

# Christopher Strachey

## – nieznanany tekst o sztucznej inteligencji

**Elektroniczne komputery same w sobie nie są zdolne do robienia czegokolwiek. Są jedynie urządzeniami, które mogą wykonywać bardzo szybko długie ciągi rozkazów.**

Christopher Strachey  
„The ‘Thinking’ Machine”, *Encounter*, nr 13/1954

Christopher Strachey był współczesnym Alana Turinga i choć trudno powiedzieć, że był jego współpracownikiem, to stykał się z nim w czasie, gdy Turing pełnił funkcję zastępcy kierownika Computing Machine Laboratory Uniwersytetu Manchester. Znali się zresztą wcześniej, bo studiowali razem w King’s College w Cambridge. Strachey tak wspomina tę znajomość w wywiadzie z Nancy Foy (*Computing Europe*, 15 sierpnia 1974), po tym, jak zapoznał się z otrzymanym od Turinga jego podręcznikiem użytkownika komputera Mark I firmy Ferranti, który uważano za kompletnie niezrozumiały, bo napisany przez matematyka:

„Pojechałem do Manchester zobaczyć maszynę i zostałem tam na krótko, aby sprawdzić jak działa, po czym wróciłem. [...] Później udałem się do Manchester ponownie zaopatrzonej w ok. 20 stron i – o ile dobrze pamiętam – 64 instrukcje na każdej. Przyjechałem wieczorem i Turing przedstawił mi typowy, bardzo szybki i bardzo górnolotny opis działania maszyny. – „Zrobisz to, to, to i to” – powiedział – „a potem stwierdzisz, że musisz to zrobić od początku, bo miałaś wyłączony przełącznik zasilania”. Po czym oddalił się, zostawiając mnie samego. [...] Wrócił następnego ranka i zapytał, czy cokolwiek osiągnąłem. „Och tak, niezły pokaz!” – powiedział entuzjastycznie w odpowiedzi. [...] Było to z mojej strony trochę aroganckie pojawić się nie wiadomo skąd i napisać działający 19-stronicowy program [...], podczas gdy najdłuższy program wykonany dotychczas na tej maszynie miał pół strony”.



Janusz Zalewski

ukończył studia na Wydziale Elektroniki, doktorat obronił na Wydziale Elektrycznym Politechniki Warszawskiej. Po studiach pracował w Instytutach Badań Jądrowych w Warszawie i Świerku, komputeryzując eksperymenty z dziedziny fizyki i chemii jądrowej, a w 1989 r. wyjechał do USA, gdzie pracował w laboratoriach jądrowych oraz uczył informatyki na uczelniach w Teksasie i na Florydzie. Jest emerytowanym profesorem Florida Gulf Coast University i profesorem informatyki na Państwowej Akademii Nauk Stosowanych im. Ignacego Mościckiego. Prywatnie zajmuje się tłumaczeniem na język polski literatury polskich Amerykanów oraz analizą twórczości literackiej amerykańskich bitników.

” *Postać Stracheya wydaje się trochę zapomniana, a był on jednym z pionierów, którzy programowali wczesne komputery. Szczególnie jednak jest wart przypomnienia z kilku istotnych powodów: jako pierwszy zaprogramował komputerową grę wideo (była to gra w warcaby, o czym opowiada we wspomnianym wywiadzie) i generował muzykę komputerowo, ale też – o czym dalej w tym artykule – praktycznie zrealizował pierwszy komputerowy program sztucznej inteligencji.*

### Kariera zawodowa

Warto wspomnieć o jego wyjątkowych dokonaniach profesjonalnych. Jako informatyk zainteresowany językami programowania znałem książkę Milne’a i Stracheya „A Theory of Programming Language Semantics” (Londyn, 1976), aczkolwiek z niej nie korzystałem. Powstała w słynnym ośrodku Programming Research Group w Oxfordzie, stworzonym przez Stracheya (po jego śmierci kierownictwo przejął Tony Hoare). Z dzisiejszej perspektywy semantyka denotacyjna, którą Strachey wprowadził (choć używał nazwy „semantyka matematyczna”), wydaje się być może jego największym osiągnięciem profesjonalnym, ale był on też autorem kilku innych szczególnych koncepcji, które w informatyce przetrwały do dzisiaj. Jedną z nich było wieloprogramowanie, które nazywał „time-sharing” (1959) i nawet wystąpił o patent, przyznany mu w połowie lat sześćdziesiątych. Największy wkład Stracheya w informatykę wiąże się właśnie z językami programowania.

Christopher Strachey był szczególnie zaangażowany w ulepszenie Algolu, o czym opublikował już w 1961 r. w *Communications of the ACM* ważny artykuł<sup>1</sup> wspólnie z M.V. Wilkesem. Nie sam Algol był jednak jego przeznaczeniem lecz pochodny, aczkolwiek nowo utworzony język CPL (Combined Programming Language), którego był głównym twórcą. Dzisiejsze znaczenie języka CPL nie sposób przecenić, jako że – mimo nieukończenia oryginalnego projektu – stał się on podstawą kolejnych języków, od BCPL do B oraz C.

” *Można więc miarodajnie stwierdzić, że bez Stracheya i jego wkładu w opracowanie CPL nie byłoby ani C++, ani języka Java.*

Strachey ma też wkład w rozwój systemów operacyjnych, co udokumentował napisanym wspólnie z J.E. Stoyem i wydrukowanym w *The Computer Journal* (1972) dwuczęściowym artykułem<sup>2</sup> nt. systemu OS6 – napisanego całkowicie w języku BCPL dla komputera Modular One.

Dla bardziej wtajemniczonych w programowanie warto dodać, że to Strachey był autorem pojęć *l-value* i *r-value* odnoszących się do operacji przypisania, co wyjaśnił w ten sposób:

„Wartość *l-value* reprezentuje obszar w pamięci komputera. Nazywamy ją komórką, a nie adresem, aby uniknąć nieporozumienia względem normalnego mechanizmu adresowania pamięci w komputerze. Dwie główne cechy komórki to: ma ona zawartość, tj. związaną z nią wartość *r-value*, i fakt, że zazwyczaj można zmienić tę zawartość przez odpowiednią operację aktualizacji”. Cytat pochodzi z artykułu *Fundamental Concepts of Programming Languages*, napisanego w latach 1967–68 lecz formalnie opublikowanego dopiero w 2000 r.

(*Higher-Order and Symbolic Computation*, vol. 13).

Dokąd nas prowadzi ta wspiana kariera Christophera Stracheya? Właściwie widać to po datach publikacji, od 1959 r. nt. wieloprogramowości do wydanej pośmiertnie semantyki języków programowania (Strachey zmarł w 1975 r.). Pełną listę prac opublikowano w artykule biograficznym w *Annals of the History of Computing*, Vol. 7, No. 1, January 1985, pp. 19–42.

Gdy odwróćmy pytanie i zapytamy nie „dokąd?” a „skąd” prowadzi nas kariera Stracheya, czeka nas spora niespodzianka. Wiedziałem o Stracheyu, że był instrumentalną postacią w rozwoju języków programowania i – podziwiając jego precyzję rozumowania w formalnych opisach języka – nie interesowałem się nim w ogóle aż do pewnego momentu, który nadszedł nader nieoczekiwanie.

### Strachey wizjoner

Są różne metody oderwania się od informatycznej nierzeczywistości kodu programów komputerowych, gdy o trzeciej lub czwartej nad ranem coś nadal nie działa i to nie wiadomo dlaczego. W szczególności pomaga przejście w inną nierzeczywistość. Zgłębiałem wraz z kolegą tajniki pewnej konstrukcji programistycznej i aby oddalić się od tematu wybraliśmy dwie nierzeczywistości o profilu literackim, każdy

<sup>1</sup> C. Strachey, M.V. Wilkes, Some Proposals for Improving the Efficiency of ALGOL 60. *Coom. of the ACM*, vol. 4, no. 11, pp. 488–191, November 1961.

<sup>2</sup> J.E. Stoy, C. Strachey, OS6 – An experimental operating system for a small computer. *The Computer Journal*, Part I, vol. 14, no. 2, pp. 117–124, Part II, vol. 14, no. 3, pp. 195–203, May & August 1972.

swoją. Jednego pociągnęło widzenie fantastyczne, głównie reprezentowane przez Stanisława Lema, Janusza Zajdla i podobnych autorów, a drugiego – widzenie magiczne pisarzy latynoamerykańskich, takich jak Gabriel Garcia Marquez, Mario Vargas Llosa, Juan Rulfo. Zawzięcie dyskutując nad tym, co jest ciekawsze i bardziej inspirujące, i skacząc sobie do oczu przy dowodzeniu własnych racji, niespodziewanie dokopaliśmy się numeru magazynu literackiego *Encounter* z 1954 r., a w nim do artykułu Christophera Stracheya „The ‘Thinking’ Machine”. Ranga tego czasopisma, aczkolwiek dosyć kontrowersyjnego, była wtedy dość duża, wystarczy powiedzieć, że w tym samym numerze opublikowali swoje teksty William Faulkner i Arnold Toynbee. Choć po tym odkryciu nasze emocje nie opadły, to spory natychmiast ustały. Ich miejsce zajęła analiza tekstu Stracheya. Ogólne przesłanie jego wypowiedzi okazuje się w wielu aspektach prorocze.

Najpierw jednak opowiem o pierwszym komputerowym programie sztucznej inteligencji, jaki kiedykolwiek został napisany. Strachey opisuje w tym artykule swój program automatycznego generowania listów miłosnych. Podając przykłady dwóch listów o długości trzydziestu kilku wyrazów, przedstawia algorytm ich tworzenia. Składają się one z nagłówka, treści i podpisu. Nagłówek jest prosty i tworzy się go z dwóch słów odpowiednio dobranych z wcześniej przygotowanej listy, jak np. „Darling Sweetheart” lub „Honey Dear” lub coś podobnego. Podpis jest zawsze ten sam i brzmi:

*Yours [przysłówek w odpowiedniej wersji], M.U.C.*

Skrót M.U.C. oznacza Manchester University Computer, ale to, co jest naprawdę ważne, to treść listów.

Strachey tak charakteryzuje algorytm generowania treści, nazywając go „dziecinnie prostym”. Istnieją tylko dwa podstawowe typy zdań. Pierwszy typ ma postać następującą (podaję wersję angielską, aby uniknąć przekłamań przy tłumaczeniu na polski):

*My \_\_\_ (przym.) \_\_\_ (rzecz.) \_\_\_ (przysł.) \_\_\_ (czas.) your \_\_\_ (przysł.) \_\_\_ (rzecz.).*

Listy odpowiednich przymiotników, rzeczowników, przysłówków i czasowników są przygotowane z góry i wybiera się z nich metodą przypadkową słowa wstawiane w puste pola. Przypadek rządzi również decyzją, czy w ogóle należy wstawić przymiotnik lub przysłówkę.

Drugi typ zdania ma po prostu inny początek, mianowicie:

*You are my \_\_\_ (przym.) \_\_\_ (rzecz.)*

przy czym w tym przypadku przymiotnik jest zawsze obecny.

Typ zdania dobiera się również przypadkowo. Jest jeszcze parę innych prostych reguł, np. określenie długości listu arbitralnie na pięć zdań. Pełnej treści listów może nie warto przytaczać, trzeba jednak dodać, że współczesne metody generowania listów miłosnych niewiele się różnią od schematu opisanego w 1954 r. przez Stracheya, choć korzystają z znacznie większego korpusu danych i o wiele bardziej złożonych algorytmów<sup>3</sup>. Można nawet zaryzykować stwierdzenie, że listy Stracheya to pierwsze przykłady utworu literackiego wygenerowanego przez komputer.

W artykule Strachey nie poprzestaje jednak na opisie działania rzeczonoego programu, ale wypowiada bardzo prze-myślane stwierdzenia o roli komputerów w wykonywaniu zadań o celu nieobliczeniowym. Rozwijając myśl zacytowaną w motcie, kontynuuje:

„Rozkazy te, znane [zbiorowo] jako program, muszą dokładnie stwierdzać, co maszyna ma robić w każdej możliwej sytuacji, jaka może ją spotkać, gdyż maszyna nie może zrobić niczego, co nie zostało jej wyrażone w programie”.

Dość przewidująco wypowiada się też o inteligencji komputerów: „Zachowuje się on tak, jak by był absolutnie posłusznym i obowiązkowym niewolnikiem, który działa z niewiarogodną szybkością, ale całkowicie nieinteligentnie. Ważne jest, aby sobie uświadomić, że maszyna sama zupełnie nie potrafi dokonać żadnego osądu swojej własnej aktywności”.

Strachey ustosunkowuje się też do testu Turinga, stwierdzając, że pytanie, czy komputery mogą myśleć, powinno być postawione inaczej i brzmieć: „Czy możemy napisać program, który spowoduje, aby komputer mógł myśleć?” Przypisując Turingowi wymyślenie pierwowzoru testu już w roku 1936, Strachey zauważa dalej, że w rzeczywistości Turing miał na myśli odpowiedź na jeszcze inaczej sformułowane pytanie: „Czy maszyna może myśleć tak dobrze, jak człowiek?” Przytaczając opinię Turinga, że upłynie jeszcze z dziesięć lat, zanim współczesne im komputery utrzymają uczestników testu w niepewności przez pięć minut, Strachey uważa, że to zbyt optymistyczna prognoza, ale nie ma wątpliwości, że będzie możliwe stworzenie takiej maszyny w niedalekiej przyszłości.

Polecam lekturę cytowanego wcześniej artykułu z *Annals of the History of Computing* oraz materiałów z konferencji poświęconej Stracheyowi, *Strachey 100. Celebrating the life and research of Christopher Strachey*, która odbyła się 18–19 listopada 2016 r., na Uniwersytecie Oxfordzkim: <https://www.cs.ox.ac.uk/strachey100/>

<sup>3</sup> Por. Y. Liang, Z. Han, Intelligent Love Letter Generator Based on GPT-2 Model, Proc. 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAL), Zhuhai, Chiny, 14–16 January 2022, pp. 562–567.