

Paweł Stacewicz

Umysł a modele
maszyn uczących się

Spis treści

Część I. Sztuczne systemy uczące się

1	Systemy informatyczne	1
1.1	Podstawowe pojęcia informatyki	1
1.1.1	Informacja	2
1.1.2	Algorytm	5
1.1.3	Automat	7
1.2	Specyfika systemów informatycznych	9
1.2.1	Systemy informatyczne a maszyny tradycyjne	9
1.2.2	Typy systemów informatycznych	11
1.3	Narzędzia czy maszyny autonomiczne?	14
1.3.1	Systemy informatyczne jako narzędzia	14
1.3.2	Ku systemom autonomicznym	16
1.4	Podsumowanie	18
2	W stronę sztucznych systemów inteligentnych	21
2.1	Zagadnienie sztucznej inteligencji	21
2.1.1	Pojęcie sztucznej inteligencji	21
2.1.2	Praktyka badawcza i mity	23
2.1.3	Na pograniczu informatyki i filozofii	24
2.2	Badania nad sztuczną inteligencją	27
2.2.1	Rys historyczny	27
2.2.2	Główne cele i obszary badawcze	30
2.2.3	Wybrane techniki „inteligentne”	33
2.3	Logicyzm czy naturalizm?	44
2.3.1	Inspiracje logiczne i empiryczne	44
2.3.2	Odniesienia filozoficzne	48
2.4	Podsumowanie	49

3	Badania nad systemami uczącymi się	51
3.1	Przedmiot i zakres badań	51
3.1.1	Cechy ogólne systemów uczących się	52
3.1.2	Od przyczyn do celów badań	54
3.1.3	Obszary badań szczegółowych	57
3.2	Strategie automatycznego uczenia się	60
3.2.1	Środowisko, nauczyciel i wiedza	60
3.2.2	Strategie logicystyczne	66
3.2.3	Strategie naturalistyczne	69
3.3	Automatyczne uczenie się pojęć	71
3.3.1	Pojęcia jako rdzeń wiedzy	72
3.3.2	Pojęcia jako binarne funkcje decyzyjne	73
3.3.3	Maszynowe reprezentacje pojęć	78
3.3.4	Przykładowe algorytmy uczenia się pojęć	83
3.4	Podsumowanie	88

Część II. Od systemów informatycznych do modeli umysłu

4	Umysł jako system poznawczy	93
4.1	Pojęcie umysłu	93
4.1.1	Intuicyjne rozumienie terminu <i>umysł</i>	93
4.1.2	Próby naukowych uściśleń	96
4.1.3	Filozoficzny wymiar umysłu	100
4.2	Interdyscyplinarne badania nad umysłem	102
4.2.1	Struktura badań	102
4.2.2	Kognitywistyka	107
4.2.3	Dociekania filozoficzne	112
4.3	Podsumowanie	114
5	Uczenie się jako przejaw otwartości umysłu	117
5.1	Pojęcie uczenia się	117
5.1.1	Intuicyjny obraz uczenia się	117
5.1.2	Uczenie się jako czynność systemu	121
5.2	Wybrane badania nad uczeniem się	124
5.2.1	Badania psychologiczne	124
5.2.2	Koncepcje dydaktyczne	136
5.2.3	Potrzeba badań metodologicznych	139
5.3	Podsumowanie	142
6	Modelowanie czynności umysłowych	145
6.1	Modelowanie	145
6.1.1	Typy modeli	146
6.1.2	Od metafory do modelu	149
6.1.3	Schemat modelowania	151

6.1.4	Modelowanie a funkcjonalizm	153
6.2	Teorioinformacyjna strategia modelowania	156
6.2.1	Informatyczny rodowód strategii	156
6.2.2	Modele teorioinformacyjne a systemy informatyczne . . .	159
6.2.3	Modele teorioinformacyjne a maszyny Turinga	162
6.3	Podsumowanie	165
7	Wybrane teorioinformacyjne modele umysłu	167
7.1	Modele regułowe	167
7.1.1	Geneza modeli	167
7.1.2	Cechy strukturalno-funkcjonalne	169
7.1.3	Zagadnienia metodologiczne i filozoficzne	172
7.2	Modele sieciowe	178
7.2.1	Geneza modeli	178
7.2.2	Cechy strukturalno-funkcjonalne	179
7.2.3	Zagadnienia metodologiczne i filozoficzne	182
7.3	Modele ewolucyjne	187
7.3.1	Geneza modeli	188
7.3.2	Cechy strukturalno-funkcjonalne	189
7.3.3	Zagadnienia metodologiczne i filozoficzne	191
7.4	Podsumowanie	198
8	Ku otwartym modelom umysłu	203
8.1	Modele otwarte a systemy uczące się	203
8.1.1	Teorioinformacyjne modele otwarte	203
8.1.2	Czy modele otwarte mogą być autonomiczne?	206
8.1.3	Dwie groźby <i>regressus ad infinitum</i>	209
8.2	W stronę otwartych modeli ewolucyjnych	212
8.2.1	Populacyjny schemat modelowania	212
8.2.2	Ocena podejścia ewolucyjnego	217
8.3	Podsumowanie	218
	Literatura	221
	Skorowidz rzeczowy	229

Systemy informatyczne

Niniejszy rozdział jest poświęcony ogólnej charakterystyce systemów informatycznych ze szczególnym uwzględnieniem procesu, który przejawia się po części w historii informatyki (komputeryzacji), a po części w „przedłużających” tę historię wizjach futurystycznych. Mamy na myśli proces wiodący od pierwszych, całkowicie reaktywnych maszyn liczących, poprzez współczesne nam multimedialne systemy interaktywne, aż do hipotetycznych maszyn autonomicznych (które byłyby zdolne do samorzutnej aktywności).

Rozdział pierwszy odzwierciedla charakter całej książki. Są w nim obecne i analizy pojęciowe, i konkretne przykłady rozwiązań technicznych, i pewne rozważania ogólne, będące owocem filozoficznego namysłu nad tymi rozwiązaniami. Przede wszystkim zaś tekst ma charakter nie tyle informatyczno-techniczny, co filozoficzno-interdyscyplinarny.

1.1 Podstawowe pojęcia informatyki

Za podstawowe pojęcie informatyki należy uznać *informację*. W rozumieniu potocznym oznacza ono treść pewnego przekazu - wizualnego, dźwiękowego lub innego - która wzbogaca wiedzę odbiorcy lub pozwala mu podjąć określone działania. Treść ta jest jak najbardziej realna. Przekonuje o tym istnienie urządzeń zdolnych do jej magazynowania i przetwarzania, spośród których na pierwszy plan wybijają się komputery cyfrowe. Oprócz *informacji* za dwa inne kluczowe pojęcia informatyczne uznajemy *algorytm* i *automat*. Wybierając je, przychylamy się do dość powszechnego poglądu, iż **badane współcześnie systemy informatyczne przetwarzają dane (tj. zakodowane informacje) w sposób automatyczny i algorytmiczny**. Nie twierdzimy jednak, iż wszelkie systemy - tak naturalne, jak sztuczne - muszą przetwarzać informacje na podstawie algorytmów.

1.1.1 Informacja

Chociaż brzmienie nazwy *informatyka* sugeruje, iż oznaczana nią dyscyplina stanowi wiodącą naukę o informacji, a to z kolei może nasunąć wniosek, iż pojęcie informacji powinno zostać zredukowane do przedmiotu zainteresowania informatyków, to istnieją rozsądne argumenty za tym, by odróżnić informację w ogóle od informacji będącej przedmiotem informatyki. O ile zaś ta pierwsza byłaby czymś wieloaspektowym, możliwym do wykorzystania i zinterpretowania na różne sposoby (jej pojęcie mieściłoby w sobie tę wielość możliwości), ta druga byłaby tworzywem systemów informatycznych, na przykład maszyn cyfrowych. Biorąc za wzór pewne sugestie W. Marciszewskiego [Marciszewski 1991], proponujemy rozróżnienie powyższe uwyraźnić, nazywając informację w drugim rozumieniu *danymi*¹. Słowo to będzie oznaczać zatem informację badaną pod kątem metod jej automatycznego kodowania, magazynowania i przetwarzania; innymi słowy: specyficzne tworzywo systemów informatycznych².

Jeśli skupimy uwagę na funkcjach danych, a więc pytaniu o to, czemu dane służą, to dostrzeżemy natychmiast ich odmienność od dwóch innych tworzyw, również elementarnych i również przetwarzanych przez maszyny, a mianowicie od *materii* i *energii*. Chociaż bowiem nośnikami danych mogą być i pewne cząstki materialne, i pewne porcje energii, to w gruncie rzeczy **służą one do sterowania zakresem i kierunkiem przemian materialno-energetycznych**. Owo sterowanie może mieć charakter bezpośredni lub pośredni. Pierwsza sytuacja występuje wtedy, gdy maszyny informatyczne sprzęga się z maszynami tradycyjnymi (np. silnikami) w celu kierowania ich pracą. Druga zachodzi, gdy systemy informatyczne przekazują wyniki swoich działań użytkownikom, a dopiero ci podejmują dalsze decyzje co do ich wykorzystania.

Tę samą różnicę, którą już wyjaśniliśmy przy pomocy terminu *sterowanie*, daje się wyrazić poprzez odniesienie do *funkcji reprezentacyjnej* danych. Mamy tu na myśli fakt, że **pewien nośnik fizyczny uznaje się za informację dopiero wtedy, gdy reprezentuje coś różnego od siebie samego**. Poprzednio przyjęliśmy, że są to bezpośrednie lub pośrednie sygnały sterujące, teraz możemy wskazać obiekty nieco konkretniejsze, np. wyniki obliczeń, dźwięki i obrazy. Warto podkreślić, że obiekty reprezentowane nie muszą przynależać do świata zewnętrznego wobec danych, nie muszą też go dotyczyć. Jak poucza praktyka programowania komputerów, pewne dane - nazwijmy je wewnętrznymi - nie „wychodzą” poza wnętrze systemu przetwarzającego;

¹ Rozróżnienie zaproponowane przez W. Marciszewskiego ma nieco inną podstawę niż nasze. Otóż *informacją* nazywa on pewne obiekty abstrakcyjne i pojmowane przez ludzi intuicyjnie (np. liczby czy wartości logiczne), a *danymi* - ich fizyczne reprezentacje (np. cyfry czy fizyczne symbole wartości logicznych). W naszym ujęciu dane - będące badanym teoretycznie tworzywem systemów informatycznych - mają również charakter abstrakcyjny.

² Mimo to - ze względów estetycznych - w dalszej części podrödziału terminem *informacja* będziemy posługiwać się niekiedy jako stylistycznym odpowiednikiem słowa *dane*.

niektóre z nich informują o stanie systemu (np. wartości zmiennych systemowych), a niektóre sterują procesem przetwarzania innych danych (np. rozkazy maszynowe). Jeśli jednak skupimy uwagę na danych zastępujących obiekty zewnętrzne, to musimy dostrzec niedoskonałość funkcji reprezentowania. Wyraża się ona w tym, że każde faktyczne odwzorowanie obiektu zewnętrznego w danych pozostaje zawsze odwzorowaniem niepełnym, wybiórczym, dokonanym ze względu na cechy wybrane przez użytkownika i/lub możliwości techniczne systemu przetwarzającego.

Maszynowa realizacja funkcji reprezentowania nie mogłaby się powieść bez wynalezienia odpowiedniego kodu. Jak wiadomo, **w informatyce przyjęła się metoda kodowania cyfrowego (liczbowego), polegająca na przyporządkowywaniu obiektom reprezentowanym pewnych liczb, zaś operacjom na tychże obiektach - obliczeń**. Ze względu na techniczne właściwości komputerów odpowiedni kod ma postać *binarną*. W aspekcie fizycznym znaczy to, że wszelkie dane są odwzorowywane w pamięci maszyny jako sekwencje dwóch podstawowych stanów elementów przetwarzających, włączony i wyłączony; w aspekcie programistycznym zaś, że wszelkie liczby odpowiadające danym są zapisywane w układzie dwójkowym. Rozwiązanie takie przyjęło się, ponieważ łączy w sobie prostotę matematyczną z gwarancją niezawodności przekazu. Chodzi o to, że dwa jedyne składniki kodu, 0 i 1, stanowią, z jednej strony, niezbędne minimum do zapisania dowolnej liczby, a z drugiej strony - dają się realizować fizycznie jako dwa wyróżnione stany układu elektronicznego, włączony i wyłączony. Te dwa stany zaś różnią się od siebie tak krańcowo, że przy ich generowaniu i rozpoznawaniu niezmiernie trudno o błędy techniczne.

Teoretycznym rozwinięciem idei kodu binarnego jest sformułowana przez C. Shannona definicja *ilości informacji* [Shannon, Weaver 1949]. Zgodnie z nią przekaz informacji ma na celu jednoznaczne wskazanie odbiorcy jednego z n równie prawdopodobnych stanów, a **ilość informacji zawartej w konkretnym przekazie to minimalna liczba bitów (zer lub jedynek) niezbędnych do wskazania na jeden z n stanów**. Precyzyjnie wyraża to wzór logarytmiczny, $I = \log_2 n$. W celu zobrazowania tej suchej formuły sugestywnym przykładem założmy, że pewien przekaz ma dotyczyć poinformowania odbiorcy o tym, która z czterech dróg prowadzi do celu. Minimalna liczba zer i jedynek umożliwiających jednoznaczne wskazanie jednej z dróg wynosi 2, czyli $\log_2 4$ (zgodnie ze wzorem na I). Wynika to stąd, że najbardziej optymalną metodą zerojedynekowego ponumerowania dróg jest numeracja 00/ 01/ 10/ 11 (znaki „/” oddzielają kody kolejnych dróg), a więc z wykorzystaniem dwóch znaków dla każdej z dróg³.

Binarna definicja ilości informacji stanowi tylko fragment - choć jest to fragment podstawowy - dociekań, które składają się na informatykę niestososo-

³ Sukcesy technologii cyfrowych - ukazujące coraz to nowe możliwości kodu binarnego - popychają też niektórych teoretyków do utożsamiania informacji z kodem binarnym. Postępowanie takie nie wydaje się całkiem zasadne. Istnieją bowiem

waną, tj. teoretyczną. Na jej polu dane są traktowane jako przedmiot abstrakcyjny, czyli „oderwany” od możliwych zastosowań i interpretacji. Kwestię tę obrazuje dobrze porównanie informatyki teoretycznej z jej bliską krewną, matematyką. Otóż podobnie jak matematycy uprawiają teorię liczb, niezależnie od tego jakie obiekty będą za jej pomocą liczone (a także mierzone, dzielone itp.)⁴, tak również **informatycy rozwijają teorię przetwarzania danych niezależnie od tego, co konkretne dane będą reprezentować**. Abstrakcja, o której mowa, ma przynajmniej dwa poziomy. Na pierwszym obmyśla się i analizuje uniwersalne struktury danych (takie jak tablice, listy czy drzewa), na drugim - rozwiązuje się pewne podstawowe i wspólne różnym zastosowaniom problemy przetwarzania danych (takie jak problem sortowania czy przeszukiwania baz danych).

Reasumując: na polu informatyki ukształtował się obraz informacji jako tworzywa maszyn oddziaływujących na swoje środowisko pośrednio, za pośrednictwem sygnałów sterujących innymi maszynami lub możliwych do zinterpretowania przez użytkownika. Tworzywo to koduje się najczęściej cyfrowo, w postaci ciągów binarnych. Teoria jego możliwych przekształceń ma charakter zbliżony do matematyki, a przez to wysoce abstrakcyjny (por. [Harel 2000 (1987)]).

Naszkirowany obraz informacji zredukowanej do danych, czyli informacji będącej przedmiotem zainteresowania informatyki, uzupełnimy na koniec wybiórczym zestawieniem określeń typowych dla innych dziedzin, czyli określeń składających się na pojęcie informacji w ogóle. Jakkolwiek informatyka wiedzy prym w badaniach nad automatycznym przetwarzaniem informacji, to dopiero połączenie jej dokonań z dorobkiem innych dziedzin może doprowadzić do uogólnionego pojęcia informacji. Spośród dziedzin tych na szczególną uwagę zasługują **cybernetyka, psychologia i filozofia**. I tak, w cybernetyce - uważanej za bezpośrednią poprzedniczkę informatyki - *informacją* nazywa się treść zaczerpniętą przez system ze środowiska w procesie jego dostosowywania się do różnych ewentualności tego środowiska i czynnego życia w nim [Wiener 1961 (1950)]. W psychologii - wykorzystującej w dużej mierze koncepcje informatyków - za *informację* uznaje się wynik procesu psychicznego polegającego na odzwierciedlaniu i kategoryzowaniu wybranych elementów otoczenia w wewnątrzmysłowym kodzie [Nosal 1990]. W filozofii z kolei *informację* traktuje się jako jedną z trzech ogólnych kategorii objaśniających rzeczywistość - kategorię równorzędną z materią i energią [Lubański 1991]. Jak widać, to ostatnie podejście ujawnia nieco inny aspekt różnicy, o której wspominaliśmy przy okazji omawiania informacji jako tworzywa maszyn

systemy - choć mniej popularne - zdolne do analogowego kodowania i przetwarzania danych. Choćby ze względu na nie, informatycy winni poszukiwać innego, uogólnionego pojęcia informacji.

⁴ Teorię liczb wymieniamy tylko dla przykładu. Jak wiadomo, wszelkie teorie matematyczne, dotyczące różnych obiektów matematycznych, mają charakter abstrakcyjny.

informatycznych (czyli informacji zredukowanej do danych). Tym razem różnica między informacją z jednej strony, a materią i energią z drugiej, dotyczy ich funkcji eksplikacyjnych, a przedstawia się następująco. Podczas gdy wymiar materialny pozwala opisywać statyczne właściwości przedmiotów (np. budowę cząsteczkową ciała), a wymiar energetyczny prawa rządzące zmianami i wzajemnymi oddziaływaniami układów materialnych (np. przemiany energii kinetycznej w elektryczną), to dopiero w płaszczyźnie informacyjnej daje się opisać prawa rządzące sterowaniem zmianami i ich komunikowaniem⁵.

1.1.2 Algorytm

Pojęcie algorytmu ma rodowód matematyczny. Do połowy XX wieku, tj. do początków współczesnej informatyki, oznaczało ono całkowicie określoną metodę rozwiązywania zagadnień algebraicznych, takich jak wyznaczanie największego wspólnego podzielnika dwóch liczb czy znajdowanie pierwiastków równań kwadratowych⁶.

Wraz z wynalezieniem pierwszych maszyn informatycznych - których zakres zastosowań ograniczał się początkowo do automatyzacji obliczeń opisanych pewnymi wzorami - termin *algorytm* przeniknął do informatyki. Nazwano nim **precyzyjny opis maszynowej realizacji zadań określonego typu**. W opisie takim należało uwzględnić wszystkie niezbędne operacje maszyny, ich kolejność, a także wszelkie sytuacje wyboru, w których są wywoływane warunkowo takie lub inne operacje. Pierwotnie algorytmy przedstawiano w formie schematów blokowych, gdzie poszczególne kroki przyszłego programu są symbolizowane odpowiednimi kształtami (rombami, kwadratami

⁵ Niektórzy teoretycy badają rozwój trójki wymienionych pojęć (tj. materii, energii i informacji), przyjmując - naszym zdaniem słusznie - że narodziny pewnych konstrukcji teoretycznych poprzedza zawsze skuteczna praktyka. Oto człowiek miałby tworzyć coraz doskonalsze narzędzia praktyczne, a wraz z upowszechnianiem się ich zastosowań zmieniałby swój sposób myślenia o świecie. Między innymi przenosiłby zasady działania tych narzędzi w dziedziny abstrakcyjne, na przykład w obszar matematyki, psychologii czy filozofii. Założenie, o którym mowa, pozwala zrozumieć, w jaki sposób kształtowały się kolejno trzy ogólne pojęcia/kategorie wyjaśniające. Pojęcie *materii* mogło wykrystalizować się najwcześniej, bo już w okresie dominacji narzędzi manualnych (ich dobrymi przykładami są koło garncarskie czy pług), pojęcie *energii* stało się dojrzałe wraz z zaistnieniem urządzeń działających dzięki przetwarzaniu energii-tworzywa (tj. silników), pojęcie *informacji* mogło ukształtować się dopiero wraz z wynalezieniem urządzeń do przetwarzania danych (przede wszystkim komputerów cyfrowych). Por. [Bolter 1990 (1984)], [Marciszewski 1998].

⁶ Sama nazwa *algorytm* wywodzi się od imienia matematyka perskiego, Al Chorezmi, który około roku 820 opisał czysto mechaniczne metody liczenia w systemie dziesiętnym (tzw. „liczenie w słupkach”). Za jego sprawą i za dalszym pośrednictwem Arabów metody te trafiły do Europy pod nazwą algorytmu (będącą pewnym zniekształceniem imienia *Al Chorezmi*). Por. [Kordos 1994].

itp), a przejścia pomiędzy nimi - strzałkami. Współcześnie natomiast zapisuje się je przeważnie w jednym z języków programowania wysokiego poziomu, np. w PASCALU; zakładając, że większość informatyków potrafi przełożyć takie zapisy na inny język programowania⁷.

Zauważmy tutaj, że precyzyjną definicję algorytmu jako schematu automatycznego przetwarzania informacji sformułowano jeszcze przed skonstruowaniem pierwszych programowanych maszyn informatycznych. W 1936 r. Alan Turing zaproponował *stricte* teoretyczny⁸ opis urządzenia zdolnego wykonywać w sposób uprzednio zaprogramowany dowolne operacje symboliczne; dziś opis ten jest znany pod nazwą *uniwersalnej maszyny Turinga* [Turing 1936]. Po faktycznym wynalezieniu komputerów i upowszechnieniu się sztuki programowania okazało się, że uniwersalna maszyna Turinga stanowi najbardziej podstawowy, logiczny schemat działania komputerów cyfrowych. Dzięki temu zaś intuicyjne pojęcie algorytmu zyskało czysto matematyczny punkt odniesienia - algorytmem można było nazwać **schemat operacji możliwych do wykonania przez uniwersalną maszynę Turinga** (ogólny opis tejże maszyny damy w podrozdziale kolejnym).

Dla zrozumienia idei algorytmu niezmiernie ważny jest fakt, że współczesne maszyny informatyczne, a więc sterowane algorytmicznie, nie służą li tylko do obliczeń matematycznych, lecz przede wszystkim do przetwarzania danych „wyższego poziomu”. Dane takie są wprawdzie kodowane liczbowo i poddawane w tej formie pewnym operacjom matematycznym, ale zarówno ich kod, jak i odpowiednie operacje, mają sens „nieliczbowy”. Na przykład w programach do zarządzania bazami danych typowe dane mają postać nazwisk, adresów i opisów tekstowych, a typowe operacje na nich - takie jak wyszukiwanie i sortowanie - wykraczają poza czyste obliczenia matematyczne.

Ponieważ algorytmy stanowią punkt wyjścia do tworzenia programów, informatycy badają je przede wszystkim ze względu na efektywność komputerowej realizacji. Miarą teże efektywności są następujące cechy: **złożoność struktury** - wpływająca na długość oraz przejrzystość kodu przyszłego programu, **złożoność pamięciowa** - tj. wymagania, co do pamięci maszyny realizującej algorytm, **zbieżność** - tj. warunki, pod jakimi algorytm zapewnia wykonanie danego zadania, i **złożoność czasowa**. Na przykładzie tej ostatniej cechy łatwo dostrzec związek pomiędzy teorią algorytmów a funkcjami komputerów.

⁷ Podkreślmy tu, że algorytm jakkolwiek może zostać przedstawiony w jakimś języku programowania, nie jest gotowym programem komputerowym. Stanowi on pewien schemat, na podstawie którego można utworzyć szczegółowy i gotowy do uruchomienia program komputerowy.

⁸ Oprócz motywacji czysto matematycznej (związanej z chęcią rozwiązania problemu nierozstrzygalności) Turingiem powodowały pewne względy praktyczne, przede wszystkim zaś faktyczne zapotrzebowanie na mechaniczne urządzenia szyfrująco-deszyfrujące. Z perspektywy czasu możemy powiedzieć jednak, że idea maszyny Turinga przyczyniła się bardziej do rozwoju teorii maszyn cyfrowych i obliczeń, niż praktyki ich konstruowania.

Złożoność czasowa danego algorytmu informuje bowiem o tym, jak wraz ze wzrostem wielkości przetwarzanych danych zmienia się liczba elementarnych operacji maszyny niezbędnych do wykonania zadania. Chodzi, rzecz jasna, o zgrubne oszacowanie typu funkcji wiążącej rozmiar danych z liczbą operacji; w przypadku algorytmów najbardziej efektywnych są to funkcje liniowe lub wielomianowe, w przypadku algorytmów najmniej sprawnych - wykładnicze⁹.

Rozważmy na koniec sprawę rzekomej niezawodności algorytmów. Otóż twierdzi się niejednokrotnie, głównie w publikacjach nie-informatycznych, że algorytm - opisujący pewnego rodzaju metodę rozwiązywania problemów - jest to schemat niezawodny, a więc gwarantujący rozwiązanie problemu danego typu w skończonej liczbie kroków (Por. [Szaniawski 1994], [Kozielecki 1992]). Czy opinia ta znajduje swoje potwierdzenie w aktualnej praktyce językowej informatyków? Okazuje się, że nie. Współcześnie bowiem mówi się o dwóch ważnych rodzajach algorytmów, które nie w każdym wypadku zapewniają rozwiązanie skojarzonego z nimi problemu. Po pierwsze, są to *algorytmy probabilistyczne*, czyli takie, w których pewne operacje są wywoływane losowo, z założonym z góry lub stopniowo modyfikowanym prawdopodobieństwem (ich ważną podklasę stanowią algorytmy ewolucyjne). Po drugie, są to *algorytmy heurystyczne*, czyli takie, które sprawdzają się wystarczająco dobrze dla większości danych początkowych, lecz nie dają stuprocentowej pewności co do każdorazowego rozwiązania problemu; najczęściej za ich pomocą rozwiązuje się tzw. zagadnienia optymalizacyjne, gdzie dąży się do rozwiązania optymalnego (czyli wystarczająco dobrego), a nie idealnego¹⁰.

1.1.3 Automat

Ogólna charakterystyka informatyki jako nauki o algorytmicznym przetwarzaniu informacji byłaby niepełna, gdybyśmy nie podkreślili faktu, że **systemy informatyczne są zdolne do działania nienadzorowanego, czyli automatycznego**. Automatyizacja, o której mowa, ma dwa aspekty: symulacyjny

⁹ Ponieważ algorytmy opisują pewnego rodzaju metody rozwiązywania problemów, funkcja złożoności czasowej dostarcza wygodnego kryterium podziału problemów. I tak: problemy, dla których istnieją algorytmy o złożoności liniowej nazywa się liniowymi, problemy, dla których nie ma algorytmów lepszych niż te o złożoności wielomianowej - wielomianowymi itd. Faktem jest, że istnieją i takie problemy, co do których udowodniono niemożność zaprojektowania jakichkolwiek algorytmów. Przynajmniej tak wygląda sprawa w przypadku algorytmów tradycyjnych (tj. nie zawierających elementów probabilistycznych). Por. [Harel 2000 (1987)]

¹⁰ Przykładem zagadnienia optymalizacyjnego jest następujący problem komiwojagera: dla danej sieci n miast o podanych odległościach między poszczególnymi miastami znaleźć najkrótszą spośród dróg, które przebiegają przez każde z miast dokładnie raz. Ponieważ problem ten ma złożoność wykładniczą, dla odpowiednio dużych n nie istnieje efektywny algorytm rozwiązania. Dlatego też proponuje się różne algorytmy heurystyczne, które pozwalają drogi możliwie najkrótsze (znajdują więc rozwiązania optymalne, a nie idealne). Por. [Bolc, Cytowski 1989 i 1991]

i realny. Z jednej strony bowiem, maszyny informatyczne są w stanie symulować działania innych urządzeń, z drugiej natomiast - mogą nimi sterować, a w ten sposób oddziaływać na swoje otoczenie faktycznie i fizycznie¹¹.

Niezależnie od aspektu automatyzacji systemy informatyczne dają możliwość automatyzowania ogromnej (być może, nieograniczonej w praktyce) liczby funkcji. Dzieje się tak wskutek uniezależnienia realizowanych funkcji od fizycznej budowy układu. Mówiąc dokładniej: o tym, jaką funkcję realizuje w danej chwili dany układ, decyduje nie jego fizyczna budowa, lecz zainstalowane w nim wymienne *oprogramowanie*.

Dla teorii automatów przetwarzających dane niezmiernie ważny jest fakt, że mimo ogromnej liczby ich zastosowań, **istnieje stosunkowo prosta konstrukcja myślowa, która obrazuje zasadniczą ideę działania pewnej podklasy tych automatów, a mianowicie komputerów cyfrowych**. Mamy na myśli wspomnianą wcześniej *uniwersalną maszynę Turinga*.

Maszyna taka składa się z rejestru stanów, głowicy odczytująco-zapisującej oraz nieskończonej¹² pokratkowanej taśmy, na której można zapisywać symbole pewnego alfabetu. Jej działanie polega na skrupulatnym wykonywaniu programu określającego ruchy głowicy, operacje zapisu/odczytu w bieżącej kratce oraz przejścia między stanami. Wszelkie instrukcje programu mają postać warunkową: „**jeśli** (*stan*=*p*), (*symbol*=*x*), **to** *zmień stan na q, zmień symbol na y, przesun głowicę w prawo lub w lewo*”. Sterowany za ich pomocą mechanizm działa następująco: gdy w bieżącej kratce „widzi” symbol *x*, wpisuje w niej inny symbol *y*, przesuwa głowicę o jedną kratkę w lewo lub w prawo, po czym zmienia swój bieżący stan *p* na inny stan *q*. Maszyna kończy pracę, gdy znajdzie się w wyróżnionym stanie końcowym; w tym momencie na taśmie pozostaje wynik, czyli gotowy do zinterpretowania ciąg symboli alfabetu.

W prostym języku stanów i symboli alfabetu daje się układać wiele programów, mniej lub bardziej skomplikowanych. Na przykład, gdyby ktoś chciał napisać program do mnożenia dwóch liczb, musiałby wybrać alfabet, w którym liczby byłyby kodowane (np. symbole 0 i 1), zdecydować się na liczbę stanów

¹¹ Zauważmy na marginesie, że w odniesieniu do ludzi „działać automatycznie” znaczy tyle, co „działać bez udziału świadomości”, czyli w sposób instynktowny lub uprzednio wyuczony. Mogłoby wydawać się zatem, że w odniesieniu do ludzi automatyzm działań i świadomość tychże kłócą się ze sobą. Tak jednak nie jest. Człowiek działający automatycznie (czyli bez udziału świadomości) może bowiem odnieść się do swoich działań *post factum*. A zatem, mimo możliwości działań automatycznych jest zdolny do świadomej refleksji nad nimi. Ponieważ nie jest wykluczone, że zdolność tę będą wykazywać hipotetyczne maszyny przyszłości, nie wolno twierdzić, że samo pojęcie automatu (maszyny działającej bez ingerencji użytkownika) wyklucza świadomość tegoż. W istocie tego właśnie problemu, tj. możliwości skonstruowania świadomych automatów, dotyczy dyskusja filozoficzna na temat sztucznej inteligencji. Por. [Penrose 2000 (1994)].

¹² Założenie nieskończonej długości taśmy ma charakter teoretyczny. W praktyce wszelkie maszyny cyfrowe - oparte de facto na modelu Turinga - dysponują pamięcią skończoną.

maszyny (np. 5) oraz zaproponować odpowiedni ciąg instrukcji zawierających symbole alfabetu i stany maszyny. Aby przeprowadzić jakieś konkretne mnożenie, musiałby umieścić na taśmie automatu dwie odpowiednio zakodowane liczby i uruchomić go; w efekcie wykonania programu na taśmie pozostałby odpowiednio zakodowany wynik¹³.

Choć opisana wyżej koncepcja oddaje istotę maszyn cyfrowych, a matematycy udowodnili, że wszelkie programy działań takich maszyn mogą zostać zapisane w postaci schematów zaproponowanych przez Turinga¹⁴, to ani sam mechanizm głowicowo-taśmowy, ani też forma kierujących nim instrukcji, nie zapewniają takiej efektywności działań, która charakteryzuje współczesne komputery. Ważny krok ku tejże efektywności stanowi kolejna koncepcja ogólna, znana szeroko pod nazwą *architektury von Neumanna*¹⁵. Koncepcja ta, jakkolwiek ogólna, określa bliżej sposób działania maszyny cyfrowej, a mówiąc konkretniej, wyznacza **wyraźny podział maszyny na bloki odpowiedzialne za jej poszczególne funkcje**. Zgodnie z nią każdy komputer cyfrowy winien zawierać: a) blok odpowiedzialny za bezpośrednią komunikację z otoczeniem (urządzenia wejścia/wyjścia), b) blok do przechowywania danych (pamięć), c) blok koordynujący wykonywanie programów (jednostka centralna). Koncepcję tę omówimy szerzej w kolejnym podrozdziale.

1.2 Specyfika systemów informatycznych

Już poprzedni podrozdział zawierał pewne rozproszone uwagi o specyfice systemów informatycznych, czyli układów przetwarzających dane automatycznie i w sposób zalgorytmizowany. Poniżej uwagi te zbierzemy i rozwiemy, w szczególności zaś odróżnimy maszyny informatyczne od tradycyjnych i wyróżnimy różne typy tych pierwszych.

1.2.1 Systemy informatyczne a maszyny tradycyjne

Filozof czy też metodolog pragnący wyjaśnić, na czym polega istota systemów informatycznych, staje przed nie lada wyzwaniem. Materiał empiryczny, do którego musi się odnieść, jest bowiem niezwykle bogaty i różnorodny. Z jednej strony składają się nań zasady działania systemów już skonstruowanych, ta-

¹³ Jako teoretyczne rozwinięcie koncepcji maszyny Turinga rozwinęła się w informatyce teoria automatów. W jej ramach bada się możliwości i ograniczenia różnego rodzaju maszyn cyfrowych, analizując ich teoretyczne modele - począwszy od automatów skończonych, przez automaty ze stosem, aż do uniwersalnych maszyn Turinga. Por. [Hopcroft. Ullman, Motwani 2001].

¹⁴ Por. [Hopcroft. Ullman, Motwani 2001].

¹⁵ Jest ona zbiorczym dziełem zespołu pracującego w latach 1944-1946 nad prototypem pierwszego zaawansowanego komputera cyfrowego. Ponieważ zespołem tym kierował John von Neumann, w nazwie koncepcji figuruje jego nazwisko.