Filozofia Nauki, ISSN 1230-6894
Rok XXIV, 2016, Nr 4(96), s. 27-47

PAWEŁ GARBACZ*

DIGITALIZACJA FILOZOFII FORMALNEJ**

Abstract

DIGITALIZATION OF FORMAL PHILOSOPHY

The paper presents a case study of digitalisation of formal philosophy. Using the theorem provers available at www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP, I show that the formal ontology presented in (Nieznański 2007) is inconsistent. I also discuss some ways to avoid this inconsistency.

Keywords: automated theorem proving, formal philosophy, consistency

Czy jest możliwe zbudowanie sztucznego filozofa, elektronicznego golema, który potrafiłby prowadzić mniej lub bardziej abstrakcyjne rozważania o bycie i niebycie, początkowo naśladując zachowania ludzkich filozofów, a w dalszej perspektywie przezwyciężając ich ograniczenia poznawcze? Ten raczej mętny problem można łatwo dookreślić, ograniczając pole możliwości do tych, które są osadzone w aktualnym stanie badań filozoficznych i informatycznych. Przy takich ograniczeniach zagadnienie filozofującego golema sprowadza się obecnie do pytania o możliwość, ewentualnie zasadność, wykorzystania systemów automatycznego dowodzenia twierdzeń w filozofii formalnej, zwanej czasami filozofią matematyczną[1]. Mimo żywotności paradygmatu logicznego w zasadzie nie spotyka się wykorzystywania komputerów jako narzędzia dowodzenia twierdzeń filozoficznych[2]. Znane są oczywiście pewne teoretyczne ograniczenia

* Katedra Podstaw Informatyki, Wydział Filozofii, Katolicki Uniwersytet Lubelski, Al. Racławickie 14, Lublin, garbacz@kul.pl.

** Artykuł został sfinansowany ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji numer DEC-2012/07/B/HS1/01938.

[1] Pomijam tu wykorzystanie komputerowych symulacji interakcji społecznych w argumentacji filozoficznej typu tej przedstawionej przez Gustafssona i Petersona (2012) oraz wykorzystanie narzędzi informatycznych do analiz statystycznych w filozofii eksperymentalnej.

[2] Stan badań przedstawiają Beavers (2011) i Portoraro (2014).

nia takich zastosowań, na przykład twierdzenie o nierozstrzygalności klasycznego rachunku logicznego. Poza dwoma wyjątkami, z których jeden omawiam w drugiej części artykułu, brak jednak bardziej praktycznie zorientowanych przedsięwzięć badawczych, które ilustrowałyby za pomocą konkretnych przykładów racjonalność (lub jej brak) tego rodzaju automatyzacji[3]. Prawdą jest, że automatyzacja dowodzenia twierdzeń nie jest zbyt popularna nawet w samej matematyce, niemniej uzyskane tam rezultaty powinny zachęcić zwolenników filozofii formalnej do odważniejszego eksperymentowania.

Formalizacja jest często postrzegana przez jej zwolenników jako zabieg podnoszący wartość poznawczą formalizowanych treści, niezależnie od jej metodologicznej proweniencji (Suppes 1986), a w skrajnej wersji tego poglądu — jako podstawowa metoda „unaukowienia" filozofii (Leitgeb 2013). Z tej perspektywy digitalizację filozofii formalnej można postrzegać jako kolejny krok w kierunku epistemicznego udoskonalenia tej dziedziny wiedzy[4]. Faktycznie budowane formalizmy, nieco wbrew deklaracjom, często zawierają mniej lub bardziej poważne usterki[5]. Współczesne systemy wspierające dowodzenie twierdzeń okazały się zdolne do rozpoznawania takich błędów — najbardziej znane przykłady dotyczą pomyłek w dowodzie praw Keplera podanym przez Newtona (Fleuriot 2001) czy braków w zaproponowanej przez Hilberta formalizacji geometrii (Meikle, Fleuriot 2003). W tego rodzaju wypadkach digitalizacja formalizmu może być jedną z metod sprawdzenia jego poprawności i ewentualnie środkiem do wprowadzenia niezbędnych poprawek. Znane są również sukcesy w zastosowaniu maszyn do dowodzenia wcześniej nieudowodnionych twierdzeń, np. słynnego twierdzenia o czterech kolorach (Appel, Haken 1989)[6] czy mniej znanego twierdzenia, że wszystkie algebry Robbinsa są algebrami Boole'a (McCune 1997). Oczywiście, komputerowo wygenerowany dowód również może być niepoprawny, np. z racji błędów w implementacji określonego algorytmu. Jednak z tego, co wiemy o dotychczasowych zastoso-

---

[3] Biorę w nawias zastosowania systemów dowodzenia twierdzeń w tzw. inżynierii ontologii (zob. np. Goczyła 2011), ponieważ metateoretyczny kontekst tworzonych tam artefaktów informatycznych jest różny od kontekstu dociekań filozoficznych.

[4] Termin „digitalizacja filozofii (formalnej)" jest neofrazeologizmem, który został sformułowany w kontekście pewnej wizji humanistyki cyfrowej — zob. Garbacz 2016. Zgodnie z tą wizją automatyzacja rozumowania jest głównym sposobem wykorzystania narzędzi informatycznych w filozofii. Megill i Linfor (w druku) przekonują, że każdy system filozoficzny, włączając w to wszystkie możliwe systemy filozoficzne, jest w tym sensie digitalizowalny.

[5] Skrajnie pesymistyczne szacunki mówią, że prawie połowa artykułów matematycznych zawiera mniej lub bardziej istotne błędy formalne (Davis 1972: 260-262).

[6] Ściślej mówiąc, pewien fragment dowodu twierdzenia o czterech kolorach ze względu na długość obliczeń potrzebnych do jego przeprowadzenia został wykonany przez maszynę cyfrową.

waniach systemów dowodzenia twierdzeń, takie przypadki są bardzo rzadkie. MacKenzie (2005) wymienia tylko jeden znany (mu) błąd tego rodzaju[7].

Mało prawdopodobne wydaje się, by wytwory filozofii formalnej wolne były od usterek formalnych lub by formalizujący filozofowie byli w stanie skuteczniej dowodzić twierdzeń niż matematycy czy logicy. Dorobek filozofii formalnej jest co prawda dużo mniejszy niż rezultaty czystej matematyki czy logiki, jednak w chwili obecnej nie jest on zaniedbywalnie mały. Dziwi więc i niepokoi stan badań nad digitalizacją filozofii formalnej, a raczej ich brak. W artykule omówię jedno z dwóch znanych mi dokonań w tym zakresie i przedstawię pewne jego uzupełnienie. Przykład ten posłuży do pokazania zysków poznawczych, które osiąga się dzięki digitalizacji, oraz przeszkód, które można w tym procesie napotkać. W szczególności zaproponuję nową metodę wykorzystania systemów automatycznego dowodzenia twierdzeń w badaniu metalogicznych własności systemów dedukcyjnych.

TYPY DIGITALIZACJI FORMALIZACJI

Z punktu widzenia metodologii nauk formalnych automatyzacji może podlegać każdy etap tworzenia teorii formalnej, a więc:

(1) definiowanie języka teorii formalnej,

(2) wybór uznanych wyrażeń tego języka i/lub reguł inferencji, np. za pomocą aksjomatyzacji,

(3) dowodzenie twierdzeń,

(4) konstruowanie metateorii tej teorii, np. budowa modelu.

Obecnie rozwijane technologie informatyczne wspierające badania w naukach formalnych koncentrują się na trzecim oraz w pewnym stopniu czwartym etapie, tj. na automatycznym generowaniu modeli. Ze względu na stopień interwencji człowieka w działanie systemu można wyróżnić trzy podstawowe typy czy poziomy takich zastosowań:

(1) dostarczanie infrastruktury do tworzenia struktur formalnych, w szczególności wspieranie logików i matematyków przy budowaniu dowodów i modeli;

---

[7] Co ciekawe, błąd ten dotyczył wspomnianego dowodu twierdzenia na temat algebry Robbinsa. Dowód został wygenerowany przez system REVEAL (zob. MacKenzie 2001: 289-290).

(2) automatyczne dowodzenie twierdzeń wskazanych przez człowieka lub znajdowanie modeli dla wskazanych zbiorów formuł;

(3) automatyczne generowanie twierdzeń oraz ich dowodów.

Pod (1) kryją się wszystkie programy komputerowe, które ułatwiają tworzenie przez człowieka teorii formalnych, w szczególności umożliwiają konstruowanie dowodów twierdzeń i sprawdzanie ich poprawności. Są to tzw. automatyczne asystenty dowodów (*computational proof assistants*). Udział takich programów w procesie twórczym jest minimalny i sprowadza się do sprawdzenia poprawności syntaktycznej dowodu skonstruowanego przez człowieka, tj. zgodności ze zbiorem reguł wnioskowania i zbiorem uznanych wyrażeń.

Typ (2) obejmuje te systemy, które poszukują dowodów lub modeli twierdzeń wskazanych przez człowieka i określają metalogiczne własności teorii formalnych, które wynikają z takich poszukiwań, np. niesprzeczność. Typ ten stawia przed maszyną więcej wymagań niż poprzedni, ponieważ do znalezienia (poprawnego) dowodu potrzeba więcej kreatywności niż sprawdzanie poprawności dowodu już znalezionego.

Ostatni poziom digitalizacji sprowadza się w zasadzie do pełnej automatyzacji procesu dowodzenia twierdzeń. W granicach zadanej przez człowieka aksjomatyzacji maszyna podpowiada człowiekowi, nie tylko *jak* można dowodzić, lecz także *co* można udowodnić[8].

Typy (1)-(3) nie są wynikiem podziału logicznego, lecz dość luźną typologią, która dopuszcza istnienie przypadków paradygmatycznych, granicznych itd. Jeżeli chodzi o zastosowania systemów wspierających dowodzenie twierdzeń w filozofii formalnej, to wszystkie istniejące zastosowania ograniczają się do typu (2). Ściślej mówiąc, znane mi przypadki digitalizacji filozofii formalnej obejmują tylko dwa przedsięwzięcia. Po pierwsze, są to digitalizacje różnych wersji formalizacji dowodu ontologicznego Gödla, realizowane przez Christopha Benzmüllera i Brunona W. Palea (2014), a po drugie jest to metafizyka obliczeniowa.

Ponieważ drugie z tych przedsięwzięć jest dużo bardziej rozbudowane, skoncentruję się właśnie na nim jako na przykładzie ukazującym możliwe zyski z digitalizacji filozofii formalnej.

---

[8] Megill i Linford (w druku) wskazują również możliwość automatycznego generowania nowych aksjomatyk. Przedstawiona przez nich metoda sprowadza się do tworzenia nowych teorii przez kombinacje znanych z historii filozofii twierdzeń filozoficznych i ich negacji. O ile mi wiadomo, nie podejmowano do tej pory prób realizacji tego rodzaju programu. Zresztą liczba niesprzecznych teorii, które można by w ten sposób uzyskać, byłaby prawdopodobnie na tyle duża, że uniemożliwiłaby wyszukanie w wygenerowanym zbiorze pomysłów wartościowych poznawczo, istotnie nowatorskich itp.

Najpierw wprowadźmy jednak pewne uściślenie terminologiczne. Podstawowe znaczenie używanego w dalszej części artykułu terminu „digitalizacja" jest związane z metalogicznym znaczeniem terminu „teoria formalna". *Digitalizacja teorii* jest tu rozumiana jako proces, na który składa się:

(1) wybór systemu automatycznego dowodzenia twierdzeń (*resp*. generowania modeli);

(2) zapis tej teorii w języku „zrozumiałym" dla tego systemu;

(3) uruchomienie procesów realizowanych automatycznie przez ten system, tj. generowania z języka tej teorii dowodów lub modeli dla formuł, które są wskazane przez człowieka, w szczególności procesu sprawdzenia niesprzeczności teorii;

(4) metalogiczna interpretacja uzyskanych rezultatów.

DIGITALIZACJA METAFIZYKI OBLICZENIOWEJ

Metafizyka obliczeniowa jest wieloletnim projektem badawczym realizowanym przez Edwarda N. Zaltę i jego współpracowników, których celem jest „implementacja i badania nad sformalizowaną aksjomatyką w środowisku automatycznego dowodzenia twierdzeń"[9].

W ramach tego projektu „zdigitalizowano" główne fragmenty metafizyki przedmiotów abstrakcyjnych (Zalta 1983) rozwijanej od wielu lat przez Zaltę, w tym platońską teorię idei, teorię światów możliwych oraz leibnizjańską teorię pojęć[10]. W zasadzie niezależnie od tych badań przeprowadzono digitalizację jednej z wersji dowodu ontologicznego na istnienie Boga. W tym miejscu szczegółowo przedstawię wyniki tego ostatniego przedsięwzięcia — pozostałe rezultaty wymagałaby bowiem zbyt obszernych wyjaśnień dość idiosynkratycznych formalizmów budowanych przez Zaltę.

W roku 1991 Paul Oppenheimer i Zalta opublikowali formalizację tego argumentu, tj. przedstawili zapis jednej z interpretacji teistycznej argumentacji inspirowanej *Proslogionem* Anzelma z Canterbury w języku logiki pierwszego rzędu z operatorem deskrypcyjnym. Przedstawiona formalizacja opierała się na trzech przesłankach:

---

[9] Dokumentacja projektu jest dostępna na stronie http://mally.stanford.edu/cm/.

[10] Ostatni z serii artykułów dotyczących metafizyki obliczeniowej to (Alama, Oppenheimer, Zalta 2015).

$$\exists x\, \mathrm{Sup}_G(x) \qquad (\textit{Przesłanka 1})$$

$$\neg \mathrm{E}(\imath x \mathrm{Sup}_G(x)) \rightarrow \exists y[\mathrm{G}(y, \imath x \mathrm{Sup}_G(x)) \wedge \mathrm{C}(y)] \qquad (\textit{Przesłanka 2})$$

$$\mathrm{G}(x, y) \vee \mathrm{G}(y, x) \vee x = y \qquad (\textit{Spójność})$$

gdzie „C(x)" to tyle co „można pomyśleć"; „G(x, y)" to „x jest większy niż y"; „E(x)" to „x istnieje", a „Sup_G" jest zdefiniowanym predykatem, którego funkcją jest oznaczenie bytu, ponad który nie można pomyśleć nic większego, czyli teistycznego Absolutu:

$$\mathrm{Sup}_G(x) \leftrightarrow \mathrm{C}(x) \wedge \neg\exists y[\mathrm{G}(y, x) \wedge \mathrm{C}(y)]$$

Oppenheimer i Zalta (1991) dowodzą, że z przedstawionych przesłanek wynika logicznie zdanie „E($\imath x \mathrm{Sup}_G(x)$)", głoszące, że istnieje byt, ponad który nie można pomyśleć nic większego.

W napisanym dwadzieścia lat później artykule ci sami autorzy zauważają, że wcześniejszą formalizację można znacznie uprościć. Pokazują, że zdanie „E($\imath x \mathrm{Sup}_G(x)$)" wynika logicznie z *Przesłanki 2*, a pozostałe dwie przesłanki są w tym dowodzie zbędne (Oppenheimer, Zalta 2011). Spostrzeżenie to „zawdzięczają" automatycznemu systemowi dowodzenia twierdzeń PROVER9. Zapisanie w nim przesłanek oraz uruchomienie automatycznych procesów wnioskowania pokazało, że PROVER9 jest w stanie udowodnić tezę o istnieniu Boga bez odwoływania się do *Przesłanki 1* czy *Spójności*.

Ponieważ uproszczenie argumentu polegające na usunięciu jednej z przesłanek, lecz nienaruszające jego konkluzywności, jest zwiększeniem jego wartości poznawczej, to wynik ten pokazuje, jak zastosowanie dość prostego narzędzia informatycznego przyczyniło się do ulepszenia jednego z bardziej doniosłych argumentów filozoficznych. Oczywiście, spostrzeżenia, którego dokonali Oppenheimer i Zalta dzięki użyciu PROVER9, można również dokonać za pomocą bardziej tradycyjnych metod (zob. Garbacz 2012). Niemniej, przez dwadzieścia lat umykało ono uwadze czytelników wcześniejszego artykułu Oppenheimera i Zalty.

Zastosowanie digitalizacji do teorii przedmiotów abstrakcyjnych pozwoliło też na odkrycie niekonkluzywności jednego z dowodów: w wypadku wyrażenia uważanego wcześniej za tezę system MACE4, program do automatycznego generowania modeli, znalazł model, w którym aksjomaty tej teorii są spełnione, a owo wyrażenie nie (zob. Fitelson, Zalta 2007: 241-242).

Podsumowując, w rozwoju teorii formalnej przejście ze stadium aksjomatycznego (abstrakcyjnego) do stadium zdigitalizowanego może wiązać się z poprawą jakości poznawczej tej teorii. Uzyskane korzyści można uważać za

niewielkie, lecz nie są one zaniedbywalnie małe. Czasami jednak digitalizacja może doprowadzić do istotniejszych spostrzeżeń.

DIGITALIZACJA SFORMALIZOWANEJ ONTOLOGII ORIENTACJI KLASYCZNEJ

Przedstawię studium przypadku, w którym różnica między wartością teorii formalnej w stadium niezdigitalizowanym a jej wartością w stadium zdigitalizowanym jest dużo większa i poznawczo bardziej doniosła. Przedmiotem rozważań będzie przedstawiona przez Edwarda Nieznańskiego (2007) tzw. sformalizowana ontologia orientacji klasycznej. Jak pisze Kordula Świętorzecka:

> Przedsięwzięcie formalizacyjne podjęte przez E. Nieznańskiego jest więc w zasadzie autorską wersją ontologii substancjalnej, respektującą jednak wykład Arystotelesa oraz wybrane koncepcje należące do nurtu filozofii klasycznej takich autorów jak: J. Łukasiewicz, J. Salamucha, J. M. Bocheński, M. A. Krąpiec, A. B. Stępień, F. Rivetti-Barbo (Świętorzecka 2008: 207).

Za Świętorzecką warto też powtórzyć, że ogół przedstawionych przez Nieznańskiego rozważań składa się z dwóch części: teorii indywiduów i ich własności (rozdziały II-V) oraz teorii relacji (rozdział VI).

Teoria indywiduów i ich własności jest teorią formalną pierwszego rzędu ukonstytuowaną przez trzydzieści osiem aksjomatów i kilkadziesiąt definicji terminów wtórnych, na podstawie których Nieznański przedstawia szkice dowodów kilkuset twierdzeń. Przeprowadzona przeze mnie digitalizacja dotyczy tylko części tej obszernej teorii, a mianowicie podteorii sformułowanej w rozdziale pierwszym zatytułowanym *Aliquid* (dalej jako TA).

Nieznański wprowadza w tym rozdziale szesnaście aksjomatów charakteryzujących znaczenia predykatu „≤", gdzie „$x \leq y$" odczytujemy jako „(istota) x jest zawarta w (istocie) y", oraz czterech symboli funkcji:

„Σ" — gdzie „Σx" odczytujemy jako „pewien x",
„Π" — gdzie „Πx" odczytujemy jako „każdy x",
„|" — gdzie „$x|y$" odczytujemy jako „x od y-a", tak jak np. w „niewolnik (od) pana", „ster (od) łodzi" itp.
„E|Z|" — gdzie „E|Z|x" odczytujemy jako „element zbioru x-ów"[11].

[11] Nieznański (2007) nie podaje wprost definicji języka formalizowanej ontologii. Podane w tekście głównym stwierdzenia są oparte na funkcjach składniowych tych symboli w formułach i dowodach tam podanych.

(A1) $x \leq y \Leftrightarrow \forall z\,[z \leq x \Rightarrow z \leq y]$

(A2) $\exists y\,\forall x\; x \leq y$

(A3) $\exists x\,\forall y\; x \leq y$

(A4) $\forall x, y\,\exists z\,[x \leq z \wedge y \leq z \wedge \forall u\,(x \leq u \wedge y \leq u \Rightarrow z \leq u)]$

(A5) $\forall x, y\,\exists z\,[z \leq x \wedge z \leq y \wedge \forall u\,(u \leq x \wedge u \leq y \Rightarrow u \leq z)]$

(A6) $\forall x, y, z\,(x+y)^*z \equiv x^*z+y^*z$

(A7) $\forall x\,\exists y\,[x+y \equiv 1 \wedge x^*y \equiv 0]$

(A8) $\neg 1 \leq 0$

(A9a) $x \leq \Sigma y \Leftrightarrow \Pi x \leq y$

(A9b) $\Pi x \leq y \Leftrightarrow x \leq y$[12]

(A10a) $x \leq \Pi y \Leftrightarrow \Sigma x \leq y$

(A10b) $\Sigma x \leq y \Leftrightarrow x \equiv y$

(A11) $x \leq y|z \Rightarrow x \leq y$

(A12) $x \varepsilon y|z \wedge z \varepsilon u \Rightarrow x \varepsilon y|u$

(A13) $x \varepsilon y \Rightarrow z|x \varepsilon z|y$

(A14) $x \varepsilon y|(u|w) \Rightarrow \exists z\,(x \varepsilon y|z \wedge z \varepsilon u|w)$[13]

(A15) $x|y \equiv x|\Sigma y$

(A16) $x \leq \mathrm{E}|\mathrm{Z}|y \Leftrightarrow x \varepsilon y$

---

[12] Aksjomaty A9a i A9b występują w (Nieznański 2007) jako jeden: $x \leq \Sigma y \Leftrightarrow \Pi x \leq y \Leftrightarrow x \leq y$. Podobna uwaga dotyczy aksjomatów A10a i A10b. Stąd „szesnaście" aksjomatów jest tu zapisane za pomocą osiemnastu formuł.

[13] Aksjomat A13 ma w (Nieznański 2007: 37) wieloznaczny składniowo poprzednik o postaci „$x \varepsilon y|u|w$", więc przedstawiona tu forma jest próbą jego ujednoznacznienia. Alternatywną formą byłoby: (A14') $x \varepsilon (y|u)|w \Rightarrow \exists z\,(x \varepsilon y|z \wedge z \varepsilon u|w)$. Główne wyniki prezentowane w tym artykule nie zależą od wyboru jednej z tych interpretacji. Warto przy tym zauważyć, że sposób przedstawiania aksjomatów, definicji, twierdzeń i ich dowodów, przyjęty w tej książce, czyli warstwa edytorska części formalnej, nie ułatwia jej lektury i może prowadzić do pomyłek w recepcji jej treści. I tak Świętorzecka (2008: 211) pisze, że ostatnim aksjomatem sformalizowanej ontologii substancjalnej wyrażonej w języku atrybutywnym jest aksjomat 37, gdy tymczasem na s. 89 pojawia się aksjomat A38, który stwierdza, że świat materialny jest przedmiotem.

Ponadto Nieznański wprowadza za pomocą definicji dwadzieścia siedem symboli wtórnych, z których sześć występuje, *explicite* lub *implicite*, w podanych wcześniej aksjomatach:

(Df.≡) $x \equiv y \Leftrightarrow x \leq y \wedge y \leq x$

(Df.0) $x \equiv 0 \Leftrightarrow \forall y\ x \leq y$

(Df.1) $x \equiv 1 \Leftrightarrow \forall y\ y \leq x$

(Df.+) $z \equiv x+y \Leftrightarrow x \leq z \wedge y \leq z \wedge \forall u\ (x \leq u \wedge y \leq u \Rightarrow z \leq u)$

(Df.*) $z \equiv x^*y \Leftrightarrow z \leq x \wedge z \leq y \wedge \forall u\ (u \leq x \wedge u \leq y \Rightarrow u \leq z)$

(Df.ε) $x \varepsilon y \Leftrightarrow \neg x \leq 0 \wedge x \leq y$

Pewnym problemem interpretacyjnym pozostają formuły oznaczone w (Nieznański 2007: 37) jako „Df.| Π" i „Df.|Σ":

(Df.|Π) $x \varepsilon y|\Pi z \Leftrightarrow x \varepsilon 1 \wedge \forall u\ (u \varepsilon z \Rightarrow x \varepsilon y|u)$

(Df.|Σ) $x \varepsilon y|\Sigma z \Leftrightarrow \Sigma u\ (u \varepsilon z \wedge x \varepsilon y|u)$

Problem polega na tym, że symbole „Σ" i „Π" zostały wprowadzone „kilkadziesiąt formuł" wcześniej (na s. 35), w sposób, jak pisze autor, „aksjomatyczny". Dodatkowo Df.|Π i Df.|Σ są definicjami cząstkowymi tych symboli, ponieważ charakteryzują tylko ich wystąpienia w pewnych miejscach pewnych formuł. Sądzę więc, że bezpiecznie będzie potraktować te definicje jako aksjomaty, które doprecyzowują ich sens określony przez aksjomaty A9-A10.

TA jest więc teorią aksjomatyczną opartą na aksjomatach A1-A16 oraz wymienionych definicjach. Przeprowadzona przeze mnie digitalizacja doprowadziła do stwierdzenia, że jest to teoria sprzeczna. W konsekwencji cała sformalizowana ontologia substancjalna zawarta w (Nieznański 2007) jest sprzeczna.

Digitalizacja TA polegała na zapisie aksjomatów i definicji w tzw. notacji TPTP, a następnie na wykorzystaniu systemów automatycznego dowodzenia twierdzeń. Systemy te są dostępne na stronie www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP. Wykorzystywane przez nie środki dowodowe nie wychodzą poza logikę pierwszego rzędu:

```
fof(axiom1, axiom, (p(X,Y) <=> ! [Z] : (p(Z,X) => p(Z,Y)))).
fof(axiom2, axiom, (? [Y] : ! [X] : p(X,Y))).
fof(axiom3, axiom, (? [X] : ! [Y] : p(X,Y))).
fof(axiom4, axiom, (! [X,Y] : ? [Z] : ((p(X,Z) & p(Y,Z)) &
     ! [U] : ((p(X,U) & p(Y,U)) => p(Z,U)) ))).
fof(axiom5, axiom, (! [X,Y] : ? [Z] : ((p(Z,X) & p(Z,Y)) &
     ! [U] : ((p(U,X) & p(U,Y)) => p(U,Z)) ))).
fof(axiom6, axiom, (! [X,Y,Z] :
     equiv(prod(sum(X,Y),Z),sum(prod(X,Z),prod(Y,Z))))).
fof(axiom7, axiom, (! [X] : ? [Y] : (equiv(sum(X,Y), jeden) &
     equiv(prod(X,Y),zero)))).
fof(axiom8, axiom, (~(p(jeden, zero)))).
fof(axiom9a, axiom, (p(X,sigma(Y)) <=> p(pi(X),Y))).
fof(axiom9b, axiom, (p(pi(X),Y) <=> p(X,Y))).
fof(axiom10a, axiom, (p(X,pi(Y)) <=> p(sigma(X),Y))).
fof(axiom10b, axiom, (p(sigma(X),Y) <=> equiv(X,Y))).
fof(axiom11, axiom, (p(X,rel(Y,Z)) => p(X,Y))).
fof(axiom12, axiom, ((epsilon(X,rel(Y,Z)) & epsilon(Z,U)) =>
     epsilon(X,rel(Y,U)))).
fof(axiom13, axiom, (epsilon(X,Y) =>
     epsilon(rel(Z,X),rel(Z,Y)))).
fof(axiom14, axiom, (epsilon(X,rel(Y,rel(U,W))) =>
     ? [Z] : (epsilon(X,rel(Y,Z)) & epsilon(Z,rel(U,W))))).
fof(axiom15, axiom, (equiv(rel(X,Y),rel(X, sigma(Y))))).
fof(axiom16, axiom, (epsilon(x,el(y)) <=> epsilon(x,y))).
fof(df_equiv, axiom, (equiv(X,Y) <=> (p(X,Y) & p(Y,X)))).
fof(df0, axiom, (equiv(X,zero) <=> ! [Y] : p(X,Y))).
fof(df1, axiom, (equiv(X,jeden) <=> ! [Y] : p(Y,X))).
fof(df_sum, axiom, (equiv(Z,sum(X,Y)) <=> ((p(X,Z) & p(Y,Z)) &
     ! [U] : ((p(X,U) & p(Y,U)) => p(Z,U))))).
fof(df_prod, axiom, (equiv(Z,prod(X,Y)) <=> ((p(Z,X) & p(Z,Y))
     & ! [U] : ((p(U,X) & p(U,Y)) => p(U,Z))))).
fof(df_epsilon, axiom, (epsilon(X,Y) <=>
     (~p(X,zero) & p(X,Y)))).
fof(df_pi, axiom, (epsilon(X,rel(Y,pi(Z))) <=> (epsilon(X,
     jeden) & ! [U] : (epsilon(U,Z) => epsilon(X,rel(Y,U)))))).
fof(df_sigma, axiom, (epsilon(X,rel(Y,sigma(Z))) <=>
     ? [U] : (epsilon(U,Z) & epsilon(X,rel(Y,U))))).
```

Ramka 1. Zapis TA w notacji TPTP

Tabela 1 podaje symbole formalne w obu notacjach:

Notacja z (Nieznański 2007)	Notacja TPTP
¬	~
∧	&
⇒	=>
⇔	<=>
∀	!
∃	?
≤	p
Π	pi
Σ	sigma
|	rel
E|Z|	el
≡	equiv
0	zero
1	jeden
+	sum
*	prod
ε	epsilon

Tabela 1. Notacja TPTP

Dowód sprzeczności TA można uzyskać przy użyciu wielu systemów automatycznego dowodzenia twierdzeń dostępnych za pośrednictwem tego portalu. Przedstawię uproszczony zapis dowodu uzyskany z systemu PROVER9. Ponieważ PROVER9 stosuje nieco inną notacją niż TPTP, Tabela 2 podaje odpowiednie tłumaczenia notacji. Dodatkowo dowód zawiera symbol „|", który oznacza alternatywę (nierozłączną).

Notacja TPTP	Notacja PROVER9
~	-
=>	->
<=>	<->
!	all
?	exists

Tabela 2. Specyficzne symbole w notacji PROVER9

Ponieważ notacje używane przez systemy automatycznego dowodzenia twierdzeń są uważane za mało czytelne (Ganesalingam, Gowers 2016), przedstawiam zapis *idei* tego dowodu w notacji z (Nieznański 2007), zachowując jednak numerację wierszy wygenerowaną przez komputer[14]:

1.	$x \leq y \Leftrightarrow \forall z\,[z \leq x \Rightarrow z \leq y]$	(A1)
11.	$\Sigma x \leq y \Leftrightarrow x \equiv y$	(A10b)
18.	$x \equiv y \Leftrightarrow x \leq y \wedge y \leq x$	(Df.≡)
20.	$x \equiv 1 \Leftrightarrow \forall y\, y \leq x$	(df1)
33.	$x \equiv 1 \Rightarrow y \leq x$	z: 20
34.	$\Sigma x \leq y \Rightarrow x \equiv y$	z: 11
42.	$x \leq y \wedge y \leq x \Rightarrow x \equiv y$	z: 18
57.	$\neg(\mathrm{f}(x,y) \leq x) \Rightarrow x \leq y$	z: 1 (oraz prawa „$[p \Leftrightarrow (q \Rightarrow r)] \Rightarrow (\neg q \Rightarrow p)$”)
58.	$\neg 1 \leq 0$	(A8)
65.	$\mathrm{f}(x,y) \leq y \Rightarrow x \leq y$	z: 1 (oraz prawa „$[p \Leftrightarrow (q \Rightarrow r)] \Rightarrow (r \Rightarrow p)$”)
99.	$\Sigma x \leq 1 \Rightarrow y \leq x$	z: 34 i 33
119.	$x \leq 1 \wedge 1 \leq x \Rightarrow y \leq x$	z: 42, 33
159.	$1 \leq 1 \Rightarrow x \leq 1$	z: 119
208.	$x \leq x$	z: 65, 57
218.	$x \leq 1$	z: 159, 208
221.	$x \leq y$	z: 99, 218
222.	sprzeczność	58 i 221[15]

[14] Luki w numeracji są rezultatem pominięcia wierszy, które zostały wygenerowane przez PROVER9, ale okazały się nieużyteczne w podanym dowodzie.

[15] Symbol „f” użyty w tym dowodzie jest dwuargumentowym symbolem funkcyjnym.

```
% Proof 1 at 002 (+ 000) seconds
% Length of proof is 17
% Level of proof is 6
% Maximum clause weight is 9000
% Given clauses 14

1 (all X all Y (p(X,Y) <-> (all Z (p(Z,X)
     -> p(Z,Y)))))                          # label(axiom1)
11 (all X all Y (p(sigma(X),Y)
     <-> equiv(X,Y)))                       # label(axiom10b)
18 (all X all Y (equiv(X,Y) <-> p(X,Y)
     & p(Y,X)))                             # label(df_equiv)
20 (all X (equiv(X,jeden)
     <-> (all Y p(Y,X))))                   # label(df1)
33 -equiv(A,jeden) | p(B,A)                 [(20)].
34 -p(sigma(A),B) | equiv(A,B)              [(11)].
42 equiv(A,B) | -p(A,B) | -p(B,A)           [(18)].
57 p(A,B) | p(f1(A,B),A)                    [(1)].
58 -p(jeden,zero)                           # label(axiom8).
65 p(A,B) | -p(f1(A,B),B)                   [(1)].
99 -p(sigma(A),jeden) | p(B,A)              [(34,33)].
119 -p(A,jeden) | -p(jeden,A) | p(B,A)      [(42,33)].
159 -p(jeden,jeden) | p(A,jeden)            [(119)].
208 p(A,A)                                  [(65,57)].
218 p(A,jeden)                              [(159,208)].
221 p(A,B)                                  [(99,218)].
222 $F.                                     [(221,58)].
```

Ramka 2. Dowód sprzeczności TA

Warto zauważyć, że znaleziony dowód pokazuje stosunkowo niewielką część TA, która jest sprzeczna — sprzeczna jest mianowicie teoria złożona z aksjomatów A1, A2, A7, A8, A10b oraz definicji Df.1 i D. ≡. Analiza dowodu prowadzi do wniosku, że najbardziej problematyczny jest A10b. Ponieważ z aksjomatu A1 wynika, że relacja ≤ jest zwrotna, więc ≡ jest również zwrotna (na mocy swej definicji). Df.1 implikuje wtedy, że $\forall y\ y \leq 1$, czyli również, że $\Sigma x \leq 1$. Ostatecznie, z A10b otrzymujemy $x \equiv 1$ (dla dowolnego x), co wobec definicji ≡ jest sprzeczne z A8.

Usunięcie A10b z TA nie usuwa jednak sprzeczności ontologii, co pokazuje kolejny dowód: również aksjomaty A1, A2, A5, A9a, A11, A13, A15 oraz definicje Df.≡ i Df.ε prowadzą do sprzeczności:

```
% Proof 1 at 40.52 (+ 0.73) seconds.
% Length of proof is 29.
% Level of proof is 6.
% Maximum clause weight is 11.
% Given clauses 912.

1 (all X all Y (p(X,Y) <-> (all Z (p(Z,X)
     -> p(Z,Y))))                          # label(axiom1)
2 (exists Y all X p(X,Y))                  # label(axiom2)
5 (all X all Y exists Z (p(Z,X) & p(Z,Y)
     & (all U (p(U,X) & p(U,Y) -> p(U,Z))))) # label(axiom5)
8 (all X all Y (p(X,sigma(Y))
     <-> p(pi(X),Y)))                      # label(axiom9a)
11 (all X all Y all Z (p(X,rel(Y,Z))
     -> p(X,Y)))                           # label(axiom11)
13 (all X all Y all Z (epsilon(X,Y) ->
     epsilon(rel(Z,X),rel(Z,Y))))          # label(axiom13)
15 (all X all Y equiv(rel(X,Y),
     rel(X,sigma(Y))))                     # label(axiom15)
16 (all X all Y (equiv(X,Y) <->
     p(X,Y) & p(Y,X)))                     # label(df_equiv)
21 (all X all Y (epsilon(X,Y) <->
     -p(X,zero) & p(X,Y)))                 # label(df_epsilon)
24 p(A,c1)                                 [(2)].
33 equiv(rel(A,B),rel(A,sigma(B)))         [(15)].
35 -p(jeden,zero).                         # label(axiom8)
36 -epsilon(A,B) | -p(A,zero)              [(21)].
37 -equiv(A,B) | p(A,B)                    [(16)].
48 p(A,sigma(B)) | -p(pi(A),B)             [(8)].
51 -p(A,rel(B,C)) | p(A,B)                 [(11)].
56 -p(A,B) | -p(C,A) | p(C,B)              [(1)].
58 epsilon(A,B) | p(A,zero) | -p(A,B)      [(21)].
60 -epsilon(A,B) |
     epsilon(rel(C,A),rel(C,B))            [(13)].
62 -p(A,B) | -p(A,C) | p(A,f3(B,C))        [(5)].
85 p(rel(A,B),rel(A,sigma(B)))             [(37, 33)].
171 -p(c1,zero)                            [(56, 24, 35)].
208 -epsilon(A,B) | -p(rel(C,A),zero)      [(60, 36)].
298 p(A,f3(c1,c1))                         [(62, 24)].
1188 p(rel(A,B),A)                         [(85, 51)].
3720 p(A,sigma(f3(c1,c1)))                 [(298, 48)].
305732 epsilon(c1,sigma(f3(c1,c1)))        [(58, 171, 3720)].
391005 -p(rel(A,c1),zero)                  [(208, 305732)].
391006 $F                                  [(391005, 1188)].
```

Ramka 3. Dowód sprzeczności fragmentu TA

O ile stwierdzenie sprzeczności, ewentualnie niesprzeczności, danej teorii za pomocą wybranego systemu automatycznego dowodzenia twierdzeń nie wymaga specjalistycznej wiedzy z zakresu informatyki czy reprezentacji wiedzy, o tyle w ogólnym przypadku nie jest możliwe automatyczne *usunięcie* sprzeczności. Można natomiast, korzystając z takiego systemu, uzyskać pewne

wyniki metalogiczne, które wskazują na zakres możliwości modyfikacji sprzecznej teorii. Możemy mianowicie za jego pomocą znaleźć maksymalne (w sensie relacji $\subseteq$) niesprzeczne podzbiory teorii sprzecznej oraz minimalne podzbiory sprzeczne. W ten sposób można wskazać formalizującemu filozofowi możliwe sposoby uniknięcia wykrytej sprzeczności.

Abstrahując od kwestii implementacyjnych, realizacja takiego zadania jest prosta. Dla danego sprzecznego zbioru aksjomatów Z szukamy — za pomocą systemu dowodzenia twierdzeń — sprzecznych podzbiorów Z (o ile takie istnieją), a za pomocą systemu generowania modeli — niesprzecznych podzbiorów Z (o ile takie istnieją). Następnie w rodzinie sprzecznych aksjomatyk znajdujemy minimalne (w sensie relacji $\subseteq$), a w rodzinie niesprzecznych aksjomatyk maksymalne (w sensie relacji $\subseteq$) podzbiory. Przy tym istotne jest, że może istnieć taki zbiór, który nie należy do żadnej z rodzin. Problem decyzyjny, czy zbiór tez wynikających z danego zbioru aksjomatów zawiera wyrażenia sprzeczne, może bowiem mieć złożoność obliczeniową większą niż zasoby użyte w procesie jego rozwiązywania. Proces poszukiwania niesprzecznych podzbiorów sprzecznej teorii może więc dać tylko częściową odpowiedź, pozostawiając niepewność co do statusu niektórych teorii.

Z racji praktycznych trzeba więc uwzględnić wielkość przeszukiwanej rodziny podzbiorów sprzecznego zbioru aksjomatów. Dla TA, która jest przecież tylko niewielkim fragmentem sformalizowanej ontologii orientacji klasycznej, rodzina podzbiorów jej zbioru aksjomatów liczy ponad 60 milionów elementów (dokładnie: 2^{26}). Przyjmując nierealistycznie optymistyczne założenie, że sprawdzenie niesprzeczności każdego takiego zbioru zajmie średnio tylko sekundę, sprawdzenie niesprzeczności wszystkich zajęłoby ponad dwa lata. Oczywiście, nie musimy rozważać wszystkich podzbiorów TA, skoro każdy podzbiór niesprzecznego zbioru aksjomatów jest niesprzeczny, a każdy nadzbiór zbioru sprzecznego jest sprzeczny.

Nie wiedząc jednak z góry, jaki jest rzeczywisty „stan metalogiczny" TA, musimy spróbować ograniczyć liczbę sprawdzanych teorii. W rozważanym przypadku sprawę ułatwia pewna modularność TA. Mianowicie, jak wskazuje Nieznański, aksjomaty A1-A8 (oraz odpowiednie definicje) stanowią aksjomatykę teorii Boole'owskiej dla relacji „$\leq$". Korzystając z jednego z narzędzi dostępnych na http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP, łatwo sprawdzić, że teoria ta jest niesprzeczna, znajdując jej model. Dlatego poszukiwania niesprzecznych podteorii TA możemy ograniczyć do nadteorii tej Boole'owskiej teorii — oznaczmy ją przez BTA. Ponadto do BTA możemy włączyć aksjomat A16, który zawiera jedyne w TA wystąpienie symbolu „E|Z|". Liczba przeszukiwanych zbiorów aksjomatów zmniejsza się więc z 2^{26} do 2^{12}.

Do sprawdzenia niesprzeczności tych podteorii TA użyłem dwóch systemów automatycznego dowodzenia twierdzeń, E oraz CVC4, oraz dwóch systemów automatycznego poszukiwania modeli, MACE4 i Paradox. Aby zautomatyzować proces przeszukiwania rodziny podteorii TA, stworzyłem niewielką aplikację JAVA, której zadaniem jest orkiestracja procesów: (i) sprawdzania niesprzeczności teorii, które jest realizowane (równolegle) przez CVC4 oraz E, (ii) poszukiwania modeli dla teorii, dla których te systemy nie znalazły dowodu sprzeczności; proces (ii) jest (równolegle) realizowany przez MACE4 i Paradox[16].

Proces przeszukiwania nadteorii BTA zawartych w TA sprawdził 196 z 4096 teorii — pozostałe teorie są albo nadteoriami teorii, które zostały zidentyfikowane jako sprzeczne, albo podteoriami teorii, które zostały zidentyfikowane jako niesprzeczne. Okazało się, że 177 spośród tych 196 teorii jest niesprzecznych, tzn. MACE4 lub Paradox znalazły dla nich modele, a 19 jest sprzecznych, tzn. E lub CVC4 udowodniły ich sprzeczność[17].

Wśród 177 niesprzecznych teorii znajduje się sześć teorii maksymalnych (Tabela 3). Podobnie wśród teorii sprzecznych istnieje sześć teorii minimalnych (Tabela 4).

	T_NSP_1	T_NSP_2	T_NSP_3	T_NSP_4	T_NSP_5	T_NSP_6
A1-A8, A16 + definicje	+	+	+	+	+	+
Df.ε	+	+	+	+		+
Df.|Π	+	+		+	+	
Df.|Σ	+	+	+		+	+
A9a	+		+	+	+	+
A9b	+		+	+	+	+
A10a	+		+	+	+	+
A10b						
A11					+	+
A12	+	+	+	+	+	+
A13	+	+	+	+	+	
A14	+	+	+	+	+	+
A15		+	+	+	+	+

Tabela 3. Maksymalne niesprzeczne podteorie TA

[16] Aplikacja wykorzystuje skrypt powłoki napisany przez Benzmüllera i Palea (2014), który umożliwia automatyczne wysyłanie zapytań do serwera http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP.

[17] Log procesu oraz jego wyniki, tj. modele teorii niesprzecznych oraz dowody sprzeczności, są dostępne na: http://metaontology.pl/deliverables/papers/digitalizacja-filozofii-formalnej/.

	T_SP_1	T_SP_2	T_SP_3	T_SP_4	T_SP_5	T_SP_6
A1-A8, A16 + definicje	+	+	+	+	+	+
Df.ε		+	+	+	+	+
Df.|Π		+		+	+	+
Df.|Σ		+			+	+
A9a					+	
A9b						+
A10a		+				
A10b	+					
A11			+	+		
A12						
A13			+			
A14						
A15		+			+	+

Tabela 4. Minimalne sprzeczne podteorie TA

Zestawiając ze sobą uzyskane wyniki, można pokusić się o pewne sugestie dotyczące możliwości usunięcia sprzeczności z TA. W zasadzie każda z teorii wymienionych w Tabeli 3 nadaje się na nową wersję TA. Poprawki wymaga niewątpliwie aksjomat A10b, który już na gruncie samej BTA prowadzi do sprzeczności. Jeżeli wielkość modyfikacji teorii mierzyć liczbą usuniętych aksjomatów, to najbardziej konserwatywną wersją TA byłaby teoria T_NSP_5. Nie zawiera ona jednak definicji Df.ε, która z intuicyjnego punktu widzenia wydaje się podstawowa dla ontologii substancjalnej.

Jeżeli uznamy, że definicje, w szczególności Df.ε, Df.|Π, Df.|Σ, wyrażają fundamentalne intuicje filozoficzne co do definiowanych relacji oraz że z tej racji są bardziej ugruntowane niż same aksjomaty, to ponieważ A11 prowadzi do sprzeczności z definicjami Df.ε i Df.|Π, powinniśmy go zmodyfikować. Dodatkowo, skoro T_NSP_1 i T_NSP_2 są jedynymi maksymalnymi niesprzecznymi teoriami zawierającymi wszystkie definicje oraz ponieważ ich część wspólną stanowią aksjomaty A12-A14, możemy uznać te aksjomaty za „bezpieczny" rdzeń niesprzecznego rozszerzenia BTA. Co więcej, różnica symetryczna między aksjomatykami tych teorii wskazuje nam dwa dalsze kierunki modyfikacji TA. Możemy do owego rdzenia dodać albo A15, albo A9a, A9b i A10a, modyfikując odpowiednio pozostałe aksjomaty. Sprzeczność T_SP_2, T_SP_5 i T_SP_6 dowodzi, że nie możemy jednocześnie obrać obu kierunków: aksjomat A15 jest sprzeczny z każdym aksjomatem A9a, A9b i A10a z osobna.

Jeśli natomiast zrównamy status epistemiczny aksjomatów i definicji, to uzyskane wyniki zdają się wskazywać na jakąś zasadniczą niezgodność między

intuicjami wyrażonymi przez te pierwsze a intuicjami wyrażonymi przez te drugie. Tylko A12-A14 są zgodne z wszystkimi definicjami, pozostałe aksjomaty wykluczają niektóre z nich w sposób określony przez zależności wynikające z przedstawionych już uwag. Sugestie te zakładają oczywiście nienaruszalność BTA jako części TA. Jeżeli będziemy gotowi zmodyfikować BTA, np. przez usunięcie niektórych aksjomatów, to możemy uzyskać niesprzeczne teorie nawet z aksjomatem A10b.

Warto zauważyć, że istotnym czynnikiem wpływającym na wynik automatycznego dowodzenia (*resp.* poszukiwania modelu) jest wybór odpowiedniego systemu dowodzenia twierdzeń (lub generowania modeli) oraz dobór właściwych parametrów dla danego procesu (o ile wybrany system dowodzenia to umożliwia). Wiele systemów dowodzenia twierdzeń dopuszcza bowiem określenie zakresu stosowanych heurystyk dowodowych, które są uważane za podstawowy czynnik efektywności danego algorytmu[18]. W rozważanym tu przypadku wybór omówionych systemów oraz dobór parametrów ich działania był wynikiem analiz wyników kilkunastu innych systemów i innych parametrów. Przykładowo, system PROVER9 sprzężony z systemem MACE4 był dużo mniej efektywny, ponieważ status 110 teorii okazał się nieustalony, tj. PROVER9 nie udowodnił ich sprzeczności, a MACE4 nie znalazł dla nich modeli.

Refleksja nad przedstawionymi tu wynikami digitalizacji filozofii formalnej pozwala dostrzec, że proces ten, przynajmniej jeśli przyjrzymy się dotychczasowym realizacjom, ma charakter pomocniczy względem filozofii formalnej. Jest to raczej przedsięwzięcie z zakresu praktyki czy inżynierii filozofii niż komponent badań teoretycznych. Z tego punktu widzenia podstawową kwestią staje się prakseologiczna ekonomiczność: wydajność i oszczędność, a ściślej pytanie o to, czy digitalizacja filozofii formalnej jest działaniem bardziej ekonomicznym, tj. bardziej wydajnym lub oszczędnym[19], niż uprawianie filozofii formalnej metodą „konwencjonalną", tj. przy użyciu ołówka i kartki papieru. Przy tym w obu wypadkach najbardziej istotnym aspektem porównywania tych działań byłby czas potrzebny do osiągnięcia danego rezultatu, np. znalezienia dowodu pewnego twierdzenia. Obecny stan badań nad możliwością digitalizacji filozofii uniemożliwia podanie empirycznie uzasadnionej odpowiedzi na tak ogólne pytanie. Najprawdopodobniej wiele zależy do rodzaju sformalizowanej teorii (i zdolności dedukcyjnych formalizującego filozofa): digitalizacja niektórych teorii może okazać się stratą czasu, gdy maszyna nie będzie w stanie znaleźć dowodu, którego poszukujemy, a który potrafimy znaleźć samodzielnie. Oczywiście, większość czasu potrzebnego na digitalizację filozofii jest zuży-

[18] Zob. uwagi van Hermelena, Lifschitza i Portera (2008: 61-64) o nieinteraktywnych systemach dowodzenia twierdzeń.

[19] Mam tu na myśli wartości prakseologiczne, o których mówił Kotarbiński (1973: 121-125).

wana na zapis formuł danej teorii w języku systemu dowodzenia twierdzeń, natomiast czas dowodzenia nie jest kosztem (czy, jak pisał Tadeusz Kotarbiński, ubytkiem). Warto więc zauważyć, że dostępność systemów takich jak portal http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP minimalizuje ten koszt. Dzięki nim, korzystając tylko z jednej notacji, możemy użyć kilkudziesięciu różnych systemów (dowodzenia twierdzeń i generowania modeli) do digitalizacji pojedynczej teorii, włączając w to systemy dowodzące twierdzeń w logice drugiego i wyższych rzędów.

W kontekście problemu ekonomiczności digitalizacji filozofii formalnej należy podkreślić, że zarówno wyniki uzyskane przez Zaltę, jak i przedstawione w tym artykule studium przypadku nie obejmują sytuacji, w których dowód wygenerowany przez maszynę byłby zbyt długi lub skomplikowany, aby *można* było go znaleźć metodą konwencjonalną[20]. Niemniej *faktem* pozostaje, że wyniki te zawdzięczamy zastosowaniu systemów automatycznego dowodzenia twierdzeń.

PODSUMOWANIE

Przedstawione dwa przykłady zastosowania systemów automatycznego dowodzenia twierdzeń w filozofii formalnej ilustrują potencjalne korzyści i praktyczne ograniczenia takich przedsięwzięć. Stosując tego rodzaju narzędzia, możemy przy stosunkowo niewielkim nakładzie pracy wskazać usterki tworzonych przez nas teorii — zarówno te mniej istotne, jak wskazana przez Zaltę współzależność przesłanek w dowodzie ontologicznym, jak i te poważniejsze, takie jak sprzeczność. W niektórych przypadkach narzędzia te mogą również dostarczyć nam wskazówek, w jaki sposób usuwać dostrzeżone usterki. Niemniej zarówno teoria złożoności obliczeniowej, jak i praktyczny kontekst implementacji algorytmów generowania dowodów (*resp.* modeli) nakładają istotne ograniczenia na możliwe zyski poznawcze: należy być przygotowanym na to, że używając systemów automatycznego dowodzenia twierdzeń, wiele pytań pozostawimy bez odpowiedzi. Przy obecnym stanie wiedzy i praktyki wykorzystanie technologii informatycznych może wspomóc rozwijanie filozofii formalnej, lecz nie zastąpi wysiłku intelektualnego formalizujących filozofów.

---

[20] Tak jak to miało miejsce w przypadku wspomnianego dowodu o czterech kolorach. Zob. również argumentację Tymoczki (1979), który twierdzi, że dowodu tego nie da się przejrzeć i sprawdzić (*non surveyable*).

BIBLIOGRAFIA

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

Appel K., Haken W. (1989), *Every Planar Map Is Four Colorable*, Providence, RI: AMS.

Beavers A. F. (2011), *Recent Developments in Computing and Philosophy*, „Journal for General Philosophy of Science" 42(2), 385-397.

Benzmüller C., Paleo B. W. (2014), *Automating Gödel's Ontological Proof of God's Existence with Higher-Order Automated Theorem Provers* [w:] *ECAI 2014. 21th European Conference on Artificial Intelligence 18-22 August 2014, Prague, Czech Republic. Proceedings*, T. Schaub, G. Friedrich, B. O'Sullivan (red.), Amsterdam: IOS Press, 93-98.

Davis P. J. (1972), *Fidelity in Mathematical Discourse. Is One and One Really Two?*, „The American Mathematical Monthly" 79(3), 252-263.

Fitelson B., Zalta E. N. (2007), *Steps Toward a Computational Metaphysics*, „Journal of Philosophical Logic" 36(2), 227-247.

Fleuriot J. (2001), *A Combination of Geometry Theorem Proving and Nonstandard Analysis, with Application to Newton's Principia*, Berlin: Springer.

Ganesalingam G., Gowers W. T. (2016), *A Fully Automatic Theorem Prover with Human-Style Output*, „Journal of Automated Reasoning", 1-39.

Garbacz P. (2012), *Prover9's Simplification Explained Away*, „Australasian Journal of Philosophy" 90(3), 585-592.

Garbacz P. (2016), *Metody sztucznej inteligencji w digitalizacji filozofii*, „Roczniki Kulturoznawcze" 7(1), 57-81.

Goczyła K. (2011), *Ontologie w systemach informatycznych*, Warszawa: Exit.

Gustafsson J. E., Peterson M. (2012), *A Computer Simulation of the Argument*, „Synthese" 184(3), 387-405.

van Harmelen F., Lifschitz V., Porter B. (2008), *Handbook of Knowledge Representation*, Amsterdam: Springer.

Kotarbiński T. (1973), *Traktat o dobrej robocie*, Wrocław: Ossolineum.

Leitgeb H. (2013), *Scientific Philosophy, Mathematical Philosophy, and All That*, „Metaphilosophy" 44(3), 267-275.

MacKenzie D. A. (2001), *Mechanizing Proof. Computing, Risk, and Trust inside Technology*, Cambridge, MA: MIT Press.

MacKenzie D. (2005), *Computing and the Cultures of Proving*, „Philosophical Transactions of the Royal Society A" 363(1835), 2335-2550.

McCune W. (1997), *Solution of the Robbins Problem*, „Journal of Automated Reasoning" 19(3), 263-276.

Megill J., Linford D. (w druku), *On Computable Metaphysics. On the Uses and Limitations of Computational Metaphysics* [w:] *Ontology of Theistic Beliefs. Meta-Ontological Perspectives*, M. Szatkowski (red.), Berlin: De Gruyter.

Meikle L., Fleuriot J. (2003), *Formalizing Hilbert's Grundlagen in Isabelle/Isar Theorem* [w:] *Proving in Higher Order Logics. 16th International Conference, TPHOLs 2003*, D. Basin, B. Wolff (red.), Berlin: Springer, 319-334.

Nieznański E. (2007), *Sformalizowana ontologia orientacji klasycznej*, Warszawa: Wydawnictwo UKSW.

Oppenheimer P. E., Zalta E. N. (1991), *On the Logic of the Ontological Argument*, „Philosophical Perspectives" 5, 509-529.

Oppenheimer P. E., Zalta E. N. (2011), *A Computationally-Discovered Simplification of the Ontological Argument*, „Australasian Journal of Philosophy" 89(2), 333-349.

Portoraro F. (2014), *Automated Reasoning* [w:] *The Stanford Encyclopedia of Philosophy* (Winter 2014 Edition), E. N. Zalta (red.), https://goo.gl/gUwlUn.

Suppes P. (1968), *The Desirability of Formalization in Science*, „The Journal of Philosophy" 65(20), 651-664.

Świętorzecka K. (2008), *Sformalizowana ontologia orientacji klasycznej, Edward Nieznański, Warszawa 2007* [recenzja], „Studia Philosophiae Christianae" 44(1), 207-212.

Tymoczko T. (1979), *The Four-Color Problem and Its Philosophical Significance*, „The Journal of Philosophy" 76(2), 57-83.

Zalta E. (1983), *Abstract Objects. An Introduction to Axiomatic Metaphysics*, Dordrecht: D. Reidel.