerokiej i solidnej bazy teoretycznej nie mniejszą rolę odgrywają wiadomości praktyczne, w szczególności umiejętność praktycznego stosowania teorii, metod i narzędzi, którymi dysponuje informatyka. (Przyjmujemy, że „informatyka" w naszym rozumieniu oznacza dokładnie to samo, co „computer science" w sensie „Curriculum").

Autorzy „Curriculum 68" podzielili całą „computer science" na trzy części, a te z kolei na mniejsze działy w następujący sposób:

I. Struktury informacyjne i ich przetwarzanie. Ta część informatyki dotyczy reprezentacji struktur informacyjnych i ich przekształceń oraz mówi o teoretycznych modelach takich reprezentacji i przekształceń.

1. *Struktury danych*: opisywanie, reprezentowanie i manipulacje na liczbach, tablicach, listach, drzewach, kartotekach itp.; organizacja pamięci, rozmieszczanie informacji w pamięci, dostęp do pamięci; tworzenie wykazów, przeszukiwanie i sortowanie; techniki generowania, modyfikacji, transformowania i kasowania struktur; statyczne i dynamiczne własności struktur; algorytmy manipulowania zbiorami, grafami i innymi strukturami złożonymi.

2. *Języki programowania*: zapis algorytmów; syntaktyczny i semantyczny opis języków; analiza wyrażeń, zdań, deklaracji, struktur sterujących i innych zapisów stosowanych w językach programowania; struktury dynamiczne powstające w trakcie obliczeń; budowa, opracowywanie i ocena języków; efektywność programu i upraszczanie programów; sekwencyjne przekształcanie struktur programowania; języki specjalizowane; relacje między językami programowania i językami formalnymi a lingwistyką.

3. *Modele narzędzi obliczeniowych*: analiza struktury i zachowania układów przełączających i maszyn sekwencyjnych; własności i klasyfikacja automatów; algebraiczna teoria automatów i teoria modeli; języki formalne i gramatyki formalne; klasyfikowanie języków przez urządzenia rozpoznające; analiza syntaktyczna; formalny opis semantyki; przetwarzanie uwarunkowane przez składnię; problemy rozstrzygalności dla gramatyk; traktowanie języków programowania jako automatów; inne teorie formalne języków programowania i obliczeń.

II. Systemy przetwarzania informacji. Ta część dotyczy systemów zdolnych do transformowania informacji. W takich systemach zwykle występuje współdziałanie sprzętu i oprogramowania.

1. *Budowa i organizacja maszyn cyfrowych*: rodzaje struktur maszyn — maszyny von Neumanna, maszyny tablicowe i maszyny z przewidywaniem; hierarchie pamięci — rejestry przerzutnikowe, pamięci ferrytowe, dyski, bębny, taśmy — oraz techniki dostępu do nich; mikroprogramowanie i realizacja funkcji sterowania; układy wykonujące działania arytmetyczne; kody rozkazowe; układy wejścia i wyjścia; struktury wieloprogramowe i z wieloprzetwarzaniem.

2. *Translatory i interpretery*: teoria i metody stosowane przy budowaniu asemblerów, kompilatorów, interpreterów, programów ładujących, programów wydawniczych i tłumaczących (zmieniających środki zapisu, format itp.)

3. *Maszyny i systemy operacyjne*: nadzór nad działaniem programu i gospodarka danymi; programy użytkowe i nadzoru; biblioteki danych i programów; modularna organizacja systemów programowania; współdziałanie modułów i komunikacja między nimi; wymagania narzucane przez warunki pracy z wielodostępem, wieloprogramowością i wieloprzetwarzaniem; opisywanie i dokumentacja wielkich systemów; techniki usuwania błędów i diagnostyka; mierzenie parametrów eksploatacyjnych.

4. *Systemy specjalizowane*: maszyny analogowe i hybrydowe; specjalizowane urządzenia do transmisji danych; urządzenia ekranowe; urządzenia peryferyjne i pomocnicze dla specjalnych zastosowań; specjalizowane oprogramowanie takich urządzeń.

III. Metodyki. Metodyki te zostały opracowane i są używane w wielu różnych dziedzinach zastosowań, w których występują jednakowe struktury, procesy lub techniki.

1. *Matematyka numeryczna*: algorytmy numeryczne oraz ich właściwości teoretyczne i obliczeniowe; analiza błędów obliczeniowych (dla błędów zaokrąglenia i obcięcia); automatyczna ocena błędu i właściwości zbieżności.

2. *Przetwarzanie danych i prowadzenie kartotek*: techniki, które można stosować do systemów informacyjnych zarządzania, bibliotecznych, biomedycznych; języki przetwarzania danych.

3. *Przetwarzanie napisów*: operacje na wyrażeniach, takie jak upraszczanie lub formalne różniczkowanie; języki przetwarzania napisów.

4. *Przetwarzanie tekstów*: redagowanie, korekta i justowanie tekstów; planowanie układu tekstów; zastosowania analizy lingwistycznej; języki do przetwarzania tekstów.

5. *Grafika maszynowa*: cyfrowy zapis danych i pamięci dla takich zapisów; wyposażenie ekranów i generowanie obrazów; kompresja rysunków i uzyskiwanie wypukłości obrazu; geometria i topologia obrazów; perspektywa, obroty; analizowanie obrazów; języki graficzne.

6. *Symulacja*: modele naturalne i operacyjne; modele symulacji dyskretnej; modele o zmianach ciągłych; języki symulacyjne.

7. *Wyszukiwanie informacji*: indeksowanie i klasyfikacja; techniki statystyczne; automatyczne klasyfikowanie; strategie porównywania i przeszukiwania; wtórne wyjścia, takie jak abstrakty lub indeksy; systemy z selektywnym rozpraszaniem; automatyczne systemy odpowiadające na pytania.

8. *Sztuczna inteligencja*: heurystyka; modele mózgu; rozpoznawanie kształtów; dowodzenie twierdzeń; rozwiązywanie problemów; rozgrywanie gier; systemy adaptacyjne i uczące się; systemy człowiek—maszyna.

9. *Sterowanie procesami*: maszynowe sterowanie obrabiarek; sterowanie eksperymentami; systemy kierowania i sterowania.

10. *Systemy uczące*; nauczanie wspomagane przez maszynę.

Zestaw problemów, wymienionych przy każdym dziale, służy do ogólnego scharakteryzowania tego działu i ułatwienia klasyfikacji; nie wyczerpuje to oczywiście wszystkich zagadnień, które mogą być w ramach tego działu przedmiotem badań lub kształcenia, ale pozwala nakreślić ogólny zarys problematyki.

W zakres informatyki w podanym sensie nie włączono tych działów, które z racji swego charakteru należą raczej do matematyki, elektroniki lub innej dziedziny:

teoretyczna analiza numeryczna, teoria układów cyfrowych i impulsowych i metody ich projektowania, teoria informacji i kodowania oraz inne. Tak samo poza tak wyznaczonym zakresem informatyki pozostają zastosowania maszyn cyfrowych i metod informatycznych w technice, ekonomii itp., jak również zagadnienia wpływu rozwoju informatyki i stosowania maszyn cyfrowych na rozwój gospodarki i nauki.

Autorzy „Curriculum", podając zestaw zalecanych wykładów informatycznych, podzielili je na trzy grupy według poziomu: podstawowe (basic, B), pośrednie (intermediate, I) i zaawansowane (advanced, A). Ich zdaniem, wykształcenie podstawowe (na poziomie „undergraduate" w anglosaskiej organizacji studiów), powinno mieć charakter ogólny; aczkolwiek przewiduje się, że po wysłuchaniu wykładów podstawowych (B) studenci będą mieli pewną swobodę wyboru wykładów typu I, jednak, jak się podkreśla, powinno się przeciwdziałać naturalnej tendencji studentów do nadmiernej specjalizacji. Niekorzystny wpływ na poziom wiedzy absolwentów może mieć również ograniczanie przekazywanej im wiedzy przez zmniejszenie ilości wykładów lub zubożenie ich treści. Oczywiście zalecenia „Curriculum" pozostawiają pewną swobodę planowania programu w zależności od lokalnych warunków uczelni, ale bez obniżenia poziomu.

Dla pierwszych czterech lat studiów (undergraduate level) zalecane są następujące wykłady (obok nazwy podano poziom i numer kolejny, oraz ilość godzin wykładu i ćwiczeń):

B1. Wstęp do przetwarzania informacji	2 - 2
B2. Maszyny i programowanie	2 - 2
B3. Wstęp do struktur cyfrowych	2 - 2
B4. Rachunek numeryczny	2 - 2
I1. Struktury danych	3 - 0
I2. Języki programowania	3 - 0
I3. Organizacja maszyn	3 - 0 lub 3 - 2
I4. Programowanie systemów	3 - 0

Prócz tych wykładów, obowiązujących dla wszystkich, są do wyboru przynajmniej po 2 z następujących wykładów:

I5. Konstrukcja translatorów	3 - 0
I6. Teoria układów przełączających	3 - 0 lub 2 - 2
I7. Maszyny sekwencyjne	3 - 0
I8. Analiza numeryczna I	3 - 0
I9. Analiza numeryczna II	3 - 0

Ponadto obowiązuje odpowiedni zestaw wykładów czysto matematycznych. Chociaż opanowanie praktycznych umiejętności programowania i posługiwania się maszyną cyfrową nie jest głównym celem tego kursu, ale jest to ważny jego produkt uboczny. Należy więc dążyć do tego, aby studenci osiągali odpowiedni poziom w tym zakresie. Celowi temu może służyć rozwiązywanie przez studentów zadań o coraz większym stopniu trudności w ramach różnych wykładów i ćwiczeń,

praktyki, udział w wykonywaniu rzeczywistych projektów, praca w czasie wakacji, współpraca z ośrodkami obliczeniowymi itp.

Uzyskane na tym szczeblu szkolenia wykształcenie ogólne można pogłębić w pewnych kierunkach, takich jak programowanie systemów, organizacja i projektowanie maszyn, programowanie dla zastosowań naukowych, programowanie dla przetwarzania danych, wysłuchując odpowiednio dobranych wykładów poziomu A oraz z matematyki i innych dziedzin.

Wykłady na poziomie zaawansowanym, A, obejmują:

A1. Języki formalne i analiza syntaktyczna	3 - 0
A2. Organizacja maszyn, kurs zaawansowany	3 - 0
A3. Maszyny analogowe i hybrydowe	2 - 2
A4. Symulacja systemów	3 - 0
A5. Organizacja i wyszukiwanie informacji	3 - 0
A6. Grafika maszynowa	2 - 2
A7. Teoria obliczalności	3 - 0
A8. Wielkie systemy przetwarzania informacji	3 - 0
A9. Sztuczna inteligencja i programowanie heurystyczne	3 - 0

Na wyższym poziomie studiów (graduate level w systemie anglosaskim) student może już koncentrować swe zainteresowania na wybranych dziedzinach, takich jak te, które przykładowo zostały opisane niżej. Na tym etapie student powinien otrzymać odpowiednio szeroki zasób wiedzy, zarówno informatycznej, jak z pokrewnych dziedzin, pogłębionej w wybranym przez niego kierunku. W każdym przypadku wykształcenie powinno obejmować pewne działy ze wszystkich trzech części informatyki (struktury informacyjne i ich przetwarzanie, systemy przetwarzania informacji i metodyki — zob. str. 42 - 43). Szczegółowy wybór działów zależy od problematyki, jaką student ma się dokładniej zajmować. Część tego niezbędnego wykształcenia ogólnego można odebrać jeszcze na wykładach szczebla I, jeśli one są prowadzone na odpowiednim poziomie, albo na wykładach bardziej zaawansowanych i specjalistycznych. Oto przykłady możliwych wyborów kierunków zainteresowań, wraz z działami, które one obejmują:

Teoria maszyn matematycznych. Struktury danych I.1, języki programowania I.2, modele narzędzi obliczeniowych I.3, przetwarzanie napisów III.3, sztuczna inteligencja III.8, analiza kombinatoryczna, logika matematyczna, kodowanie i teoria informacji.

Języki programowania, programowanie z zastosowaniami. Struktury danych I.1, języki programowania I.2, budowa i organizacja maszyn cyfrowych II.1, translatory i interpretery II.2, maszyny i systemy operacyjne II.3, przetwarzanie napisów III.3, symulacja III.6, teoria optymalizacji, logika matematyczna.

Systemy cyfrowe i ich zastosowania. Struktury danych I.1, modele narzędzi obliczeniowych I.3, budowa i organizacja maszyn cyfrowych II.1, maszyny i systemy operacyjne II.3, grafika maszynowa III.5, teoria optymalizacji, logika matematyczna, układy cyfrowe i impulsowe, teoria informacji i kodowania.

Matematyka numeryczna. Struktury danych I.1, języki programowania I.2, budowa i organizacja maszyn cyfrowych II.1, maszyny i systemy operacyjne II.3, matematyka numeryczna III.1, symulacja III.6, teoretyczna analiza numeryczna, metody matematyki stosowanej, teoria optymalizacji.

Wykorzystanie systemów. Struktury danych I.1, języki programowania I.2, budowa i organizacja maszyn cyfrowych II.1, systemy specjalizowane II.4, symulacja III.6, sterowanie procesami III.9, metody matematyki stosowanej, teoria optymalizacji, teoria łączności i sterowania.

Systemy informacyjne. Struktury danych I.1, języki programowania I.2, budowa i organizacja maszyn cyfrowych II.1, maszyny i systemy operacyjne II.3, przetwarzanie danych i prowadzenie kartotek III.2, przetwarzanie tekstów III.4, wyszukiwanie informacji III.7, teoria optymalizacji, logika matematyczna.

Zagadnienia kształcenia informatycznego były po 1968 r. przedmiotem wielu studiów i badań, wykonywanych przez różne organizacje i w różnych krajach. Między innymi Międzynarodowa Federacja Przetwarzania Informacji (IFIP, *International Federation for Information Processing*), która dla informatyki ma podobne znaczenie, jak Międzynarodowa Unia Matematyczna dla matematyki, zorganizowała 3 konferencje poświęcone temu zagadnieniu: Western European Symposium on Computer Education, Londyn, marzec 1969, Eastern European Symposium on Computer Education, Balatonszeplak, Węgry, wrzesień 1969, IFIP World Conference on Computer Education, Amsterdam, sierpień 1970. Omawiano na nich aktualny stan kształcenia informatycznego w krajach o różnym stopniu zaawansowania oraz potrzeby i plany na przyszłość.

Z dostępnych materiałów wynika, że sposób kształcenia informatycznego na uniwersytetach zawsze mocno zależy od wyposażenia uczelni w maszyny cyfrowe. Te uczelnie, które mają dostateczny zasób własnych maszyn lub korzystają w odpowiednim zakresie z międzyuczelnianych wielomaszynowych systemów cyfrowych, realizują, ogólnie rzecz biorąc, program zbliżony do podanego w „Curriculum", z różnymi odchyleniami zależnymi od lokalnych warunków, personelu, tradycji, organizacji uczelni. Natomiast uczelnie źle wyposażone w maszyny z reguły tego całego programu nie realizują, ograniczając się często (choć niewątpliwie są to przypadki skrajne) tylko do bardzo uproszczonej nauki o metodach numerycznych i programowaniu, uzupełnionej ewentualnie elementarnymi wiadomościami o prostych maszynach cyfrowych i pewnymi informacjami czysto teoretycznej natury. Jest oczywiste, że takie ograniczenia decydują o poziomie wykształcenia w zakresie informatyki, odbieranego na tych uczelniach.

W przeszłości we wszystkich chyba krajach Europy informatyka rozwijała się „pod parasolem" matematyki i nauczanie uniwersyteckie informatyki zaczynało się często właśnie na wydziałach matematycznych. Ta sytuacja miała swoje niewątpliwe zalety, zwłaszcza we wczesnych stadiach rozwoju, ale miała także wady, które mogły odbić się negatywnie na rozwoju kształcenia. Jedną z nich było to,

że specjaliści matematycy mieli wyraźną tendencję ograniczania informatyki (i kształcenia informatycznego) tylko do metod rozwiązywania problemów matematycznych na maszynach cyfrowych. Mówiąc obecnie o stosunku wzajemnym informatyki i matematyki, warto zauważyć, że te dziedziny są bardzo bliskie sobie, gdyż metody badawcze i sposób rozwiązywania problemów w informatyce są niewątpliwie matematyczne w swej istocie. W tym sensie można uważać informatykę za część matematyki. Ale, co trzeba wyraźnie podkreślić, informatyka nie może być traktowana jako część matematyki w jej ujęciu tradycyjnym.

Z planowaniem kształcenia informatycznego wiąże się kwestia oszacowania aktualnego i przyszłego zapotrzebowania na specjalistów. To z kolei zależy od wyposażenia w maszyny matematyczne, od postępów informatyki i rozwoju zastosowań maszyn liczących. Sytuacja jest różna w różnych krajach, ale na ogół przyjmuje się (zob. np. [10]), że prędzej czy później każdy kraj osiągnie taki poziom wyposażenia w maszyny cyfrowe, że stosunek liczby zainstalowanych maszyn do liczby ludności będzie taki sam, jak w USA w 1970 r. (60 tysięcy maszyn na 200 mln mieszkańców). Wobec tego, że rozwój tej dziedziny jest bardzo szybki, realna ocena przyszłych zmian i perspektyw rozwoju różnych krajów w tej dziedzinie jest bardzo trudna i wszelkie oszacowania można robić tylko na stosunkowo krótkie okresy — nie więcej niż 5 lat. Istnieją czynniki, których działania nie można z należną dokładnością przewidzieć nawet w najstaranniej przygotowanych prognozach. Do takich czynników należą m. in. możliwe decyzje rządów wywierające bezpośrednio lub pośrednio wpływ na przyspieszanie lub zahamowanie rozwoju nauki lub pewnych jej dziedzin w różnych krajach.

Odnosząc się z należytą ostrożnością do rezultatów badań nad obecnym i przyszłym rozwojem tej dziedziny, można próbować wyciągnąć z nich pewne wnioski ogólne. Badania holenderskie [9] wykazały, że krzywa wzrostu stosunku liczby zainstalowanych maszyn cyfrowych do liczby ludności ma taki sam kształt dla poszczególnych krajów Europy Zachodniej i USA; krzywe te są tylko poprzesuwane względem siebie w czasie i wobec tego takie przesunięcie pomiędzy dwiema krzywymi dla dwu dowolnych krajów można by określić jako opóźnienie (lub wyprzedzenie) jednego kraju względem drugiego. Na przykład w 1969 r. oceniano tak rozumiane opóźnienie Holandii względem USA na 5 lat. W każdym razie możemy przyjąć, że proces rozwoju zastosowań maszyn matematycznych ma, ogólnie rzecz biorąc, zawsze taki sam charakter, niezależnie od tego, gdzie on przebiega; natomiast w pewnych krajach proces ten może przebiegać z opóźnieniem w stosunku do innych.

Według badań amerykańskich [4] koszty oprogramowania maszyn cyfrowych w 1950 r. stanowiły 5% całkowitych kosztów produkcji, utrzymywania i wykorzystywania maszyn, w 1965 r. 50 %, a obecnie ocenia się je na 80%. Równolegle z tym silnie wzrasta zapotrzebowanie na specjalistów, którzy mogliby projektować i opracowywać oprogramowanie maszyn i systemów cyfrowych. Badania nad efektywnością wykorzystania zainstalowanych maszyn wykazały, że bardzo często są one źle wykorzystywane i pracują poniżej swych możliwości; jedną z głównych przyczyn

tego stanu rzeczy jest niedobór odpowiednio wysoko wykwalifikowanych specjalistów oprogramowania.

Dane te są o tyle interesujące dla naszych rozważań, że właśnie uniwersytety mają kształcić takich fachowców od projektowania, wykonywania i wykorzystania oprogramowania; można również przyjąć, że większość tych ludzi będzie pracowała raczej na rzecz użytkowników maszyn cyfrowych, a nie producentów. Obserwacja zatrudnienia absolwentów sekcji „metod numerycznych" uniwersytetów polskich i oceny naszych potrzeb również potwierdzają wzrost zapotrzebowania na fachowców tego typu. Jednak zanim sformułujemy wnioski, dotyczące kształcenia informatyków, które mogłyby ze stwierdzenia takich potrzeb wynikać, musimy najpierw sprecyzować sens terminów „programowanie" i „oprogramowanie", gdyż w praktyce może tu powstać równie wiele istotnych nieporozumień, jak przy interpretowaniu innych terminów, związanych z informatyką.

W prymitywnej fazie rozwoju maszyn i ich zastosowań programy były pisane najczęściej w językach symbolicznych albo takich, jak FORTRAN czy ALGOL i na ogół nie zawierały więcej niż kilka tysięcy zdań w tych językach. Różne programy były tworzone oddzielnie i rzadko tworzono z nich większe jednolite systemy programowania, jak również rzadko opracowywano je dla większych systemów cyfrowych. Program taki mógł być napisany przez jednego człowieka lub przez bardzo mały zespół, a w każdym razie jeden człowiek o odpowiednich kwalifikacjach mógł ogarnąć i zrozumieć we wszystkich szczegółach działanie tego programu.

Obecnie, kiedy dość powszechnie stosuje się duże systemy cyfrowe ze zintegrowanym oprogramowaniem, skala problemów jest zupełnie inna i zupełnie inne są metody ich rozwiązywania. J. D. Aron [8] z IBM Federal Systems Center przyjmuje następującą klasyfikację programów: zawierające 10 000 zdań uważa się jeszcze za małe, średnie zawierają od 30 000 do 500 000 zdań, duże powyżej 500 000. Jak dotychczas, największy opracowany tam program liczył 6 000 000 zdań. Jest oczywiste, że w tej sytuacji żaden człowiek nie może objąć wszystkich szczegółów działania całości, ani nawet wszystkich szczegółów projektu takiego programu. Projektować i wykonywać takie programy mogą tylko odpowiednio zorganizowane i kierowane duże zespoły, składające się z ludzi wyspecjalizowanych w różnych kierunkach i w różnych technikach rozwiązywania problemów, jakie w toku takiego opracowania mogą powstać.

Jeżeli więc ma się kształcić specjalistów, którzy by mieli odpowiednie kwalifikacje do brania udziału w takich pracach, to trzeba im dać wiedzę odpowiednio szeroką, dobrze ugruntowaną i opartą na solidnej podbudowie teoretycznej.

Przyglądając się pracy absolwentów uniwersyteckich wydziałów matematyki w ośrodkach obliczeniowych i podobnych instytucjach, korzystających z maszyn cyfrowych, można by dojść do wniosku, że fakty przeczą tak sformułowanej tezie. Rzeczywiście, nasi absolwenci są w większości zatrudniani po studiach jako programiści i chyba nie wymaga się od nich ani zbyt szerokiej wiedzy informatycznej, ani, tym bardziej, gruntownych wiadomości teoretycznych. Powinni raczej sprawnie

posługiwać się językami programowania, używanymi w ich ośrodku obliczeniowym i umieć je wykorzystywać przy pisaniu programów dla określonych zastosowań. Wszystko to prowadzi do wniosku, że nasze ośrodki obliczeniowe potrzebują przede wszystkim ludzi, którzy opanowali praktycznie sztukę programowania i prostego posługiwania się maszynami liczącymi i ich oprogramowaniem — reszta ma raczej znaczenie drugorzędne, jak pewnego rodzaju wykształcenie ogólne. Wiadomo, co prawda, że niektórzy nasi absolwenci — przede wszystkim ci wybitniejsi — wykonują tu i ówdzie prace bardzo zaawansowane, opracowując na przykład nowe i skomplikowane systemy oprogramowania dla różnych zastosowań, ale są to nieliczne wyjątki, nie zmieniające ogólnego obrazu sytuacji.

Pomińmy tu kwestię rodzaju zadań wykonywanych obecnie w ośrodkach obliczeniowych; postęp w tej dziedzinie jest raczej nieuchronny. Zwróćmy za to uwagę na inną ważną przyczynę takiego stanu zatrudnienia absolwentów: słabość szkolnictwa średniego, które ma przygotowywać kadry programistów. Jest jasne, że nie wszystkie prace przy oprogramowaniu i przy wykorzystywaniu maszyn dla zastosowań muszą wykonywać specjaliści z wyższym wykształceniem — wiele z nich mogą wykonywać ludzie mający wykształcenie średnie zawodowe odpowiedniej specjalności, a do niektórych wystarczą nawet absolwenci kursów zawodowych. Ocenia się (zob. np. [10]), że w przeciętnym ośrodku obliczeniowym tylko około 25 % pracowników, mających kontakt z maszyną, musi mieć wyższe wykształcenie; procent niezbędnych specjalistów o najwyższych kwalifikacjach jest jeszcze niższy. Ale żeby zachować takie proporcje, trzeba mieć skąd brać te pozostałe 75 % pracowników bez wyższego wykształcenia, ale o kwalifikacjach odpowiednich do ich stanowiska. Poza tym, jeżeli nie ma żadnych względów, które by przeciwko temu przemawiały, kierownicy i dyrektorzy oczywiście chętniej zatrudnią absolwentów uniwersytetów niż wychowanków średnich szkół zawodowych. Wszystko to przyczynia się do powstania tego pozornego zapotrzebowania na raczej wąsko wyspecjalizowanych „praktyków", o którym była mowa wyżej.

Moglibyśmy zastanowić się nad pytaniem, od którego właściwie można było zacząć nasze rozważania: czy w ogóle warto kształcić informatyków na uniwersytetach? Ponieważ zapotrzebowanie na specjalistów informatyków istnieje i wszystko wskazuje na to, że będzie stale rosło, problem sprowadza się tylko do tego, kto i jak będzie ich kształcił. Doświadczenie innych krajów wskazuje, że jeżeli nie zorganizuje się tego kształcenia w należyty sposób, będą się nim zajmować różne wydziały rozmaitych uczelni, każdy niezależnie i z osobna, a także najrozmaitsze instytucje, między którymi mogą się znaleźć nawet takie, które nie mają ani dostatecznych możliwości organizacyjnych, ani nie mogą zapewnić odpowiedniego zakresu i poziomu szkolenia. Dochodzi wtedy do znacznego rozproszenia środków, a co za tym idzie, znacznej podwyżki ogólnych kosztów kształcenia przy obniżonej efektywności. Rozproszenie to utrudnia również właściwe wykorzystanie kadr wykładowców o odpowiednio wysokich kwalifikacjach. Poza tym instytucje czy wydziały uczelni, które zajmują się kształceniem informatycznym jako swego rodzaju działalnością uboczną, mogą przejawiać silne tendencje do ograniczania jego zakresu. Z jednej

strony może to wynikać z chęci podporządkowania celów kształcenia potrzebom, związanym z ich działalnością podstawową, z drugiej strony faktem, że na ogół łatwiej im o fachowców ze swej dziedziny (oraz zastosowań informatyki w tej dziedzinie) niż o dostateczną kadrę informatyków sensu stricto.

Wszystkie te względy przemawiają za tym, żeby kształcenie informatyków na wyższym poziomie prowadzić w sposób jednolity i odpowiednio zorganizowany, a działalność w tym kierunku powinna być w rozsądny sposób scentralizowana, aby unikając rozproszenia środków i kadr fachowych wykładowców można było nimi możliwie efektywnie i wszechstronnie gospodarować. W tych warunkach jest także łatwiej utrzymywać stały kontakt z nowymi osiągnięciami w tej bardzo szybko rozwijającej się dziedzinie i łatwiej wprowadzać wynikające stąd niezbędne zmiany, unowocześniające program szkolenia.

Zarówno dotychczasowy rozwój szkolenia informatycznego, jak sam charakter tej dziedziny zadecydowały o tym, że jest ona najsilniej związana z matematyką. Kształcenie informatyczne powinno zawsze opierać się na gruntownej wiedzy matematycznej. Najłatwiej to uzyskać, jeżeli prowadzi się je na tej samej uczelni, gdzie jest wydział matematyki (o ile tym kształceniem zajmuje się osobny, wyspecjalizowany wydział informatyki), albo, tak jak to jest między innymi w Polsce, na wydziałach matematyki. W kraju zajmują się tym sekcje tych wydziałów, zwane tradycyjnie sekcjami metod numerycznych, choć obecnie nazwa ta może mieć niewiele wspólnego z rzeczywistym zakresem szkolenia.

Podkreślając ścisłe związki organizacyjne z matematyką i podobny charakter metod badawczych musimy jednak pamiętać o tym, że w kształceniu informatycznym musi być zachowana właściwa równowaga między wiedzą teoretyczną a praktyczną. Umiejętności praktyczne, potrzebne informatykom, nie mogą być im przyswojone na wykładach teoretycznych, za pomocą tylko kredy i tablicy. Niezbędny jest do tego stały aktywny kontakt ze sprzętem liczącym i jego oprogramowaniem oraz skuteczne opanowanie zaawansowanych metod posługiwania się nimi. Student musi nauczyć się rozwiązywać — skutecznie i całkowicie, nie zadowalając się samym tylko istnieniem algorytmu lub stwierdzeniami ogólnikowymi — problemy, które można napotkać przy wykorzystywaniu istniejących systemów oprogramowania lub tworzeniu nowych.

To wszystko decyduje o tym, że kształcenie informatyczne musi być przynajmniej częściowo oparte na innych zasadach niż kształcenie w zakresie samej matematyki. Te same cechy odróżniające występują także w działalności zawodowej informatyków. Ze względu na szybki rozwój informatyki i jej ważność dla gospodarki, niezbędne jest prowadzenie szkolenia specjalistów tej dziedziny w ten sposób, aby nie tylko zaznajomić ich z aktualnym stanem wiedzy i zastosowań, ale także dać im niezbędne podstawy dalszego samodzielnego pogłębiania swej wiedzy i odnawiania jej już w czasie pracy zawodowej. Trzeba sobie zdawać sprawę z tego, że wiele wyników i metod, które są teraz w programach nauczania informatyki i są normalnie wykładane na uniwersytetach, nie istniało jeszcze dziesięć lat temu. A tempo rozwoju tej dziedziny nie słabnie.

Do trudności, wynikających z silnego uzależnienia jakości kształcenia od posiadania odpowiedniego sprzętu, dochodzą jeszcze dodatkowe utrudnienia, wynikające z nowości całej dziedziny i jej bardzo szybkiego rozwoju. To także, na równi z względami wymienionymi wyżej, przemawia za koniecznością oparcia kształcenia informatycznego na szerokich podstawach i unikania zbyt wąskiej specjalizacji. Silne oparcie o matematykę i istniejący dorobek matematyczny jest niewątpliwie dużą pomocą dla takiego kształcenia, chociaż nie rozwiąże wszystkich problemów — zwłaszcza tych, które wynikają ze specyficznych cech informatyki i jej rozwoju. Ale te problemy będą musiały być rozwiązane już przez samych informatyków, stosownie do potrzeb i posiadanych możliwości.

Prace cytowane

[1] *Curriculum 68*, Recommendation for Academic Programs in Computer Science, A Report of the ACM CCCS, Communications of the ACM, vol. 11, No. 3 1969, str. 151–197.

[2] G. E. Forsythe, *Computer science and education*, Information Processing 68, Proc, of the IFIP Congress, ed. A. J. H. Morrell, North Holland, Amsterdam 1969, str. 1025–1039.

[3] A. B. Frielink, et al., *Education in informatics in the Netherlands*, IAG Journal, vol. 4, No. 1. 1971, str. 74–89.

[4] C. E. Joseph, *Computers: Trends toward the future*, Information Processing 68, Proc. of the IFIP Congress, Amsterdam, 1969, str. 665–680.

[5] P. Naur, *„Datalogy", the science of data and data processes, and its place in education*, Information Processing 68, Proc. of the IFIP Congress, Amsterdam 1969, str. 1383–1387.

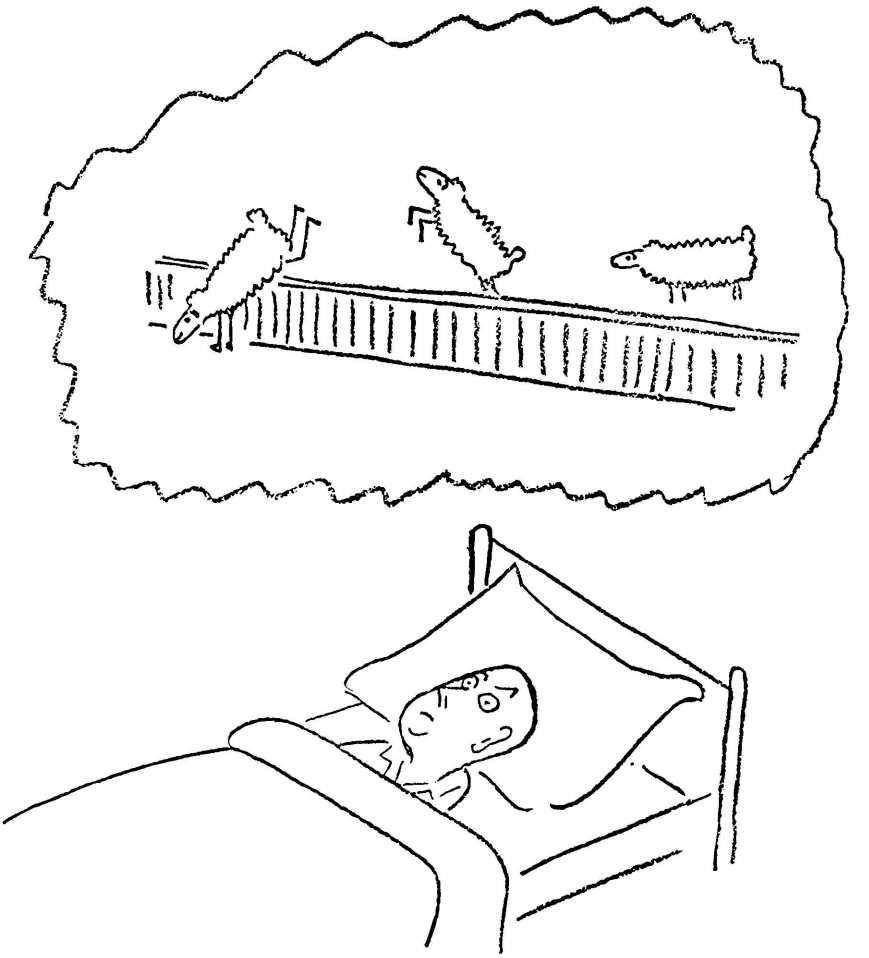
[6] B. Scheepmaker, *Functions in the field of computers in data processing*, Western European Symposium on Computer Education, London, March 1969.

[7] — and K. L. Zinn, *General Report on the IFIP World Conference on Computer Education 1970*, IAG Journal, vol. 4, No. 1 1971, str. 96–112.

[8] B. Shaw, ed., *The teaching of programming at university level*, Proc. of the Joint IBM–University of Newcastle upon Tyne Seminar, September 1970, University of Newcastle upon Tyne, 1971.

[9] D. Wolbers, referat nie publikowany, Eastern European Symposium on Computer Education, Balatonszeplak, September 1969.

[10] H. Zemanek, *Computer technology in education — how to make it viable*, Proc. of the IFIP World Conference on Computer Education 1970, North Holland, Amsterdam 1970, str. 344–433.



"$\ldots, \omega+2, \omega+3, \omega+4$"

R. C. BUCK

Reprinted from the Mathematics Magazine
Vol. 37, No. 5, November–December, 1964