

Monika Walczak

Algorytmizowalność reguł heurystycznych. (Trudności z algorytmizowaniem odkryć naukowych)

Wśród metodologów i filozofów nauki przeważa opinia, że odkrycia naukowe mają charakter niealgorytmiczny, czy są wręcz ze swej natury niealgorytmizowalne¹. Chciałabym tę tezę poddać analizie i wskazać teoretyczne trudności, związane z ustaleniem jej prawdziwości albo fałszywości. Interesuje mnie przede wszystkim algorytmizowalność reguł heurystycznych. Problem polega nie tyle na tym, czy reguły heurystyczne są czy nie są algorytmiczne, ale na tym, czy istnieje możliwość ich zaalgorytmizowania.

1. Pojęcie algorytmu

Pojęcie algorytmu występuje głównie w matematyce, logice i informatyce². Najogólniej mówiąc, algorytm jest skończonym zbiorem dokładnie określonych reguł, wyrażonych w języku symbolicznym, pozwalających rozwiązać pewne klasy problemów, jest więc specyficzną metodą rozwiązywania pewnego typu zagadnień. Zgodnie ze słownikową definicją A. Latawiec³, algorytm powinien posiadać następujące cechy: 1) dyskretność, czyli krokowość (etapowość), a więc musi dać się rozłożyć na skończoną ilość elementarnych czynności; 2) jednoznaczność, czyli konieczność jednoznacznego określenia poszczególnych kroków,

¹ Na przykład: K. Popper, *Logika odkrycia naukowego*, przeł. U. Niklas, Warszawa 1977; Tenże, *Wiedza obiektywna. Ewolucyjna teoria epistemologiczna*, przeł. A. Chmielewski, Warszawa 1992; Th. Kuhn, *Dwa bieguny. Tradycja i nowatorstwo w badaniach naukowych*, przeł. S. Amsterdamski, Warszawa 1995; K. Szaniawski, *Metoda i twórczość w nauce*, w: K. Szaniawski, *O nauce, rozumowaniu i wartościach. Pisma wybrane*, oprac. J. Woleński, Warszawa 1994, s. 68-76.

² Twórcami teorii algorytmów są między innymi S. C. Kleene, A. Church, A. M. Turing i E. L. Post. W niniejszym artykule nie odwołuję się do formalizacji pojęcia algorytmu poruszając na jego intuicyjnym sensie.

³ A. Latawiec, *Algorytm*, w: *Słownik pojęć filozoficznych*, red. W. Krajewski, Warszawa 1996, s. 13-14.

tak aby każdorazowe jego wykonanie przebiegało w identyczny sposób; 3) zupełność, czyli możliwość przewidzenia wszelkich możliwych sytuacji wraz z określeniem sposobu postępowania w przypadku ich wystąpienia; 4) finitystyczność (skończoność), czyli konieczność ścisłego wyznaczenia punktu rozpoczęcia oraz punktu końcowego skończonej ilości kroków; 5) uniwersalność, czyli możliwość zastosowania do określonej grupy problemów, co oznacza, iż przepis służący do rozwiązania jednego tylko problemu nie jest algorytmem.

Według A. A. Markowa⁴ algorytm jest to dokładna recepta określająca proces obliczeniowy, który prowadzi od danych początkowych do żadanego rezultatu. Według P. Dembińskiego⁵ algorytm jest zbiorem reguł, które określają ciąg akcji (działań), jakie mają być wykonane dla rozwiązania pewnego problemu lub wykonania pewnego zadania. Każdy algorytm, poza ustaleniem kolejności wykonywania akcji, określa obiekty, na których działa (dane wejściowe) i zbiory wartości, które obiekty te mogą przyjmować (dane wyjściowe). Opis obiektów, na których wykonuje się czynności, odwołuje się do rozpoznawalnych w sposób łatwy i (praktycznie) niezawodny cech tych obiektów. Takimi cechami są w szczególności cechy fizyczne, np. algorytmiczny opis postępowania towarzyszący dowodom sformalizowanym odwołuje się do kształtu i położenia symboli, a nie do ich znaczeń. Każda reguła występująca w algorytmie musi być całkowicie i jednoznacznie rozumiana przez człowieka lub maszynę realizującą algorytm. Obliczenia algorytmu powinny być wykonywane w skończonej liczbie kroków. Każda operacja algorytmu musi być nie tylko dobrze określona, lecz także efektywna.

A. Góralski definiuje algorytm jako metodę rozwiązywania wyróżnionej klasy zadań. Metoda ta zawiera: 1) jednoznaczny opis procedury, tj. wyszczególnienie i charakterystykę etapów procesu rozwiązywania oraz jednoznaczne określenie układu (kolejności) i współzależności etapów; 2) szczegółowy wykaz czynności gwarantujących efektywność metody, tj. zapewniających możliwość pokonania określonego etapu procesu rozwiązywania zadania lub wykazania, że nie jest to osiągalne⁶.

D. Harel⁷ opisuje algorytmy stosowane w informatyce jako rozwiązania pewnych rodzajów zadań, zwanych zadaniami algorytmicznymi albo obliczeniowymi. Na kontekst związany z rozwiązaniem zadania algorytmicznego składają się dane wejściowe (procesu prowadzącego do rozwiązania zadania) i dane wyjściowe (wynik). Harel oscyluje między dwoma znaczeniami terminu algorytm. Z jednej strony algorytmem nazywa sam przepis jak działać, który jest w pewnym sensie bytem abstrakcyjnym (znaczenie reguł składających się na algorytm?), z drugiej

⁴ A. A. Markow, *Elementy matematycznej logiki*, Moskwa 1984, s. 73.

⁵ P. Dembiński, *Algorytmy*, w: W. Marciszewski (red.), *Logika formalna. Zarys encyklopedyczny z zastosowaniem do informatyki i lingwistyki*, Warszawa 1987, s. 192-196.

⁶ A. Góralski, *Działanie twórcze*, w: A. Góralski (red.), *Zadanie, metoda, rozwiązanie. Zbiór 2*, Warszawa 1978, s. 53-54.

⁷ D. Harel, *Rzecz o istocie informatyki. Algorytmika*, Warszawa 1992.

strony — formalnie spisana wersję tego przepisu, odpowiadającą programowi komputerowemu. „Mówiąc o oprogramowaniu, myślimy najczęściej o rzeczywistych programach — precyzyjnych reprezentacjach algorytmów, zapisanych w specjalnych językach czytelnych dla komputerów, a nie o samych algorytmach”⁸.

Rozwiązanie zadania algorytmicznego stanowi algorytm złożony z elementarnych instrukcji (reguł) zadających akcje (czynności) z ustalonego zbioru. Algorytm ten, wykonany dla dowolnego dopuszczalnego zestawu danych wyjściowych, rozwiązuje zadanie dostarczając wynik zgodnie z oczekiwaniami. Według Harela, aby stworzyć algorytm dysponować trzeba: 1) charakterystyką dopuszczalnego, być może nieskończonego zbioru potencjalnych zestawów danych wejściowych; 2) specyfikacją pożądaných wyników jako funkcją danych wejściowych; 3) zestawem akcji (działań) podstawowych (czynności elementarnych), przy czym różnego typu problemy/zadania wiążą się z różnymi rodzajami akcji podstawowych (elementarnych czynności); 4) dokładną kolejnością wykonania tych akcji.

Elementarne instrukcje (opisujące operacje podstawowe) oraz instrukcje sterujące (kolejnością wykonywania operacji) algorytmu powinny według Harela charakteryzować się następującymi cechami: 1) poziom szczegółowości instrukcji musi być starannie dobrany, tak aby odpowiadał szczególnym możliwościom danego sprzętu; powinien również odpowiadać zdolności pojmowania, jaką ma potencjalny użytkownik tego algorytmu; 2) precyzją, jasnością i jednoznacznością; w algorytmie (odpowiednim dla komputera) nie ma miejsca na mętne sformułowania i niejednoznaczności; 3) każda z operacji podstawowych musi być wykonana w skończonym czasie, w przeciwnym razie algorytm nigdy nie dobiegłby końca.

Dane wyjściowe powinny być ściśle określone, zależnie od zastosowania algorytmu. Na dopuszczalny rodzaj danych wejściowych nakłada się pewien rodzaj specyfikacji. Algorytm powinien zachowywać się zadowalająco dla nieskończonej liczby zestawów danych odpowiadających specyfikacji (uniwersalność). Bazy danych powinny mieć przejrzystą strukturę, pozwalającą odszukiwać dane i nimi manipulować. Algorytm musi także „wiedzieć”, jakimi danymi może manipulować. „Tak jak w przypadku struktur danych, dobry projekt bazy danych — to projekt przejrzysty i łatwy do zapisania; najważniejszą jednak jego zaletą jest duża sprawność działania i wykonalność opartego na tym projekcie systemu zarządzania bazą danych, który potrafi odpowiedzieć na zadane mu pytania w sensownie krótkim czasie”⁹.

Według Harela, algorytmy muszą być zapisane w jednoznaczny i formalny sposób. Aby opisać algorytm na użytek komputera, trzeba posłużyć się językiem programowania. Program to formalna wersja algorytmu, nadająca się do wyko-

⁸ Tamże, s. 17-18.

⁹ D. Harel, dz. cyt., s. 62.

niania przez komputer. Język programowania składa się z symboli i reguł, według których pisze się program. Język programowania wiąże się zwykle ze sztywną składnią, dopuszczającą używanie jedynie specjalnych kombinacji wybranych symboli i słów kluczowych. Wyposażenie języka w precyzyjną składnię stanowi tylko część zadania stojącego przed projektantem języka programowania. Bez formalnej i jednoznacznej semantyki, a więc bez znaczenia każdego wyrażenia dozwolonego składniowo, składnia jest bezwartościowa. Gdyby nie podano znaczenia danej instrukcji w języku programowania, byłoby ono niezrozumiałe. Język programowania wymaga nie tylko sztywnych reguł definiowania postaci poprawnego programu, ale także równie sztywnych reguł definiowania jego znaczenia¹⁰.

2. Reguły heurystyczne

Reguła jest wyrażeniem: 1) nakazującym realizację pewnych celów oraz 2) wskazującym, co powinno się zrobić, aby osiągnąć dany cel, albo czego pod groźbą nieskuteczności robić nie należy¹¹. Pojęcie reguły jest związane z pojęciem celu i skuteczności działania. Poszczególne reguły charakteryzują się różnym stopniem (nie)zawodności, czyli skuteczności w prowadzeniu do wyznaczonego celu. Maksymalną niezawodność posiadają reguły algorytmiczne, które zawsze prowadzą do wyznaczonego celu.

Termin „reguła heurystyczna” zarówno od strony treściowej, jak i zakresowej bywa charakteryzowany różnie, i to w sposób nieostry i niejasny. Bazować będę głównie na pojęciu reguł heurystycznych Imre Lakatosa i jego kontynuatorów, Gary Guttinga i Elie Zahara.

Pojęcie reguł heurystycznych pojawia się u Gary Guttinga¹² i Elie Zahara¹³ w kontekście charakterystyki rozumowań odkrywczych. Według Guttinga rozumowaniami odkrywczymi kierują reguły heurystyczne, formułowane jako wypowiedzi metajęzykowe, takie jak np. zasada analogii, dyrektywa zastępowania bardziej złożonych zadań prostszymi, zasada przegrupowywania itp. Zasady te nazywane są heurystycznymi, ponieważ stosowane wybiórczo, w konkretnych rozumowaniach odkrywczych, jako przesłanki (punkt wyjścia) rozumowania wraz z regułami wyznaczającymi ostateczne cele nauki (np. prawdę) oraz zasadami

¹⁰ Tamże, s. 65-69.

¹¹ T. Pszczółowski, *Zasady sprawnego działania. Wstęp do prakseologii*, Warszawa 1976⁵, s. 61-62; Tenże, *Mała encyklopedia prakseologii i teorii organizacji*, Wrocław 1978, s. 53-54, 205.

¹² G. Gutting, *The Logic of Invention*, w: T. Nickles (ed.), *Scientific Discovery, Logic and Rationality*, Dordrecht 1980, s. 221-233.

¹³ E. Zahar, *Logic of Discovery or Psychology of Invention*, „British Journal for the Philosophy of Science” 34 (1983), nr 3, s. 243-261; Tenże, *The Popper – Lakatos Controversy*, „Fundamenta Scientiae” 1 (1982).

kosmologicznymi, będącymi założeniami dotyczącymi natury świata, prowadzą do odkryć naukowych.

W ujęciu Zahara w punkcie wyjścia rozumowania odkrywczego znajdować się mogą takie zasady heurystyczne, jak: 1) zasada dostosowywania teorii do faktów; 2) zasada korespondencji lub odpowiedniości sformułowana przez N. Bohra w 1923 r.; 3) zasada identyczności (wedle której różnorodność zjawisk wyprowadza się z jednego stałego zbioru); 4) zasada jedności (która ustanawia wymóg podporządkowania wszelkich zjawisk jednej wszechogarniającej teorii nadrzędnej); 5) zasada proporcjonalności skutków i przyczyn; 6) zasada racji dostatecznej; 7) reguła kierowania się intuicyjnym pojęciem prawdopodobieństwa itp. Przy czym geneza zasad heurystycznych jest następująca: ontologia implikuje pewne zasady heurystyczne zgodnie ze schematem: metafizyka (nienaruszalny rdzeń programu badawczego) → postulaty → metawypowiedzi (pozytywna heurystyka).

I. Lakatos posługuje się terminem „reguły heurystyczne” w kontekście charakterystyki programów badawczych oraz ich heurystyki negatywnej i pozytywnej. Reguły heurystyczne są dla niego rodzajem reguł metodologicznych. Według Lakatosa program badawczy „składa się z reguł metodologicznych: część mówi nam, jakich dróg badań unikać (*heurystyka negatywna*), a część, jakimi drogami podążać (*heurystyka pozytywna*)”¹⁴. „Nawet naukę jako całość można uważać za ogromny program badawczy, z naczelną regułą heurystyczną Poppera: «wymyślaj przypuszczenia, które mają bogatszą treść empiryczną od przypuszczeń wcześniejszych». Takie reguły metodologiczne można formułować, jak zauważył Popper, w postaci zasad metafizycznych”¹⁵. „Negatywna heurystyka programu zakazuje nam kierować *modus tollens* w ten «twardy» rdzeń. Zamiast tego musimy zrobić użytek z naszej pomysłowości i wyartykułować, a nawet wymyślić, «hipotezę pomocniczą», tworzącą wokół tego rdzenia *pas ochronny i tam* musimy skierować *modus tollens*”¹⁶. „Heurystyka negatywna określa «twardy rdzeń» programu, który jest «nieobalalny» na mocy decyzji metodologicznej jego zwolenników; heurystyka pozytywna składa się z częściowo wyartykułowanego zbioru sugestii lub wskazówek, w jaki sposób zmieniać, rozwijać «obalalne wersje» programu badawczego, jak modyfikować, udoskonalać (obalalny) *pas ochronny*”¹⁷.

Przytoczone przykłady reguł heurystycznych pokazują, że pojęcie to jest zakresowo szerokie i że nie jest kryterialne. Nie da się odróżnić, posługując się nim,

¹⁴ I. Lakatos, *Pisma z filozofii nauk empirycznych*, wybrał, przełożył i wstępem poprzedził W. Sady, Warszawa 1995, s. 72.

¹⁵ Tamże, s. 72.

¹⁶ Tamże, s. 73.

¹⁷ Tamże, s. 77.

reguł heurystycznych od reguł metodologicznych¹⁸. Lakatos — jako przykład reguły heurystycznej — podaje regułę Poppera: „wymyślaj przypuszczenia, które mają bogatszą treść empiryczną od przypuszczeń wcześniejszych”, która ma tę samą treść, co reguła metodologiczna wyznaczająca cel poznania naukowego: „celem nauki jest wymyślanie przypuszczeń, które mają bogatszą treść empiryczną od przypuszczeń wcześniejszych”¹⁹. Podobnie ma się rzecz np. z wymienioną przez Zahara wśród reguł heurystycznych zasadą jedności. Być może omawiane podobieństwo świadczy o tym, że przydawka „heurystyczny” zastosowana do reguł metodologicznych wskazuje na jedną z funkcjonalnych cech tychże reguł, a więc, że każda reguła metodologiczna w pewnych warunkach może stać się regułą heurystyczną.

Stosunki między regułami heurystycznymi a regułami metodologicznymi można próbować ustalić za pomocą relacji między zbiorami. Można, po pierwsze, przyjmować, że zbiór reguł metodologicznych i zbiór reguł heurystycznych wykluczają się (lub uważać, że zbiór reguł heurystycznych jest zbiorem pustym). Tak uważają ci, którzy akceptują różnicę między kontekstem uzasadnienia i kontekstem odkrycia. Wówczas odpowiedź na pytanie o algorytmizowalność reguł heurystycznych jest negatywna, gdyż samo pojęcie reguły heurystycznej jest *contradictio in adiecto*. Po drugie, możliwe jest pojęcie reguł heurystycznych, przy czym relacje zakresowe między pojęciem reguły metodologicznej i reguły heurystycznej są następujące: zbiór reguł heurystycznych jest podrzędny względem zbioru reguł metodologicznych, a więc są takie reguły metodologiczne, które nie są regułami heurystycznymi, ale nie odwrotnie. Po trzecie, można twierdzić, że zbiory reguł metodologicznych i heurystycznych krzyżują się, co znaczy tyle, że istnieją reguły metodologiczne, które są regułami heurystycznymi, ale istnieją też reguły heurystyczne, które nie są regułami metodologicznymi i odwrotnie. Można też, po czwarte, przyjmować, tożsamość zbioru reguł metodologicznych i zbioru reguł heurystycznych. Sytuacja ta zachodzi, kiedy przyjmuje się bardzo szerokie pojęcie reguł heurystycznych. Ostatnia z możliwych sytuacji: zbiór reguł heurystycznych jest nadrzędny względem zbioru reguł metodologicznych, tzn. każda reguła metodologiczna jest regułą heurystyczną, ale nie odwrotnie. Przy trzech ostatnich wersjach omawianych relacji, odpowiedź na interesujące nas pytanie może być twierdząca, jednak odpowiedź twierdząca jest najbardziej prawdopodobna przy ostatnim rozstrzygnięciu.

Kwestia, czy istnieją reguły heurystyczne inne niż reguły metodologiczne przyjmowane w ramach poszczególnych nauk, jest wtórna w stosunku do pytania, czy istnieją jakieś szczególne, osobne czynności odkrywcze różne od czynności naukowych, takich jak eksperymentowanie, formułowanie praw naukowych

¹⁸ Można tak zdefiniować termin „reguła” i termin „heurystyka”, że samo połączenie w postaci *reguł heurystycznych* staje się *contradictio in adiecto*.

¹⁹ Przypuszczenia rozumiane są tutaj jako hipotezy w sensie Poppera.

czy sprawdzanie ich wartości epistemicznej lub czy istnieją jakieś osobne metody odkrywcze, czyli układy czynności odkrywczych różne od „zwykłych” metod naukowych.

Należałoby zatem sprecyzować pojęcie czynności odkrywczych, co znów odсылa do pojęcia odkrycia naukowego. Według E. Pietruskiej-Madej o odkryciu *sensu stricto* naukowym możemy mówić wtedy, gdy „nowa idea pojawi się w nauce i zacznie funkcjonować w zobiektywizowanej postaci”²⁰. „Odkrycie naukowe — to zaistnienie w nauce nowej (dla nauki) idei”²¹. Coś jest zatem odkrywcze, jeżeli jest nowe, oryginalne. Obydwa pojęcia, nowości i oryginalności są trudne do sprecyzowania. Mają charakter relacyjny. Coś jest nowe lub oryginalne w stosunku do czegoś innego. Na przykład, nowe jest jakieś twierdzenie w stosunku do twierdzeń już istniejących w nauce²². Nie chodzi tu o nowość czy oryginalność czynności odkrywania, ale ich rezultatu w jakiejkolwiek postaci — czy to twierdzenia, czy pytania, czy eksperymentu.

Określenie czynności jako odkrywczych jest wtórne w stosunku do przypisania kwalifikacji bycia odkryciem jakimś rezultatu czynności naukowych²³. Zakłada ono, że rezultaty owych czynności zostały wcześniej ujętykowane, co stanowi warunek konieczny spełniania przez dany wytwór warunków/reguł go obowiązujących oraz warunek konieczny, aby dane odkrycie zaczęło funkcjonować w postaci intersubiektywnie dostępnej. Wydaje się, że nie ma jakichś osobnych czynności odkrywczych, różniących się w sposób zasadniczy od „zwykłych” czynności naukowych.

Według H. A. Simona odkrycie naukowe jest odmianą rozwiązywania problemów, a procesy, za pomocą których nauka jest uprawiana, mogą być wyjaśnione w terminach, których używa się do wyjaśnienia procesów rozwiązywania problemów²⁴. Również przy takim ujęciu odkrycia, nie traktuje się czynności odkrywczych jako zasadniczo różnych od pozostałych czynności naukowych.

²⁰ E. Pietruska-Madej, *Odkrycie naukowe. Kontrowersje filozoficzne*, Warszawa 1990, s. 232.

²¹ Tamże, s. 242.

²² Ostatecznie, jeśli próbuje się sprecyzować pojęcie nowości, trzeba przyjąć istnienie jakiegoś obiektywnego rezerwuaru wiedzy, np. w postaci Popperowskiego trzeciego świata.

²³ A. Góralski określa działanie twórcze (nie „odkrywcze” — choć mniej więcej o to samo chodzi) jako „działanie warunkowane przez konieczność osiągnięcia wartości, realizowane w warunkach niedostateczności środków lub sposobów”. (A. Góralski, *Działanie twórcze*, w: A. Góralski [red.], *Zadanie, metoda, rozwiązanie. Zbiór 2*, Warszawa 1978, s. 13-14). Jest to próba określenia działania twórczego bez bezpośredniego odniesienia do wytworu tego działania. Definicja ta równocześnie przesądza/zakłada, że działanie twórcze (odkrywcze) nie może mieć charakteru metodycznego (a tym bardziej algorytmicznego), gdyż jest to właśnie działanie wykonywane w warunkach „niedostateczności” metody.

²⁴ H. A. Simon, *Models of Discovery and other Topics in the Methods of Science*, Dordrecht 1977.

Mozna próbować wyróżnić w ramach procesu prowadzenia badań naukowych jeden typ działań i nazywać je odkrywczymi, na przykład te, w które zaangażowana jest (twórcza) intuicja (co to jest), a równocześnie prowadzące do rozwiązania, które jest nowatorskie w stosunku do dotychczasowej wiedzy. Jednak wydaje się, że bardziej adekwatne będzie przyjęcie, że czynności odkrywcze nie pozostają w izolacji od pozostałych czynności naukowych, lecz są przez nie warunkowane, zatem nie są czymś zasadniczo różnym od pozostałych czynności poznawczych.

3. Trudności z algorytmizacją reguł heurystycznych

O algorytmizacji (w pewnym sensie dyskursywizacji) można mówić w przypadku zbiorów reguł, a nie pojedynczej reguły. Algorytmizacja polega między innymi na pewnego typu uporządkowaniu tego zbioru. To samo dotyczy reguł heurystycznych. To ewentualne zbiory reguł heurystycznych bądź zbiory reguł, wśród których niektóre są regułami heurystycznymi, są algorytmizowalne albo niealgorytmizowalne.

W przytoczonych wcześniej określeniach algorytmu zwraca się uwagę na to, że algorytmy dostarczają rozwiązań określonego typu zadań, niekiedy zwanych wprost zadaniami algorytmicznymi. Czym są zadania algorytmiczne? Jak je scharakteryzować? Czym się one różnią od innych typów zadań? Czy każdy typ zadania (problemu) nadaje się do algorytmizacji, także zadania odkrywcze (o ile takie są)? Na czym polega algorytmizacja pewnego problemu? Być może samo postawienie problemu ma charakter twórczy i w ten sposób proces odkrywania nie zaczyna się wcale od prób rozwiązania określonego zadania, ale od znalezienia samego pytania? A więc samo zadanie ma już charakter algorytmiczny? Czy da się podać algorytmiczne rozwiązanie dla niealgorytmicznego zadania? Co jest pierwotnie, a co wtórne: zadanie jest algorytmiczne, bo istnieje algorytm będący jego rozwiązaniem, czy też tylko algorytmiczne zadania posiadają rozwiązanie w postaci algorytmów? Na te pytania należałoby odpowiedzieć, aby ustalić prawdziwość zdania, że reguły heurystyczne są (nie)algorytmizowalne.

Algorytmy muszą być zapisane w jednoznaczny i formalny sposób. Czy w przypadku reguł heurystycznych da się stworzyć język o wyliczonych cechach? (Czy zapisanie reguł w języku sformalizowanym jest warunkiem koniecznym ich algorytmiczności?).

Przejście od punktu wyjścia (danych początkowych algorytmu) do punktu dojścia algorytmu (rezultatu) w przypadku odkrycia naukowego natrafia na trudności znane pod nazwą paradoksu Menona. Punktem wyjścia dla tego paradoksu jest pytanie: jak w ogóle możliwe jest odkrywanie (poznawanie)? Jak możliwe jest przejście od stanu niewiedzy do stanu wiedzy? Platon kwestionuje jakąkolwiek możliwość przejścia od stanu niewiedzy do stanu wiedzy, ponieważ „[...] nie może człowiek szukać ani tego, co zna, ani tego, czego nie zna. Bo jak zna to

przecież nie będzie szukał. Już to zna. Takemu nie trzeba dopiero szukać. Ani tego, czego nie zna. Bo wtedy nie wie tego nawet, *czego ma szukać*²⁵. Różnie próbowano przezwyciężyć ten paradoks²⁶. Czy jest on do przezwyciężenia, jeśli wiedza w punkcie dojścia jest zasadniczo nowa, tzn. nie powstaje w wyniku przekształcenia danych początkowych? Wydaje się, że nie. W tym punkcie także powstają wątpliwości co do możliwości algorytmizacji reguł heurystycznych, gdyż dane wejściowe algorytmu (dla komputera) są zawsze punktem wyjścia dla przekształceń prowadzących do żądanego rezultatu. Jeśli odkrycie naukowe jest rzeczywiście odkryciem, jest ono czymś zasadniczo różnym od danych wejściowych. Jeśli potraktować algorytm jako rozumowanie (dedukcyjne?), to teza poprzedniego zdania brzmiałaby następująco: nie ma niczego we wniosku, czego nie było wcześniej w przesłankach.

Jako jedną z ważnych cech algorytmu wymienia się to, że prowadzi on niezawodnie do żądanego rezultatu. Efektywność algorytmu polega na zgodności rezultatu działań algorytmicznych z zamierzonym celem tych działań. Przy takim pojęciu efektywności trzeba wiedzieć, co jest rezultatem działań algorytmicznych, a w przypadku odkryć dysponuje się jedynie ogólną charakterystyką kryteriów, jakie powinien spełniać rezultat, a nie jego dokładnym opisem. Ponadto pojawiają się trudności z kryterialnością wartości epistemicznych (np. prawdziwością, stopniem pewności), którymi trzeba się posłużyć oceniając rezultat działania algorytmu jako funkcji danych wejściowych.

Trudności są także z uporządkowaniem zbioru czynności (reguł), gdyż wymaga ono znajomości nie tylko efektu końcowego, do którego ma prowadzić algorytm, ale także określenia celów częściowych, do których prowadzą poszczególne czynności algorytmu. W przypadku czynności odkrywczych cele częściowe również nie są znane. Dlatego trudno zaplanować całość algorytmicznego działania. Trudności te są częściowo eliminowane przez przyjmowanie — w ramach wiedzy wyjściowej dla algorytmu — reguł dostarczających kryteriów wartościowania rezultatów czynności algorytmicznych. Jednak natrafia się tu na wspomniane problemy z kryterialnością wartości epistemicznych.

Cechą charakteryzującą algorytm jest zupełność. Jeśli zdefiniuje się ją jako możliwość przewidzenia wszelkich możliwych sytuacji (decyzyjnych) wraz z określeniem sposobu postępowania w przypadku ich wystąpienia, to, jak już wcześniej wspomniano, należałoby znać wszelkie możliwe cele częściowe, do których prowadzą poszczególne czynności i ich funkcję w dążeniu do celu nadrzędnego. Natomiast w przypadku procesu odkrywania nie da się tych celów ustalić.

Algorytm ma cechę uniwersalności, tzn. możliwość zastosowania go do określonej grupy problemów. Algorytmem nie jest zbiór reguł stosowalnych jedno-

²⁵ Platon, *Menon*, 80, D, F, przeł. W. Witwicki, Warszawa 1959, s. 34.

²⁶ Próby te omawia krótko E. Pietruska-Madej, dz. cyt., s. 211-224.

razowo, lecz taki, który pozwala na wielokrotne stosowanie. W samym pojęciu reguły stosowalnej jednorazowo tkwi sprzeczność. Jeżeli istnieją reguły heurystyczne, to powinny one spełniać warunek uniwersalności, natomiast odkrycie naukowe z natury jest jednostkowe.

Z jakich czynności elementarnych/reguł składałyby się algorytmy heurystyczne, gdyby istniały? Jeżeli za jedną z elementarnych czynności odkrywczych uzna się akty intuicji, to nie ma (algorytmicznych) reguł intuicji.

Wydaje się, że możliwą odpowiedź na pytanie o algorytmizowalność odkryć naukowych należałoby zrelatywizować do typologii odkryć naukowych, uwzględniającej stopniowalny charakter nowości odkrycia.

Próba algorytmizowania odkryć naukowych jest próbą przeniesienia problematyki odkryć z płaszczyzny pragmatycznej na płaszczyznę syntaktyczną. Jeśli możliwa jest taka redukcja, to istnieje także możliwość algorytmizacji reguł heurystycznych/odkryć naukowych.

Hermeneutic Algorithms

The author is interested in the question, if it is possible to discover any patterns in the process of making scientific discoveries. Usually it is claimed that discoveries are unpredictable and therefore cannot be generated by applying algorithmic methods. Whether this belief is true or not, is one problem, whether it is well supported or convincing, is another. After the preliminary elucidation of fundamental concepts, the author compares the views of I. Lakatos, G. Gutting and E. Zahar to illustrate some specific problems. She reaches a bold conclusion that the chances of finding heuristic algorithms depend on the possibility of transposing pragmatic problems into syntactic ones. Such reduction is not impossible, claims the author, and consequently says that the hope for finding heuristic algorithms cannot be rejected out of hand.