HUMANISTYKA I PRZYRODOZNAWSTWO 25 | Olsztyn 2019 | doi: 10.31648/hip.5040

Jarosław Strzelecki
ORCID 0000-0002-1308-0746

Uniwersytet Warmińsko-Mazurski w Olsztynie
Instytut Filozofii

University of Warmia and Mazury in Olsztyn
Institute of Philosophy

MODELOWANIE W RAMACH FILOZOFII *IN SILICO* NA PODSTAWIE WYBRANYCH ELEMENTÓW Z *ETYKI* SPINOZY

Modelling within the philosophy *in silico* on the basis of selected elements from Spinoza's *Ethics*

Słowa kluczowe: filozofia, logika, Prolog, *in silico*, metafizyka

Key words: philosophy, logic, Prolog, *in silico*, metaphisics

Streszczenie

W artykule przedstawiono koncepcję filozofii *in silico* wraz z implementacją tej idei do fragmentów *Etyki* Barucha Spinozy. Zbudowano model deklaratywny w języku Prolog. Sprawdzono jego zgodność z danym fragmentem metafizyki Spinozy. Zaproponowano dalszy rozwój deklaratywnego modelowania w filozofii w kierunku dydaktycznym lub teoretycznym.

Abstract

The article presents a concept of philosophy *in silico* which is subsequently applied to fragments of the *Ethics* of Baruch Spinoza. A declarative model was built in the Prolog language. The compatibility of the model with a given fragment of Spinoza's metaphysics was checked. Further development of declarative modelling in philosophy in didactic or theoretical direction is proposed.

W XX wieku ogromny wpływ na filozofię wywarła logika formalna wraz z semiotyką, co zaowocowało powstaniem ważnego nurtu nazwanego filozofią analityczną. Być może w XXI wieku warto byłoby spróbować podobnego zabiegu na filozofii. Tym razem z zastosowaniem teorii i narzędzi informatycznych. Niniejszy artykuł jest właśnie taką próbą. Składa się z dwóch części. W pierwszej przedstawiono ideę filozofii *in silico*. W drugiej podjęto się budowy deklaratywnego modelu początkowych tez metafizycznych zawartych w *Etyce* Barucha Spinozy.

1. Nauka *in silico*

Współcześnie wyróżnia się w naukach biologicznych trzy kategorie eksperymentów: badania *in vitro*, badania *in vivo*, badania *in silico*. Każdy z nich ma odmienną charakterystykę (Marshall 2019).

Wyrażenie „in vitro" pochodzi z języka łacińskiego i znaczy tyle, co „w szkle". W naukach biologicznych oznacza metodę polegającą na przeprowadzaniu badania na zewnątrz żywego organizmu w kontrolowanym środowisku. Wiele eksperymentów w biologii komórkowej wykonuje się poza organizmami albo poza komórkami. Ponieważ eksperymenty są przeprowadzane poza organizmami, metoda ta uniemożliwia dokładne odtworzenie warunków występujących w żywych organizmach. Dlatego też wyniki uzyskane tą metodą mogą nie odpowiadać realnej sytuacji organizmu.

Odmienny sposób przeprowadzania badania postuluje technika *in vivo*. Sam termin również pochodzi z języka łacińskiego i dosłownie znaczy „w żywym". Metoda ta polega na badaniu całego żywego organizmu w przeciwieństwie do badania jego części albo organizmu martwego. Metoda *in vivo* stosowana jest głównie w badaniach klinicznych albo w badaniach na zwierzętach. Mimo pewnej – w niektórych przypadkach – przewagi tej metody nad *in vitro*, nie jest ona pozbawiona możliwości uzyskania błędnych wniosków, szczególnie w sytuacji dużej rozpiętości czasowej (np. terapia przynosi pozytywne skutki w krótkim czasie, ale po upływie dłuższego okazuje się mieć negatywny wpływ na stan zdrowia chorego).

Nowym sposobem przeprowadzania eksperymentów jest trzecia z przedstawianych – metoda zwana *in silico*. Etymologicznie wyrażenie to znaczy „w krzemie". Technicznie oznacza badanie wykonywane na komputerze lub za pomocą symulacji komputerowej. Choć z perspektywy filozoficznej idea eksperymentów wirtualnych wzbudza kontrowersje (Winsberg 2015), to jednak zdobywa coraz większą popularność przede wszystkim w naukach biologicznych (Marshall 2019).

Sam termin „in silico" został po raz pierwszy użyty publicznie w 1989 roku podczas warsztatów *Cellular Automata: Theory and Applications* w Los Alamos w Nowym Meksyku (Ibidem). Terminem tym Pedro Miramontes nazwał te biologiczne eksperymenty, które w całości zostały przeprowadzone w komputerze.

Dzisiaj stworzono wiele różnych technik eksperymentów *in silico*. Do ważniejszych z nich należą:

– sekwencjonowanie DNA i RNA bakterii,

– modelowanie molekularne (np. wykazanie interakcji między pewnymi lekami a receptorami jądrowymi komórek),

– symulacja całych komórek (np. zbudowano model komórki reagującej na cukier).

Badania metodą *in vivo* sprowadzają się do eksperymentów na żywych organizmach, *in vitro* do badań pozaustrojowych, *in silico* zaś do badań w wirtualnej rzeczywistości.

2. Filozofia *in silico*

Ekstrapolacja metod z biologii na filozoficzne dociekania wydaje się niemożliwa. Biolog raczej zadaje pytania typu „jak". Natomiast filozof stara się znaleźć odpowiedzi na najbardziej fundamentalne pytania o rację, czyli stawia pytania typu „dlaczego".

W filozofii informacji zaleca się stosowanie informatycznych schematów pojęciowych (Floridi 2002, 2012). Skoro metoda *in silico* w schemacie teoria-matematyka-doświadczenie/rzeczywistość_realna zamienia ostatni człon na rzeczywistość wirtualną, to w przypadku filozofii znaczyłoby, że przedmiotem filozoficznego poznania mają być szeroko pojęte obiekty wirtualne.

Analizując współczesne badania, można wyróżnić przykładowo cztery ogólne obszary zagadnień należących do filozofii *in silico*:

1) filozoficzne badania przedmiotów wirtualnych,

2) symulacja przedmiotów opisywanych przez twierdzenia filozoficzne,

3) reinterpretacja tez filozoficznych przy użyciu kategorii z paradygmatu obiektowego,

4) deklaratywne modelowanie przedmiotów opisywanych przez twierdzenia filozoficzne.

Pierwszy punkt w zasadzie nie wymaga komentarza. Filozofię *in silico* uprawia ten, kto przedmiotem swoich analiz uczynił obiekty wirtualne. Rozprawy o takiej tematyce już powstają (Bondecka-Krzykowska 2016; Przegalińska 2014; Stacewicz 2019).

W drugim punkcie chodziłoby o budowanie komputerowych symulacji rzeczywistości opisywanych i wyjaśnianych w ramach danej teorii filozoficznej (np. symulacje różnych teorii bytu). Podejmowane są pierwsze próby wizualizacji filozoficznych twierdzeń (Jernajczyk

2016a, 2016b), jak również bardziej formalno-informatyczne symulacje (Błądek, Komosinski, Miazga 2019).

Trzeci punkt postuluje reinterpretację koncepcji filozoficznych przy użyciu aparatu pojęciowego z obiektowego paradygmatu programowania. Prowadziłoby to do powstania filozofii obiektowej (*object-oriented philosophy*, OOPh) przez analogię do programowania obiektowego (*object-oriented programming*, OOP). Prace na temat relacji między filozofią a OOP już powstają (Janusz 2002; Rayside, Campbell 2000; Rayside, Kontogiannis 2001). Można również spotkać pierwsze próby obiektowej reinterpretacji filozoficznych koncepcji (Strzelecki 2017).

Co do ostatniego punktu, należy stwierdzić, że być może niniejsza praca jest jedną z pierwszych, które realizują czwarty postulat filozofii *in silico* (odmienne podejście do programowania w logice znaleźć można w: Garbacz 2016; Alama, Oppenheimer, Zalta 2015; Fitelson, Zalta 2007).

Są przynajmniej dwa sposoby na rozwinięcie tego pomysłu: programowanie w logice oraz programowanie funkcyjne. Ten artykuł poświęcony jest pierwszemu zagadnieniu. Używając języka Prolog, można wirtualizować filozoficzne problemy jako systemy dedukcyjne z rozumowaniem przez rezolucję (Bramer 2013; Clocksin, Mellish 2003; Fulmański 2009; Niederliński 2014). Co do drugiej drogi, wydaje się, że bardzo użytecznym – ale w inny sposób niż Prolog – językiem programowania może być Haskell (Lipovača 2017; O'Sullivan, Goerzen, Stewart 2008). Uniwersum Haskella zawiera czyste funkcje i dane. Haskell umożliwia definiowanie własnych typów. W ten sposób dowolny obiekt filozoficzny może być wirtualizowany jako specyficzny dla Haskella typ danych lub instancja takiego typu.

Ogólnie rzecz ujmując, filozofia *in silico* ma być filozofią uprawianą komputerowo. Tak pojęta jest częścią ogólnego projektu filozofii informacji autorstwa Luciano Floridiego (2002, 2012). Nie trudno też zauważyć, że wpisuje się w pewną zauważalną tendencję szeroko pojętej informatyzacji filozofii (Marciszewski, Stacewicz 2011; Mycka, Stacewicz 2015).

3. Deklaratywny paradygmat programowania

Przez „paradygmat" w świecie informatycznym rozumie się utrwalony sposób myślenia, w którym występują pewne założenia; one zaś nie podlegają krytycznej refleksji (Moczurad, Moczurad 2006; Sebesta 2016).

I tak np. w systemach UNIX uznano, że dobrze zbudowany program powinien składać się z niewielkich modułów, które użytkownicy mogą łączyć w działające aplikacje (Pfaffenberg 1999: 213).

Paradygmat programowania jest pewnym sposobem myślenia. To, o czym się myśli, jest programem komputerowym (rozumianym zarówno jako wytwór, czyli konkretny kod, jak i czynność, czyli sposób pisania kodu). Pewien wzorzec myślenia o programie komputerowym staje się paradygmatem (podobnie jak w przypadku innych nauk), gdy jest ceniony przez znaczącą grupę informatyków (Płoski 1999: 272).

Można wyróżnić przynajmniej trzy podstawowe paradygmaty programowania: imperatywny, deklaratywny, obiektowy. W tym artykule zajmiemy się wyłącznie paradygmatem deklaratywnym.

W paradygmacie deklaratywnym – jak sama nazwa wskazuje – program jest ciągiem opisów, swoistej deklaracji tego, co wie programista (Triska 2020). Program nie jest ciągiem instrukcji definiujących sposób działania (jak ma to miejsce w paradygmacie imperatywnym). Nie jest też systemem obiektów współdziałających ze sobą (jak ma to miejsce w paradygmacie obiektowym). Program jest opisem tego, co wiadome.

Do języków programowania w omawianym paradygmacie zalicza się języki funkcyjne (np. LISP, Haskell), języki bazodanowe (np. SQL) oraz języki logiczne (np. Prolog). W językach logicznych podstawowymi konceptami są: fakty, reguły, relacje, zapytania. W pierwszych opisuje się pojedyncze zależności/relacje (np. to, że taki to a taki przedmiot ma określoną cechę, bądź pozostaje w danej relacji z innym indywiduum). Reguły pozwalają na definiowanie ogólnych związków. Do opisu faktów i reguł wykorzystuje się odpowiednio dostosowaną logikę symboliczną. Odpowiedzi na zadane przez użytkownika pytania uzyskuje się dzięki mechanizmom wynikania logicznego zaimplementowanym w danym języku (Sebesta 2016: 728).

Zdaniem w sensie logicznym nazywa się zdanie orzekające, które może być prawdziwe lub fałszywe. Zdanie składa się z nazw oznaczających obiekty i wyrażeń oznaczających relacje między tymi obiektami. W językach logicznych tymi nazwami mogą być stałe (oznaczają dany obiekt) albo zmienne (reprezentują różne obiekty w różnych sytuacjach).

Zdanie proste – zwane atomowym – składa się z funktora i jego parametrów. Funktor wskazuje na relacje. Parametry oznaczają przedmioty, między którymi zachodzi relacja, na którą wskazuje funktor. I tak przykładowo zdanie proste napisane w języku Prolog „człowie-

k(jan)" może mieć sens taki, jak zdanie z języka naturalnego „Jan jest człowiekiem", a „lubi(jan, genowefa)" może oznaczać relację lubienia zachodzącą między obiektem Jan i obiektem Genowefa.

Zdanie złożone składa się z przynajmniej dwóch zdań prostych (np. „Jan jest człowiekiem i lubi Genowefę", w Prologu „czlowiek(jan), „lubi(jan, genowefa)". W zdaniu zmienne pojawiają się wówczas, gdy zostanie użyty jeden z kwantyfikatorów. Może to być kwantyfikator ogólny albo egzystencjalny. Logika predykatów pierwszego rzędu, w której występują wspomniane kwantyfikatory, jest podstawą języków programowania w logice (Ibidem: 731).

4. Prolog

Prolog jest komputerowym językiem programowania. Służy do rozwiązywania problemów związanych z obiektami i relacjami między nimi (Clocksin, Mellish 2003: 2). Prolog bazuje na informatycznych badaniach prowadzonych w Europie w latach 60. i 70. XX wieku – szczególnie na uniwersytetach w Marsylii, Londynie i Edynburgu (Bramer 2013: 9). Został stworzony przez Alaina Colmerauera i Philippe'a Roussela w 1971 roku (Fulmański 2009: 1).

Prolog jest bardzo przydatnym narzędziem do opisywania wiedzy na temat problemu, który jest rozważany. Opis jest modelem problemu (Niederliński 2014: 13–14) i składa się z klauzul Horna (fakty i reguły). W Prologowym kompilatorze znajduje się system inferencyjny, który bazuje na logicznej rezolucji (Clocksin, Mellish 2003: rozdz. 10). Użytkownik zadaje pytania, a Prologowy system inferencyjny, używając bazy wiedzy, poszukuje odpowiedzi (np. przeprowadza dowód).

Fakty mają następującą strukturę: *predykat(argument, ...)*. Na przykład zdanie „Sokrates jest inteligentny" w notacji Prologu ma następujący wygląd *inteligentny(sokrates)*. Zakłada się, że każdy fakt jest spełniony w modelu.

Reguły są zdaniami warunkowymi ze znakiem implikacji „:-". Ich struktura jest następująca: *wniosek :- warunek1, ...* Na przykład zdanie: „Jeżeli ktoś jest podobny do Sokratesa, to jest inteligentny", w notacji Prologu przedstawia się następująco: *inteligentny(Ktos) :- podobny(Ktos, sokrates)*. Jeżeli warunek *podobny(Ktos, sokrates)* jest prawdziwy, to wniosek *inteligentny(Ktos)* również jest prawdziwy.

Obok znaku implikacji „:-" występują również znaki koniunkcji „,", alternatywy „;" oraz negacji „\+".

Użytkownik kontaktuje się z programem przez zadawanie pytań. Są to pytania rozstrzygnięcia (np. „Czy Jaś odrobił lekcje?" odpowiednik Prologowy „?- odrobil(jas, lekcje).") lub dopełnienia (np. „Co Zosia jadła na kolację?" odpowiednik Prologowy „?- jadla(zosia, kolacja, Cos).").

5. *Etyka* Spinozy w Prologu – ustalenia wstępne

Filozoficzny model *in silico* został zbudowany na podstawie czterech tłumaczeń *Etyki*. Są to dwa tłumaczenia polskie (Spinoza 1888; Spinoza 1927), jedno francuskie (Spinoza 1993), jedno angielskie (Spinoza 2009).

Powstały już prace, w których podjęto się logicznej analizy *Etyki* Spinozy (Jarrett 1978) lub formalizacji wybranych fragmentów tego dzieła przy użyciu logiki predykatów pierwszego rzędu (Blum, Malinovich 1993), czy też zastosowania systemu *Prover9* do pierwszych 11 twierdzeń z pierwszej części pracy Spinozy (Janowicz i in. 2017). Jednak autor tego artykułu zdecydował się na powrót do samych źródeł myśli Spinozy. Dlatego też pozwolił sobie na niezapośredniczanie Prologowego modelu fragmentu *Etyki* przez dotychczas wykonane formalizacje. Rzecz w tym, że język programowania Prolog, choć jest ufundowany na języku predykatów pierwszego rzędu, to ze względu na swoją specyfikę (fakty, reguły, zapytania, predykaty wbudowane, zmienność bazy wiedzy w trakcie przeprowadzania dowodów) nie jest prostym, komputerowym odwzorowaniem logiki pierwszego rzędu i samo przeprowadzanie dowodów nie jest istotą tego języka. Oczywiście porównanie formalizacji czysto predykatowych, jak i tych dostosowanych do systemu *Prover9* z odpowiednimi formalizacjami z Prologu mogłoby być treścią bardzo interesującego artykułu, a nawet książki.

W języku Prolog nie występują polskie znaki, więc wszelkie formuły napisane w tym języku są ich pozbawione. Choć może to wydawać się dla polskiego czytelnika irytujące, specyfika języka programowania, w którym stworzono model, wymusza właśnie taki zapis.

Model *Etyki* został napisany w języku Prolog w wersji SWI Prolog. Wszelkie informacje na temat tej wersji języka deklaratywnego znajdzie czytelnik na stronie projektu: https://www.swi-prolog.org.

Wszystkie wyrażenia napisane takim `fontem` będą oznaczały formułę z języka Prolog.

Spinoza często posługuje się terminem „pojmowanie". Przyjęto założenie, że na potrzeby wirtualnego modelu można to pojęcie zdefiniować jako uzyskiwanie pozytywnej odpowiedzi na zadane pytanie do bazy wiedzy, która jest zbiorem klauzul napisanych w języku Prolog. I tak np., gdyby w bazie wiedzy znajdował się fakt `czlowiek(sokrates)`, to na pytanie `filozof(sokrates)` (Czy Sokrates jest filozofem?), przy założeniu, że w bazie wiedzy nie ma ani reguły, ani faktu wskazującego na to, że Sokrates jest filozofem, Prolog udzieli negatywnej odpowiedzi. W przyjętej interpretacji będzie to oznaczało, że Prolog nie ma pojęcia o tym, że Sokrates jest filozofem.

Na koniec tych uwag należałoby wspomnieć o metodzie abstrakcji, która jest jednym z fundamentów programowania w ogóle, zatem również występuje w programowaniu deklaratywnym. Abstrakcja polega na pomijaniu cech nieistotnych z danej perspektywy epistemologicznej, przez co uwypukla się te własności badanego przedmiotu, które są ważne z przyjętego punktu widzenia (Floridi 2012: 46–79). I tak w Prologu można byłoby podać fakt *czlowiek(platon*), który byłby odwzorowaniem faktu ze świata rzeczywistego, że Platon jest człowiekiem. Jednak na pytanie *atenczyk(platon)* („Czy Platon jest ateńczykiem?") otrzyma się negatywną odpowiedź (przy założeniu, że taki fakt nie występuje w bazie wiedzy). Choć zdanie „Platon jest ateńczykiem" jest prawdziwe w świecie rzeczywistym, to w świecie wirtualnym, a dokładnie w określonej bazie wiedzy, jest fałszywe. Jednakże można abstrahować od przynależności Platona do Aten, o ile przyjęty w badaniach poziom abstrakcji nie uwzględnia tej własności jako istotnej. Podobnie jest z budowanym modelem fragmentu *Etyki* Spinozy. Zyskuje on na precyzji formalnej, jednocześnie traci na wielości znaczeń zawartych w dziele Spinozy.

6. Prologowy model *Etyki* Spinozy

Najpierw przedstawione i przeanalizowane zostaną deklaratywne odpowiedniki definicji zawartych na początku *Etyki* Spinozy. Choć polscy tłumacze zdecydowali się użyć słowa „określniki" na oznaczenie tych pojęć, to autor tego artykułu odwołuje się do terminu występującego w tłumaczeniu zarówno angielskim, jak i francuskim. Nazwa

„definicje" lepiej oddaje sens zawarty w tych pierwszych punktach Spinozjańskiej *Etyki*. Następnie zaprezentowane zostaną Prologowe odpowiedniki niektórych Spinozjańskich pewników/aksjomatów. Na koniec do budowanego modelu zostaną dołączone dwa pierwsze twierdzenia.

6.1. Definicje

Zostanie przełożonych na język Prologu pięć pierwszych definicji. Będą to: „causa sui", być ograniczonym w swoim rodzaju, substancja, atrybut, modus.

Definicja pierwsza – pojęcie „causa sui".

„Przez «przyczynę samej siebie» rozumiem to, czego treść zawiera w sobie istnienie, czyli to, czego naturę można pojąć tylko jako istniejącą" (Spinoza 1927). Dany byt nazywa się przyczyną samej siebie, o ile jest to byt, którego treść zawiera w sobie istnienie. W języku francuskim występuje tu wyrażenie *l'essence enveloppe l'existence* (Spinoza 1993). Czasownik *envelopper* wskazuje na czynność owijania, zawijania, otulania, okrążania, wmieszania, zapakowania. Rzeczownik *essence* oznacza istotę bytu, czyli to w bycie, co czyni go właśnie tym bytem. W tłumaczeniu angielskim występuje czasownik *involve* – angażować, obejmować, żądać, potrzebować, wplątywać (Spinoza 2009). Można by zatem – nieco metaforycznie – powiedzieć, że byt, którego istota „żąda" istnienia, jest przyczyną siebie samego.

Przechodząc do modelowania w Prologu, pierwszą definicję można by sformalizować w następujący sposób:

```
causa_sui(X):-egzystuje(X).
egzystuje(_).
```

O pewnym bycie *X* powiemy, że jest *causa sui*, o ile jest on bytem, który egzystuje – taki jest sens pierwszej reguły. Użyto słowa „egzystuje" zamiast „istniejący", aby z jednej strony być syntaktycznie bliżej łacińskiemu pierwowzorowi, który w języku angielskim przyjął formę *existent*, a we francuskim *existante*, a z drugiej, aby uwypuklić specyfikę istnienia tak pojętego bytu, którego istota zawiera istnienie.

Jednak sama reguła nie wystarczy, ponieważ Prolog nie ma definicji predykatu „egzystuje", czyli Prolog nie pojmuje – w rozumieniu

wcześniej przyjętej konwencji dla tego terminu – czym jest byt egzystujący, a bez tego pojęcia, predykat „causa sui" również nie jest pojmowany. Dlatego też wprowadzono pewien fakt do bazy wiedzy – `egzystuje(_)`. Należy go rozumieć jako odpowiednik twierdzenia, że istnieje coś (symbol „_" oznacza cokolwiek), co egzystuje. Dzięki takiemu zabiegowi na pytanie, czy istnieje coś, co byłoby swoją własną przyczyną, otrzymujemy pozytywną odpowiedź:

```
?- causa_sui(X).
true.
```

Warto zaznaczyć, że wprowadzony zabieg narusza Spinozjańską definicję, ponieważ nie tylko definiuje termin „causa sui", ale również wprowadza twierdzenie, że w przyjętym uniwersum istnieje coś, co jest swoją własną przyczyną. Zamiast dopisanego faktu można byłoby wprowadzić dodatkową definicję:

```
egzystuje(X):-causa_sui(Y).
```

O danym bycie można orzec, że egzystuje, o ile jest on *causa sui*. Jednak takie rozwiązanie prowadzi do nieskończonej rekurencji. Prolog bez końca będzie próbował znaleźć w bazie wiedzy taki byt, który jest *causa sui*, o ile egzystuje, oraz egzystuje, o ile jest *causa sui*. Rekurencję można przerwać, wprowadzając fakt `egzystuje(_)`. W ten sposób pojawia się epistemologiczny problem podstaw wiedzy, o których już rozprawiali Arystoteles i Platon, a który daleko wykracza poza ramy tego artykułu.

Definicja druga – pojęcie „być ograniczonym w swoim rodzaju"

„Taka rzecz nazywa się w «swoim rodzaju skończoną», którą może ograniczać inna rzecz tej samej natury. Tak np. ciało nazywa się skończonym, ponieważ pojmujemy inne, zawsze większe. Podobnie jednak myśl ogranicza drugą. Jednakże ciało nie podlega ograniczaniu przez myśl, ani myśl przez ciało" (Spinoza 1927).

Sformułowana definicja, aby być względnie precyzyjną, wymagałaby dodatkowych wyjaśnień dotyczących pojęć „ograniczanie" (ang. *be limited*; fr. être *limité*) oraz „natura" (ang. i fr. *nature*). Podobnie rzecz się ma z podanym przykładem. Chcąc go poprawnie zrozumieć, nale-

żałoby zrekonstruować Spinozjańską teorię ciała oraz Spinozjańską teorię myśli. Wskazane trudności można ominąć przy użyciu metody abstrakcji, o której już wspomniano:

```
skonczony_w_swoim_rodzaju(X):-
ta_sama_natura(X,Y),
ogranicza(Y, X).
```

O danym bycie *X* można powiedzieć, że jest skończony w swoim rodzaju, o ile istnieje pewien byt *Y*, który jest tej samej natury co *X*, oraz jest tak, że *Y* ogranicza *X* – w ten sposób można odczytać powyższą regułę.

Jeżeli predykaty „ta_sama_natura" oraz „ogranicza" nie zostałyby zdefiniowane, Prolog na pytanie związane z predykatem „skonczony_w_swoim_rodzaju" zgłosi błąd, ponieważ „ta_sama_natura" oraz „ogranicza" nie zostały określone w bazie wiedzy.

Trudność tę można rozwiązać na trzy sposoby. Pierwszy z nich wymagałby rekonstrukcji teorii natury w filozofii Spinozy, a następnie zbudowania deklaratywnego modelu tej teorii. To podejście daleko wykracza poza jeden artykuł.

Drugi z nich polegałby na podaniu faktów, w których wspomniane predykaty występowałyby, np.:

```
ta_sama_natura(cialo_Klepatry, palac).
ograniczna(palac, cialo_Klepatry).
```

Wówczas na pytanie, czy ciało Kleopatry jest bytem ograniczonym w swoim rodzaju, uzyskano by pozytywną odpowiedź.

Trzeci sposób byłby pośredni. Predykat „ogranicza" byłby dookreślany przez podanie faktów. Natomiast predykat „ta_sama_natura" miałby swoją własną definicję. To podejście prowadzi do nowego problemu. Prolog w zasadzie jest komputerową implementacją rachunku predykatów pierwszego rzędu. Predykat „ta_sama_natura" wymagałby operacji na innych predykatach, co w rachunku predykatów pierwszego rzędu jest niewykonalne. Jednak, jak już zostało to napisane, Prolog to nie tylko prosta implementacja wspomnianego rachunku logicznego. Występują w nim wbudowane operatory, które umożliwiają operowanie na samych predykatach (np. „asserta", „assertz", „retract" odpowiadają za zmianę zawartości bazy wiedzy; „funktor", „arg", „=.." umożliwiają bezpośrednie tworzenie i zmianę struktury predykatów).

Definicja trzecia – pojęcie „substancja”

„Przez Substancyę rozumiem to, co istnieje samo w sobie i pojmowanem bywa przez siebie; czyli to, co dla swego pojęcia nie potrzebuje pojęcia innej rzeczy, z którego by się wytworzyć musiało” (Spinoza 1888). Terminem „substancja” Spinoza oznacza ten przedmiot, który spełnia dwa warunki: istnieje sam w sobie (ang. *is in itself* (Spinoza 2009), fr. *est en soi* (Spinoza 1993)) oraz jest pojmowany przez siebie (ang. *is conceived through itself* (Spinoza 2009), fr. *est conçu par soi* (Spinoza 1993)).

Najprostsza formalizacja polegałaby na dosłownym odwzorowaniu definicji Spinozy:

```
substancja(X):-istnieje_samo_w_sobie(X),przez_siebie_
jest_pojmowane(X).
```

Jednakże takie podejście wymuszałoby dookreślenie dwóch dodatkowych predykatów, co byłoby niezgodne z definicją, w której mowa o tym, że pojęcie substancji nie wymaga żadnych dodatkowych pojęć.

Lepszym rozwiązaniem byłoby uznanie, że predykat „istnieje_samo_w_sobie” jest swoistą „własnością” substancji, a sens predykatu „przez_siebie_jest_pojmowane” mógłby być oddany przez wpisanie do bazy wiedzy faktu, że coś jest substancją. Wówczas pojmowanie zostałoby ujęte w konwencji wprowadzonej na początku tej części artykułu. Uwzględniając te uwagi, Prologowa definicja substancji przedstawiałaby się następująco:

```
istnieje_samo_w_sobie(Substancja):-substancja(Substancja).
substancja(_).
```

Podaną regułę w języku naturalnym można odczytać następująco: o pewnym bycie można powiedzieć, że istnieje sam w sobie, o ile jest substancją. W fakcie `substancja(_)` stwierdza się, że istnieje coś, co jest substancją.

Rozpatrując powyższą regułę czysto formalnie – a jest ona odwróconą implikacją (jeżeli coś jest substancją, to istnieje samo w sobie) – dopuszcza się możliwość, w której dany przedmiot istniałby sam w sobie, ale nie byłby substancją. Jednak w bazie wiedzy nie ma więcej ani reguł, ani faktów związanych z predykatem „istnieje_samo_w_sobie”. W ten sposób gwarantuje się wyłączność związku tego predykatu z predykatem „substancja”.

Na pytanie, czy w zbudowanym modelu istnieje byt, o którym można powiedzieć, że istnieje sam w sobie, otrzyma się pozytywną odpowiedź:

```
?- istnieje_samo_w_sobie(X).
true.
```

Warto zauważyć, że otwartą kwestią pozostaje związek predykatu „substancja" z predykatem „egzystuje".

Definicja czwarta – pojęcie „atrybut"

„Przez atrybut, czyli przymiot rozumiem to, co umysł pojmuje jako stanowiące istotę substancyi" (Spinoza 1888). Atrybutem jest przedmiot, który stanowi istotę substancji, a związek z substancją jest taki, że gdy umysł postrzega atrybut, od razu wie, że ów przedmiot jest atrybutem substancji:

```
atrybut(X) :- substancja(Y), konstytuuje_istote(X, Y).
```

W języku naturalnym regułę tę można odczytać następująco: o danym bycie *X* powiemy, że jest atrybutem, o ile istnieje substancja *Y*, oraz jest tak, że *X* konstytuuje istotę tej substancji. Predykat „substancja" już został zdefiniowany. Niestety predykat „konstytuuje_istote" jeszcze nie występuje w bazie wiedzy ani jako fakt, ani jako reguła. Rekonstrukcja i formalizacja Spinozjańskiej teorii konstytucji istoty, o ile w ogóle jest to możliwe, daleko wykracza poza niniejszy artykuł. Dlatego na potrzeby testów poprawności modelu można wprowadzić fakt `konstytuuje_istote(mysl,_)`, czyli myśl konstytuuje istotę pewnego przedmiotu:

```
?- atrybut(mysl).
true.
```

Myśl jest atrybutem w modelu, ponieważ istnieje pewien przedmiot, który jest substancją, oraz myśl konstytuuje istotę tego przedmiotu.

Definicja piąta – pojęcie „modus"

„Przez objaw rozumiem poruszenia substancyi, czyli to, co istnieje w czemś innem, za pomocą czego też zostaje pojmowanem" (Spinoza 1888). W tłumaczeniach angielskim i francuskim termin „objaw" oddawany jest przy użyciu nazwy *mode* („moda", „sposób", „tryb"). Modusem – inaczej sposobem działania – substancji są jej poruszenia (ang. *Modifi-*

cations (Spinoza 2009), fr. *affections* (Spinoza 1993), staropolskie „afekcje" – skłonności ku czemuś, komuś). Modus substancji charakteryzuje się tym, że istnieje w czymś innym, czyli nie jest prawdą, że istnieje sam w sobie oraz jest pojmowany przez coś innego:

```
modus(X) :- substancja(Y),
            afekcja(X, Y),
            \+ substancja(X).
```

Regułę tę można odczytać następująco: byt *X* jest modusem, o ile istnieje *Y*, który jest substancją, a *X* jest afekcją *Y* oraz *X* nie jest substancją (znak „\+" oznacza negację).

Przyjmijmy – na potrzeby weryfikacji modelu – fakt `afekcja(pragnienie,_)`, czyli że pragnienie jest afekcją jakiegoś bytu. Na pytanie o to, czy pragnienie jest modusem, otrzymamy pozytywną odpowiedź, więc predykat „modus" jest pojmowany przez predykaty „substancja", „\+", „afekcja":

```
?- modus(pragnienie).
true.
?- modus(_).
false.
```

Drugie z pytań dotyczy tego, czy istnieje coś, co jest modusem. Interpreter Prologu generuje odpowiedź „false", czyli w przyjętym modelu nie istnieje modus. W oczywisty sposób taki wniosek nie jest zgodny z wprowadzonym faktem, że pragnienie jest afekcją, a więc również modusem. Trudność wynika ze specyfiki języka Prolog i jego zmiennej nieokreślonej „_". Co gorsza, na pytanie, czy pragnienie jest substancją, Prolog odpowiada, że tak. Wszystkie wymienione trudności znikną, gdy w fakcie `substancja(_)` zamiast zmiennej nieokreślonej zostanie użyta stała – np. „egzystujacy" `(substancja(egzystujacy))`.

6.2. Pewniki

W tej części artykułu przedstawione zostaną trzy pewniki: o istnieniu w czymś albo w sobie, o pojmowaniu, o przyczynowości.

Pewnik pierwszy – być w sobie albo być w czymś innym

„Wszystko, co jest, jest bądź samo w sobie, bądź w czymś innym" (Spinoza 1927). Przełożenie tego pewnika na język deklaratywny nie

nastręcza trudności, skoro predykat „istnieć samemu w sobie" już został zdefiniowany:

```
przedmiot(X) :-
  (\+(istnieje_samo_w_sobie(X)), !);
  istnieje_samo_w_sobie(X).
```

Predykat „przedmiot" oznacza cokolwiek. O *X* powiemy, że jest przedmiotem, o ile nie jest prawdą, że istnieje sam w sobie albo istnieje sam w sobie (znak „!" to tzw. znak odcięcia, który sprawia, że Prolog – po spełnieniu celu – przestaje przeszukiwać bazę wiedzy, aby znaleźć alternatywne rozwiązania).

Po wprowadzeniu do bazy wiedzy faktu `dziecko(janek)`, otrzymuje się następujące odpowiedzi na zadane pytania:

```
?- przedmiot(dziecko(janek)).
true.
[debug]  ?- substancja(dziecko(janek)).
false.
```

W skonstruowanym modelu Janek jest dzieckiem, więc jest przedmiotem. Jednak nie jest substancją. Takie twierdzenie jest w pełni zgodne z metafizyką Spinozy.

Pewnik drugi – pojmowanie

„To, co się nie da pojąć przez coś innego, musi być pojmowanem samo przez się" (Spinoza 1888). W przeciwieństwie do pierwszego pewnika, który można nazwać ontologicznym, ten pewnik należy do sądów epistemologicznych. Zgodnie z przyjętą w tym artykule konwencją dotyczącą rozumienia terminu „pojmowanie" (ang. *conceive* – pojąć, wyobrażać sobie, rozumieć rozumieć (Spinoza 2009); fr. *concevoir* – postrzegać, zrozumieć, pojąć, czuć (Spinoza 1993)), pewnik ten nie musi być definiowany *explicite* w samym programie. Predykat, który nie jest określany przez inne predykaty, musi występować w programie jako fakt. Wówczas jest pojmowany sam przez się. Wszystkie pozostałe predykaty są pojmowane przez ich relacje do innych predykatów.

Pewnik trzeci – przyczynowość

„Z przyczyny danej, określonej, niezbędnie wynika skutek, i odwrotnie: jeżeli nie jest dana żadna określona przyczyna, wtedy i skutek nastąpić nie może" (Spinoza 1927). Ten pewnik należy do kategorii ontologicznych. Wyrażenie „niezbędnie wynika" w języku angielskim oddane jest przez *necessarily follows* (*follow* – następować, wynikać, iść za) (Spinoza 2009), a we francuskim *suit nécessairement* (*suivre* – następować, iść za, podążać) (Spinoza 1993). Spinoza jest pewien, że po danej przyczynie koniecznie następuje skutek oraz, gdy brak przyczyny, skutek nie występuje.

Jak wcześniej zostało zauważone, bez twierdzeń dookreślających pojęcia występujące, czy to w definicji, czy w pewniku, nie sposób względnie precyzyjnie odwzorować teorię Spinozy w modelu deklaratywnym. W przypadku analizowanego pewnika brakującym elementem jest koncepcja przyczynowości. Z przytoczonego pewnika wiadomo jedynie, że między czymś, co nazywa się przyczyną, a tym, co nazywa się skutkiem, występuje swoista relacja. Gdy pojawia się przyczyna, skutek również się pojawia. Gdy nie pojawia się przyczyna, skutek wystąpić nie może.

Najprostszą Prologową reprezentacją tej relacji byłaby złożona struktura `przyczyna(X,Y)`, gdzie zmienna *X* reprezentowałaby przyczynę, a zmienna *Y* skutek. Jednak zaproponowana reguła nie wykluczałaby sytuacji, w której zmienne *X* i *Y* przyjęłyby te same wartości. Wówczas predykat „przyczyna" odpowiadałby predykatowi „causa sui". Stosując zasadę życzliwości, należałoby uznać, że Spinoza relację przyczynowości odróżniał od bycia przyczyną samej siebie, skoro informacje na ich temat umieścił w dwóch różnych miejscach swojej pracy. Uwzględniając tę uwagę, do modelu można byłoby wprowadzić regułę `przyczyna(X,Y):-X\=Y`, czyli *X* jest przyczyną *Y*, o ile są one różnymi przedmiotami. Dokładniej rzecz ujmując, relacja „\=" tworzy prawdziwą klauzulę, gdy termy występujące z lewej i prawej strony symbolu nie mogą być uzgodnione. W Prologu występuje relacja „\==", która tworzy prawdziwą klauzulę, gdy jej argumenty nie są identyczne. Jednak wprowadzenie relacji „\==" zamiast „\=" prowadzi do powstawania niezgodnych z filozofią Spinozy twierdzeń. Samo zagadnienie jest ciekawe i wymagałoby dokładniejszych analiz. Jednak nie należy ono do głównego tematu artykułu.

Język Prolog jest na tyle elastyczny, że umożliwia zawarcie w ramach jednego modelu sytuacji, w której nie odróżnia się przyczynowości typu *causa sui* od przyczynowości między dwoma różnymi przedmiota-

mi. Wystarczy do bazy wiedzy dopisać dodatkową regułę `przyczyna(X,X):-` causa_sui(X). Przedmiot *X* jest przyczyną siebie samego, o ile prawdą jest, że przedmiot *X* jest *causa sui*.

Maksymalne odwzorowanie pewnika trzeciego wymagałoby uwzględnienia pojęcia skutku. Można to zrobić przez wprowadzenie dodatkowej reguły dla predykatu „skutek":

```
skutek(X) :- przedmiot(Y), przyczyna(Y, X).
przyczyna(X,Y) :- X\=Y.
przyczyna(X,X):- causa_sui(X).
```

Przedmiot *X* jest skutkiem, o ile istnieje pewien przedmiot (substancja albo nie-substancja) *Y*, który jest przyczyną *X*.

Niestety podane reguły obarczone są podstawową wadą. W takim modelu dopuszczalna jest sytuacja skutku bez przyczyny. Przy założeniu, że do bazy wiedzy wprowadzono fakt `skutek(zniszczone_drzewo)`, na pytanie o przyczynę zniszczonego drzewa Prolog odpowiada „false":

```
?- przyczyna(X, zniszczone_drzewo).
false.
```

Taką odpowiedź można interpretować dwojako. W danym modelu nie istnieje przyczyna niszczenia drzewa, czyli została naruszona Spinozjańska zasada przyczynowości. Druga interpretacja otrzymanego wyniku jest epistemologiczna. W danym modelu nie jest znana przyczyna zniszczenia drzewa, czyli że zasada przyczynowości nie została naruszona. Wszystko zatem zależy od interpretacji, jaką nałoży się na wynik operacji obliczeniowych dokonanych przez interpreter języka Prolog.

6.3. Twierdzenia

Ze względu na ograniczenia, jakie narzuca forma artykułu, wybrano dwa pierwsze twierdzenia z *Etyki* Spinozy. Pierwsze z nich dotyczy bytowego prymatu substancji nad modusami. W drugim z kolei twierdzi się, że substancje, mające różne atrybuty, nie mają ze sobą nic wspólnego.

Twierdzenie pierwsze

„Substancya z natury swej istnieje przed swemi objawami" (Spinoza 1888). Dowodem tego twierdzenia mają być definicje trzecia i piąta. Substancja istnieje sama w sobie, a objaw/modus istnieje w czymś innym.

Jeżeli w bazie wiedzy nie występuje dany predykat, Prolog zwróci błąd. Dlatego należałoby zdefiniować relację „istnieć przed". Odpowiednikami polskiego słowa „przed" w tłumaczeniu francuskim jest *antérieure* (Spinoza 1993), a w angielskim *prior* (Spinoza 2009). Oba niepolskojęzyczne terminy wskazują na relację czasową, czyli w porządku czasowym najpierw istnieje substancja, a potem pojawia się objaw/modus.

Jednak taka czasowa interpretacja nie jest zgodna z definicjami, na które powołuje się Spinoza. Uwzględniając definicję trzecią, w której mowa o tym, że substancja istnieje w sobie, relację „istnieć przed" należałoby rozumieć ontologicznie lub epistemologicznie. Substancja istnieje przed modusem, czyli, aby mógł zaistnieć objaw, najpierw musi istnieć substancja (wymiar ontologiczny). Chcąc pojąć modus, wcześniej należy pojąć m.in. substancję, której pojęcie współtworzy pojęcie modusu (wymiar epistemologiczny). Kwestią otwartą pozostaje, czy w metafizyce Spinozy dopuszczalne jest istnienie bytów, które ontologicznie i epistemologicznie ufundowane są nie na substancji, ale na bytach, które nie istnieją same w sobie.

Reguła, która odwzorowywałaby twierdzenie pierwsze i jednocześnie ograniczała relację „istnieć przed" do substancji i afekcji, może mieć następującą postać:

```
istnieje_przed(X, Y) :-
   substancja(X),
   afekcja(Y, X).
```

Między dwoma dowolnymi bytami *X* i *Y* zachodzi relacja „istnieć przed", o ile pierwszy z tych bytów jest substancją, a drugi afekcją tej substancji.

Przy założeniu, że w bazie wiedzy występują fakty `substancja(egzystujacy)` i `afekcja(pragnienie, egzystujacy)`, na pytanie o byt, który istnieje przed pragnieniem, Prolog formułuje odpowiedź, że tym bytem jest `egzystujacy`:

```
?- istnieje_przed(X,pragnienie).
X = egzystujacy.
```

Choć bezpośrednio nie zdefiniowano relacji „istnieć przed" w stosunku do substancji, to Prolog, na pytanie o to, czy istnieje coś przed substancją, odpowiada zgodnie z metafizyką Spinozy, że taki byt nie istnieje:

```
?- istnieje_przed(X,egzystujacy).
false.
```

Twierdzenie drugie

„Dwie substancye, mające różne przymioty, nie mają ze sobą nic wspólnego” (Spinoza 1888). Dowodem tego twierdzenia – według Spinozy – jest definicja trzecia, na którą powoływał się w pierwszym twierdzeniu.

Termin „przymiot” jest – w rozpatrywanej metafizyce – synonimem terminu „atrybut”, zatem – wbrew deklaracji Spinozy – dowód tego twierdzenia wymaga nie tylko definicji trzeciej, ale również czwartej, w której mowa o pojęciu atrybutu.

Tak jak w poprzednich przypadkach, tak i tym razem należy wprowadzić dodatkowe pojęcia do bazy wiedzy, aby dowieść prawdziwości tego twierdzenia w deklaratywnym modelu:

```
brak_zwiazku(Substancja_1, Substancja_2) :-
   substancja(Substancja_1), substancja(Substancja_2),
   \+(ten_sam_atrybut(Substancja_1, Substancja_2)).
ten_sam_atrybut(Substancja_1, Substancja_2):-
   konstytuuje_istote(Atrybut_1, Substancja_1),
   konstytuuje_istote(Atrybut_1, Substancja_2).
```

Między dwoma bytami `(Substancja_1, Substancja_2)` nie zachodzi żaden związek, o ile są one substancjami, oraz nie istnieje nawet jeden wspólny atrybut.

Przy założeniu, że w bazie wiedzy występują następujące fakty: `substancja(sub1).substancja(sub2).konstytuuje_istote(myslenie,sub1).konstytuuje_istote(przestrzennosc, sub2)`, czyli istnieją dwie substancje o odmiennych atrybutach, Prolog na pytanie, czy między tymi dwoma substancjami nie ma żadnego związku, odpowie pozytywnie:

```
?- brak_zwiazku(sub1, sub2).
true.
```

Metafizyka Spinozy jest monistyczna, ponieważ istnieje tylko jedna substancja. Tymczasem takiego twierdzenia nie można wyprowadzić z bazy wiedzy, opierając się wyłącznie na definicjach, pewnikach i dotychczas udowodnionych twierdzeniach. Zamiana faktu:

```
konstytuuje_istote(przestrzennosc,sub2)
na
konstytuuje_istote(myslenie,sub2),
```

czyli uznanie, że istnieją dwie różne substancje, które mają przynajmniej jeden wspólny atrybut, nie prowadzi do sprzeczności w tym modelu:

```
?- brak_zwiazku(sub1, sub2).
false.
```

W dotychczas zbudowanym modelu dopuszczalna jest sytuacja, w której istnieją dwie substancje o przynajmniej jednym wspólnym atrybucie. Wniosek taki nie jest zgodny z monizmem metafizyki Spinozy. Oczywiście nie jest to zarzut w stosunku do tej teorii bytu, ponieważ dowód tego, że istnieje tylko jedna substancja, Spinoza przeprowadza w dalszych częściach swojego traktatu.

Podsumowanie

Rozwój modeli deklaratywnych może iść w dwóch kierunkach. Pierwszy byłby dydaktyczny. Zbudowanie pełnego modelu np. metafizyki Spinozy umożliwiłoby adeptom filozoficznego myślenia poznawanie tej teorii przez interaktywne zadawanie pytań do bazy wiedzy, w której zapisano by Spinozjańską teorię bytu.

Drugi kierunek byłby teoretyczny. Budowanie deklaratywnego modelu uwypukla niejasności terminologiczne wirtualizowanej teorii filozoficznej. Wskazuje na przesłanki entymematyczne. Umożliwia dostrzeżenie zależności między różnymi predykatami. Dysponując różnymi modelami deklaratywnymi, można byłoby poszukiwać podobieństw na poziomie czysto formalnym między danymi teoriami. Pytania o granice deklaratywnej wirtualizacji teorii filozoficznych również pozwoliłyby rzucić nowe światło na same teorie filozoficzne.

Bibliografia

Alama J., Oppenheimer P.E., Zalta E.N. (2015), *Automating Leibniz's Theory of Concepts*, [in:] *Proceedings of the 25th International Conference on Automated Deduction*, A.P. Felty, A. Middeldorp (eds.), Springer, Dordrecht: 73–97.

Blum A., Malinovich S. (1993), *A formalization of a segment I of Spinoza's Ethics*, "Metalogicon" 1: 1–14.

Błądek I., Komosinski M., Miazga K. (2019), *Mappism: Formalizing classical and artificial life views on mind and consciousness*, "Foundations of Computing and Decision Sciences" 44(1): 55–99, doi: 10.2478/fcds-2019-0005.
Bondecka-Krzykowska I. (2016), *Z zagadnień ontologicznych informatyki*, Wydawnictwo Naukowe UAM, Poznań.
Bramer M. (2013), *Logic Programming with Prolog*, Springer, London.
Clocksin W., Mellish C. (2003), *Programming in Prolog*, Springer, London.
Fitelson B., Zalta E.N. (2007), *Steps Toward a Computational Metaphysics*, "Journal of Philosophical Logic" 36(2): 227–247.
Floridi L. (2002), *What is the philosophy of information*, „Metaphilosophy" 33: 123–145.
Floridi L. (2012), *The philosophy of information*, Oxford University Press, Oxford.
Fulmański P. (2009), *Programowanie w logice. Prolog*, URL=http://fulmanski.pl/zajecia/prolog/wyklad.pdf [dostęp z dnia 4.02.2019].
Garbacz P. (2016), *Digitalizacja filozofii formalnej*, „Filozofia Nauki" 4(96): 27–47.
Janowicz M., Ochnio L., Chmielewski L., Orłowski A. (2017), *Application of Automated Theorem-Proving to Philosophical Thought: Spinoza's Ethics*, doi: 512-518. 10.1007/978-981-10-4154-9_59.
Janusz R. (2002), *Program dla Wszechświata. Filozoficzne aspekty języków obiektowych*, Ignatianum, Kraków.
Jarrett C. (1978), *The logical structure of Spinoza's Ethics. Part I*, „Synthese" 37: 15–65.
Jernajczyk J. (2016a), *Obraz wyczerpujący*, URL=http://www.grapik.pl/grapik-pl/prace-artystyczne/obraz-wyczerpujacy/ [dostęp z dnia 2.02.2019].
Jernajczyk K. (2016b), *Wizualizacja zagadnień naukowych*, URL=http://www.grapik.pl/grapik-pl/projekty-badawcze/wizualizacja-zagadnien-naukowych/ [dostęp z dnia 2.02.2019].
Lipovača M. (2017), *Learn You a Haskell for Great Good!*, URL=http://learnyouahaskell.com [dostęp z dnia 1.03.2018].
Marciszewski W., Stacewicz P. (2011), *Umysł, komputer, świat: o zagadce umysłu z informatycznego punktu widzenia*, Akademicka Oficyna Wydawnicza EXIT, Warszawa.
Marshall T. (2019), *Differences between in vitro, in vivo, and in silico studies*, URL=https://mpkb.org/home/ patients/assessing literature/in vitro studies [dostęp z dnia 10.12.2019].
Moczurad A., Moczurad W. (2006), *Paradygmaty programowania*, URL=http://wazniak.mimuw.edu.pl/index.php?title=Paradygmaty_programowania [dostęp z dnia 11.11.2017].
Mycka J., Stacewicz P. (2015), *Informatyka a filozofia: od informatyki i jej zastosowań do światopoglądu informatycznego*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa.
Niederliński A. (2014), *Programowanie w logice z ograniczeniami*, URL=http://www.pwlzo.pl [dostęp z dnia 9.09.2017].
O'Sullivan B., Goerzen J., Stewart D.B. (2008), *Real World Haskell. code you can believe in*, Ebook ISBN: 978-0-5965-5430-9, 9780596554309.

Płoski Z. (1999), *Słownik encyklopedyczny*, Wydawnictwo Europa, Wrocław.
Pfaffenberg B. (1999), *Słownik terminów komputerowych*, Wydawnictwo Prószyński i S-ka, Warszawa.
Przegalińska A.K. (2014), *Fenomenologia istot wirtualnych*, URL=https://depotuw.ceon.pl/handle/item/463 [dostęp z dnia 6.10.2018].
Rayside D., Campbell G.T. (2000), *An aristotelian understanding of object--oriented programming*, [in:] *Proceedings of the conference on Object-oriented programming, systems, languages, and applications*, ACM Press: 337–353.
Rayside D., Kontogiannis K. (2001), *On the syllogistic structure of object-oriented programming*, doi: 10.1109/ICSE.2001.919086.
Sebesta R. (2016), *Concepts of programming languages*, URL=https://vulms.vu-.edu.pk/Courses/CS508/Downloads/Concepts%20of%20Programming%20Languages%2011th%20Ed.pdf [dostęp z dnia 9.12.2019].
Spinoza B. (1888), *Etyka sposobem geometrycznym wyłożona*, przeł. A. Paskal, Skład Główny w Księgarni E. Wendego i S-ki, Warszawa.
Spinoza B. (1927), *Etyka w porządku geometrycznym dowiedziona*, przeł. I. Myślicki, URL=https://wolnelektury.pl/katalog/lektura/spinoza-etyka.html [dostęp z dnia 4.10.2019].
Spinoza B. (1993), *L'éthique: démontrée selon la méthode géométrique et divisée en cinq parties*, przeł. A. Guérinot, URL=http://classiques.uqac.ca/classiques/spinoza/ethique/ethique.html [dostęp z dnia 6.12.2019].
Spinoza B. (2009), *The Ethics (Ethica Ordine Geometrico Demonstrata*), przeł. R.H.M. Elwes, URL=https://www.gutenberg.org/files/3800/3800-h/3800-h.htm [dostęp z dnia 5.11.2019].
Stacewicz P. (2019), *Wirtualność w perspektywie obliczeniowej*, [w:] *Przedmioty wirtualne*, P. Stacewicz, B. Skowron (red.), Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa: 36–46.
Strzelecki J. (2017), *monada = Monada() – interpretacja obiektowa*, [w:] *Filozofia i technika*, J. Sobota, G. Pacewicz (red.), Wydawnictwo IF UWM, Olsztyn.
Triska M. (2020), *The power of Prolog*, URL=https://www.metalevel.at/prolog [dostęp z dnia 1.02.2020].
Winsberg E. (2015), *Computer simulations in science*, Summer Edition, E.N. Zalta (ed.), URL=https://plato.stanford.edu/archives/sum2015/entries/simulations-science/ [dostęp z dnia 3.04.2018].