WITOLD MARCISZEWSKI

O rachunku zdań w perspektywie informatycznej

Uzupełnienie do rozdziału III – *Klasyczny rachunek zdań. Świat funkcji prawdziwościowych.*

1. Miejsce rachunku zdań w całości logiki

1.1. Irlandczycy mają godne uwagi przysłowie, że kiedy Bóg stwarzał czas, stworzył go pod dostatkiem, nie ma więc obawy, że go zabraknie. Coś podobnego da się powiedzieć o teoriach logicznych; stworzono ich pod dostatkiem, stąd nie ma obaw, że zabraknie nam reguł do rozumnego kierowania inteligencją (jak trafnie nazwał Kartezjusz rolę logiki w tytule swego traktatu *Regulae ad directionem ingenii*). Kierować inteligencją to przede wszystkim tak kontrolować rozumowania, żeby nie rozumować niepoprawnie, to znaczy, nie rozumować w sposób, który by od prawdy prowadził do fałszu.

Da się naliczyć kilka tuzinów teorii logicznych, z których każda ujmuje jakąś część tego rodzaju reguł. Mamy reguły dotyczące rozumowań o możliwości i konieczności (logika modalna), reguły dotyczące rozumowań o tym, co obowiązujące, co dozwolone itp. (logika deontyczna, pożyteczna w prawie i etyce), reguły uwzględniające prócz prawdy i fałszu wartości pośrednie (rozległa klasa logik wielowartościowych) itd. Da się jednak w tej wielości wyodrębnić główny trzon, którego znajomość jest konieczna, żeby zgłębiać inne logiki oraz (na szczęście) jest wystarczająca, żeby sobie radzić z kontrolą rozumowań najczęściej występujących w naukach i w życiu codziennym.

Główny trzon to **klasyczna logika pierwszego rzędu**. Klasyczna znaczy mniej więcej to samo, co stanowiąca ów trzon główny. A co znaczy „pierwszego rzędu” da się wyjaśnić dopiero po jej wyłożeniu (logiki wyższych rzędów wydatnie zasilają to bogactwo logik, o którym była wyżej mowa).

Na klasyczną logikę pierwszego rzędu składają się dwie teorie: logika zdań i logika predykatów. Nazwy te są dobrane w zależności od tego, która z kategorii syntaktycznych brana jest pod uwagę w regułach rozumowania czyli wnioskowania teorii (używamy raczej terminu „reguły wnioskowania” ze względu na zgodność brzmieniową ze słowem „wniosek”, oznaczającym wynik wnioskowania).

Zamiast „logika” mówimy często w obu przypadkach „rachunek”. W obecnym jednak wykładzie różnicuje się tę terminologię, jak następuje. Termin „rachunek” stosowany jest tylko w kontekście nazwy „Rachunek zdań”, dla drugiej zaś z rozważanych teorii przyjmuje się nazwę „Logika predykatów” (nie zaś „Rachunek predykatów”). Bierze się to z nastawienia na problematykę algorytmów, Mianowicie, słowo „rachunek”, gdy brać je w najściślejszym sensie, oznacza operowanie algorytmami, co w pełni się realizuje tylko w rachunku zdań.

Logika predykatów natomiast, jak to okazało się dopiero po pół wieku od jej powstaniu, nie jest w pełni algorytmiczna. Jest to odkrycie, które zaskoczyło cały świat naukowy, a jest tak ważne dla logiki. informatyki, teorii poznania, filozofii nauki i filozofii umysłu, że zasługuje na nieustanne przypominanie, także za pomocą odpowiedniej terminologii. Unikając określania logicznej teorii predykatów mianem rachunku, przypominamy w ten sposób doniosłe odkrycie niedoboru algorytmiczności (rachunkowości) w logice predykatów.

1.2. Druga część tytułu w rozdziale o rachunku zdań, brzmiąca „Świat funkcji prawdziwościowych”, również wiąże się z zagadnieniem algorytmiczności. Ma ona przypominać coś co można oddać pytaniem: co widziała Alicja (ta z krainy czarów) po drugiej stronie lustra? Algorytm sam z siebie nic nie mówi o jakiejkolwiek rzeczywistości, jest to twór czysto syntaktyczny, w którym określone są sposoby mechanicznego przekształcania ciągów symboli. Ale konstruujemy go z rozmysłem w taki sposób, żeby odzwierciedlał interesującą nas żywo rzeczywistość; w owym systemie symboli ma się przeglądać jakiś świat. Tylko wtedy będzie on przydatny w poznawaniu i przekształcaniu świata. Tak np. działania arytmetyczne wykonywane na symbolach w algorytmach dodawania, mnożenia etc. są wiernym odwzorowaniem odpowiednich relacji zachodzących w świecie liczb.

Z tych to powodów, naturalnym dopełnieniem rozważań o algorytmach są rozważania o świecie, który się w nich „odbija”. Dla rachunku zdań jest to świat funkcji prawdziwościowych. W tym miejscu przyda się uwaga, że należy się oswoić z używaniem słowa „świat” także w odniesieniu do dziedzin wysoce abstrakcyjnych, jakimi są, w szczególności, różne dziedziny liczbowe

Dziedzina, którą oddaje język rachunku zdań składa się z dwóch obiektów abstrakcyjnych zwanych prawdziwością (krócej, prawdą) i fałszywością (fałszem) oraz funkcji określonych na tych obiektach i przybierających jako swe wartości prawdę lub fałsz. Nazywamy je **funkcjami prawdziwościowym**. Takie odwzorowanie mając na widoku, konstruujemy język dla algorytmów w sposób opisany w następnym ustępie.

2. Algorytmy budowania i rozpoznawania formuł rachunku zdań

2.1. Określenie „formuła” stosujemy do wyrażeń z kategorii zdaniowej zawierających symbole zmienne. W literaturze logicznej można się spotkać z podziałem formuł na poprawnie zbudowane (ang. *well-formed formulae*) i pozostałe. Można też – co czyni się w obecnym tekście – przyjąć, że jeśli jakieś zestawienie symboli nie jest poprawnie zbudowane, to nie jest formułą, podobnie jak nie jest budowlą ani to, co jest dopiero w toku budowy, ani to, co jest ruiną dawnej budowli.

> Mamy tu do czynienia w odwieczną w języku dwuznacznością wymagającą wyboru, czy wartościowanie pozytywne zawrzeć w samej nazwie, czy oddać je dopiero dołączoną do nazwy przydawką. Podobnie jest ze słowami „pogoda”, „metoda” i wieloma innymi. W jednych kontekstach „pogoda” znaczy tyle, co „dobra pogoda” (np. w tytule serialu „Pogoda dla bogaczy”), w innych zaś mówimy o złej pogodzie, a więc prześlizgujemy się do innego sensu. A gdy recenzent zarzuca pracy magisterskiej brak metody, mając na myśli brak dobrej metody, to używa tego terminu w sensie wartościującym pozytywnie, nie zaś neutralnym. Brak jest, z kolei, takiego pozytywnego momentu wtedy, gdy mówi się o złej metodzie, dzieląc tym samym metody na złe i dobre.

Żeby uchwycić rys algorytmiczny w budowaniu formuł, porównajmy je z rozwiązywaniem łamigłówki-skladanki (ang. *puzzle*). To drugie wymaga pomysłowości, wymaga też wiedzy o tym, jak wygląda konstruowany obiekt, a dokonuje się metodą prób i błędów. Żadnej z tych trzech rzeczy nie ma w algorytmie.

Do istoty algorytmu należy to, że obiekty, na których dokonuje się przekształceń są niezawodnie rozpoznawalne co do kształtu. Drugą cechą istotną jest to. że w każdym z kolejnych kroków dokładnie wiadomo, którą należy zastosować spośród reguł (tj. przepisów postępowania) składających się na dany algorytm (czego nie ma w „puzzlu”). Zobaczmy, jak spełnia te warunki algorytm syntaktyczny tworzenia formuł rachunku zdań.

2.2. W rachunku zdań obiektami przekształceń podstawowymi, to znaczy, danymi w punkcie wyjścia są litery przeznaczone do tego, żeby symbolizować dowolne zdania. Istotnie, jak wymaga sie algorytmie, są one łatwo i niezawodnie rozpoznawalne. Nazywamy je **zmiennymi zdaniowymi** i traktujemy jako **wyrażenia atomowe** rachunku zdań, z nich bowiem jak z atomów buduje się wszelkie inne wyrażenia tego języka.

Najczęściej przyjmuje się w tej roli literę „p” (kojarzącą się z łacińskim *propositio* – zdanie) zaopatrzoną w numer, co pozwala uzyskać tyle zmiennych, ile jest liczb nadających się do numerowania, a więc dowolnie wiele. Gdy potrzebujemy nie więcej niż kilka zmiennych, zamiast indeksowanej litery „p_n” używa się tej litery i kilku po niej następujących („q”, „r” etc.) bez indeksów, co jest praktycznie wygodniejsze. Ale w postępowaniu teoretycznym, jak w poniższej definicji formuły, posługujemy się literami indeksowanymi. W ten sposób uwydatniamy tę właściwość języka rachunku zdań, że ma on w swym słowniku nieskończenie wiele (tyle, co liczb naturalnych) wyrażeń atomowych.

Oprócz wyrażeń atomowych z kategorii zdaniowej, stanowiących argumenty operacji syntaktycznych, mamy w rozważanym języku funktory zdaniotwórcze od argumentów zdaniowych. Za ich pomocą tworzy się formuły złożone, nazywane czasem molekularnymi przez analogię do powstawania molekuł z atomów. Takich funktorów jest w klasycznym rachunku zdań dwadzieścia, co wynika z pewnej prostej kombinatoryki (zob. rozdział III, ustęp 3.3).

W praktyce używa się najczęściej pięciu z nich, co odpowiada potrzebom formułowania i analizowania rozumowań przeprowadzanych w klasycznym rachunku logicznym. Z tych standardowych pięciu tylko dwa są absolutnie niezbędne do wyrażenia wszystkich praw rozumowania; takich par, które się do tego nadają jest trzy. Wybrawzy jedną z nich, za jej pomocą definiujemy pozostałe funktory, gdy zechcemy – dla większej operatywności – zbogacić o nie nasz język. Funktory wybrane jako wyjściowe i użyte do zdefiniowania pozostałych nazywają się **terminami pierwotnymi**, pozostałe zaś **terminami zdefiniowanymi** (dokonany niżej wybór terminów pierwotnych motywowany jest pewnym względem dydaktycznym).

Wśród trzech wspomnianych par są symbole „$\neg$” i „ $\wedge$ ” zwane, odpowiednio, funktorami **negacji** i **koniunkcji** (ich znaczenie zostanie podane niebawem; w obecnym punkcie, pozostającym w tematyce czysto syntaktycznej, byłoby to zbędne). Dołączywszy je do zbioru zmiennych zdaniowych, dysponujemy obecnie słownictwem niezbędnym do zapisania wszystkich twierdzeń (czyli praw) rachunku zdań. Przyjąwszy ten słownuik, definiujemy w następujący sposób pojęcie formuły, którą ze względu na występowanie wyłącznie symboli negacji i koniunkcji nazywamy formułą negacyjno-koniunkcyjnego języka rachunku zdań (n.k.r.z.).

Ciąg symboli jest **formułą n.k.r.z** wtedy i tylko wtedy, gdy jest:
1. zmienną zdaniową „p_1”, „p_2”, „p_3”,..
lub
2. formułą poprzedzoną przez symbol „$\neg$”
lub
3. zestawieniem dwóch formuł poprzedzonym przez symbol „ $\wedge$ ”.

Przepis 3 odwołuje się do notacji prefiksowej. W praktyce będziemy korzystać z infiksowej jako dogodniejszej dla ludzkiego oka, ale z innego względu dogodniejsza jest notacja prefiksowa, która uwalnia od potrzeby wprowadzania do słownika nawiasów (a takiej pedanterii nie dałoby się uniknąć, jeśli nasza definicja ma dostarczyć algorytmu dla komputera).

W powyższej definicji zawiera się przepis tworzenia formuł algorytmicznego, czyli całkowicie mechanicznego, bez potrzeby odwoływania się do rozumienia znaczeń i bez namysłu nad wykonywanym postępowaniem. Przypuśćmy, że poleca się komputerowi utworzyć wszystkie formuły liczące nie więcej niż pięć symboli i korzystające z dwu różnokształtnych zmiennych. Maszyna traktując symbol „p_1” jako formułę na mocy przepisu 1, tworzy według przepisu 2 coraz bardziej złożone negacje: „$\neg p_1$”, „$\neg\neg p_1$” itd., aż do czterech symboli „$\neg$” poprzedzających „p_1”, a następnie czyni to samo z formułą atomową „p_2”. Działając następnie według przepisu 3, tworzy najpierw najkrótsze, a więc trzywyrazowe koniunkcje z obu branych pod uwagę formuł atomowych, potem produkuje formuły czterowyrazowe, jak negacja koniunkcji „$\neg\ \wedge\ p_1p_2$” i koniunkcje z jednym członem zanegowanym (np, „$\wedge\neg p_1p_2$”). Dochodząc do pięciowyrazowych, maszyna produkuje koniunkcje dwóch negacji i podwójne negacje koniunkcji. wszystko czyniąc raz w kolejności od „p_1” do „p_2” i raz w odwrotnej.[1]

Definicja formuły n.k.r.z. dostarczy też maszynie danych do realizowania algorytmu Ajdukiewicza, w który trzeba by ją zaopatrzyć jako osobny program nadbudowany na tej definicji. Algorytm Ajdukiewicza, przypomnijmy, służy do badania poprawności syntaktycznej ciągu wyrazów, a więc, w obecnym przypadku, bo badania, czy dany ciag symboli jest formułą n.k.r.z. Warunki poprawności syntaktycznej podaje nasza definicja formuły, operująca notacją prefiksową, stąd zastosowanie Algorytmu Ajdukiewicza nie wymaga nawet kroków przygotowawczych.

I tak, okaże się, że np. ciąg symboli „$\wedge\neg p_1\neg p_2$” jest formułą, ponieważ odpowiadający mu ciąg wskaźników (kategorii syntaktycznej). uszeregowanych w powyższej kolejności symboli, redukuje się do pojedynczego wskaźnika z. Ten sam test wykaże, że nie jest formułą, powiedzmy, ciąg symboli „$p\neg\ \wedge\ p_1p_2$”, którego redukcja kończy się na dwóch, nieredukowalnych już dalej, wskaźnikach kategorii zdaniowej.[2]

3. Semantyka negacyjno-koniunkcyjnego rachunku zdań

3.1. Obecny odcinek jest pierwszym w tych wykładach (jeśli nie liczyć sporadycznych wzmianek), który wprowadza pojęcia semantyczne i samą ideę semantyki. Gdyby w jakiejś opowieści typu science fiction logika, jej działy i rozważane w niej przedmioty uległy personifikacji, to główna intryga i napięcia rozgrywałyby się w fabule dotyczącej relacji między syntaktyką i semantyką. Istotnie, w najnowszej historii logiki (liczącej półtora stulecia) relacja ta stała się przedmiotem sensacyjnych odkryć (nawiązuje do nich wyżej w ustępie 1.1 krótka wzmianka dotycząca nazewnictwa „rachunek zdań” i „logika predykatów”; o semantykę potrąca także przypowieść o lustrze w 1.2).

Rozróżnienie syntaktyki i semantyki ma tym większe znaczenie obecnie, większe nawet niż wtedy, gdy je wprowadzano, ponieważ przyczynia sie do ujęcia natury algorytmu, a więc kluczowego pojęcia nie tylko logiki lecz także informatyki. Mianowicie, algorytm jest obiektem zamkniętym bez reszty w sferze syntaktycznej.

Syntaktyka jest zbiorem teorii dotyczącyh sposobów kształtowania (inaczej: tworzenia, budowania, formowania) oraz przekształcania (transformowania) wyrażeń. Teorią należącą do syntaktyki jest każda teoria gramatyczna. Semantyka natomiast dotyczy stosunków między językiem i rzeczywistością, czy to światem empirycznym, uchwytnym po części zmysłowo, czy światem abstrakcyjnym, takim jak dziedzina matematyki, etyki etc. Nie tylko język naukowy lecz także potoczny dysponuje określeniami różnych relacji semantycznych. Oto kilka przykładów.

— „Wilno” jest nazwą stolicy Litwy.
— „Wilno” oznacza stolicę Litwy.
— „Leniwiec” oznacza pewien gatunek małp.
— π podpada pod termin „liczba rzeczywista”.
— π nie podpada pod termin „liczba naturalna”.
— Słowo „drań” nie odnosi się do dżentelmenów.
— Zdanie „Książę Witold knował z Krzyżakami” jest prawdziwe wtedy i tylko wtedy, gdy Książę Witold knował z Krzyżakami.

[1] Zapisanie prefiksowe, a następnie infiksowe, wszystkich tak wyprodukowanych formuł warto wykonać jako łatwe lecz kształcące ćwiczenie syntaktyczne.

[2] Wykonanie tych i podobnych redukcji w języku n.k.r.z. to kolejny temat do ćwiczeń.

— Wielkie twierdzenie Fermata jest prawdziwe wtedy i tylko wtedy, nie istnieją liczby naturalne x, y, z i $n > 2$ takie, że $x^n + y^n = z^n$.

Dwa ostatnie przykłady ilustrują fakt, że nazwy wyrażeń dla potrzeb semantyki można tworzyć różnymi metodami; jedną z nich jest nazwanie pewnego wyrażenie jakimś umownym terminem, inną jest utworzenie nazwy wyrażenia (będącego np. zdaniem) za pomocą cudzysłowów.

Te same przykłady dotyczą semantycznej relacji prawdziwości zachodzącej między zdaniem atomowym i jakimś stanem rzeczy: mówimy, że zdanie jest prawdziwe, gdy jego treść jest z danym stanem rzeczy zgodna. Oba przykłady ilustrują tę właśnie myśl. O zdaniach atomowych była mowa w tekście pt. "O gramatyce, logice, algorytmach i cywilizacji informatycznej" w ustępie 9.3. Przypomnijmy, że zdanie atomowe jest to wyrażenie utworzone przez pojedynczy predykat, czyli funktor zdaniotwórczy od jednego lub więcej argumentów nazwowych.

Mając określone pojęcie prawdy dla zdań atomowych, rozszerzamy je na takie zdania złożone, z jakimi będziemy mieć do czynienia w danej teorii logicznej. W języku n.k.r.z. mamy dwa rodzaje zdań złożonych: negacje i koniunkcje. Oto definicja prawdy dla każdego z tych rodzajów.[3]

Negacja zdania „p”, symbolicznie „$\neg p$” jest prawdziwa wtedy i tylko wtedy, gdy zdanie „p” jest fałszywe.

Koniunkcja zdań „p”, „q” jest prawdziwa wtedy i tylko wtedy, gdy „p” jest prawdziwe i „q” jest prawdziwe.

Funktor negacji ma w języku polskim postać zwrotu „nie jest prawdą, że” lub któregoś z wyrażeń z nim równoznacznych, a funktorowi koniunkcji odpowiada spójnik „i” (w jednym ze swych znaczeń) oraz spójniki z nim równoznaczne.

Powyższe definicje przyjęło się zapisywać w skrótowy i przejrzysty sposób za pomocą tabelek, w których używamy symbolu „1” na oznaczenie prawdy i symbolu „0” na oznaczenie fałszywości. (zob. rozdział III, ustęp 2.1). Nazywają się one **tabelkami prawdziwościowymi**, ponieważ opisują, w jaki sposób prawdziwośc zdania złozonego zależy od prawdziwości kego składników.

3.2. Definicję prawdy dla zdań złożonych można kontynuować, biorąc pod uwagę inne jeszcze funktory występujące powszechnie w rozumowaniach. Najczęściej bierze się pod uwagę następujące:

funktor alternatywy „$\vee$”, który odczytujemy za pomocą spójnika „lub”
funktor implikacji „$\Rightarrow$”, który odczytujemy za pomocą spójnika „jeśli, to”
funktor równoważności „ $\Leftrightarrow$ ”, który odczytujemy za pomocą spójnika „wtedy i tylko wtedy, gdy”.

Zamiast tworzyć definicje wedle schematu zastosowanego w przypadku negacji i koniunkcji, można za pomocą tych dwóch pierwotnych, już danych, zdefiniować nowe funktory, jak następuje.

Alternatywa: Niech „$(p \vee q)$” znaczy tyle, co „$\neg(\neg p \wedge \neg q)$”.
Implikacja: Niech „$(p \Rightarrow q)$” znaczy tyle, co „$\neg(p \wedge \neg q)$”.
Równoważność: Niech „$(p \Leftrightarrow q)$” znaczy tyle, co „$(p \Rightarrow q) \wedge (q \Rightarrow p)$”.

Powyższy sposób zapisywania definicji dobrze się spisuje jako wyraz intencji kierującej tego rodzaju definiowaniem. Mianowicie, jest to rodzaj definicji polegający na ustanowieniu, że wyrażenia po „niech”, wolno w danej teorii używać zamiennie z drugim z wymienionych wyrażeń. Dzięki tej zamienności uzyskujemy skrócenie napisów, czasem bardzo znaczne i zwiększające przejrzystość formuł.

[3] Od tego punktu, po wykorzystaniu walorów teoretycznych notacji prefiksowej, zaczyna być stosowana notacja infiksowa jako znacznie w praktycznym użyciu dogodniejsza.

Nie jest to jednak najwygodniejszy sposób zapisywania definicji, toteż przyjął się jako standard taki schemat definiowania, że o roli definicyjnej świadczy specjalnie do tego celu wprowadzony funktor. Gdy łączy nazwy, ma on postać „$=_{df}$”, a gdy łączy zdania, ma postać „$\Leftrightarrow_{df}$”; ponieważ funktor ten wskazuje, że mowa jest o stosunku między wyrażeniami (a nie rzeczami), zbędne stają się cudzysłowy.

Przepiszmy na ten zwięźlejszy sposób definicje naszych nowych funktorów. Otrzymujemy następujące zdania.

Alternatywa: $(p \vee q) \Leftrightarrow_{df} \neg(\neg p \wedge \neg q)$.

Implikacja: $(p \Rightarrow q) \Leftrightarrow_{df} \neg(p \wedge \neg q)$.

Równoważność: $(p \Leftrightarrow q) \Leftrightarrow_{df} (p \Rightarrow q) \wedge (q \Rightarrow p)$.

Funktory te, podobnie jak negację i koniunkcję, charakteryzuje się za pomocą tabelek prawdziwościowych; uzyskuje się je biorąc pod uwagę definicje prawdy dla koniunkcji i negacji oraz definicję (w terminach koniunkcji i negacji) tego funktora, dla którego sporządzamy tabelkę (tabelki dla alternatywy, implikacji i równoważności są zawarte w rozdziale III, odcinek 3).

Po wprowadzeniu stosowanego w rozumowaniach zestawu funktorów, potrafimy pełniej określić semantykę rachunku predykatów. Dotąd mamy z ustaleń semantycznych to jedno, że zmiennym zdaniowym są przyporządkowane wartości logiczne. Mówiąc dokładniej, zmienne te reprezentują wartości logiczne – w takim sensie, w jakim np. zmienne indywiduowe arytmetyki reprezentują liczby. Wprawdzie liczb, np. naturalnych, jest nieskończenie wiele, a wartości logicznych tylko dwie (w logice klasycznej), ale w obu przypadkach mamy ten sam stosunek reprezentowania.

Jeśli zmiennym języka rachunku zdań odpowiadają, po stronie semantycznej, wartości logiczne, to co odpowiada po tejże stronie funktorom? Są to abstrakcyjne obiekty będące funkcjami. Argumentami tych funkcji są prawda lub fałsz, a wartościami, jakie przybierają funkcje są także prawda lub fałsz. Ponieważ z tych dwu wartości wyróżniamy (jako pożądaną) prawdziwość. mówimy w tym przypadku o **funkcjach prawdziwościowych**, zaś oznaczające je symbole noszą miano **funktorów prawdziwościowych**.