

Paweł Stacewicz

ALGORYTMY. UWAGI Z POGRANICZA INFORMATYKI I METODOLOGII NAUK

Niniejsze uwagi kieruję przede wszystkim do studentów nauk technicznych (np. informatyki), którzy chcieliby uzyskać szerszą wiedzę o algorytmach niż ta, która wynika ze znajomości różnych technik programistycznych. Uwagi te dotyczą po części historii algorytmów, a po części, wykorzystywania algorytmów w innych dziedzinach niż sama informatyka (ogólnie zaś: w nauce). Uwagi są dość ogólne, choć syntetyczne (pierwotnie, w wersji nieco rozszerzonej, były wygłaszane na pewnym seminarium filozoficznym).

Zagadnienia: historia pojęcia algorytmu, definicja algorytmu, typy i własności algorytmów, złożoność czasowa, problemy algorytmicznie nierozwiązywalne (zasadniczo i praktycznie), wiedza naukowa (opisowa i proceduralna), algorytmiczny dostęp do wiedzy, algorytmiczny wzorzec wiedzy, ograniczenia metody algorytmicznej.

I. O INFORMATYCZNYM POJĘCIU ALGORYTMU

1. Mimo szerokiej obecności we wszystkich niemal naukach ścisłych pojęcie algorytmu przynależy przede wszystkim do **informatyki**, a oznacza schemat maszynowej realizacji zadań określonego typu, możliwy do zakodowania w postaci **programu** komputerowego. Pojęcie to należy rozpatrywać w ramach **triady** (informacja, algorytm, automat) – która wyraża skrótowo fakt, że informatyka zajmuje się algorytmicznym przetwarzaniem informacji (danych) za pomocą określonego rodzaju automatów (komputerów)¹.

2. Informatyczne pojęcie algorytmu ma **rodowód matematyczny**, a jego historia pokazuje jasno, w jaki sposób matematyczne próby uściślenia terminów „algorytm” oraz „procedura mechaniczna” doprowadziły do powstania komputerów i informatyki.

2a. Historię tę można podzielić na trzy nierównomiernej długości **etapy**.

ETAP I (Od starożytności po początek wieku XX)

Algorytmy są używane spontanicznie (bez ogólnej definicji algorytmu) do rozwiązywania różnorodnych problemów matematycznych: geometrycznych, arytmetycznych i algebraicznych (typowy problem to rozwiązywanie równań i ich układów). Sama nazwa „algorytm” pojawia się wieku IX, w wyniku fonetycznej przeróbki imienia uczonego arabskiego (al-Chwarizmiego), który w jednym ze swoich dzieł podał mechaniczne metody działań na liczbach zapisywanych dziesiętnie.

¹ Drugi element w/w triady wiąże informatykę z matematyką – po pierwsze z tego względu, że duża część algorytmów służy do rozwiązywania problemów matematycznych (np. równań i ich układów), a po drugie dlatego, że różne własności algorytmów (np. złożoność czasowa) są badane za pomocą narzędzi matematycznych (np. miarą złożoności czasowej są różne matematyczne funkcje). Natomiast element trzeci, tj. automat, wiąże informatykę z elektroniką – to zaś dlatego, że dopiero fizyczne automaty do przetwarzania danych (a współcześnie są to automaty elektroniczne) umożliwiają realne zastosowania informatyki. Mówiąc krótko: informatyka jest dziedziną usytuowaną pomiędzy matematyką (z której wywodzi się pojęcie algorytmu) oraz elektroniką (która pozwala konstruować fizyczne automaty do przetwarzania danych).

- ETAP II (Od roku 1936 do lat 50-tych XX wieku)
W wyniku prac takich matematyków jak Alan Turing, Alonzo Church i Emil Post pojęcie algorytmu – rozumianego jeszcze jako mechaniczna procedura rozwiązywania zagadnień matematyczno-logicznych – zostaje sprecyzowane, a następnie wykorzystane w badaniach nad rozstrzygalnością² systemów formalnych (największą rolę odegrała tu konstrukcja Turinga, nazywana dziś uniwersalną maszyną Turinga)³. W tym samym czasie zostają wykryte istotne ograniczenia metody algorytmicznej (np. problem stopu).
- ETAP III (Czasy najnowsze, od zakończenia II wojny światowej).
Pojęcie algorytmu przenika do informatyki, stając się jej pojęciem naczelnym. Opracowuje się coraz to nowe algorytmy dla coraz bardziej zaawansowanych sprzętowo i programistycznie maszyn; stosuje się je głównie poza matematyką (do problemów kodowanych liczbowo, ale mających sens nie-liczbowy). Bada się różne własności algorytmów – związane przede wszystkim z ich efektywnością.

2b. Wydaje się, że obecnie, wskutek spontanicznego rozwoju technik **programistycznych**, dostosowanych do różnych typów maszyn, narasta potrzeba syntezy materiału empirycznego i wyabstrahowania zeń **nowego** pojęcia algorytmu – ogólniejszego niż definicja dana przez Turinga w roku 1936.

3. Niezależnie od sugestii zawartej w punkcie (2b) trzeba stwierdzić, że obecnie na polu informatyki (czy też metodologii informatyki) współlistnieją ze sobą **dwa** pojęcia algorytmu.

3a. W sensie **wąskim** i jednocześnie najbardziej precyzyjnym algorytmem nazywa się każdy ogólny schemat procedury możliwej do wykonania przez **uniwersalną maszynę Turinga** (UMT). Ze względu na obliczeniową równoważność UMT i komputerów cyfrowych jest to pojęcie algorytmu dla maszyn cyfrowych.

3b. W sensie **szerszym** (a precyzowanym częściowo w ramach różnych modeli obliczeń) algorytmem nazywa się ogólny schemat procedury możliwej do wykonania przez **pewną maszynę**, niekoniecznie cyfrową i niekoniecznie deterministyczną (np. analogową, kwantową, ewolucyjną – lista nie jest zamknięta).

3c. Pytaniem otwartym pozostaje, w jakim sensie drugie (ogólniejsze) pojęcie algorytmu jest **sprowadzalne** do pierwszego (węższego). Choć bowiem istnieją takie problemy, które są nierozwiązywalne lub trudno-rozwiązywalne za pomocą maszyn Turinga, a są rozwiązywalne lub łatwiej-rozwiązywalne za pomocą maszyn innych, to pozostaje pytanie o to, czy są to problemy **istotne** z praktycznego punktu widzenia.

4. Z ukazaną wyżej niejednoznacznością informatycznego pojęcia algorytmu współgra dobrze informatyczna **praktyka**, w której mamy do czynienia z bardzo **dużą różnorodnością** algorytmów (a właściwie ich typów) badanych ze względu na różne własności.

4a. Identyfikując różne **typy algorytmów**, można podać następujące pary przeciwieństw (są one przykładowe i wyróżnione ze względu na różne kryteria):

² Rozstrzygalność jest to pewna własność systemów aksjomatycznych (np. rachunku zdań w logice). System jest rozstrzygalny, jeśli dla każdej poprawnie zbudowanej formuły tego systemu, można sprawdzić w sposób mechaniczny oraz w skończonej liczbie kroków, czy formuła ta jest twierdzeniem systemu (może zostać wyprowadzona z aksjomatów).

³ Dopowiedzieć trzeba, że zarówno Turingowi, jak również innym współczesnym mu matematykom, udało się sprecyzować pojęcie algorytmu dla maszyn cyfrowych (przetwarzających dyskretne dane symboliczne), a nie jakichś innych (np. analogowych). Zob. pkt. 3a.

(1) algorytmy cyfrowe vs analogowe, (2) deterministyczne vs niedeterministyczne, (3) szeregowo vs równoległe, (4) sekwencyjne vs rekurencyjne, (5) linearne vs populacyjne, (6) algorytmy działania vs algorytmy uczenia się⁴.

4b. Jeśli chodzi o badane przez informatyków ogólne **własności algorytmów**, to najważniejsze pośród nich są:

- (1) skuteczność w rozwiązywaniu określonych klas problemów (czy problem pewnego typu jest w ogóle rozwiązywalny przez dany algorytm?),
- (2) stabilność numeryczna (czy pewne wartości danych wejściowych, np. bardzo małe, nie powodują utraty skuteczności danego algorytmu?),
- (3) złożoność (strukturalna, pamięciowa i czasowa; zob. dalej).

4c. Powyższe własności bada się dla algorytmów pewnego typu, związanych z określonym **modelem obliczeń**. Najczęściej są to algorytmy dla maszyn cyfrowych (związane z modelem uniwersalnej maszyny Turinga).

5. Algorytmy rozwiązujące dany problem P najczęściej porównuje się ze sobą ze względu na czas rozwiązywania P dla różnych danych, czyli ze względu na ich **złożoność czasową**. Za miarę złożoności czasowej algorytmu uznaje się **funkcję**, która określa zależność między rozmiarem danych wejściowych algorytmu a czasem potrzebnym na uzyskanie wyniku. Zależnie od **typu funkcji** opisującej złożoność mówi się o złożoności liniowej, wielomianowej, logarytmicznej, wykładniczej itp.

5a. Cechę złożoności przypisuje się również samym **problemom**: złożoność czasowa problemu P jest to złożoność najbardziej efektywnego (czyli najmniej złożonego) spośród wszystkich algorytmów rozwiązujących problem P.

Na przykład: problem o złożoności wielomianowej to taki problem, dla którego istnieją rozwiązujące go algorytmy o złożoności wielomianowej (czyli niskiej).

6. Problemy o złożoności co najmniej wykładniczej uznaje się – ze względu na bardzo długi czas rozwiązywania nawet dla stosunkowo „małych” danych – za **praktycznie nierozwiązywalne**. Typowe przykłady to: problem komiwojażera i problem spełnialności formuł logiki zdań.

7. Oprócz problemów nierozwiązywalnych praktycznie istnieją w informatyce (a szerzej: w nauce) problemy **nierozwiązywalne zasadniczo**, zwane także nieobliczalnymi. Są to takie zagadnienia, dla których nie istnieje (w sensie obiektywnym) algorytm rozwiązujący je we wszystkich przypadkach szczególnych.

Typowy przykład z dziedziny problemów dla maszyn cyfrowych to **problem stopu**: określenia algorytmu sprawdzającego, czy dany program dla określonych danych wejściowych kończy pracę⁵.

⁴ Z każdą ze wskazanych opozycji wiąże się szereg ważnych i ciekawych zagadnień filozoficznych, na przykład: ad 1) pod jakimi warunkami co do struktury świata/materii techniki analogowe są realizowalne w praktyce (z punktu widzenia teorii techniki te są istotnie odmienne od techniki cyfrowych)?; albo ad 6) w jakim sensie i pod jakimi warunkami algorytmy uczenia się mogą skutkować inwencją maszyn?

⁵ Problemy algorytmicznie nierozwiązywalne lub nieobliczalne nazywa się także *algorytmicznie niedostępnymi*. Nazwa ta wydaje się trafna, bo choć w sensie obiektywnym pewne problemy mogą mieć rozwiązania (jak problem stopu maszyny Turinga), to okazuje się, że za pomocą żadnego (lub: żadnego dostatecznie efektywnego) algorytmu nie możemy uzyskać dostępu do wiedzy o tym rozwiązaniu.

II. ALGORYTMY I WIEDZA

W drugiej części szkicu potraktujemy algorytmy szerzej niż to ma miejsce w informatyce (czy matematyce). Będziemy je rozumieć jako pewne sformalizowane, schematyczne i możliwe do zautomatyzowania **metody zdobywania wiedzy naukowej**.

8. Przez **wiedzę typu naukowego** rozumie się najczęściej ogół sądów i metod akceptowanych w różnych naukach (a zatem: dobrze uzasadnionych ze względu na kryteria obowiązujące w tychże naukach)⁶.

W najbardziej nas tu interesującym aspekcie **pragmatycznym** wiedza naukowa służy do skutecznego rozwiązywania właściwych danej nauce **problemów** (m.in. praktycznych).

8a. Niezależnie od dziedziny badań wiedza naukowa przyjmuje **dwie**, powiązane ze sobą, postaci:

(a) **opisową** – tzw. „wiedza że”, opisująca właściwe danej dyscyplinie fakty i zależności między nimi (np. prawa fizyczne czy twierdzenia matematyczne);

(b) **proceduralną** – tzw. „wiedza jak”, składająca się z metod, które prowadzą w sposób niezawodny lub wiarygodny do „wiedzy że” (przy czym fakt zastosowania odpowiedniej metody stanowi jednocześnie dobre uzasadnienie danego elementu wiedzy).

8b. W aspekcie **pragmatycznym** pojęcie wiedzy wiąże się z pojęciem **problemu**: na wiedzę z zakresu danej nauki składa się ogół sądów (o faktach) i metod zapewniających rozwiązania interesujących daną naukę problemów. W ten kontekst wpisuje się doskonale pojęcie **algorytmu** jako schematycznego opisu metody rozwiązywania problemów określonego typu.

9. Rozwijając wątek pragmatycznych funkcji wiedzy, możemy podać szereg narzucających się **związków** między pojęciami wiedzy i algorytmu. Oto ich prowizoryczne zestawienie.

9a. Algorytm stanowi pewną **formę zapisu** wiedzy. Jest to forma niezwykle **ekonomiczna** – jeden uniwersalny schemat reprezentuje bowiem nieskończoną liczbę rozwiązań problemów określonego typu (rozwiązania te uzyskujemy, stosując algorytm do odpowiednich danych)⁷.

9b. Wiedza zapisana algorytmicznie stanowi coś **pośredniego** między wiedzą opisową i proceduralną. Każdy algorytm bowiem jest ogólnym **opisem procedury**, która po zastosowaniu do odpowiednich danych doprowadza do nowych **sądów** o faktach, czyli nowej wiedzy opisowej⁸.

⁶ Oprócz wspomnianej w głównym tekście wiedzy naukowej można wyróżnić: a) wiedzę potoczną (inaczej zdroworozsądkową, podzielaną przez pewną społeczność, ale bez statusu naukowej), oraz b) indywidualną (właściwą poszczególnym ludziom; może ona wywodzić się zarówno z wiedzy naukowej, jak i potocznej).

⁷ Ze względu na wspomnianą wyżej ekonomię zapisu wiedzy (co można nazwać kompresją wiedzy) algorytmy przypominają pojęcia (jedno pojęcie reprezentuje nieskończenie wiele obiektów podobnych do siebie pod pewnym względem), przy czym o ile pojęcie jest statycznym składnikiem wiedzy, to algorytm jest składnikiem dynamicznym.

⁸ Widać zatem, że m.in. za sprawą algorytmów – opisujących procedury dochodzenia do wiedzy – granica między wiedzą opisową („wiedzą że”) i proceduralną („wiedzą jak”) jest rozmyta. W matematyce, na przykład, bardzo wiele twierzeń – wydawałoby się opisowych – można przedstawić jako schematyczne metody rozwiązywania problemów określonego typu, czyli algorytmy.

9c. Algorytmy prowadzą do wiedzy w sposób **niezawodny** oraz **intersubiektywnie** dostępny – to znaczy: każda osoba, stosując ten sam algorytm do tych samych danych, dojdzie do tego samego wyniku.

10. Podkreślić trzeba, że używane w takiej czy innej dziedzinie algorytmy zapewniają dwojakiego rodzaju **dostęp do wiedzy**:

(1) każde zastosowanie algorytmu do nowych danych (których jest potencjalnie nieskończenie wiele) skutkuje **nową wiedzą** (tzn. rozwiązaniem pewnego szczególnego przypadku problemu, którego dotyczy algorytm);

(2) trafnie dobrany zbiór algorytmów ułatwia penetrację danej dziedziny na **nowym jakościowo poziomie** – umysł badacza może posługiwać się tymi algorytmami bezrefleksyjnie, koncentrując uwagę na nowych problemach i nowych pomysłach ich rozwiązania.

11. Wskazane wyżej kwestie poznawczej ekonomii, intersubiektywności i niezawodności algorytmów prowadzą wprost do ciekawego pytania o to, czy w nauce obowiązuje **algorytmiczny wzorzec wiedzy**?

A zatem: czy w swojej praktyce badawczej naukowcy powinni **dążyć** do zapisu wiedzy w postaci algorytmicznej?

W poniższych punktach przedstawimy pewne argumenty za odpowiedzią **pozytywną**.

11a. Wiedza zapisana w postaci algorytmicznej i rozbudowywana w oparciu o algorytmy spełnia szerokie wymagania **racjonalizmu** (które muszą – niejako z definicji – charakteryzować każdą naukę): chodzi o wymóg **intersubiektywnej** komunikowalności i sprawdzalności⁹.

11b. Zapis wiedzy w postaci algorytmicznej jest bardzo **efektywny** ze względu na możliwość nowych **odkryć** – dzięki algorytmom część pracy naukowej zostaje **zautomatyzowana** (współcześnie: skomputeryzowana), a to pozwala poświęcić więcej czasu i energii na twórczość.

11c. Z pragmatycznego punktu widzenia dana gałąź wiedzy staje się **dojrzała**, gdy przyjmuje (przynajmniej częściowo) postać **rachunku**, to znaczy systemu formalnie zapisanych faktów (podstawowych) i formalnych reguł dochodzenia do nowych faktów (inaczej: rozwiązań właściwych danej nauce problemów). **Algorytmizację** wiedzy można uznać za zbudowanie takiego rachunku¹⁰.

12. Mimo oczywistej przydatności algorytmów do poszerzania ludzkiej wiedzy (zob. pkt 10) metoda algorytmiczna ma poważne **ograniczenia** formalne. Są one formalne, bo wiążą się z ogólnymi własnościami algorytmów, jak np. złożoność czasowa (niektóre algorytmy mają

⁹ Oczywiście nie jest to jedyna postać wiedzy spełniająca szerokie wymagania racjonalizmu.

¹⁰ Siła „rachunkowej koncepcji wiedzy dojrzałej” ujawnia się najpełniej w matematyce, gdzie buduje się różne szczegółowe rachunki, czyli zbiory reguł/algorytmów pozwalających efektywnie przekształcać obiekty z danego działu matematyki (np. granice czy całki). Wymieńmy przykładowo: rachunek prawdopodobieństwa, rachunek granic, rachunek różniczkowo-całkowy, rachunek zdań (logicznych).

Dodajmy, że algorytmizacja danej dyscypliny – czyli zapis jej metod a pośrednio i wyników w postaci algorytmów – jest możliwa pod warunkiem formalizacji tej dyscypliny w pewnym precyzyjnym języku (np. języku pewnego rachunku logicznego, wzbogaconego o terminy dla danej nauki specyficzne).

zbyt dużą złożoność czasową) czy warunek stopu (o niektórych algorytmach, nie można algorytmicznie stwierdzić, dla jakich danych się zatrzymują, a dla jakich nie). Zob. pkt 6 i 7.

Ponadto: metoda algorytmiczna nie może być traktowana jako w pełni **automatyczna**. Nawet dobór algorytmu adekwatnego do problemu wymaga pewnej decyzji poznawczej badacza, a wytworzenie nowego algorytmu wymaga inwencji badacza.