

Rafał Kasterski

PRZEDMIOT OCHRONY PRAWNEJ PROGRAMÓW KOMPUTEROWYCH

I

Dyskusja na temat potrzeby i środków ochrony prawnej programów komputerowych trwa już wiele lat. Wiele kwestii dotąd pozostało nierozstrzygniętych, jedna wszakże wydaje się nie budzić już dziś większych wątpliwości, a mianowicie konieczność poddania programów skutecznej ochronie prawnej¹.

Podstawą takiego zapatrywania są przede wszystkim względy natury gospodarczej i to zarówno w odniesieniu do interesów indywidualnego producenta, jak i w skali całej gospodarki². Poważne argumenty prze-

¹ Spośród bogatej literatury por. zwi. A. Kopff, *Wpływ postępu techniki na prawo autorskie*, Warszawa—Kraków 1988, s. 31—40; przeglądowe opracowanie M. Kindermanna, *The International Copyright of Computer Software — History, Status and Developments*, „Copyright” 1988, No 4, s. 201 i n. oraz M. Röttingera, *Rechtsschutz von Computerprogrammen, Die Diskussion in der Bundesrepublik Deutschland, Österreich und der Schweiz in der vergangenen 20 Jahren*, „Informatik und Recht” 1986, Nr. 7—8, s. 293 i n.; 1987, Nr. 3, s. 93 i n. oraz Nr. 4, s. 139 i n. Niniejszy artykuł został przygotowany w 1990 r.

² Co ciekawe, względy gospodarcze bywają koronnymi argumentami zarówno zwolenników, jak i przeciwników poddania oprogramowania ochronie prawnej — por. np. J. Waluszewski, *Ochrona programów dla maszyn cyfrowych w Ustawie Wzorcowej Światowej Organizacji Własności Intelektualnej*, PiP 1980, nr 8, s. 81—83 i 88—89. Ostatnie lata przyniosły jednak szereg publikacji obciążających częścią odpowiedzialności za stan polskiej informatyki fakt braku jasnych i skutecznych mechanizmów „prawno-ekonomicznych” — por. Uchwałę II Zjazdu Polskiego Towarzystwa Informatycznego, „Informatyka” 1989, nr 2, s. 7—8; A. Blikle, *O uwarunkowaniach rozwoju systemów informatycznych i przemysłu oprogramowania*, „Informatyka” 1989, nr 2, s. 4 i 13; J. Irlik, *Ochrona prawna programów na świecie*, [w:] *Współczesne kierunki rozwoju informatyki*, Mrągowo 1988; *Stanowisko Polskiego Towarzystwa Informatycznego w sprawie ochrony prawnej programów komputerowych*, „Przegląd Techniczny” 1989, nr 14, s. 20.

mawiają też za koniecznością poddania ochronie majątkowych i niemajątkowych interesów twórcy.

Jak przy tym wiadomo, programy komputerowe są wyjątkowo łatwym obiektem kopiowania. Rezultatem takiego stanu rzeczy jest opanowanie krajowych targów przez „adaptacje”, „aplikacje” czy wręcz zwyczajne kopie programów i literatury zachodniej. Do rangi symbolu urasta fakt, że oprogramowanie tzw. komputera dla szkół oparte było w znacznej mierze na programach „zaadaptowanych” bez wiedzy i zgody ich producenta. Tymczasem zagraniczni kontrahenci coraz częściej żądają udokumentowania przez polskich programistów legalności wykorzystywanych przy opracowywaniu programów „narzędzi”, a więc kompilatorów używanych języków programowania etc.

Względy te, w moim przekonaniu, przesądzają o potrzebie poddania programów komputerowych ochronie prawnej. Ustalić natomiast należy najwłaściwszą jej formę z punktu widzenia istoty dobra, jakim jest program, a także zapewnienia adekwatnej ochrony interesu autorów, producentów i użytkowników oprogramowania. Potrzeby w tym zakresie najpełniej określiła Komisja ds. Prawnych Uwarunkowań Informatyki działająca przy Zarządzie Głównym Polskiego Towarzystwa Informatycznego w *Stanowisku w sprawie ochrony prawnej programów komputerowych*. Postuluje się w nim, by twórcy programu przysługiwało prawo do ochrony dzieła oraz by uprawnienia majątkowe do programu posiadała osoba, która poniosła nakłady na jego opracowanie. Prawa te powinny być zbywalne w całości lub w części.

Takie działania, jak nieautoryzowane korzystanie z programu, kopiowanie, odtwarzanie z kodu wynikowego programu w formie źródłowej oraz wykorzystywanie istotnych rozwiązań przy opracowywaniu innych programów, stanowiąc mają naruszenie uprawnień do programu.

W artykule nie będę szczegółowo omawiał możliwych do zastosowania form ochrony prawnej programów. Dziś wiadomo już, że pomimo wielu wątpliwości teoretycznych, orzecznictwo, a następnie ustawodawstwo wielu krajów zastosowało do ochrony oprogramowania system ochrony uniwersalnej zapewnianej przez prawo autorskie. Wspomniane przeze mnie wątpliwości dotyczą przede wszystkim kwestii, czy program komputerowy można uznać za utwór w rozumieniu prawa autorskiego, a jeżeli tak, to czy jest w nim miejsce na wyrażenie indywidualnego wkładu (piętna) twórcy.

Sądzę, że możliwe do przyjęcia jest stanowisko akceptujące tezę, że programy są utworami zbliżonymi do dzieł naukowych, posiadającymi często znaczący walor estetyczny. Są one więc przedmiotem prawa autorskiego o tyle, o ile są oryginalne, tzn. jeżeli elementy ich formy i (lub)

treści noszą piętno indywidualności twórcy i nie są w pełni wyznaczone przez elementy wchodzące w skład domeny publicznej³.

Problematyka uprawnień majątkowych producenta oprogramowania zatrudniającego programistę, zakresu prawa do integralności programu, ograniczenia dozwolonego użytku prywatnego jedynie do sporządzenia kopii bezpieczeństwa (*back-up copy*) oraz czasu trwania ochrony ma w moim przekonaniu charakter wtórny. Kwestie te możliwe są ponadto do rozstrzygnięcia na gruncie obowiązującego prawa autorskiego.

W dalszych rozważaniach chciałbym się jednak skoncentrować na zagadnieniu dotychczas w piśmiennictwie pomijanym. Dyskusja na temat środków ochrony prawnej programów toczy się bowiem — paradoksalnie — bez wcześniejszego jednoznacznego ustalenia zakresu takich pojęć, jak program komputerowy i algorytm, często przy tym używanych zamiennie.

Brak pogłębionej analizy przedstawionego tu zagadnienia jest charakterystyczny nie tylko dla znaczącej części naszego piśmiennictwa, lecz również dla wypowiedzi wielu przedstawicieli nauki zachodniej⁴. Rezultatem takiego stanu rzeczy jest, prócz zrozumiałych trudności i niejasności w toku rozważań szczegółowo zajmujących się kwestią środków ochrony prawnej programów, niemożność znalezienia wspólnej płaszczyzny porozumienia ze środowiskiem informatyków, którzy ze zdziwieniem konstatują, że powszechnie używane przez nich pojęcia nabierają w ustach prawników zupełnie nowych znaczeń.

II

Punktem wyjścia naszych rozważań będzie stwierdzenie, że algorytm jest sposobem rozwiązania określonego zadania. Ta niezwykle ogólna definicja wymaga dookreślenia, jednak dzięki swej lapidarności jest przydatna do przedstawienia pierwszych pojawiających się wątpliwości.

Mówiąc bowiem o algorytmie jako o sposobie rozwiązania danego zadania możemy, opierając się na „słownikowej” definicji algorytmu, rozumieć przez to sposób realizacji ściśle określonego zadania przez wykonanie sekwencji działań elementarnych⁵.

³ Por. B. Czachórska, *Umowy w zakresie informatyki i ochrona programów komputerowych*, Wrocław 1980, s. 72; Kopff, *Wpływ...*, s. 34—35; E. Ulmer, G. Kollé, *Copyright Protection of Computer Programs*, IIC 1983, nr 2, s. 172 i n.

⁴ Por. H. Haberstumpf, *Computerprogramm und Algorithmus*, UFITA 1983, nr 95, s. 221 i n.

⁵ Por. *Encyklopedia powszechna*, Warszawa 1973, s. 58; *Słownik języka polskiego*, Warszawa 1988, s. 33; J. Irlik, *Kwalifikacja prawna programu komputerowe-*

Wyjaśnienia wymaga pojawiające się w tym ujęciu pojęcie „działania elementarnego”. Jest to działanie, którego sposób wykonania jest dla wykonującego je oczywisty. Zatem to, czy określony plan działania jest dla danego adresata algorytmem, zależy od tego, czy operacje planu są dla niego oczywiste, a w każdym razie możliwe do wykonania. Poziom szczegółowości algorytmu jest więc wyznaczany przez repertuar nazw czynności i klasę obiektów nie wymagających dla konkretnego zbioru adresatów dalszych uściśleń⁶. Informatyk piszący program dla komputera układa algorytm na podstawie zbioru operacji elementarnych maszyny lub zestawu takich operacji, jakie komputer może wykonać opierając się na przekazanych wcześniej algorytmach (chodzi tu o instrukcje języka programowania, w którym program może być napisany)⁷. Programista musi sformułować algorytm tak dokładnie, by według jego instrukcji komputer mógł działać automatycznie. Innymi słowy, algorytm należy zapisać jako program w języku programowania — specjalnej symbolice, w której formuluje się algorytm na poziomie szczegółowości, odpowiednim do mechanicznej interpretacji⁸. Program jest więc jednym ze sposobów zapisania algorytmu, jedną z postaci jego utrwalenia.

W powyższych rozważaniach kilka razy pojawiło się pojęcie „języka programowania”. Przekazując maszynie algorytm, musimy posłużyć się językiem, który będzie dla niej „zrozumiały”. Język taki, służący do formułowania algorytmów przeznaczonych do wykonania przez maszynę nazywamy językiem programowania. Nie można jeszcze do tego celu użyć języka naturalnego, ponieważ jego składnia i semantyka są zbyt złożone. Język, w którym jedynymi dopuszczalnymi wyrażeniami są instrukcje należące do repertuaru instrukcji, w jakie konstrukcyjne została wyposażona maszyna, nazywamy językiem maszynowym.

Dla ułatwienia pracy programisty opracowano tzw. języki wysokiego poziomu. Ich przyswojenie jest dużo prostsze niż języka maszynowego. Język taki nie jest jednak bezpośrednio „zrozumiały” dla maszyny. Konieczne jest „przetłumaczenie” algorytmu zapisanego w języku wysokiego poziomu na język maszynowy. Służą do tego specjalne

go, „Informatyka” 1985, nr 7, s. 6; J. C. Faure, *Zarys oprogramowania maszyn cyfrowych*, Warszawa 1977, s. 21.

⁶ Irlik, *Kwalifikacja...*, s. 6; W. M. Turski, *Propedeutyka informatyki*, Warszawa 1977, s. 53.

⁷ J. Irlik, *Umowy zawierane w obrocie oprogramowaniem*, [maszyn. niepubl.], Katowice 1984, s. 2.

⁸ S. Alagic, M. A. Arbib, *Projektowanie programów poprawnych i dobrze zbudowanych*, Warszawa 1982, s. 9; tak też sądzą P. Dembiński, J. Małuszkański, *Matematyczne metody definiowania języków programowania*, Warszawa 1981, s. 13.

programy, które zamieniają ciąg poleceń wydanych w formie instrukcji charakterystycznych dla danego języka na ciąg instrukcji w kodzie maszynowym. Z punktu widzenia programisty odbywa się to „automatycznie”. Można więc obrazowo powiedzieć, że „widzi” on komputer przez pryzmat użytego języka programowania, pisząc zaś program uszczegóławia go do poziomu wybranego języka.

Konstrukcja ta, jakkolwiek dość powszechnie przyjęta w opracowaniach z dziedziny informatyki, nie znalazła szerszej aprobaty w piśmiennictwie prawniczym. Większość autorów zdaje się przyjmować, że algorytm jest jedynie matematyczno-logiczną koncepcją sposobu postępowania⁹, abstrakcyjną ideą rozwiązania pewnej klasy zadań, sformułowaną przy tym na dość wysokim stopniu ogólności, a więc jest swego rodzaju pomysłem leżącym u podstaw programu, który jest jego dookreśleniem, uszczegółowieniem.

Co więcej, niejednoznaczne jest też pojęcie zadania jakie stoi przed programistą. Ujęcie wąskie koncentruje się jedynie na podstawowym, ogólnie sformułowanym problemie, bez odwoływania się np. do szczególnych potrzeb użytkownika czy uwarunkowań sprzętowych.

Ujęcie szerokie obejmuje swoim zakresem również szczegółowo opisane w specyfikacji programu wymagania, takie jak określone zachowanie w szczególnych sytuacjach, określona reakcja na błędy danych, błędy obsługi, a także tzw. przyjazność wobec użytkownika. W piśmiennictwie zachodnim określa się ją mianem „look and feel”, co polscy autorzy¹⁰ definiują jako wygodę użytkowania systemu, na którą składa się przejrzystość i czytelność „menu”, prostota zasad komunikowania się z systemem, łatwość opanowania umiejętności formułowania poleceń, przejrzystość odpowiedzi. Na „przyjazność” składa się również szybkość reakcji systemu, która jest uzależniona m.in. od zastosowanych w programie algorytmów obliczeniowych, określonej jego struktury itd., a także walor estetyczny, w tym szata graficzna systemu.

Konsekwencją przedstawionych tu rozróżnień są dwie spotykane w piśmiennictwie koncepcje relacji między algorytmem a programem.

Pierwsza z nich zakłada, że algorytm to ogólna metoda rozwiązywania określonego typu ogólnie sformułowanych zadań, mająca charakter matematyczny lub logiczny, zaś program to rozwinięcie, dookreślenie algorytmu dla potrzeb konkretnego, szczegółowo opisanego zadania¹¹.

⁹ Np. Czachórska, *Umowy...*, s. 32; W. Linden, *Technologia informatyczna a rozwój ochrony prawnej*, ZNUJ 1990, *Prace z Wynalazczości i Ochrony Własności Intelektualnej*, z. 52, s. 29.

¹⁰ Np. J. Irlík, *Wypowiedź w kwestii ochrony oprogramowania*, [w:] *Współczesne kierunki rozwoju informatyki*, Mrągowo 1987, s. 99.

¹¹ Por. Czachórska, *Umowy...*, s. 32; S. Sołtysiński, *Ochrona progra-*

Koncepcja druga przyjmuje, że algorytm to szczegółowy przepis postępowania, doprowadzony do poziomu działań elementarnych, obejmujący opis obiektów łącznie z opisem czynności, jakie należy wykonać z tymi obiektami, by osiągnąć z góry określony cel¹². Program w tym ujęciu to jedna z form zapisu algorytmu.

W moim przekonaniu, jeżeli mówimy o algorytmie dla komputera, to należy przyjąć koncepcję drugą. Ujęcie to jest poprawniejsze z semantycznego punktu widzenia, lepiej też oddaje rzeczywiste relacje między algorytmem a programem. Dużą jego zaletą jest również to, że opiera się na słownikowych, powszechnie przyjętych definicjach, a przez to jest zrozumiałe nie tylko dla prawników. Przyjęcie tej koncepcji pociąga za sobą jednak potrzebę objęcia ochroną prawną obok programu także, a nawet przede wszystkim, algorytmu.

Potrzebę i rolę przedstawionych wyżej uściśleń terminologicznych najłatwiej ukazać na przykładach z piśmiennictwa. Wielu autorów podejmując próbę oceny możliwości poddania programów komputerowych ochronie prawnej, rozpoczyna swoje rozważania od zdefiniowania algorytmu, opierając się przy tym przeważnie na ujęciu „słownikowym”. Dowiadujemy się więc, że algorytm jest metodą rozwiązywania określonego typu zagadnień, podającą w języku symbolicznym zbiór przepisów postępowania określających kolejno po sobie następujące czynności, które trzeba wykonać, by rozwiązać zadanie pewnego typu¹³. Algorytm jest również definiowany jako podstawa rozwiązywania konkretnego zindywidualizowanego zadania lub klasy zadań¹⁴, wreszcie jako dokładny przepis wykonania w określonym porządku skończonej liczby operacji, pozwalający na rozwiązanie każdego zadania danego typu¹⁵. W toku dalszych rozważań przedstawiany jest proces opracowywania programu, podzielony na takie etapy, jak np.: oznaczenie problemu i jego analiza, projektowanie technologii przetwarzania danych, kodowanie, testowanie i opracowywanie dokumentacji programu¹⁶.

mów do maszyn cyfrowych, ZNUJ 1973, Prace z Wynalazczości i Ochrony Własności Intelektualnej, z. 1, s. 368; B. Gawlik, *Ochrona programów komputerowych na gruncie prawa autorskiego*, [w:] *Współczesne kierunki rozwoju informatyki*, Mąragowo 1987, s. 174—179.

¹² Tak cytowany już Turski, *Propedeutyka...*, s. 53; Irlik, *Umowy...*, s. 2; Alagic, Arbib, *Projektowanie...*, s. 9; Haberstumpf, *Computerprogramm...*, s. 223—225.

¹³ J. Waluszewski, *Programy dla maszyn cyfrowych jako przedmiot prawa*, ZNUJ 1975, Prace z Wynalazczości i Ochrony Własności Intelektualnej, z. 5, s. 70.

¹⁴ J. Waluszewski, *Ochrona programów dla maszyn cyfrowych*, [w:] *Informacja naukowo-techniczna a prawo*, Katowice 1984, s. 130.

¹⁵ Waluszewski, *Ochrona programów dla maszyn cyfrowych*, s. 130.

¹⁶ Por. Czachórska, *Umowy...*, s. 32—33.

Przy analizie opisów tych etapów dochodzimy do wniosku, że powołani autorzy niepostrzeżenie przechodzą do pierwszej z wyżej wymienionych koncepcji algorytmu. Najczytelniej czyni to B. Gawlik, który wręcz dzieli rezultaty pracy programisty na algorytm, ogólny schemat blokowy, szczegółowy schemat blokowy i program dla maszyny¹⁷.

W tym kontekście pojawia się teza głosząca, że ten sam algorytm może stanowić podstawę do opracowania różnych szczegółowych opisów, a więc istotnie odmiennych programów. Teza ta stoi jednak w sprzeczności z przyjętą przez nas koncepcją algorytmu, gdyż — jak powiedzieliśmy wcześniej — program jest formą zapisu algorytmu. Źródło sprzeczności leży, jak sądzę, w równoczesnym używaniu pojęcia „algorytm” w różnych znaczeniach. Mianowicie albo jako metody rozwiązania jednego z wielu problemów cząstkowych, jakie składają się na zadanie (w tym właśnie znaczeniu mówimy o „użytych algorytmach obliczeniowych”), albo jako metody rozwiązania całego zadania. O nieporozumienie tu nietrudno, tym bardziej że w literaturze informatycznej propaguje się tzw. programowanie zstępujące (analityczne), polegające w najogólniejszym zarysie na rozkładaniu problemu na określone podproblemy i dowodzeniu, że jeżeli każdy podproblem jest rozwiązany prawidłowo, to cały problem też jest rozwiązany poprawnie¹⁸. Sposób rozwiązania każdego z podproblemów jest oczywiście algorytmem w szerokim znaczeniu tego pojęcia, jednak dopiero o sposobie rozwiązania całego problemu (jeżeli został on doprowadzony do poziomu działań elementarnych) można powiedzieć, że jest on algorytmem rozwiązania określonego zadania przeznaczonym dla danej maszyny.

III

Można zapytać, jakie jest praktyczne znaczenie przedstawionych różnic. W moim przekonaniu grają one niebagatelną rolę przy ustalaniu zakresu ochrony prawnej. W piśmiennictwie dotyczącym prawa autorskiego spotykamy się bowiem z tezą, że idee leżące u podstaw programów komputerowych powinny pozostać poza ochroną¹⁹. Czy dotyczy to również algorytmów? Dość powszechne jest przekonanie, że tak²⁰. Teza ta może

¹⁷ Gawlik, *Ochrona...*, s. 174—179.

¹⁸ Alagic, Arbib, *Projektowanie...*, s. 9—10.

¹⁹ Por. Model Provisions on the Protection of Computer Software, „Copyright” 1978, No 1, s. 12 i n. (Art. 4).

²⁰ Ulmer, Kolle, *Copyright...*, s. 182; Waluszewski, *Ochrona programów dla maszyn cyfrowych*, s. 140; por. też Kopff: *Wpływ...*, s. 35; Linden, *Technologia...*, s. 29.

się jednak okazać trudna do obrony, jeśli zważywszy, że po ustaleniu algorytmu do zrobienia pozostają już tylko rutynowe czynności zawodowego programisty. Jak pisze Gawlik, indywidualne cechy działalności programisty mogą wyczerpywać się w opracowaniu „opisu programu”, zaś samo sporządzenie „programu dla maszyny” stanowi wówczas jedynie reprodukcję opisu w formie czytelnej dla komputera²¹. Z dotychczasowych rozważań wynika jednak, że nie będzie to, jak chce ww. autor, wyjątkowy wypadek, lecz sytuacja normalna, zaś „opis programu”, to po prostu jedna z możliwych form zapisu algorytmu rozwiązania zadania.

IV

Omawiając istotę algorytmu należy postawić pytanie, czy nie jest on formułą matematyczną opisującą obiektywnie istniejącą rzeczywistość, a więc, czy nie jest odkryciem naukowym lub zasadą naukową. Rozstrzygnięcie tej kwestii będzie miało duże znaczenie dla ewentualnej ochrony algorytmu i programu, ponieważ powszechnie uznaje się, że odkrycia naukowe powinny pozostawać poza ochroną prawa.

Jakkolwiek nie sposób w krótkim artykule przedstawić szczegółowej analizy tego zagadnienia, niemniej dla zarysowania problemu należy odwołać się do przyjętego w piśmiennictwie prawniczym znaczenia pojęcia „odkrycie naukowe”. Zostało ono zdefiniowane jako stwierdzenie określonych właściwości lub zjawisk występujących w przyrodzie, które były dotychczas nie znane nauce. Zasady naukowe natomiast, to próby określenia prawidłowości zachodzących w świecie materii i świecie pojęć ujętych w formę praw fizyki i chemii, matematyki i logiki, estetyki i nauk społecznych etc.²² Należałoby zatem rozważyć, czy algorytmy spełniają przedstawione wyżej warunki. Jak już wspomniałem, rozważania dotyczące tego zagadnienia przekraczają ramy krótkiego szkicu. Ze swojej strony chciałbym raczej wnieść do dyskusji kilka dodatkowych argumentów niż starać się wyciągnąć ostateczne wnioski.

W piśmiennictwie można spotkać się z dwoma ujęciami problemu „naukowości” algorytmu. Pierwsze akceptuje tezę, że algorytm może być rozpatrywany w kategoriach odkrycia naukowego. Ujęcie drugie, bardziej rozpowszechnione, w zasadzie nie ustosunkowuje się do wspomnianej kwestii, akceptując wszakże autorską lub wynalazczą ochronę oprogramowania, pośrednio odrzuca stanowisko pierwsze.

²¹ Por. Gawlik, *Ochrona...*, s. 177–178; tak też uważa Irlik, *Kwalifikacja...*, s. 6.

²² S. Sołtysiński, *Prawo wynalazcze, Komentarz*, Warszawa 1975, s. 13.

Przykładem rozwinięcia pierwszego stanowiska może być argumentacja J. Irlika²³. Jego zdaniem algorytm jest obiektywnie sprawdzalnym twierdzeniem ustalającym fakt dotyczący obiektywnie istniejącej rzeczywistości, stwierdzający, że dane zadanie może być zrealizowane przez wykonanie sekwencji działań elementarnych.

Moim zdaniem jednak, rozumowanie Irlika nie uwzględnia dwóch ważnych aspektów problemu. Po pierwsze należy, jak sądzę, konsekwentnie przyjąć szerokie ujęcie problemu (zadania) stojącego przed programistą. Wiemy, że algorytm jest szczegółowym opisem jego rozwiązania. Opis taki z reguły jest w stanie napisać przeciętny programista, znikome jest przy tym niebezpieczeństwo, że będzie on „taki sam” jak inne algorytmy rozwiązujące ten sam problem. Gdyby zastosować analogię do wywodów Irlika można by powiedzieć, że sformalizowany, szczegółowy opis budowy mostu też jest obiektywnie sprawdzalnym twierdzeniem ustalającym fakt dotyczący rzeczywistości (w tym wypadku, że wykonanie określonej sekwencji działań elementarnych doprowadzi do rezultatu jakim będzie dany most). Wyraźnie jednak widać, że formalizm zapisu spowodowany specyfiką adresata (w przypadku oprogramowania adresem jest maszyna) nie może przysłać nam znikomej naukowej doniosłości takiego ustalenia. Nie sposób oczywiście wykluczyć, że przy pracy nad algorytmem może dojść do sformułowania ogólniejszej zasady, prawa, metody. Sytuację taką należy jednak odróżnić od mniej lub bardziej nowatorskiego zastosowania danej zasady dla osiągnięcia konkretnej funkcji. Algorytm dla komputera *per se* nie jest z reguły ani zasadą, ani prawem²⁴.

Po wtóre, nie bez znaczenia jest cel działalności programisty. Jest to cel użytkowy. W praktyce spotykamy programy, które lepiej lub gorzej spełniają swoje zadania. W niedoskonałości tej (zaznaczam, że nie mam tu na myśli niedoskonałości powodującej, że program w ogóle nie spełnia swoich funkcji) wyraża się m. in. oryginalność algorytmu, kwestię tę pozostawiać jednak jedynie zasygnalizowaną.

Z wywodów powyższych należy, jak sądzę, wyciągnąć wniosek, że z reguły algorytm dla komputera nie jest odkryciem naukowym, brak zatem podstaw, by pozostawiać go poza ochroną prawną. Pomimo przed-

²³ Por. Irlik, *Umowy...*, s. 2.

²⁴ Podobna myśl przyświecała zapewne twórcom ustaw autorskich wyłączających spod ochrony „języki programowania” (por. hiszpańskie prawo autorskie z 1987 r., południowokoreański *Computer Program Protection Act* z 1986 r.). Języki, czyli słowniki dozwolonych słów, reguły składni i semantyki, a nie programy tłumaczące z języka wysokiego poziomu na język maszynowy, potocznie a niepoprawnie nazywane „językami programowania”.

stawionych wątpliwości, możliwa jest do zastosowania ochrona zapewniana przez prawo autorskie. Problematyka jej zakresu w odniesieniu do algorytmów i programów będących ich zapisem musi stać się przedmiotem odrębnych rozważań.

Katedra Prawa Gospodarczego i Handlowego
Uniwersytetu Łódzkiego

Rafał Kastorski

SUBJECT OF LEGAL PROTECTION OF COMPUTER SOFTWARE

The need for and means of legal protection of computer software has been under discussion for years. Despite theoretical doubts adjudication and then jurisdiction of many countries have been applying the system of universal protection provided by the copyright to protect software. However, in the studies so far published the problem of mutual relationship between algorithm and computer program has not been sufficiently investigated.

The writer of this paper distinguishes two conceptions of solving the presented problem.

The first conception assumes that an algorithm is a general method of solving a particular type of generally formulated problem, whereas a program is development and adaptation of an algorithm to the needs of a definite thoroughly described problem. In this formulation algorithm is an abstract idea of solving a certain class of problems, being a certain idea on which a program is based.

The other conception admits that an algorithm is a detailed instruction at the level of elementary operation, covering a description of objects together with a description of activities to be performed on the objects to achieve a presumed goal. In this formulation the program is one of the forms of notation of algorithm. Elementary operation is such an operation whose way of performing is obvious to the performer. Thus in this formulation the level of minuteness of detail of the algorithm is determined by a list of operations and class of objects, not requiring further specification for a particular addressee. An algorithm for a computer must then refer to a set of elementary operations of the machine or a set of such operations which the computer can perform basing upon the algorithms adopted before (e.g. translators or compilers of programming languages). In this formulation the program is one of the ways of notation of algorithm, one of the forms of its transcription, proper for mechanical interpretation and performing by computer.

In the opinion of the writer, if we mean an algorithm for computer, the second conception should be admitted, which is more accurate from the semantic point of view, better expressing the actual relationship between algorithm and program, basing upon vocabulary and rather popular definitions, thus also clear for non-jurists.

Accepting that conception leads to a need for covering with legal protection not only the program, but first of all the algorithm.