Digitalizacja filozofii formalnej*

Czy jest możliwe zbudowanie sztucznego filozofa, elektronicznego golema, który byłby w stanie prowadzić mniej lub bardziej abstrakcyjne rozważania o bycie i niebycie, początkowo naśladując zachowania naturalnych filozofów, a w dalszej perspektywie przezwyciężając ich ograniczenia poznawcze? Ten raczej mętny problem można łatwo uściślić ograniczając pole możliwości do tych, które są osadzone w aktualnym stanie badań: filozoficznych i informatycznych. Przy takich ograniczeniach wydaje się, że 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]

Pomimo żywotności paradygmatu logicznego w uprawianiu filozofii, wykorzystanie komputerów jako narzędzia dowodzenia formułowanych tam twierdzeń jest w zasadzie nieznane – zob. np. stan badań przedstawiony w (Beavers 2011) czy (Portoraro 2014). Znane są oczywiście pewne teoretyczne ograniczenia takich zastosowań, na przykład twierdzenie o nierozstrzygalności klasycznego rachunku logicznego. Poza dwoma wyjątkami, z których jeden omawianym w sekcji drugiej poniżej, brak jest jednak bardziej praktycznie zorientowanych przedsięwzięć badawczych, które ilustrowałyby racjonalność (lub jej brak) tego rodzaju automatyzacji za pomocą konkretnych przykładów.[2] Prawdą jest, iż automatyzacja dowodzenia twierdzeń nie jest zbyt popularna nawet w samej matematyce, niemniej uzyskane tam rezultaty powinny zachęcić zwolenników filozofii formalnej do bardziej odważnego eksperymentowania. Formalizacja jest często postrzegana przez jej zwolenników jako zabieg podnoszący wartość poznawczą formalizowanych treści, niezależnie od jej metodologicznej proweniencji, (Suppes

---

* Publikacja została sfinansowana 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 w (Gustafsson i Peterson 2012) oraz wykorzystanie narzędzi informatycznych do analiz statystycznych w filozofii eksperymentalnej.

[2] Stwierdzenie to ignoruje zastosowania systemów dowodzenia twierdzeń w tzw. inżynierii ontologii – zob. np. (Goczyła 2011) – gdyż metateoretyczny kontekst tworzonych tam artefaktów informatycznych jest odmienny od kontekstu dociekań filozoficznych.

1968), a w skrajnej wersji tego poglądu, jako podstawowa metoda „unaukowienia” filozofii (Leitgeb 2013). W takiej perspektywie digitalizację filozofii formalnej można postrzegać jako kolejny krok w kierunku „melioracji epistemicznej” tej dziedziny wiedzy.

Faktycznie budowane formalizmy, nieco wbrew głoszonym manifestom, często zawierają mniej lub bardziej poważne usterki.[3] Współczesne systemy wspierające dowodzenie twierdzeń okazały się zdolne do identyfikacji takich błędów – najbardziej znane przykłady takich sukcesów dotyczą błędów w dowodzie praw Keplera podanego przez I. Newtona (Fleuriot 2001) czy braków w D. Hilberta formalizacji geometrii (Meikle i Fleuriot 2003). W tego rodzaju przypadkach 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 i Haken 1989)[4] czy mniej słynnego 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. Takie przypadki są jednak, w świetle wiedzy o dotychczasowych zastosowaniach systemów dowodzenia twierdzeń, bardzo rzadkie. (MacKenzie 2005) wymienia tylko jeden znany (mu) przypadek błędu tego rodzaju.[5]

Mało prawdopodobna wydaje się możliwość, że wytwory filozofii formalnej wolne są od usterek formalnych lub że formalizujący filozofowie są w stanie dowodzić twierdzeń bardziej efektywnie niż matematycy czy logicy. „Urobek epistemiczny” filozofii formalnej jest co prawda dużo mniejszy niż rezultaty czystej matematyki czy logiki, jednak w chwili obecnej nie jest on zaniedbywalnie mały. Z tych racji dziwi i niepokoi stan badań nad digitalizacją filozofii formalnej, a raczej jego brak. W tym artykule chciałbym przedstawić jedno z dwóch znanych mi dokonań w tym zakresie oraz przedstawić swój własny przyczynek, które (owo dokonanie i ów przyczynek), mają ilustrować możliwe zyski poznawcze, jakie można na tej drodze osiągnąć, jak i przeszkody, jakie można w tym procesie napotkać. [6]

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

[4] Ściślej, pewien fragment dowodu twierdzenia o czterech kolorach został wykonany przez maszynę cyfrową z racji długości obliczeń potrzebnych do jego wykonania.

[5] Interesujący może być fakt, iż ów błąd dotyczył dowodu wspomnianego wcześniej twierdzenia dotyczącego algebry Robbinsa, który został wygenerowany przez system REVEAL - zob. (MacKenzie 2001, 289-290).

[6] (Megill i Linford (w druku)) argumentują, iż każdy system filozoficzny, włączając w to wszystkie *możliwe* systemy filozoficzne, jest w tym sensie digitalizowalny.

Typy digitalizacji formalizacji

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

1. definicja języka teorii formalnej
2. wybór uznanych wyrażeń tego języka i/lub reguł inferencji, za pomocą, na przykład, aksjomatyzacji
3. dowodzenie twierdzeń
4. konstrukcja metateorii tej teorii, np. budowa modelu.

Obecnie rozwijane technologie informatyczne wspierające badania w naukach formalnych koncentrują się na trzecim oraz w pewnym stopniu na 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. dostarczenie infrastruktury do tworzenie struktur formalnych, w szczególności wsparcie dla logików czy matematyków przy budowaniu dowodów i modeli;
2. automatyczne dowodzenie twierdzeń wskazanych przez człowieka lub znajdowanie modeli dla wskazanych zbiorów formuł;
3. automatyczne generowanie twierdzeń oraz ich dowodów.

W obrębie typu czy poziomu pierwszego znajdują się te wszystkie programy komputerowe, które ułatwiają tworzenie przez człowieka teorii formalnych, w szczególności ułatwiają konstruowanie dowodów twierdzeń i sprawdzanie ich poprawności – 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. jego zgodności ze zbiorem reguł wnioskowania i zbiorem uznanych wyrażeń.

Typ (poziom) drugi digitalizacji obejmuje te systemy, które poszukują dowodów lub modeli twierdzeń wskazanych przez człowieka i określają te metalogiczne własności teorii formalnych, które wynikają z takich poszukiwań, np. niesprzeczność. Typ ten stawia przed maszyną więcej wymagań niż poprzedni, gdyż znalezienie (poprawnego) dowodu wymaga więcej kreatywności od sprawdzenia poprawności znalezionego dowodu.

Ostatni poziom digitalizacji sprowadza się w zasadzie do pełnej automatyzacji procesu dowodzenia twierdzeń. Maszyna nie tylko podpowiada człowiekowi, jak można dowodzić, ale i co można udowodnić – w granicach zadanej przez człowieka aksjomatyzacji. [7]

Przedstawiony wyżej podział nie jest podziałem logicznym, lecz dość luźną typologią, która dopuszcza istnienie paradygmatycznych przypadków, przypadków granicznych, itd.

Jeżeli chodzi o zastosowania systemów wspierających dowodzenie twierdzeń w filozofii (formalnej), to wszystkie istniejące zastosowania ograniczają się do tych z poziomu drugiego. Ściślej, znane mi przypadki digitalizacji filozofii formalnej obejmują tylko dwa przedsięwzięcia:

1. digitalizacje różnych wersji formalizacji dowodu ontologicznego K. Gödla realizowane przez Christopha Benzmüllera i Bruna Woltzenlogla Paleo – zob. np. (Benzmuller i Paleo 2014);
2. metafizykę obliczeniową.

Ponieważ owo drugie przedsięwzięcie jest dużo bardziej rozbudowane niż pierwsze, w tym artykule skoncentruję się właśnie na nim jako na przykładzie pokazującym możliwe zyski z digitalizacji filozofii formalnej.

Najpierw jednak chciałbym wprowadzić pewne uściślenie terminologiczne. Podstawowe znaczenie terminu „digitalizacja” w dalszej części tego artykułu 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 dowodów lub modeli dla formuł z języka tej teorii, które są wskazane przez człowieka, w szczególności procesu sprawdzenia niesprzeczności teorii;
4. metalogiczna interpretacja uzyskanych rezultatów.

[7] (Megill i Linford (w druku)) wskazują również na możliwość automatycznego generowania nowych aksjomatyk. Przedstawiona przez nich metoda sprowadza się tworzenia nowych teorii z kombinacji 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 istotnie nowatorskich.

Digitalizacja metafizyki obliczeniowej

Metafizyka obliczeniowa jest wieloletnim projektem badawczym realizowanym przez E. Zaltę (i jego współpracowników), którego celem jest „implementacja i badania nad sformalizowaną aksjomatyką w środowisku automatycznego dowodzenia twierdzeń" – zob. dokumentację projektu dostępną na http://mallystanfordedu/cm/ (link aktywny 4. marca 2016 r.).

W ramach tego projektu „zdigitalizowano" centralne fragmenty metafizyki przedmiotów abstrakcyjnych – zob. (Zalta 1983) – rozwijanej od wielu lat przez E Zaltę, w tym: platońską teorię idei, teorię światów możliwych, oraz leibnizjańską teorię pojęć.[8] Nieco niezależnie do tych badań przeprowadzono digitalizację jednej z wersji dowodu ontologicznego na istnienie Boga. W tym miejscu chciałbym bardziej szczegółowo przedstawić wyniki tego ostatniego przedsięwzięcia – pozostałe rezultaty wymagałaby bowiem zbyt obszernych wyjaśnień dość idiosynkratycznych formalizmów budowanych przez Zaltę.

W roku 1991 P. Oppenheimer i E. 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 – zob. (Oppenheimer i Zalta 1991). Przedstawiona w tym artykule formalizacja była oparta na trzech przesłankach:

$$\exists x \, \mathrm{Sup_G}(x). \qquad (\textit{Przesłanka 1})$$

$$\neg \mathrm{E}(\iota x \mathrm{Sup_G}(x)) \rightarrow \exists y[\mathrm{G}(y, \iota x \mathrm{Sup_G}(x)) \wedge \mathrm{C}(y)]. \qquad (\textit{Przesłanka 2})$$

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

gdzie:

- "$\mathrm{C}(x)$" to tyle, co „można pomyśleć x";
- „$\mathrm{G}(x, y)$" to tyle, co „x jest większy niż *y*";
- „$\mathrm{E}(x)$" to tyle, co „*x* istnieje";
- a „$\mathrm{Sup_G}$" 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 powyższych trzech przesłanek wynika logicznie zdanie

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

„ $E(\iota x \mathrm{Sup}_G(x))$ ”, głoszące, iż 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, mianowicie dowodzą, że zdanie „ $E(\iota x \mathrm{Sup}_G(x))$ ” wynika logicznie z *Przesłanki 2*, a pozostałe dwie przesłanki są w tym dowodzie zbędne (Oppenheimer i Zalta 2011). Spostrzeżenie to „zawdzięczają” automatycznemu systemowi dowodzenia twierdzeń PROVER9. Zapisanie powyższych przesłanek w tym systemie 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, które nie narusza jego konkluzywności, jest zwiększeniem jego wartości poznawczej, zatem wynik uzyskany w (Oppenheimer i Zalta 2011) pokazuje, jak zastosowanie dość prostego narzędzia informatycznego przyczyniło się do ulepszenia jednej z bardziej doniosłych egzemplifikacji argumentacji filozoficznej. Oczywiście, spostrzeżenie, którego dokonali Oppenheimer i Zalta przy użyciu PROVER9, jest dostępne również dla bardziej tradycyjnych metod – zob. w tej sprawie (Garbacz 2012). Niemniej, spostrzeżenie to przez dwadzieścia lat umykało uwadze czytelników (Oppenheimer i Zalta 1991).

Natomiast zastosowanie digitalizacji do teorii przedmiotów abstrakcyjnych pozwoliło na odkrycie niekonkluzywności jednego z dowodów: dla 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 jest – zob. (Fitelson i Zalta 2007, 241-242).

Podsumowując, przejście ze stadium aksjomatycznego (abstrakcyjnego) do stadium zdigitalizowanego w rozwoju teorii formalnej może wiązać się z poprawą jakości poznawczej tej teorii. Uzyskane korzyści można uważać za niewielkie, lecz nie są one niezaniedbywalnie małe. Czasami jednak digitalizacja może doprowadzić do bardziej istotnych spostrzeżeń.

Digitalizacja sformalizowanej ontologii orientacji klasycznej

W tej części artykułu chciałbym przedstawić studium przypadku, w którym różnica pomiędzy wartością teorii formalnej w stadium niezdigitalizowanym a jej wartością w stadium zdigitalizowanym jest dużo większa i poznawczo bardziej doniosła.

Przedmiotem rozważań będzie tzw. sformalizowana ontologia orientacji klasycznej przedstawiona w (Nieznański 2007). Jak pisze w swojej recenzji K 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 powtarzam, iż ogół przedstawionych tam rozważań składa się 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. Przeprowadzone przeze mnie studium przypadku digitalizacji filozofii formalnej dotyczy tylko części tej obszernej teorii, mianowicie podteorii sformułowanej w rozdziale pierwszym zatytułowanym *Aliquid* – dla uproszczenia będę się do niej odnosił za pomocą skrótu „TA".

Nieznański wprowadza w tym rozdziale szesnaście aksjomatów charakteryzujących znaczenia jednego predykatu:

- „$\leq$" – gdzie „$x \leq y$" czytamy „(istota) x jest zawarta w (istocie) y"

oraz czterech symboli funkcji:

- „Σ" – gdzie „Σx" czytamy „pewien x",
- „Π" – gdzie „Πx" czytamy „każdy x",
- „|" – gdzie „$x|y$" czytamy „x od y-a", tak jak np. w „niewolnik (od) pana", „ster (od) łodzi", itp.
- „E|Z|" – gdzie „E|Z|x" czytamy „element zbioru x-ów".[9]

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

[9] (Nieznański 2007) nie zawiera jawnej definicji języka formalizowanej ontologii. Podane w tekście głównym stwierdzenie są oparte na funkcjach składniowych tych symboli w formułach i dowodach tam podanych.

(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) $\Pi x \leq y \Leftrightarrow x \leq y$[10]

(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) $x \varepsilon y|(u|w) \Rightarrow \exists z\ (x \varepsilon y|z \wedge z \varepsilon u|w)$[11]

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

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

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) $x \equiv 0 \Leftrightarrow \forall y\ x \leq y$

---

[10] 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 dotycz aksjomatów A10a i A10b. Stąd owe „szesnaście” aksjomatów jest tu zapisane za pomocą osiemnastu formuł.

[11] 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 byłaby:

(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.

Przy okazji chciałbym jednak zauważyć, iż sposób prezentowania 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.

(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:

1. symbole „Σ” i „Π” zostały wprowadzone „kilkadziesiąt formuł” wcześniej, na s 35, w sposób, jak pisze Autor, „aksjomatyczny”;
2. definicje Df.|Π i Df.|Σ są definicjami cząstkowymi tych symboli, bo 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 wyżej definicjach.

Przeprowadzona przeze mnie digitalizacja TA 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 dostępnych na stronie: http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP.

```
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))))).
```

Zapis TA w notacji TPTP

Tabela 1 podaje symbole formalne w obu notacjach:

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

Tabela 1 Notacja TPTP

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

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

Tabela 2 Specyficzne symbole w notacji PROVER9

```
% 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)].
```

Dowód sprzeczności TA

Warto zauważyć, iż znaleziony dowód pokazuje stosunkowo niewielką część TA, która jest sprzeczna – sprzeczna jest mianowicie teoria złożona z aksjomatów A1, A2, A7, A8, A10b i definicji Df.1 oraz D.$\equiv$. Analiza dowodu prowadzi do wniosku, że najbardziej problematyczny jest A10b. Ponieważ z aksjomatu A1 wynika, że relacja $\leq$ jest zwrotna, więc $\equiv$ 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, A10b daje nam, iż $x \equiv 1$ (dla dowolnego x), co wobec definicji $\equiv$ jest sprzeczne z A8.

Niemniej usunięcie A10b z TA nie usuwa sprzeczności ontologii, co pokazuje dowód podany na następnej stronie: aksjomaty A1, A2, A5, A9a, A11, A13, A15 oraz definicje Df.$\equiv$ i Df.ε również 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)].
```

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, 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 na możliwe sposoby uniknięcia wykrytej sprzeczności.

Abstrahując od kwestii implementacyjnych realizacja takiego zadania jest prosta. Dla danego sprzecznego zbioru aksjomatów:

- za pomocą systemu dowodzenia twierdzeń szukamy jego sprzecznych podzbiorów (o ile takie istnieją);
- za pomocą systemu generowania modeli szukamy jego niesprzecznych podzbiorów (o ile takie istnieją);

Następnie w rodzinie sprzecznych aksjomatyk znajdujemy minimalne (w sensie relacji $\subseteq$), a w rodzinie niesprzecznych aksjomatyk maksymalne podzbiory (w sensie relacji $\subseteq$). Przy tym, istotne jest, iż 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 sprawdzania 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 jedną 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 również sprzeczny.

Nie wiedząc jednak z góry, jaki jest faktyczny „stan metalogiczny” TA, musimy spróbować ograniczyć liczbę sprawdzanych teorii. W rozważanym przypadku sprawę ułatwia pewna modularyzacja TA. Mianowicie, jak wskazuje Nieznański, aksjomaty A1-A8 (oraz odpowiednie definicje) stanowią aksjomatykę teorii Boole'owskiej dla relacji „≤". 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 owej 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;
- 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 są realizowane (równolegle) przez CVC4 oraz E (ii) poszukiwania modeli dla teorii, dla których te systemy nie znalazły dowodu sprzeczności, który to proces jest (równolegle) realizowany przez MACE4 i Paradox.[12]

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. Wśród tych 196 teorii:

1. 177 okazało się niesprzeczne, tzn. MACE4 lub Paradox znalazł dla nich modele;
2. 19 okazało się sprzeczne, tzn. E lub CVC4 udowodnił ich sprzeczność.[13]

Wśród owych 177 niesprzecznych teorii znajduje się sześć teorii maksymalnych – zob. tabelę 3. Podobnie wśród teorii sprzecznych istnieje sześć teorii minimalnych – tabela 4.

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

[13] Log procesu oraz jego wyniki są dostępne na: http://metaontology.pl/deliverables/papers/digitalizacja-filozofii-formalnej/.

	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

	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 co do możliwości usunięcia sprzeczności z TA.

W zasadzie każda z teorii wymienionych w tabeli 3 nadaje się na poprawioną wersję TA. W szczególności, 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. Teoria ta nie zawiera jednak definicji Df.ε, która, z intuicyjnego punktu widzenia, wydaje się fundamentalna dla ontologii substancjalnej.

Jeżeli uznamy, że definicje, w szczególności Df.ε, Df.$|\Pi$, Df.$|\Sigma$, wyrażają fundamentalne intuicje filozoficzne co do definiowanych relacji oraz że, z tej racji, są bardziej ugruntowane niż same aksjomaty, to:

1. Ponieważ A11 prowadzi do sprzeczności z definicjami Df.ε i Df.$|\Pi$, powinniśmy go zmodyfikować.
2. Ponieważ teorie 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.
3. Różnica symetryczna pomiędzy aksjomatykami tych teorii wskazuje nam dwa dalsze kierunki modyfikacji TA. Możemy mianowicie 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.

Jeżeli natomiast zrównamy status epistemiczny aksjomatów i definicji, to możemy jedynie powiedzieć, iż uzyskane wyniki zdają się wskazywać na jakąś zasadniczą niezgodność pomiędzy 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 powyższych uwag.

Sugestie te zakładają oczywiście nienaruszalność BTA jako części TA. Jeżeli będziemy gotowi zmodyfikować BTA, np. przez usunięcie pewnych aksjomatów, możemy uzyskać niesprzeczne teorie nawet z aksjomatem A10b.

Warto zauważyć, iż 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 umożliwia bowiem określenie zakresu stosowanych heurystyk dowodowych, które są uważane za kluczowy czynnik efektywności danego algorytmu – zob. uwagi o nieinteraktywnych systemach dowodzenia twierdzeń w (van Harmelen, Lifschitz i Porter 2008, 61-64). W rozważanym tu przypadku wybór wymienionych wcześniej systemów oraz dobór parametrów ich działania był wynikiem analiz wyników z kilkunastu wcześniejszych prób wyboru innych systemów i innych parametrów. Przykładowo, wybór systemu PROVER9 sprzężonego z systemem MACE4 okazał się dużo mniej efektywny, gdyż status 110 teorii okazał się nieustalony, tj. PROVER9 nie udowodnił ich sprzeczności, a MACE4 nie znalazł dla nich modeli.

Reflektując nad przedstawionymi tu wynikami digitalizacji filozofii formalnej łatwo dostrzec, że proces ten, przynajmniej w zaistniałych do tej pory egzemplifikacjach, ma charakter pomocniczy względem rozwoju filozofii formalnej. Jest to więc raczej przedsięwzięcie z zakresu praktyki czy inżynierii filozofii formalnej niż komponent badań teoretycznych. Z tego punktu widzenia kluczowa staje się kwestia jego prakseologicznej ekonomiczności: wydajności i oszczędności, a ściślej pytanie o to, czy digitalizacja filozofii formalnej jest działaniem bardziej ekonomicznym, tj. bardziej wydajnym lub oszczędnym[14], niż uprawianie filozofii formalnej „konwencjonalną metodą”, tj. przy użyciu ołówka i kartki papieru. Przy tym, w obu przypadkach najbardziej istotnym aspektem porównywania tych działań byłby czas potrzebny na osiągnięcie danego rezultatu, np. znalezienie 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 jesteśmy w stanie znaleźć „metodą konwencjonalną”. Oczywiście, większość czasu potrzebnego na digitalizację filozofii jest „zużywana” na zapis formuł danej teorii w języku systemu dowodzenia twierdzeń, natomiast czas dowodzenia nie jest kosztem (czy, jak pisał, T. Kotarbiński, ubytkiem). Warto więc zauważyć, iż dostępność systemów 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, iż zarówno wyniki uzyskane przez Zaltę, jak i przedstawione w tym artykule studium przypadku, nie obejmują

[14] Mam tu na myśli wartości prakseologiczne, o których mówi T. Kotarbiński w (Kotarbiński 1973, 121-125).

takiego przypadku, w którym dowód wygenerowany przez maszynę byłby zbyt długi lub skomplikowany, aby *można* było go znaleźć „metodą konwencjonalną”.[15] Niemniej, *faktem* pozostaje, iż wyniki te zawdzięczamy zastosowaniu systemów automatycznego dowodzenia twierdzeń.

Podsumowanie

Mam nadzieję, że przedstawione w tym artykule 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, zidentyfikować usterki tworzonych przez nas teorii – zarówno te mniej istotne, jak wskazana przez E. Zaltę współzależność przesłanek w dowodzie ontologicznym, jak i te bardziej poważne, takie jak sprzeczność. W niektórych przypadkach narzędzia te mogą również dostarczyć nam wskazówek co do sposobów usunięcia tych usterek. 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 można powiedzieć, iż wykorzystanie technologii informatycznych może wspomóc rozwijanie filozofii formalnej, lecz nie jest w stanie zastąpić intelektualnego wysiłku formalizujących filozofów.

[15] Tak jak to miało miejsce w przypadku wspomnianego dowodu o czterech kolorach - zob. również (Tymoczko 1979), który twierdzi, że ów dowód jest nieweryfikowalny (*non surveyable*).

Bibliografia

Alama, Jesse, Paul E. Oppenheimer, i Edward N. Zalta. 2015. "Automating Leibniz's Theory of Concepts." *Proceedings of the 25th International Conference on Automated Deduction.* Dordrecht: Springer. 73-97.

Appel, Kenneth, i Wolfgang Haken. 1989. *Every Planar Map is Four Colorable.* Providence: AMS.

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

Benzmuller, Christoph, i Bruno Woltzenlogel Paleo. 2014. "Automating Godel's Ontological Proof of God's Existence with Higher-order Automated Theorem Provers." (red.) Torsten Schaub, Gerhard Friedrich i Barry O'Sullivan. *ECAI 2014.* 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, Branden, i Edward N. Zalta. 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.

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

Goczyła, Krzysztof. 2011. „Ontologie w systemach informatycznych". Warszawa. Exit

Gustafsson, Johan E., i Martin Peterson. 2012. "A computer simulation of the argument." *Synthese* 184 (3): 387-405.

Kotarbiński, Tadeusz. 1973. *Traktat o dobrej robocie.* V. Wrocław: Ossolineum.

Leitgeb, Hannes. 2013. "Scientific Philosophy, Mathematical Philosophy, And All That." *Metaphilosophy* 44 (3): 267-275.

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

MacKenzie, Donald. 2005. "Computing and the cultures of proving." *Philosophical Transactions of the Royal Society A* 363: 2335-2550.

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

Megill, Jason, i Daniel Linford. [w druku] . “On Computable Metaphysics: On the Uses and Limitations of Computational Metaphysics.” [w:] Ontology of Theistic Beliefs: Meta-Ontological Perspectives, (red.) M. Szatkowski. De Guyter.

Meikle, L., i J. Fleuriot. 2003. “Formalizing Hilbert’s Grundlagen in Isabelle/Isar Theorem.” In *Proving in Higher Order Logics: 16th International Conference, TPHOLs 2003*, 319–334. Berlin: Springer.

Nieznański, Edward. 2007. *Sformalizowana ontologia orientacji klasycznej.* Warszawa: Wydawnictwo UKSW.

Oppenheimer, Paul E., i Edward N. Zalta. 2011. “A Computationally-Discovered Simplification of the Ontological Argument.” *Australasian Journal of Philosophy* 89 (2): 333-349.

Oppenheimer, Paul E., i Edward N. Zalta. 1991. “On the Logic of the Ontological Argument.” *Philosophical Perspectives* 5: 509-529.

Portoraro, Frederic. 2014. “Automated Reasoning.” Edited by Edward N. Zalta. Winter. http://plato.stanford.edu/archives/win2014/entries/reasoning-automated/.

Suppes, Patrick. 1968. “The Desirability of Formalization in Science.” *The Journal of Philosophy* 65 (20): 651-664.

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

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

van Harmelen, Frank, Vladimir Lifschitz, i Bruce Porter. 2008. *Handbook of Knowledge Representation.* Amsterdam: Springer.

Zalta, Edward. 1983. *Abstract Objects: An Introduction to Axiomatic Metaphysics.* Dordrecht: D. Reidel Publishing Company.