Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie

Skrypt

Złożoność obliczeniowa

Marcin Stawiski

WMS
AGH

Wydział Matematyki ~~Stosowanej~~

Kraków/Graz 2026

Spis treści

II. Teoria obliczalności

III. Złożoność obliczeniowa

0. Formalności

- Prowadzący: Marcin Stawiski.
- E-mail: stawiski@agh.edu.pl.
- Konsultacje: pokój 6.17 w budynku C7 po wcześniejszym umówieniu się mailowo. Termin 10:00 w środy lub 17:30 w czwartki.

0.1. Zasady zaliczenia egzaminu

- Egzamin ustny.
- Na egzamin obowiązuje teoria z wykładów oraz ze skryptu.
- Warunkiem uczestnictwa w egzaminie jest pozytywna ocena z ćwiczeń.
- Ocena końcowa to zaokrąglone ($0,3$ oceny z ćwiczeń plus $0,7$ średniej ocen ze wszystkich terminów egzaminów). Przy czym warunkiem koniecznym i wystarczającym zdania przedmiotu jest istnienie terminu egzaminu, z którego ocena jest pozytywna.
- Obecność na wykładzie nie jest obowiązkowa, ale bieżąca wiedza z wykładu JEST obowiązkowa na ćwiczeniach.
- Przedmiot jest trudny i będzie wymagający na egzaminie. Studenci muszą znać treść wykładu ZE ZROZUMIENIEM, co jest istotne. Jeśli student będzie znał treść wykładu bez zrozumienia, to nadal może nie dostać pozytywnej oceny z egzaminu.
- Studenci na bieżąco powinni wyjaśniać wszelkie wątpliwości i zadawać pytania. W razie problemów ze zrozumieniem materiału lub w razie innych pytań zapraszam na konsultacje, o których wspomniałem wcześniej.
- Należy śmiać się z żartów prowadzącego!

0.2. Ćwiczenia

- Kolokwium na 60 punktów oraz projekt na 40 punktów.
- Ocenę może podnieść także aktywność podczas ćwiczeń i WYKŁADÓW.
- Zamiast kolokwium zaliczeniowego proponuję drugi projekt.

0.3. Bibliografia

1. J. E. Hopcroft, R. Motwani, J. D. Ullman, *Wprowadzenie do teorii automatów, języków i obliczeń*, PWN, 2020.
2. G. S. Boolos, J. P. Burgess, R. C. Jeffrey, *Computability and Logic*, Cambridge University Press, 2007.
3. M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
4. A. Błaszczyk, S. Turek, *Teoria Mnogości*, PWN, 2015.
5. A. Piękosz, *Wstęp do teorii modeli*, Wydawnictwo Politechniki Krakowskiej, 2008.

Program wykładu będzie istotnie różny od tego z poprzednich lat, więc notatki z poprzednich lat nie są ani konieczne, ani wystarczające. W zamian mamy pewną swobodę w doborze materiału i możemy pewne fragmenty skrócić, wydłużyć, dołożyć bądź usunąć w zależności od tego, jak będzie przebiegało tempo wykładu.

Ważniejsze jest dla mnie to, aby studenci zrozumieli treść wykładu, a nie żebyśmy przerobili jak najwięcej materiału. Wykład wraz ze skryptem jest samowystarczalny, to znaczy, że podana literatura nie jest obowiązkowa, ale może ułatwić zrozumienie pewnych pojęć i uzupełnić wiedzę.

Mam nadzieję, że przedmiot Wam się spodoba. Życzę miłej lektury!

1. Wprowadzenie

1.1. O przedmiocie

Jakie działy matematyki zawierają się w podstawach matematyki?

Podstawy matematyki (foundations):
- ~~analiza matematyczna~~,
- ~~fizyka matematyczna~~,
- logika matematyczna,
- teoria mnogości,
- teoria modeli,
- teoria dowodu,
- logika algebraiczna,
- złożoność obliczeniowa i obliczalność.

Będziemy zajmować się złożonością obliczeniową jako podstawą informatyki, ale także, a może przede wszystkim, złożonością obliczeniową jako częścią podstaw matematyki. Informatyka teoretyczna jest w istocie działem matematyki mocno związanym z logiką matematyczną. Granica między tym, co należy do informatyki, a co do matematyki, często jest umowna, „sztuczna" i płynna. Prawie wszystkie nazwiska, które będą się przewijać w trakcie wykładu czy ćwiczeń, to logicy matematyczni.

Jako że ten przedmiot zajmuje się teoretycznymi podstawami informatyki, to zaczynamy w tej kwestii od zera i nie ma żadnych wymagań wstępnych odnośnie informatyki, chociaż pewna intuicja związana z programowaniem czy algorytmami może być pomocna. Wynika to z tego, że do wielu zagadnień można podejść z punktu widzenia logiki matematycznej lub z punktu widzenia bardziej informatycznego.

Jeśli chodzi o matematykę, to niezbędna będzie podstawowa wiedza ze wstępu do logiki i teorii mnogości oraz teorii grafów (wystarczy znajomość pojęć ze wstępu do matematyki dyskretnej). Większość tego, co będzie potrzebne w zakresie logiki i teorii mnogości, zostanie jednak przedstawiona na wykładzie i ćwiczeniach.

Na jakie pytania odpowiemy na wykładzie?

1. Czy dla wszystkich problemów decyzyjnych (tj. pytań, dla których odpowiedź to „tak" lub „nie") istnieje algorytm pozwalający odpowiedzieć na to pytanie?
2. Co to znaczy, że problem jest obliczalny? Obliczalność w sensie maszyn Turinga oraz języków rekurencyjnych.
3. Co to znaczy, że problem jest wielomianowy?
4. Co oznacza „P vs. NP"? Jeden z problemów milenijnych. Czy słyszeli Państwo o tym problemie? Co oznaczają P i NP?
5. Twierdzenia Gödla-Rossera: Czy zbiór aksjomatów matematyki (teorii mnogości) jest niesprzeczny lub zupełny?
6. Co oznacza, że zdanie jest niezależne od danych aksjomatów? (hipoteza continuum, aksjomat wyboru)

Z czego będzie składał się wykład? Trzy części: podstawy matematyki, obliczalność i złożoność obliczeniowa.

1. Podstawy matematyki. Elementy logiki matematycznej, teorii dowodu, teorii modeli oraz teorii mnogości.
2. Różne modele obliczeniowe (maszyny Turinga i jej modyfikacje, funkcje rekurencyjne), formalizacja pojęcia obliczalności, równoważność omawianych modeli (teza Churcha).
3. Problemy nieobliczalne w informatyce.
4. Wpływ nieobliczalności na matematykę. Niedowodliwość w teoriach matematycznych: Twierdzenia Gödla-Rossera.
5. „Właściwa" złożoność obliczeniowa - (heurystycznie) ilość zasobów (czasu, pamięci, operacji, ...) potrzebnych do rozwiązania danego problemu obliczalnego. Klasy P i NP.
6. Problemy NP-zupełne.

1.2. Podstawowa terminologia i ujednolicenie oznaczeń:

Definicja 1.1. *Przez zbiór liczb naturalnych* $\mathbb{N}$ *będziemy rozumieć zbiór liczb naturalnych włącznie z zerem, czyli* $\mathbb{N} = \{0, 1, \dots\}$. *Jest to zbiór wszystkich skończo-*

nych liczb porządkowych (wyjaśnienie tego terminu odbędzie się w części poświęconej liczbom porządkowym).

Zbiór liczb naturalnych oznaczamy także w zależności od kontekstu przez ω albo ω_0, gdy chcemy podkreślić jego własności jako liczby porządkowej, lub przez $\aleph_0$, gdy mówimy o liczbach kardynalnych lub o mocach zbiorów.

Przez logarytm log będziemy rozumieć logarytm o podstawie 2.

Dla ustalonej liczby naturalnej n przez $\bar{x} \in X^n$ będziemy oznaczać wektor $(x_1, \ldots, x_n) \in X^n$. Przez wektor rozumiemy krotkę, a nie element przestrzeni wektorowej. Wektor $(x_1, \ldots, x_n, y)$ będziemy oznaczać przez $(\bar{x}, y)$, natomiast wektor $(x_1, \ldots, x_n, y_1, \ldots y_m)$ dla $x,\ y \in \mathbb{N}$, $\bar{x} \in X^n$ oraz $\bar{y} \in Y^m$ będziemy oznaczać przez $(\bar{x}, \bar{y})$.

Definicja 1.2. *Dla liczby naturalnej n przez relację n-argumentową o dziedzinie X będziemy rozumieć dowolny podzbiór iloczynu kartezjańskiego X^n.*

Definicja 1.3. *Funkcją n-argumentową F o dziedzinie $\mathrm{Dom}(F) \subseteq X^n$ i obrazie $\mathrm{Ran}(F) \subseteq Y$ nazywamy relację $n+1$ argumentową zawartą w $X^n \times Y$ taką, że spełnione są wszystkie z następujących warunków:*

1. *Jeśli $(\bar{x}, y) \in F$, to $\bar{x} \in \mathrm{Dom}(F)$ oraz $y \in \mathrm{Ran}(F)$.*
2. *Jeśli $\bar{x} \in \mathrm{Dom}(F)$, to istnieje dokładnie jeden $y \in \mathrm{Ran}(F)$, taki że $(\bar{x}, y) \in F$.*
3. *Jeśli $y \in \mathrm{Ran}(F)$, to istnieje co najmniej jeden $\bar{x} \in \mathrm{Dom}(F)$, taki że $(\bar{x}, y) \in F$.*

Definicja 1.4. *Przeciwdziedziną funkcji F nazywamy dowolny nadzbiór zbioru $\mathrm{Ran}(F)$. Funkcję F o dziedzinie D i przeciwdziedzinie P oznaczamy w następujący sposób: $F\colon D \to P$.*

Definicja 1.5. *Zbiór nazywamy przeliczalnym, jeśli jest zbiorem pustym lub wszystkie jego elementy da się ustawić w ciąg (niekoniecznie bez powtórzeń). W przeciwnym przypadku mówimy, że zbiór jest nieprzeliczalny.*

W przypadku dowodów twierdzeń przyjmujemy następujące oznaczenia:

Koniec dowodu. ■

Twierdzenie z dowodem na ćwiczeniach lub do samodzielnej pracy. ■

Twierdzenie bez dowodu. ■

Część I
Logika i podstawy

2. Logika pierwszego rzędu

Naszym pierwszym celem w tej części wykładu będzie „sformalizowanie matematyki", a przynajmniej znaczącej części matematyki. W celu nadania matematycznym sformułowaniom dokładnego, formalnego i jednoznacznego znaczenia potrzebny będzie nam pewnego rodzaju język. Język polski, język angielski, język kaszubski czy jakikolwiek inny „naturalny" język, w zwykłej formie nie nadają się do tego zadania. Wynika to z tego, że zdania w naturalnym języku nie muszą być ani dokładne, ani formalne, i przede wszystkim bardzo często bywają niejednoznaczne. Dodatkowym problemem w używaniu naturalnych języków do opisu matematyki jest problem z tłumaczeniem, który pogłębia wymienione wcześniej trudności.

W tym rozdziale opiszemy system zwany logiką pierwszego rzędu lub rachunkiem predykatów. Podamy sposoby konstrukcji poprawnych sformułowań matematycznych w logice pierwszego rzędu, czyli pewnych zdań lub ogólniej formuł logicznych.

W logice istotne jest rozróżnienie między składnią danego języka a jego znaczeniem w danym kontekście. Wprowadzamy rozróżnienie pomiędzy zdaniem a jego znaczeniem czy też prawdziwością. Zdanie uzyska dokładne znaczenie dopiero wtedy, gdy przypiszemy wszystkim symbolom dokładne i jednoznaczne znaczenie oraz uzgodnimy, po jakim zbiorze przebiegają kwantyfikatory. Zawsze będziemy jednak zakładać, że wszystkie kwantyfikatory przebiegają po tym samym, niepustym zbiorze.

Przykład 2.1. *Zdanie „$\varphi = (\forall x\ \forall y\ x + y = y)$" jest dobrze sformułowane języku każdej grupy addytywnej $(G, +)$ oraz każdego cia $(K, +, \cdot)$. Zdanie φ nie jest jednak poprawnie sformułowane dla grupy multiplikatywnej $(G, \cdot)$, dla grafu (G, E), ani dla grupy addytywnej $(G, \oplus)$. W ciele liczb rzeczywistych zdanie φ jest fałszywe, ale jest ono prawdziwe w grupie $\mathbb{Z}_0$.*

Przykład 2.2. *W klasycznym rachunku zdań mówimy o tautologii, gdy formuła logiczna jest prawdziwa niezależnie od wartościowania zmiennych. Przykładem tautologii jest formuła* $\psi = p \vee \neg p$*. Gdy mówimy o tautologii i prawdziwości formuły* ψ*, mamy tak naprawdę na myśli prawdziwość formuły* $\forall p_1 \forall p_2 \ldots \psi$*, gdzie* p_1, p_2 *są wszystkimi zmiennymi zdaniowymi w formule* ψ*, w strukturze odpowiadającej klasycznemu rachunkowi zdań.*

2.1. Języki pierwszego rzędu

Językiem (pierwszego rzędu) L będziemy nazywać (niekoniecznie skończony) zbiór symboli postaci $Fun_L \cup Const_L \cup Rel_L$, gdzie Fun_L, $Const_L$ i Rel_L są parami rozłącznymi zbiorami, wraz z określoną na $Fun_L \cup Const_L \cup Rel_L$ funkcją arności (argumentowości) zwaną też sygnaturą lub typem $\sigma = \sigma_L$, która jest funkcją o dziedzinie $Fun_L \cup Const_L \cup Rel_L$ i przeciwdziedzinie $\mathbb{N}$. Elementy Fun_L nazywamy symbolami funkcyjnymi, elementy $Const_L$ nazywamy symbolami stałych, a elementy Rel_L nazywamy symbolami relacyjnymi. Sygnatura działa w sposób następujący:

1. Każdemu symbolowi F z Fun_L przypisuje liczbę $\sigma(F) \in \mathbb{N}$, która odpowiada liczbie argumentów funkcji, które będzie można przypisać do F.
2. Każdemu symbolowi C z $Const_L$ przypisuje liczbę 0.
3. Każdemu symbolowi R z Rel_L przypisuje liczbę $\sigma(R) \in \mathbb{N}$, która odpowiada liczbie argumentów relacji, które będzie można przypisać do F (tutaj i w dalszej części wykładu relacje mogą być nie tylko dwuargumentowe, ale zawsze mają skończenie wiele argumentów).

W praktyce zwykle możemy się domyśleć, jakiej argumentowości są odpowiednie symbole. Będziemy więc pomijać sygnaturę i po prostu utożsamiać język L ze zbiorem $Fun_L \cup Const_L \cup Const_L$.

Przykład 2.3. *Przykładowym językiem jest* $L = \{0, 1, +, \leqslant, \}$*, gdzie* 0 *oraz* 1 *są symbolami stałych,* $+$ *jest dwuargumentowym symbolem funkcyjnym, a pozostałe symbole są dwuargumentowymi symbolami relacyjnymi.*

Elementy języka L możemy interpretować jako pewne symbole, którym następnie przypisujemy jakieś znaczenie. W szczególności symbole funkcyjne nie są funkcjami, a jedynie ich oznaczeniami. Wartość funkcji arności dla symbolu funkcyjnego informuje nas, ile argumentów będzie miała funkcja, której dopiero przypiszemy jakieś znaczenie. Analogiczna sytuacja dotyczy relacji i symboli relacyjnych. Stałe możemy też interpretować jako funkcje stałe, natomiast funkcje jako relacje, ale w praktyce wygodniej jest rozdzielić funkcje od pozostałych relacji i stałych.

2.2. Logika pierwszego rzędu

Do konstrukcji systemu logiki pierwszego rzędu podamy potrzebne nam symbole, które będą pełnić rolę podobną do alfabetu, a następnie reguły tworzenia „napisów": termów oraz formuł.

Dla danego języka L będziemy używać następujących symboli do jego opisu:
1. Symbole logiczne: $\wedge,\ \vee, \neg, \forall,\ \exists,\ =$.
2. Symbole zmiennych: $x,\ y,\ z, \ldots,\ x_1,\ x_2, x_3, \ldots$ (w przeliczalnej liczbie).
3. Symbole pozalogiczne, czyli symbole z języka L.
4. Symbole pomocnicze, czyli nawiasy oraz przecinek.

Uwaga: W logice pierwszego rzędu wszystkie stałe i zmienne są tego samego typu. Dla przykładu w teorii mnogości nie rozróżniamy między obiektami-zbiorami a obiektami-elementami. Każdy zbiór może być elementem innego zbioru, a każdy element zbioru sam jest zbiorem.

Uwaga: Symbol relacji „=" będzie traktowany jako symbol logiczny. Odnotujmy, że implikację oraz równoważność można zapisać przy pomocy pozostałych symboli logicznych, dlatego nie ma ich na liście.

Zdefiniujemy teraz pojęcie termu. O termie możemy myśleć jako o „wartości funkcji", ale dopiero wtedy, gdy będziemy mieli daną interpretację odpowiednich symboli.

Definicja 2.4. *Zbiorem termów języka L nazywamy skończone ciągi symboli powstałe w następujący sposób:*
1. *Każdy symbol stałej $C \in Const_L$ jest termem.*
2. *Każdy symbol zmiennej x jest termem.*
3. *Jeśli dany jest symbol funkcyjny $F \in Fun_L$ argumentowości n oraz termy $t_1, \ldots, t_n$, to $F(t_1, \ldots, t_n)$ też jest termem.*
4. *Tylko te ciągi symboli, które powstały w skończeniu wiele krokach z powyższych reguł są termami języka L.*

Przykład 2.5. *W języku $(2, 1, 0, \arccos x)$ termami są przykładowo „$1 + 2$", „y_2", „$\arccos(x_{21} + x_{37})$", „0", „x".*

Uwaga: Dla symboli funkcji dwuargumentowych często stosuje się zapis infiksowy, np. piszemy $1 + 2$ zamiast $+(1, 2)$.

Definicja 2.6. *Termem stałym nazywamy term niezawierający symboli zmiennych.*

Podamy teraz definicję formuły. O formułach możemy myśleć jako o relacjach, które zachodzą (lub nie) pomiędzy termami, ale znów dopiero wtedy, gdy będziemy mieli interpretację odpowiednich symboli.

Definicja 2.7. *Formułą atomiczną języka L nazywamy każdy skończony ciąg znaków postaci:*

$t_1 = t_2$, *gdzie* t_1 *oraz* t_2 *są termami* L,
$R(t_1, \ldots, t_n)$, *gdzie* R *jest symbolem relacyjnym argumentowości* n, *a* $t_1, \ldots, t_n$ *są termami* L.

Definicja 2.8. *Zbiorem formuł języka* L *(ozn.* $Form_L$*) nazywamy zbiór otrzymany w następujący sposób:*
1. *Każda formuła atomiczna* L *jest formułą* L,
2. *Każdy ciąg symboli postaci* $\varphi \wedge \psi$, $\varphi \vee \psi, \neg\varphi, \varphi \Rightarrow \psi, \phi \Leftrightarrow \psi, \forall x\ \varphi, \exists x\ \varphi$ *jest formułą* L, *gdzie* φ *i* ψ *są formułami języka* L, *a* x *jest dowolnym symbolem zmiennej,*
3. *Tylko te ciągi symboli, które powstały w skończeniu wiele krokach z powyższych reguł są formułami języka* L.

Definicja 2.9. *Podformułą formuły* φ *języka* L *nazywamy każdą formułę, która posłużyła do skonstruowania formuły* φ *włącznie z samą* φ.

Definicja 2.10. *Mówimy, że zmienna* x *jest wolna w termie* φ, *jeśli występuje w termie* φ *i nie jest związana w* φ *kwantyfikatorem* $\forall x$, *ani* $\exists x$. *Zmienne, które występują w formule* φ, *ale nie są wolne, nazywamy związanymi. Jeśli* $x_1, \ldots$ *są zmiennymi wolnymi formuły* φ, *to przez* $\varphi(y_1, \ldots)$ *oznaczamy formułę powstałą przez zastąpienie* $(x_1, \ldots)$ *przez* $(y_1, \ldots)$.

Definicja 2.11. *Zdaniem nazywamy formułę bez zmiennych wolnych. Zbiór zdań języka* L *oznaczamy przez* $Sent_L$.

Porównaj z definicją zdania jako „obiektu” prawdziwego lub nie.

3. Teoria dowodu

W tym rozdziale omówimy formalnie pojęcie dowodu w systemie formalnym. Dzięki wprowadzonym pojęciom możemy uzyskać system aksjomatyczny. Różni się od systemu „półaksjomatycznego" tym, że mamy ściśle zdefiniowane nie tylko aksjomaty danej teorii, ale także ściśle zdefiniowane reguły wnioskowania, czyli to, w jaki sposób uzyskujemy nowe twierdzenia z aksjomatów i poprzednio udowodnionych twierdzeń. Przykładem systemu półaksjomatycznego jest geometria euklidesowa.

Definicja 3.1. *Systemem formalnym nazywamy zbiór* $\mathcal{S} = (L,\ A,\ B,\ \Sigma)$*, gdzie*
1. L *jest pewnym językiem.*
2. A *jest zbiorem aksjomatów logicznych.*
3. B *jest zbiorem reguł wnioskowania.*
4. Σ *jest pewnym zbiorem zdań języka* L*, a jego elementy nazywamy aksjomatami systemu* $\mathcal{S}$*.*

Zdefiniujemy teraz formalnie pojęcie dowodu. Poniższą definicję możemy zobrazować w taki sposób, że twierdzenie w jest dowodzone z aksjomatów lub poprzednio udowodnionych twierdzeń za pomocą reguł wnioskowania.

Definicja 3.2. *Dowodem formalnym lub po prostu dowodem zdania* w *ze zbioru zdań* Σ *nazywamy skończony ciąg zdań* $w_1, \ldots, w_n = z$*, taki że dla każdego* $i \leqslant n$ *zdanie* w_i *jest aksjomatem systemu formalnego, elementem* Σ *lub istnieją liczby* $j_1, \ldots j_k \leqslant i$*, takie że* w_i *można uzyskać z* $w_{j_1}, \ldots, w_{j_k}$ *za pomocą jednej z reguł wnioskowania.*

Najważniejszą z reguł wnioskowania jest modus ponens, który mówi o tym, że jeśli mamy implikację i poprzednik implikacji jest prawdziwy, to możemy wywnioskować następnik implikacji.

Definicja 3.3. *Modus ponens (lub reguła odrywania) to następująca reguła wnioskowania, jeśli $p \Rightarrow q$ oraz p, to wnioskujemy q.*

Uwaga: dla uproszczenia zwykle często zakłada się, że jedyną regułą wnioskowania jest modus ponens.

Definicja 3.4. *Jeśli istnieje dowód zdania w ze zbioru zdań Σ, to mówimy, że w jest konsekwencją logiczną zbioru Σ i piszemy $\Sigma \vdash w$.*

Definicja 3.5. *Jeśli zdanie w posiada dowód w systemie $\mathcal{S}$, to w nazywamy twierdzeniem systemu $\mathcal{S}$.*

Definicja 3.6. *Domknięciem dedukcyjnym lub zbiorem konsekwencji logicznych zbioru zdań Σ nazywamy zbiór $Col(\Sigma) = \{w \in Sent_L \colon \Sigma \vdash w\}$.*

Definicja 3.7. *Mówimy, że zbiór zdań Σ jest dedukcyjnie domknięty, jeśli*

$$Col(\Sigma) = \Sigma.$$

Definicja 3.8. *Mówimy, że zbiór zdań Σ jest sprzeczny (syntaktycznie), jeśli istnieje zdanie $\varphi \in Sent_L$, takie że $\Sigma \vdash \varphi$ oraz $\Sigma \vdash \neg\varphi$. W przeciwnym przypadku mówimy, że Σ jest niesprzeczny (syntaktycznie).*

Definicja 3.9. *Mówimy, że zdanie φ jest niezależne od zbioru zdań Σ, jeśli $\Sigma \nvdash \varphi$ oraz $\Sigma \nvdash \neg\varphi$.*

Najbardziej znanym zdaniem niezależnym od standardowych aksjomatów teorii mnogości jest hipoteza continuum, która mówi o tym, że nie ma zbiorów o mocy pośredniej między mocą zbioru liczb naturalnych a mocą zbioru liczb rzeczywistych. Powiemy o niej nieco więcej w dalszej części skryptu.

Twierdzenie 3.10. *Jeśli zdanie φ jest niezależne od zbioru zdań Σ wtedy i tylko wtedy, gdy zarówno $\Sigma \cup \{\varphi\}$ jak i $\Sigma \cup \{\neg\varphi\}$ są niesprzeczne.* ■

Poniższe twierdzenie, zwane twierdzeniem o zwartości, jest jednym z najważniejszych twierdzeń w logice matematycznej. Jego nazwa pochodzi od oryginalnej metody dowodowej, które opierało się na twierdzeniu Tichonowa o zwartości (udowodnione jako pierwsze przez Čecha). Przeliczalną wersję twierdzenia o zwartości udowodnił w 1930 roku Gödel, a pełną wersję w 1936 roku udowodnił Malcew.

Twierdzenie 3.11 (Gödel 1930 [17], Malcew 1936 [30], o zwartości)**.** *Zbiór zdań Σ jest niesprzeczny wtedy i tylko wtedy, gdy każdy jego skończony podzbiór jest niesprzeczny.*

Dowód. Implikacja w prawo jest oczywista, bo dowód sprzeczności w podzbiorze Σ' zbioru Σ jest też dowodem w Σ. Załóżmy więc dla dowodu nie wprost, że zbiór Σ jest sprzeczny, i niech $w_1, \ldots, w_n$ będzie dowodem zdania $\varphi \vee \neg\varphi$ w Σ dla pewnego $\varphi \in Sent_L$. Wówczas dowód $w_1, \ldots, w_n$ jest skończony, więc znajduje się w nim tylko skończona liczba zdań z Σ. Oznaczmy go przez Σ'. Wówczas $w_1, \ldots, w_n$ jest także dowodem zdania $\varphi \vee \neg\varphi$ w Σ' oraz Σ' jest skończone. ■

Definicja 3.12. *Zbiór zdań jednocześnie niesprzeczny i dedukcyjnie domknięty nazywamy teorią.*

Definicja 3.13. *Mówimy, że zbiór zdań Σ jest zupełny, jeśli dla każdego zdania $\varphi \in Sent_L$ zachodzi $\Sigma \vdash \varphi$ lub $\Sigma \vdash \neg\varphi$.*

Obserwacja 3.14. *Jeśli zbiór zdań niesprzeczny jest zupełny, to jest dedukcyjnie domknięty.* ■

Poniższe twierdzenie przypisywane jest (między innymi przez Tarskiego) Lindenbaumowi, ale po raz pierwszy pojawia się w jego wspólnych pracach właśnie z Tarskim. Zwane jest ono lematem Lindenbauma albo twierdzeniem Lindenbauma.

Twierdzenie 3.15 (Lindenbaum, Tarski 1930/1935 [27,28], lemat Lindenbauma)**.** *Każdy niesprzeczny zbiór zdań można rozszerzyć do teorii zupełnej. To znaczy: dla każdego niesprzecznego zbioru zdań Σ istnieje zbiór Σ', taki że $\Sigma \subseteq \Sigma'$, Σ' jest niesprzeczny oraz Σ' jest zupełny.* ■

4. Teoria modeli

Teoria modeli jest działem podstaw matematyki zajmującym się związkami między składnią języka, a jego znaczeniem, czyli semantyką. Na razie poznaliśmy sposoby tworzenia „sensownych" napisów w logice pierwszego rzędu. W kolejnym rozdziale podamy, w jaki sposób możemy powiązać składnię, formuły czy termy z ich znaczeniem w danym kontekście matematycznym.

4.1. Modele

Definicja 4.1. *Niech M, I, J, K będą zbiorami parami rozłącznymi. Ponadto niech $M \neq \emptyset$. Strukturą nazywamy zbiór*

$$(M, \{f_i\}_{i\in I} \cup \{c_j\}_{j\in J} \cup \{r_k\}_{r\in K}), \text{ gdzie:}$$

1. *M nazywamy zbiorem podkładowym lub uniwersum struktury M.*
2. *Każde f_i jest odwzorowaniem $f_i \colon M^{a(i)} \to M$ zwanym funkcją pierwotną struktury $\mathcal{M}$, a liczbę $a(i) \in \mathbb{N}$ nazywamy arnością (argumentowością) funkcji f_i.*
3. *Każde c_j jest elementem M zwanym stałą struktury $\mathcal{M}$.*
4. *Każde r_k jest relacją $r_k \subseteq M^{a(k)}$ zwaną relacją pierwotną struktury $\mathcal{M}$, a liczbę $a(k) \in \mathbb{N}$ nazywamy arnością lub argumentowością relacji r_k.*

Przykład 4.2. *Przykładami struktur są $(\mathbb{N}, 0, 1, +, \cdot)$, $(\mathbb{R}, 1, \leqslant)$. Każda grupa, pierścień czy ciało również są strukturami. Podobnie strukturami są grafy, gdzie zbiór krawędzi będziemy interpretować jako pewną dwuargumentową relację, która jest przeciwzwrotna i symetryczna.*

W celu powiązania modeli z językami przydatne będzie pojęcie L-struktury. W teorii modeli, jeśli x jest pewnym „obiektem", to jego odpowiednik w strukturze $\mathcal{M}$ oznaczamy przez $x^{\mathcal{M}}$.

Przykład: Dla struktury $\mathcal{R} = (\mathbb{R}, +, 0)$, gdzie $\mathbb{R}$ jest zbiorem liczb rzeczywistych z zerem oraz naturalnym dodawaniem liczb rzeczywistych, symbolowi 0 przypisujemy element $0^{\mathcal{M}}$ zbioru $\mathbb{R}$. Symbolowi + odpowiada działanie $+^{\mathbb{R}}$.

Definicja 4.3. *Strukturą języka L lub L-strukturą nazywamy zbiór*

$$\mathcal{M} = (M, \{F^{\mathcal{M}}\}_{F \in Fun_L} \cup \{C^{\mathcal{M}}\}_{C \in Const_L} \cup \{R^{\mathcal{M}}\}_{R \in Rel_L}), \text{ gdzie:}$$

1. *M jest niepustym zbiorem będącym uniwersum L-struktury $\mathcal{M}$ lub jego zbiorem pokładowym.*
2. *Każdemu symbolowi funkcyjnemu $F \in Fun_L$ przypisujemy funkcję pierwotną $F^{\mathcal{M}} \colon M^{\sigma_L(F)} \to M$ struktury $\mathcal{M}$ zwaną interpretacją symbolu F w $\mathcal{M}$.*
3. *Każdemu symbolowi stałej $C \in Const_L$ przypisujemy stałą pierwotną $C^{\mathcal{M}} \in M$ struktury $\mathcal{M}$ zwaną interpretacją symbolu F w $\mathcal{M}$.*
4. *Każdemu symbolowi relacyjnemu $R \in Rel_L$ przypisujemy relację pierwotną $R^{\mathcal{M}} \subseteq M^{\sigma_L(R)}$ struktury $\mathcal{M}$ zwaną interpretacją symbolu R w $\mathcal{M}$.*

W teorii grafów przez $|G|$ oznaczaliśmy nie tyle moc grafu jako zbioru, ponieważ graf jako para uporządkowana ma zawsze 2 elementy, ale moc jego zbioru wierzchołków, a więc w języku teorii modeli - uniwersum grafu. Podobna konwencja dotyczy także innych struktur.

Definicja 4.4. *Mocą struktury $\mathcal{M}$ nazywamy moc jej uniwersum i oznaczamy ją przez $|\mathcal{M}|$.*

Gdy dla języka L mamy daną jego strukturę $\mathcal{M}$, to możemy nadać termom i formułom tego znaczenie w naturalny sposób. Możemy wówczas mówić o wartości logicznej zdań, czyli o tym, czy są prawdziwe w strukturze $\mathcal{M}$, czy też są fałszywe. Szczegóły pominiemy. Dla przykładu możemy pytać, czy zdanie $\exists x\ \forall y\ x \cdot y$ jest prawdziwe w $\mathcal{M}$ dla struktury języka $L = \{\cdot\}$. Jeśli zdanie φ jest prawdziwe w strukturze $\mathcal{M}$, to piszemy $\mathcal{M} \Vdash \varphi$.

Definicja 4.5. *Zbiór zdań prawdziwych w strukturze $\mathcal{M}$ nazywamy teorią struktury $\mathcal{M}$ i oznaczamy ją przez $\mathcal{T}h(\mathcal{M}) = \{\varphi \colon \mathcal{M} \Vdash \varphi\}$.*

Uwaga: Zwróćmy uwagę, że teoria w nauce często jest rozumiana w niewłaściwy sposób. Teoria nie oznacza koniecznie czegoś teoretycznego, czego nie potrafimy dowieść, jako przeciwieństwo czegoś praktycznego. Teoria oznacza natomiast pewien system, którego założenia i wnioski z nich płynące można często zweryfikować doświadczalnie. Matematyczne pojęcie teorii (wraz z systemem logicznym) jest bliskie temu pojęciu.

Definicja 4.6. *Niech Σ będzie zbiorem pewnych zdań języka L (zbiorem aksjomatów). Mówimy, że struktura $\mathcal{M}$ jest modelem zbioru zdań Σ, jeśli $\Sigma \subseteq \mathcal{T}h(\mathcal{M})$.*

Definicja 4.7. *Mówimy, że zbiór zdań Σ jest niesprzeczny semantycznie, jeśli posiada model.*

Przyjrzyjmy się teraz związkowi pomiędzy niesprzecznością semantyczną i niesprzecznością syntaktyczną, oraz wynikaniem semantycznym a syntaktycznym. Jeśli mamy twierdzenie matematyczne, które mówi np. o grafach, to w dowodzie często ustalamy jakiś graf i udowadniamy, że dla niego zachodzi to twierdzenie. To, że tego typu rozumowanie jest prawidłowe wynika z następujących dwóch twierdzeń:

Twierdzenie 4.8 (Hilbert, Ackermann 1928 [22], Gödel 1930 [17], o poprawności)**.** *Jeśli φ jest konsekwencją logiczną zbioru zdań Σ, to jest ono prawdziwe we wszystkich modelach zbioru zdań Σ. Równoważnie, jeśli φ jest niesprzeczne semantycznie, to jest niesprzeczne syntaktycznie.* ■

Twierdzenie o poprawności mówi intuicyjnie, że z aksjomatów nie da się dowieść z aksjomatów niczego fałszywego, np. z teorii grup nie wyniknie coś, co jest fałszywe dla grup. Poniższe twierdzenie o pełności mówi coś odwrotnego:

Twierdzenie 4.9 (Gödel 1930 [17], o pełności)**.** *Jeśli φ jest prawdziwe we wszystkich modelach zbioru zdań Σ, to jest ono konsekwencją logiczną zbioru zdań Σ. Równoważnie, jeśli φ jest niesprzeczne syntaktycznie, to jest niesprzeczne semantycznie.* ■

Okazuje się więc, że oba typy niesprzeczności są tożsame, jeśli założymy aksjomat wyboru. Podobnie oba typy wynikania (semantyczne i syntaktyczne) są równoważne. Twierdzenie o poprawności nie wymaga aksjomatu wyboru i jest prawdziwe w ZFC, ale twierdzenie o pełności dla logiki pierwszego rzędu jest równoważne aksjomatowi wyboru nad ZF! Poza wyszczególnionymi przypadkami nie będziemy pracować nad ZF, więc w praktyce zwykle nie musimy rozróżniać obu typów niesprzeczności czy wynikania. Nie będziemy dowodzić żadnego z tych twierdzeń, bo dowód twierdzenia o pełności jest nietrywialny, a dowód twierdzenia o poprawności jest nudny.

Bardzo ważnymi pojęciami w matematyce są pojęcia homomorfizmu oraz izomorfizmu, które w szczególnych wersjach poznaliście między innymi na algebrze czy teorii grafów. Podamy teraz informacyjnie ich ogólniejszą formę.

Definicja 4.10. *Niech $\mathcal{M}$ oraz $\mathcal{N}$ będą L-strukturami. Wówczas odwzorowanie $h\colon M \to N$ nazywamy homomorfizmem z $\mathcal{M}$ do $\mathcal{N}$, gdy spełnione są następujące warunki:*
1. *Dla każdego symbolu stałej C mamy $h(C^{\mathcal{M}}) = C^{\mathcal{N}}$.*

2. *Dla każdego symbolu funkcyjnego F arności n oraz $a_1, \ldots, a_n \in M$ mamy $h(F^{\mathcal{M}}(a_1, \ldots, a_n)) = F^{\mathcal{N}}(h(a_1), \ldots, h(a_n))$.*
3. *Dla każdego symbolu relacyjnego R arności n oraz $a_1, \ldots, a_n \in M$, z faktu, że $(a_1, \ldots, a_n) \in R^{\mathcal{M}}$, wynika $(h(a_1), \ldots, h(a_n)) \in R^{\mathcal{N}}$.*

Definicja 4.11. *Mówimy, że bijektywny homomorfizm z $\mathcal{M}$ do $\mathcal{N}$ jest izomorfizmem, jeśli jego funkcja odwrotna jest homomorfizmem z $\mathcal{N}$ do $\mathcal{M}$. Ponadto, izomorfizm z $\mathcal{M}$ do $\mathcal{M}$ nazywamy automorfizmem struktury $\mathcal{M}$.*

4.2. Zastosowania teorii modeli

W tym krótkim rozdziale podamy parę zastosowań teorii modeli w innych działach matematyki. Warto wspomnieć, że poza samą matematyką teoria modeli ma zastosowanie między innymi w bazach danych czy sztucznej inteligencji.

4.2.1. Analiza niestandardowa

Niech $\mathcal{M} = (\mathbb{R}, +, -, \cdot, 0, 1, c, \leqslant)$ będzie standardowym uporządkowanym ciałem liczb rzeczywistych z dodatkowym symbolem stałej c oraz niech $\Sigma = \mathcal{T}h(\mathcal{M}) \cup \{\varphi_n \colon n \in \mathbb{N}\}$, gdzie φ_n jest zdaniem „$c > 1 + \cdots + 1$” (n jedynek).

Każdy model $\mathcal{N}$ zbioru zdań Σ jest ciałem uporządkowanym, które zawiera w sobie uporządkowane ciało liczb rzeczywistych. Jednak w przeciwieństwie do ciała liczb rzeczywistych w $\mathcal{N}$ istnieją elementy nieskończenie duże, a więc także nieskończenie małe. Jest to przykład tak zwanego modelu niestandardowego.

4.2.2. Algebra abstrakcyjna

Definicja 4.12. *Niech P będzie pierścieniem z jedynką. Najmniejszą liczbę jedynek, które należy do siebie dodać, aby otrzymać 0 nazywamy charakterystyką pierścienia P. Jeśli taka liczba nie istnieje, to przyjmujemy, że charakterystyka pierścienia P wynosi 0.*

Twierdzenie 4.13. *Charakterystyka pierścienia z jedynką jest zawsze liczbą pierwszą lub zerem.* ■

Twierdzenie 4.14 (Ax 1936 [3], O charakterystyce, pierwsze)**.** *Jeśli zdanie $\varphi \in Sent_{\{+,-,\cdot,0,1\}}$ jest prawdziwe we wszystkich pierścieniach z jedynką charakterystyki zero, to jest prawdziwe we wszystkich pierścieniach z jedynką dostatecznie dużej charakterystyki.*

Dowód. Niech $\Sigma = F \cup \{\psi_p \colon p \in \mathbb{P}\}$, gdzie F jest aksjomatyką teorii pierścieni z jedynką, a zdanie ψ_p mówi, że suma p jedynek jest różna od zera. Wówczas

$\Sigma \vdash \varphi$ wtedy i tylko wtedy, gdy pewien skończony podzbiór zbioru Σ dowodzi φ. Niech $\Sigma' = F \cup \{\psi_{p_1}, \dots \psi_{p_n}\}$ dla pewnych liczb pierwszych $p_1, \dots p_n$. Zdanie φ, które jest prawdziwe we wszystkich pierścieniach z jedynką charakterystyki 0 jest prawdziwe we wszystkich pierścieniach z jedynką charakterystyki większej niż $\max\{p_1, \dots, p_n\}$. ■

Twierdzenie 4.15 (Ax 1965 [3], O charakterystyce, drugie)**.** *Jeśli zdanie $\varphi \in Sent_{\{+,-,\cdot,0,1\}}$ jest prawdziwe w pewnych pierścieniach z jedynką dowolnie dużej charakterystyki, to jest prawdziwe w pewnych pierścieniach z jedynką charakterystyki zero dowolnie dużej mocy.* ■

4.2.3. Grafy losowe

Definicja 4.16. *Ustalmy liczbę rzeczywistą $p \in (0,1)$. Niech $V_n = \{0, 1 \dots, n-1\}$ i niech E_n będzie zbiorem dwuelementowych podzbiorów zbioru V_n. Dla każdego $e \in E_n$ definiujemy przestrzeń probabilistyczną $(\Omega, \mathcal{P}(\Omega_e, p_e))$, gdzie $\Omega_e = (0_e, 1_e)$, $p(1_e) = p$, $p(0, e) = 1-p$. Grafem losowym $\mathcal{G}_{n,p}$ na n wierzchołkach nazywamy przestrzeń produktową $\Pi\{(\Omega, \mathcal{P}(\Omega), p_e) \colon e \in E_n\}$. Zdarzenia elementarne przestrzeni $\mathcal{G}_{n,p}$ utożsamiamy z grafami.*

Definicja 4.17. *Mówimy, że własność $\varphi(G)$ zachodzi dla prawie wszystkich grafów, jeśli dla ustalonego p i n dążącego do nieskończoności prawdopodobieństwo $p(\varphi(G))$ dąży do 1.*

Niech v będzie wierzchołkiem grafu losowego. Oznaczmy przez $\Psi_{k,l}$ zdarzenie mówiące o tym, że każdy wierzchołek jest połączony z co najmniej k wierzchołkami i nie jest połączony z co najmniej l wierzchołkami.

Twierdzenie 4.18. *Dla każdego k oraz l prawdopodobieństwo zdarzenia $\Psi_{k,l}$ dąży do 1.* ■

Definicja 4.19. *Grafem Rada (Ackermanna) nazywamy graf przeliczalny, który ma własność $\Psi_{k,l}$ dla każdego k, l.*

Podamy teraz konstrukcję grafu Rada pochodzącą od Ackermanna.

Definicja 4.20. *Rodziną zbiorów dziedzicznie skończonych $\mathbb{HF}$ nazywamy najmniejszą rodzinę zbiorów powstałych w sposób następujący:*
1. $\emptyset \in \mathbb{HF}$,
2. *Jeśli $a_1, \dots, a_k \in \mathbb{HF}$, to $\{a_1, \dots, a_k\} \in \mathbb{HF}$.*

Twierdzenie 4.21. *Niech $V(G) = \mathbb{HF} \setminus \emptyset$ i niech dwa wierzchołki grafu G będą połączone ze sobą wtedy i tylko wtedy, gdy jeden z nich jest elementem drugiego jako zbiór. Wówczas G jest grafem Ackermanna.* ■

Twierdzenie 4.22 (Ackermann 1937 [2], Rado 1964 [33])**.** *Graf Rada jest jedyny z dokładnością do izomorfizmu.* ■

Graf Rada jest modelem zbioru zdań $\Sigma = \{\Psi_{k,l} \colon k, l \in \mathbb{N}\} \cup F$, gdzie F jest zbiorem aksjomatów teorii grafów, tj. mówiącymi o tym, że relacja E jest symetryczna i antyzwrotna. Co więcej, jest to jedyny przeliczalny graf spójny o tej własności.

Definicja 4.23. *Mówimy, że zbiór zdań Σ jest kategoryczny w mocy κ, jeśli ma jedyny z dokładnością do izomorfizmu model mocy κ.*

Formalnie pojęcie mocy potrzebne do zrozumienia pełnej wersji powyższej definicji i poniższego twierdzenia będzie w późniejszym rozdziale. Dla naszych potrzeb wystarczy jednak, że ograniczymy się do nieskończonych zbiorów przeliczalnych, czyli tych mocy $\aleph_0$. Wówczas zbiór kategoryczny w mocy $\aleph_0$, to po prostu jedyny nieskończony, przeliczalny model danej mocy.

Twierdzenie 4.24. *Jeśli zbiór zdań jest kategoryczny w dowolnej mocy nieskończonej większej lub równej od mocy języka, to jest on zupełny.* ■

Teoria grafu Rada nie jest kategoryczna w mocy $\aleph_0$. Uzasadnij dlaczego! Pomimo tego prawdziwe jest następujące twierdzenie.

Twierdzenie 4.25 (Rado 1964 [33])**.** *Teoria grafu Rada jest zupełna.* ■

Twierdzenie 4.26 (Fagin 1976 [15], Prawo zero-jedynkowe dla własności pierwszego rzędu, wersja pierwsza)**.** *Niech φ będzie własnością grafów pierwszego rzędu. Wówczas prawie wszystkie grafy mają własność φ lub prawie wszystkie grafy mają własność $\neg\varphi$.*

Dowód. Z twierdzenia 4.25 wynika, że skoro Σ jest zupełne, to albo φ, albo $\neg\varphi$ może być wywnioskowane z Σ. Bez straty dla ogólności przyjmijmy, że jest to φ. Postępujemy podobnie jak w dowodzie twierdzenia o zwartości. Zdanie φ posiada skończony dowód, więc użyte jest w nim tylko skończenie wiele zdań ze zbioru Σ. Niech $\Sigma' = \{\Psi_1 \cup \ldots \Psi_k\} \subseteq \Sigma$ będzie skończonym podzbiorem zbioru Σ, który pozwala na wywnioskowanie φ. Zbiór Σ' jest skończony, a prawdopodobieństwo

każdego zdania ze zbioru Σ dąży do jeden, więc prawdopodobieństwo przecięcia tych zdarzeń dąży do jeden. Wynika z tego, że z prawdopodobieństwem dążącym do jeden możemy wywnioskować φ. ■

Co więcej, możemy sprawdzić, czy dana własność grafów pierwszego rzędu zachodzi dla prawie wszystkich grafów, przez zbadanie, czy graf Ackermanna ją posiada.

Twierdzenie 4.27 (Fagin 1976 [15], Prawo zero-jedynkowe dla własności pierwszego rzędu, wersja druga)**.** *Niech φ będzie własnością grafów pierwszego rzędu. Wówczas prawie wszystkie grafy mają własność φ wtedy i tylko wtedy, gdy ma ją graf Rada.* ■

5. Arytmetyka Peana

Za przykład zastosowania logiki pierwszego rzędu posłuży nam arytmetyka liczb naturalnych pochodząca od Peana.

5.1. Aksjomaty Peana

Definicja 5.1. *Niech* $L = \{+, \cdot, 0, 1, \leqslant\}$. *Arytmetyką Peana (pierwszego rzędu) (w skrócie PA) nazywamy następujący zbiór aksjomatów:*
1. $\forall x\ x + 1 \neq 0$,
2. $\forall x\ 0 \leqslant x$,
3. $\forall x\ (x = 0) \vee \neg(x \leqslant 0)$,
4. $\forall x\ x + 0 = x$,
5. $\forall x\ x \cdot 0 = 0$,
6. $\forall x\ \forall y\ x + (y + 1) = (x + y) + 1$,
7. $\forall x\ \forall y\ x \cdot (y + 1) = x \cdot y + x$,
8. $\forall x\ \forall y\ (x + 1 = y + 1) \Rightarrow x = y$,
9. $\forall x\ \forall y\ (y \leqslant x + 1) \Leftrightarrow (y \leqslant x \vee y = x + 1)$,
10. $\forall x\ \forall y\ (x + 1 \leqslant y) \Leftrightarrow (x \leqslant y \wedge x \neq y)$,

oraz następujący schemat aksjomatów indukcji mówiący o tym, że dla każdej formuły $\varphi(x_1, \ldots, x_n, y)$ *języka* L *mamy aksjomat postaci:*

$$\forall x \ldots\ \forall x_n\ [(\varphi(x_1, \ldots, x_n, 0) \wedge$$
$$\forall y\ (\varphi(x_1, \ldots, x_n, y) \Rightarrow (\varphi(x_1, \ldots, x_n, y + 1)))) \Rightarrow \forall y\ (\varphi(x_1, \ldots, x_n, y)].$$

Pytanie: ile mamy aksjomatów w arytmetyce Peana?

Uwaga, ważne: schemat aksjomatów indukcji nie jest pojedynczym aksjomatem. Dla każdej formuły mamy osobny aksjomat! Dlatego też mówimy o schemacie aksjomatów indukcji, a nie o aksjomacie indukcji.

Uwaga: Istotne jest, że zbiór aksjomatów w arytmetyce Peana jest nie tylko przeliczalny, ale i rekurencyjny, czyli „prosty" w sensie obliczalności (pojęcie rekurencyjności poznamy przy okazji omawiania maszyn Turinga).

Dla pewnych zbiorów zdań Σ na gruncie aksjomatyki Peana (i silniejszych systemów jak ZFC) da się skonstruować zdanie mówiące o tym, że zbiór zdań Σ jest niesprzeczny. Zdanie to oznaczamy przez $Con(\Sigma)$. Dokładniej, w aksjomatyce Peana da się ponumerować dowody (a jest ich przeliczalna liczba) oraz zdania i wyrazić formalnie zdania „A jest dowodem zdania B" oraz „istnieje dowód A zdania B". W szczególności można wyrazić formalnie zdanie „istnieje dowód A zdania „$0 = 1$"". Jeśli zdanie to ma dowód w zbiorze Σ, to zbiór Σ jest sprzeczny.

5.2. Twierdzenie Goodsteina

Arytmetyka Peana jest „dziurawa". W tym sensie, że istnieją twierdzenia w „standardowej" teorii mnogości, które można wyrazić w języku arytmetyki Peana, ale nie można ich udowodnić za pomocą aksjomatów Peana. Historycznie pierwszym takim przykładem jest twierdzenie udowodnione w 1944 roku przez Goodsteina. Opiszemy poniżej jego treść wraz z przykładem:

Wybieramy pewną liczbę naturalną k. Zapisujemy k używając tylko dodawania i naturalnych potęg liczby 2. Dla przykładu zacznijmy od liczby 11, która będzie pierwszym wyrazem G_0 pewnego ciągu:

$$G_0 = 11 = 2^{2+2^0} + 2^1 + 2^0.$$

Następnie zmieniamy wszystkie wystąpienia liczby 2 na 3 i odejmujemy od otrzymanej liczby 1, a następnie zapisujemy nowo otrzymaną liczbę w sposób podobny jak wcześniej, ale używając tylko potęg liczby 3, otrzymując kolejny wyraz G_1 ciągu.

$$G_1 = 3^{3+3^0} + 3^1 + 3^0 - 1 = 3^{3+3^0} + 3^1 = 1027.$$

Powtarzamy powyższą procedurę aż nie dostaniemy liczby 0.

Twierdzenie 5.2 (Goodstein 1944)**.** *Dla każdej liczby naturalnej powyżej opisany ciąg osiąga liczbę* 0. ■

Kirby i Paris w 1982 roku udowodnili, że powyższego twierdzenia nie da się dowieść na gruncie arytmetyki Peana, mimo iż jest ono „intuicyjnie prawdziwe", a także prawdziwe w ZFC. Dlatego nawet, gdy zajmujemy się tylko liczbami naturalnymi, może się okazać, że potrzebny nam będzie system silniejszy niż aksjomatyka Peana.

Twierdzenie 5.3 (Kirby, Paris 1982 [23])**.** *Istnieje model arytmetyki Peana, w której nie jest prawdziwe twierdzenie Goodsteina.*

5.3. Logiki wyższych rzędów

Dotychczas mieliśmy do czynienia z logiką pierwszego rzędu (rachunkiem predykatów). Istnieją jednak inne „logiki". Najważniejszym z rozszerzeń logiki pierwszego rzędu jest logika drugiego rzędu. W logice pierwszego rzędu mogliśmy kwantyfikować po zmiennych, ale nie mogliśmy kwantyfikować po zbiorach, ani po formułach logicznych. W logice drugiego rzędu mamy dwa rodzaje zmiennych oraz stałych: x - element oraz X - zbiór. Mamy też dodatkowy symbol $\in$. Zmiennych i stałych obu typów nie można ze sobą mieszać tj. termowi odpowiada albo element, albo zbiór. Wprowadza się też dodatkowe aksjomaty logiczne (których nie będziemy podawać). Logika drugiego rzędu pozwala nam wyrazić w postaci zdań to, co było nieosiągalne w logice pierwszego rzędu. Dla przykładu, aby w arytmetyce Peana pierwszego drugiego rzędu należy zastępujemy schemat aksjomatów indukcji przez pojedynczy aksjomat indukcji:

$$\forall A\ (0 \in A \wedge (x \in A \Rightarrow (x+1) \in A) \Rightarrow \forall y\ y \in A).$$

Łatwo sprawdzić, że powyższy aksjomat indukcji implikuje schemat aksjomatów indukcji pierwszego rzędu. Zwróćmy uwagę, że arytmetyka Peana drugiego rzędu będąca rozszerzeniem arytmetyki Peana pierwszego rzędu ma skończenie wiele aksjomatów. Pokazuje to większą moc ekspresywną logiki drugiego rzędu.

Poza logiką drugiego rzędu istnieją jeszcze logiki wyższych rzędów. W logice trzeciego rzędu możemy mówić o zbiorach zbiorów obiektów, w logice czwartego rzędu o zbiorach zbiorów zbiorów obiektów, itd.

6. Aksjomatyczna teoria mnogości

Współczesna matematyka opiera się na logice i teorii mnogości. W pierwszym semestrze uczyli się Państwo o tzw. naiwnej teorii mnogości. Jest to teoria zbiorów bez formalnie zdefiniowanych aksjomatów. Potrzebę aksjomatyzacji teorii mnogości dobrze wyjaśnia sformułowany w 1901 roku paradoks Russela. Jest on znany w następującej opisowej formie (paradoksu golibrody) pochodzącej z książki „Pi razy drzwi" autorstwa Barrowa:

„Cyrulik sewilski goli w Sewilli wszystkich tych i tylko tych, którzy nie golą się sami. Czy cyrulik goli się sam?".

Jeśli cyrulik goli sam siebie, to nie może sam się golić. Jeśli cyrulik nie goli siebie, to musi sam się golić. Dochodzimy więc do sprzeczności. W języku teorii mnogości paradoks Russela przyjmuje następującą formę $X = \{x \colon x \notin X\}$. Analogicznie jak wcześniej, jeśli x należy do X, to X nie należy do X, a jeśli X nie należy do X, to X należy do X.

Naiwna teoria mnogości prowadzi więc do sprzeczności. Potrzebujemy zatem bardziej restrykcyjnej wersji teorii mnogości, która z jednej strony będzie pozwalała na możliwie „najbogatszą" matematykę, a z drugiej strony pozwoli uniknąć sprzeczności jak tej z paradoksu Russella.

W tym rozdziale poznamy współczesną wersję teorii mnogości, która opiera się na aksjomatyce Zermela-Fraenkla-Skolema (ZFC, gdzie C pochodzi od Choice - aksjomatu wyboru). Pierwotna wersja aksjomatów Zermela powstała jako zbiór aksjomatów, które zostały wykorzystane przez Zermela w dowodzie twierdzenia o tym, że każdy zbiór da się dobrze uporządkować. Aksjomatyka ta została następnie zmodyfikowana przez Skolema, Fraenkla i von Neumanna. Aksjomatykę ZFC możemy wyrazić w logice pierwszego rzędu. W związku z tym w ZFC istnieją obiekty tylko jednego typu - zbiory. Każdy więc obiekt, który się pojawia, jest zbiorem. Nie rozróżniamy więc między zbiorami a elementami. Do języka aksjomatyki ZFC

należy tylko jeden symbol $\in$, który odpowiada dwuargumentowej relacji należenia. AWszystkie aksjomaty ZFC można wyrazić w logice pierwszego rzędu. Aksjomatyka ZFC opiera się na następujących aksjomatach:

I Aksjomat ekstensjonalności,
II Aksjomat pary,
III Aksjomat sumy,
IV Aksjomat zbioru potęgowego,
V Aksjomat nieskończoności,
VI Aksjomat regularności,
VII Schemat aksjomatów rozdzielania,
VIII Schemat aksjomatów zastępowania,
IX Aksjomat wyboru (AC).

I Aksjomat ekstensjonalności mówi o tym, że dwa zbiory są sobie równe wtedy i tylko wtedy, gdy mają dokładnie te same elementy. Ekstensjonalność jest przeciwieństwem intencjonalności. W systemie intencjonalnym dwa zbiory o tych samych elementach mogłyby się od siebie różnić, jeśli byłyby inaczej zdefiniowane („z innego powodu", „intencji"). Dla przykładu zbiór liczb naturalnych oraz zbiór tych liczb całkowitych, które są większe lub równe od zera, mogłyby być innymi zbiorami, ponieważ są inaczej zdefiniowane (inna była intencja oraz formuła je definiująca).

Aksjomat ekstensjonalności może wydawać się aksjomatem zubożającym teorię, ponieważ nie dopuszczamy istnienia nieekstensjonalnych zbiorów. Nie jest to jednak istotne ograniczenie z punktu widzenia matematyki. Dotyczy to zarówno teorii mnogości, jak i pozostałych działów matematyki. W teoriach ekstensjonalnych nie możemy wprawdzie uzyskać dwóch różnych zbiorów o dokładnie tych samych elementach, ale możemy stworzyć „kopię" dowolnego niepustego (ważne!) zbioru, która będzie miała analogiczne pożądane przez nas własności (np. algebraiczne, teoriografowe czy porządkowe). Dla przykładu możemy skonstruować tyle kopii grafu K_1, ile byśmy chcieli.

Dodajmy jeszcze, że aksjomat ekstensjonalności jest niezależny od pozostałych aksjomatów ZFC.

II Aksjomat pary mówi o tym, że jeśli mamy dwa zbiory x i y, to istnieje zbiór Z, którego elementami są dokładnie x i y. Zbiór Z nazywamy parą (nieuporządkowaną) i oznaczamy go przez $\{x, y\}$. Zwróćmy uwagę, że w aksjomacie pary nie zakładamy, że zbiory x i y są od siebie różne. Również w definicji pary nie ma takiego założenia. Oznacza to, że zbiór Z, którego jedynymi elementami są zbiory x oraz x, czyli $Z = \{x, x\}$ również jest parą, chociaż ma tylko jeden element! Zbiór $Z = \{x, x\}$ oznaczamy przez $Z = \{x\}$ i nazywamy go singletonem zbioru x. Aksjomat pary pozwala nam także na zdefiniowanie pary uporządkowanej. Niech a oraz

b będą dowolnymi zbiorami. Wówczas zbiór $\{a, \{a, b\}\}$ nazywamy parą uporządkowaną o elementach a i b i oznaczamy ją przez (a, b). Powyższa definicja pochodzi od Kuratowskiego. Czasami używane są także inne definicje pary uporządkowanej, z których najważniejszą jest para w sensie Morse'a-Kelleya - szczególnie przydatna, gdy mamy do czynienia z teorią klas (ogólniejszą niż teoria zbiorów). Najważniejsze własności, które powinna spełniać para uporządkowana, są następujące. Po pierwsze, para uporządkowana powinna być możliwa do utworzenia dla dowolnych i niekoniecznie różnych elementów a i b. Po drugie musi zachodzić własność: dla każdych a, b, c, d mamy $(a, b) = (c, d)$ wtedy i tylko wtedy, gdy $a = c$ oraz $b = d$. Po trzecie przydatne może być to, żeby była ona prosta z teoriomnogościowego punktu widzenia.

Jak najprościej zdefiniować k-krotki uporządkowane (dla naturalnej liczby $k \geqslant 3$)?

III Aksjomat sumy mówi o tym, że dla dowolnej rodziny $\mathcal{A}$ istnieje zbiór Y, którego elementami są dokładnie te zbiory a, takie że $a \in A \in \mathcal{A}$ dla pewnego $A \in \mathcal{A}$. Innymi słowy zbiór Y składa się z tych zbiorów, które należą do pewnego zbioru z rodziny $\mathcal{A}$. Zbiór Y, którego istnienie zapewnia nam aksjomat, nazywamy sumą zbioru $\mathcal{A}$ i oznaczamy go przez $Y = \bigcup \mathcal{A}$.

IV Aksjomat zbioru potęgowego mówi o tym, że dla zbioru X istnieje zbiór $P(X)$, którego elementami są dokładnie wszystkie podzbiory X. Zbiór $P(X)$ nazywamy zbiorem potęgowym zbioru X. Zwróćmy uwagę, że sam aksjomat zbioru potęgowego nie mówi nam nic o postaci podzbiorów X, ani nawet o ich istnieniu. Do tego będą nam potrzebne inne aksjomaty.

V Aksjomat nieskończoności mówi o tym, że istnieje zbiór nieskończony. Może być on wyrażony w różnej postaci. Prawdopodobnie najpopularniejszą formą tego aksjomatu jest postulowanie istnienia zbioru S, do którego należy zbiór pusty, a jeśli x należy do zbioru S, to $x \cup \{x\}$ również należy do zbioru S. Zbiór S o tej własności nazywamy zbiorem induktywnym. Dla nas najwygodniej będzie przyjąć, że aksjomat nieskończoności postuluje istnienie zbioru liczb naturalnych, który formalnie zdefiniujemy w części poświęconej liczbom porządkowym.

VI Aksjomat regularności mówi o tym, że w każdym niepustym zbiorze X istnieje element minimalny ze względu na $\in$ (traktowanej jako relacja silnego częściowego porządku na zbiorze X). Dokładniej mówi on, że dla każdego niepustego zbioru X istnieje jego element z, taki że jeśli $u \in X$, to $u \notin z$. Jedną z konsekwencji aksjomatu regularności jest to, że dla każdego zbioru x mamy $x \notin x$. W teorii, która zawiera aksjomat wyboru (o którym będzie mowa dalej) aksjomat regularności można sformułować w nieco bardziej intuicyjnej postaci: „Nie istnieje nieskończony zstępujący ciąg zbiorów, tj. ciąg $(x_0, \dots)$, taki że $x_0 \ni x_1 \ni x_2 \ni \dots$". Na ćwiczeniach poznamy dokładniej zastosowania aksjomatu regularności, jego

wpływ na matematykę i to, dlaczego znalazł się wśród aksjomatów ZFC. Aksjomat regularności jest niezależny od pozostałych aksjomatów ZF i od aksjomatu wyboru.

VII Aksjomat rozdzielania, zwany także aksjomatem podzbiorów lub aksjomatem wycinania, dla formuły pierwszego rzędu φ mówi o tym, że dla każdego zbioru Y istnieje zbiór Z dokładnie tych elementów zbioru Y, które mają własność φ, czyli $Z = \{y \in Y : \varphi(z)\}$. Mówimy o schemacie aksjomatów rozdzielania, ponieważ w logice pierwszego rzędu nie możemy kwantyfikować po formułach logicznych, więc dla każdej formuły logicznej mamy osobny aksjomat rozdzielania.

W naiwnej teorii mnogości zbiory definiowane były jako ogół tych elementów, które spełniają jakąś formułę, czyli $Z = \{x : \varphi(x)\}$. Jest to tak zwany aksjomat nieograniczonego wycinania, który, jak pokazuje paradoks Russella, prowadzi do sprzeczności. Aksjomaty wycinania mają na celu uniknięcie paradoksu Russella przez ograniczenie się jedynie do elementów pewnego istniejącego zbioru, a także mają nam zapewnić istnienie możliwie wielu „różnych" (w znaczeniu rozmaitych) zbiorów. W części drugiej skryptu postaramy się udzielić odpowiedzi na pytanie, czy zastąpienie aksjomatów nieograniczonego wycinania aksjomatami wycinania faktycznie pozwoliło nam uniknąć sprzeczności w aksjomatycznej teorii mnogości.

Zanim przedstawimy aksjomaty zastępowania, podamy definicję formuły funkcyjnej. Mówimy, że formuła φ jest funkcyjna, jeśli dla każdego x istnieje dokładnie jeden y, taki że $\varphi(x, y)$.

VIII Aksjomat zastępowania dla formuły φ mówi, że jeśli φ jest formułą funkcyjną, to dla każdego A istnieje zbiór B, którego elementami są dokładnie te zbiory b, dla których istnieje $a \in A$, takie że $\varphi(a, b)$. Podobnie jak dla aksjomatów rozdzielania, mamy osobny aksjomat dla każdej formuły. Innymi słowy, schemat aksjomatów zastępowania mówi, że jeśli funkcja jest zadana przez dziedzinę oraz formułę funkcyjną (np. przez wzór funkcji), to istnieje obraz tej funkcji (jako zbiór).

Selektorem rodziny $\mathcal{A}$ nazywamy zbiór S, taki że dla każdego $A \in \mathcal{A}$ istnieje dokładnie jeden $a \in A$, taki że $a \in S$. Selektor jest tym samym, co znany ze wstępu do matematyki dyskretnej system różnych reprezentantów. Używa się także nazwy transwersala. Funkcją wyboru f dla rodziny $\mathcal{A}$ nazywamy funkcję o dziedzinie $\mathcal{A}$, która każdemu zbiorowi A należącemu do rodziny $\mathcal{A}$ przyporządkowuje jej element $a \in A \in \mathcal{A}$. Odnotujmy, że jeśli rodzina $\mathcal{A}$ jest rodziną zbiorów parami rozłącznych, to $\mathcal{A}$ ma funkcję wyboru wtedy i tylko wtedy, gdy istnieje selektor rodziny $\mathcal{A}$.

IX Aksjomat wyboru mówi, że dla każdej rodziny zbiorów parami rozłącznych istnieje jej selektor. Równoważna jego wersja mówi, że każda rodzina ma funkcję wyboru. Zwróćmy uwagę, że w drugiej formie aksjomatu wyboru nie ma założenia o tym, że elementy rodziny są zbiorami parami rozłącznymi. Jest ona więc „pozornie"

silniejsza od pierwszej formy, ale tylko pozornie, ponieważ obie wersje okazują się sobie równoważne. Aksjomat wyboru omówimy dokładniej w dalszej części skryptu oraz na ćwiczeniach.

Aksjomat wyboru jest niezależny od pozostałych aksjomatów ZFC. Teorię, otrzymaną z ZFC poprzez usunięcie ze zbioru aksjomatów ZFC aksjomatu wyboru, oznaczamy przez ZF. To, że dane zdanie φ jest równoważne aksjomatowi wyboru, należy rozumieć tak, że są one równoważne sobie na gruncie aksjomatyki ZF, tj. $Con_L(ZF \cup \{\varphi\}) = Con_L(ZFC)$, gdzie L jest językiem teorii mnogości ZFC. Dla przykładu obie podane wersje aksjomatu wyboru są sobie równoważne. Nieco później poznamy inne zdania równoważne aksjomatowi wyboru.

6.1. Teoria klas NBG

Aksjomatyczna teoria mnogości była potrzebna między innymi dlatego, że w naiwnej teorii mnogości pojawiły się paradoksy - sprzeczności związane np. ze zbiorem wszystkich zbiorów (nie zawsze jednak paradoks oznacza sprzeczność, patrz podany później paradoks Skolema). Czasami przydatny jest jednak na przykład obiekt, który przypomina zbiór wszystkich zbiorów albo (omówione później) zbiory wszystkich liczb porządkowych czy wszystkich liczb kardynalnych. Jedną z teorii rozszerzających ZFC, która zajmuje się takimi obiektami, jest teoria klas NBG pochodząca od von Neumanna, Bernaysa i Gödla. Podstawowymi obiektami w tej teorii są klasy, które rozszerzają pojęcie zbioru w tym sensie, że każdy zbiór jest klasą, ale nie każda klasa jest zbiorem. Zbiorem w NBG nazywamy taką klasę, która należy do pewnej klasy. Klasą właściwą nazywamy klasę, która nie jest zbiorem. Przykładem klasy właściwej jest klasa wszystkich zbiorów, która nie może być zbiorem, bo wtedy pojawia się paradoks Russela. Oprócz symbolu należenia $\in$ potrzebny nam jest też predykat jednoargumentowy T, który będzie zaznaczał, czy klasa jest zbiorem. W NBG mamy podobne aksjomaty, co w ZFC. Najważniejszą różnicą jest to, że dzięki klasom można podać skończony zbiór aksjomatów tego systemu. NBG jest konserwatywnym rozszerzeniem ZFC, co oznacza, że każde twierdzenie NBG, które jest wyrażalne w języku ZFC (a więc nie mówi o klasach, ale wyłącznie o zbiorach) jest twierdzeniem ZFC. Pozwala to nam na nieformalne używanie pojęcia klasy w dowodach twierdzeń w ZFC. Obiekty typu funkcja czy relacja można w naturalny sposób rozszerzyć na klasy. Pozwala to między innymi na zastąpienie schematów zastępowania dla zbiorów poprzez pojedynczy aksjomat zastępowania.

Aksjomaty NBG są następujące:

I Aksjomat sortowania,
II Aksjomat ekstensjonalności dla klas,
III Schemat aksjomatów istnienia klas,

IV Aksjomat pary,
V Aksjomat sumy,
VI Aksjomat zbioru potęgowego,
VII Aksjomat nieskończoności,
VIII Aksjomat regularności,
IX Aksjomat zastępowania,
X Aksjomat wyboru (AC),
XI Aksjomat ograniczenia rozmiaru (axiom of limitation of size).

I Aksjomat sortowania mówi o tym, że klasa jest zbiorem wtedy i tylko wtedy, gdy należy do pewnej klasy.

II Aksjomat ekstensjonalności dla klas mówi o tym, że dwie klasy są sobie równe wtedy i tylko wtedy, gdy mają te same elementy.

III Aksjomat istnienia klas dla formuły φ mówi o tym, że istnieje klasa $\mathcal{V}$ dokładnie tych zbiorów, które mają własność φ. Zwróćmy uwagę, że klasa $\{x\colon x = x\}$ jest klasą wszystkich zbiorów. Jej istnienie nie powoduje paradoksu Russela, ponieważ nie może ona być swoim elementem jako klasa właściwa. Klasę tę oznaczamy przez $\mathbb{V}$ i nazywamy klasą uniwersalną, uniwersum von Neumanna lub po prostu klasą wszystkich zbiorów. Podobnie jak w ZFC mamy tu do czynienia ze schematem aksjomatów. W tym jednak przypadku jesteśmy w stanie zastąpić schemat aksjomatów pojedynczym aksjomatem wyczerpywania:

$$\forall\, R\,(((\forall z\ z \in R \Leftrightarrow (\exists x \exists y\ z = (x,y)))) \Rightarrow \exists A\ \forall x\,(x \in A\ \Leftrightarrow (x,x)))\,.$$

Lewa strona zewnętrznej implikacji mówi o tym, że R jest relacją dwuargumentową. Prawa strona tej implikacji mówi o tym, że istnieje klasa złożona ze wszystkich x takich, że (x,x) należy do R.

IV Aksjomat pary mówi o tym, że dla każdych dwóch (niekoniecznie różnych) zbiorów istnieje zbiór, którego są one jedynymi elementami.

V Aksjomat sumy mówi o tym, że dla danego zbioru jego suma mnogościowa jest zbiorem.

VI Aksjomat zbioru potęgowego mówi o tym, że dla danego zbioru istnieje jego zbiór potęgowy.

VII Aksjomat nieskończoności mówi o tym, że istnieje zbiór induktywny.

VIII Aksjomat regularności mówi o tym, że w każdej niepustej klasie istnieje element $\in$-minimalny.

IX Aksjomat zastępowania mówi o tym, że dla danej klasy F oraz zbioru X, jeśli F jest funkcją oraz $X \subseteq Dom(F)$, to obraz $F(X)$ jest zbiorem.

X Aksjomat wyboru mówi o tym, że dla każdej rodziny zbiorów niepustych istnieje jej selektor.

XI Aksjomat ograniczenia rozmiaru mówi o tym, że klasa jest właściwa wtedy i tylko wtedy, gdy istnieje surjekcja z tej klasy na klasę wszystkich zbiorów.

Często do NBG dodaje się jeszcze rozszerzenie aksjomatu wyboru na klasy, czyli aksjomat globalnego wyboru (AGC). Mówi on o tym, że dla każdej klasy zbiorów niepustych istnieje jej selektor. Aksjomat globalnego wyboru jest niezależny od aksjomatów NBG. Po dodaniu do teorii NBG tego aksjomatu pozostaje ona konserwatywnym rozszerzeniem ZFC. podobnie jak dla aksjomatu wyboru istnieją równoważne wersje aksjomatu globalnego wyboru. Między innymi wersja twierdzenia Zermela mówiąca o tym, że każdą klasę da się dobrze uporządkować oraz wzmocnienie aksjomatu ograniczenia rozmiaru z surjekcji na bijekcję.

6.2. Zbiory dobrze uporządkowane

Przypomnimy najpierw podstawową terminologię dotyczącą zbiorów częściowo uporządkowanych.

Uwaga! Przyjmujemy tutaj konwencję, że częściowym porządkiem będziemy określać relację $<$ silnego porządku, a nie relację $\leqslant$ słabego porządku. Nieco później wyjaśnimy, dlaczego tak będzie nam wygodniej.

Definicja 6.1. *Niech R będzie relacją dwuargumentową określoną na zbiorze X. Wówczas:*

I *Jeśli dla każdego elementu $x \in X$ zachodzi xRx, to mówimy, że relacja R jest zwrotna.*

II *Jeśli dla każdego elementu $x \in X$ zachodzi $\neg xRx$, to mówimy, że relacja R jest przeciwzwrotna.*

III *Jeśli dla każdych dwóch elementów $x, y \in X$ zachodzi $xRy \Rightarrow yRx$, to mówimy, że relacja R jest symetryczna.*

IV *Jeśli dla każdych dwóch różnych elementów $x, y \in X$ zachodzi $xRy \Rightarrow \neg yRx$, to mówimy, że relacja R jest antysymetryczna.*

V *Jeśli dla każdych trzech elementów $x, y, z \in X$ zachodzi $xRy \wedge xRz \Rightarrow xRz$, to mówimy, że relacja R jest przechodnia.*

VI *Jeśli dla każdych dwóch elementów $x, y \in X$ zachodzi $xRy \vee yRx$, to mówimy, że relacja R jest spójna.*

Definicja 6.2. *Niech $<$ będzie relacją dwuargumentową określoną na zbiorze X. Jeśli $<$ jest przechodnia, antysymetryczna i przeciwzwrotna, to $<$ nazywamy częściowym porządkiem na X. Wówczas parę uporządkowaną $(X, <)$ nazywamy zbiorem częściowo uporządkowanym. Ponadto przez $\leqslant$ oznaczamy relację słabego częściowego porządku na X skojarzoną z $<$. To znaczy $x \leqslant y \Leftrightarrow x < y \vee x = y$.*

Definicja 6.3. *Spójny częściowy porządek $<$ na zbiorze X nazywamy porządkiem liniowym, a $(X, <)$ nazywamy zbiorem liniowo uporządkowanym. Jeśli pewien podzbiór Y wraz z porządkiem $<_Y$ indukowanym przez częściowy porządek $<$ na X jest zbiorem liniowo uporządkowanym, to Y nazywamy łańcuchem.*

Definicja 6.4. *Niech $<$ będzie relacją częściowego porządku określoną na zbiorze X. Wówczas:*

I *Mówimy, że x i y są porównywalne, jeśli $x = y$, $x < y$ lub $y < x$.*

II *Mówimy, że x jest elementem minimalnym, jeśli nie istnieje element $y \in X$, taki że $y < x$.*

III *Mówimy, że x jest elementem najmniejszym, jeśli dla każdego elementu $y \in X$ różnego od X zachodzi $x < y$.*

IV *Mówimy, że x jest elementem maksymalnym, jeśli nie istnieje element $y \in X$, taki że $x < y$.*

V *Mówimy, że x jest elementem największym, jeśli dla każdego elementu $y \in X$ różnego od X zachodzi $y < x$.*

Definicja 6.5. *Niech $<$ będzie relacją częściowego porządku określoną na zbiorze X i niech Y będzie pewnym podzbiorem zbioru X. Wówczas:*

I *Mówimy, że n jest minorantą zbioru Y, jeśli $x < y$ dla każdego elementu $y \in Y$.*

II *Niech N będzie zbiorem minorant zbioru Y, jeśli n jest elementem najmniejszym zbioru N, to n nazywamy infimum zbioru Y lub kresem dolnym zbioru Y i przyjmujemy oznaczenie $n = \inf Y$.*

III *Mówimy, że m jest majorantą zbioru Y, jeśli $y < x$ dla każdego elementu $y \in Y$.*

IV *Niech M będzie zbiorem majorant zbioru Y, jeśli m jest elementem najmniejszym zbioru M, to m nazywamy supremum zbioru Y lub kresem górnym zbioru Y i przyjmujemy oznaczenie $m = \sup Y$.*

Definicja 6.6. *Mówimy, że zbiór częściowo uporządkowany $(X, <)$ jest dobrze uporządkowany, jeśli każdy niepusty podzbiór zbioru X ma element najmniejszy. Wówczas $<$ nazywamy dobrym porządkiem na X.*

Teraz udowodnimy kilka faktów o zbiorach dobrze uporządkowanych. Dzięki przyjętej przez nas konwencji homomorfizm zbiorów częściowo uporządkowanych jest tym samym, co funkcja rosnąca.

Twierdzenie 6.7. *Niech $(X, <)$ będzie zbiorem dobrze uporządkowanym. Jeśli $f\colon X \to X$ jest funkcją rosnącą, to dla każdego $x \in X$ mamy $f(x) \geqslant x$.*

Dowód. Załóżmy, że zbiór $W = \{x \in X\colon f(x) < x\}$ jest niepusty. Niech z będzie jego elementem najmniejszym. Oznaczmy $w = f(z)$. Wówczas $f(w) < f(z)$, bo

$w < z$, a f jest funkcją rosnącą, ale $f(w) \geqslant w$, bo w nie należy do zbioru W, co prowadzi do sprzeczności. ■

Twierdzenie 6.8. *Jedynym automorfizmem zbioru dobrze uporządkowanego jest identyczność.* ■

Obserwacja 6.9. *Jeśli W_1 i W_2 są z izomorficznymi zbiorami dobrze uporządkowanymi, to izomorfizm z W_1 do W_2 jest jedyny.* ■

Definicja 6.10. *Jeśli W jest zbiorem dobrze uporządkowanym i u jest jego elementem, to zbiór $\{x \in W : x < u\}$ nazywamy odcinkiem początkowym zbioru W zadanym przez u.*

Twierdzenie 6.11. *Zbiór dobrze uporządkowany nie jest izomorficzny ze swoim odcinkiem początkowym.* ■

Twierdzenie 6.12 (Hessenberg 1906 [21])**.** *Niech W_1 oraz W_2 będą zbiorami dobrze uporządkowanymi. Wówczas zachodzi dokładnie jeden z przypadków:*

I *W_1 jest izomorficzne z W_2,*

II *W_1 jest izomorficzne z pewnym odcinkiem początkowym W_2,*

III *W_2 jest izomorficzne z pewnym odcinkiem początkowym W_1.*

Dowód. Dla $u \in W_i$ oznaczmy przez $W_i(u)$ odcinek początkowy W_i zadany przez u. Niech

$$f = \{(x, y) \in W_1 \times W_2 : W_1(x) \cong W_2(y)\}.$$

Z poprzedniego twierdzenia wynika, że f jest injekcją. Ponadto, f i f^{-1} są funkcjami rosnącymi.

Jeśli dziedziną f jest W_1 i obrazem f jest W_2, to W_1 i W_2 są izomorficzne. Zachodzi pierwszy przypadek.

Jeśli dziedziną f jest W_1, ale obrazem f nie jest W_2, to obrazem f jest odcinek początkowy W_2 (dlaczego? ćwiczenie). Zachodzi drugi przypadek.

Jeśli dziedzina f nie jest równa W_1, to podobnie jak wyżej dziedziną f jest odcinek początkowy W_1, a obrazem f jest W_2. Zachodzi trzeci przypadek. ■

6.3. Liczby porządkowe

W tym rozdziale zdefiniujemy liczby porządkowe i podamy ich podstawowe własności. Idea liczb porządkowych jest taka, aby każdemu zbiorowi dobrze uporządkowanemu przypisać pewną liczbę porządkową w taki sposób, że dwa zbiory mają przypisaną tę samą liczbę porządkową wtedy i tylko wtedy, gdy są one izomorficzne. Pomysł liczb porządkowych pochodzi od Cantora, a jego współczesne ujęcie pochodzi od von Neumanna.

Definicja 6.13. *Mówimy, że zbiór A jest tranzytywny lub przechodni, jeśli każdy jego element zbioru A jest równocześnie jego podzbiorem.*

Definicja 6.14. *Zbiór α nazywamy liczbą porządkową, jeśli jest tranzytywny i jest dobrze uporządkowany przez relację $x < y \Leftrightarrow x \in y$.*

Liczby porządkowe oznaczamy małymi greckimi literami. Klasę wszystkich liczb porządkowych oznaczamy przez $\mathbb{ON}$. Liczby porządkowe nie tworzą zbioru, co wynika z następnego twierdzenia.

Twierdzenie 6.15. *Jeśli α jest liczbą porządkową, to $\alpha \cup \{\alpha\}$ jest liczbą porządkową.* ■

Otrzymaną liczbę porządkową $\alpha \cup \{\alpha\}$ nazywamy następnikiem liczby α i oznaczamy ją przez $\alpha + 1$.

Definicja 6.16. *Zbiorem liczb naturalnych nazywamy najmniejszy w sensie inkluzji zbiór zawierający 0 oraz zamknięty na operację brania następnika.*

Zbiór liczb naturalnych w zależności od kontekstu oznaczamy przez $\mathbb{N}, \omega, \omega_0$ lub przez $\aleph_0$. Odnotujmy, że zbiór liczb naturalnych ZAWSZE jest włącznie z zerem. Istnienie zbioru liczb naturalnych wynika z aksjomatu nieskończoności. Zauważmy, że jedyną liczbą naturalną, która nie jest następnikiem żadnej innej liczby naturalnej (ani porządkowej) jest zero.

Definicja 6.17. *Jeśli liczba porządkowa α jest postaci $\alpha = \beta + 1$ dla pewnego, to α nazywamy liczbą porządkową następnikową. W przeciwnym przypadku mówimy, że α jest liczbą porządkową graniczną.*

Poniższe twierdzenie jest rozszerzeniem twierdzenia o indukcji zupełnej na wszystkie liczby porządkowe.

Twierdzenie 6.18 (von Neumann 1923 [41])**.** *Niech α będzie dowolną liczbą porządkową i niech A będzie jej podzbiorem. Załóżmy, że dla każdego $\beta \in \alpha$ z faktu,*

że wszystkie liczby porządkowe mniejsze od β należą do A, wynika, że β należy do A. Wówczas $A = \alpha$. ■

Z powyższego twierdzenia możemy korzystać w sposób następujący: Załóżmy, że własność φ, która jest zdefiniowana dla liczb porządkowych mniejszych niż α, ma następujące własności:

LP 1 Własność φ zachodzi dla zera.

LP 2 Jeśli własność φ zachodzi dla liczby porządkowej β, to zachodzi dla jej następnika, o ile ten jest mniejszy niż α.

LP 3 Jeśli dla liczby porządkowej granicznej β własność φ zachodzi dla wszystkich liczb porządkowych od niej mniejszych, to zachodzi dla liczby porządkowej β.

Wówczas każda liczba porządkowa mniejsza niż α ma własność φ.

Czasami chcemy, aby pewna własność φ zachodziła nie tylko dla pewnych liczb porządkowych mniejszych niż pewne α, ale dla wszystkich liczb porządkowych. Użyteczny jest w tym następujący schemat twierdzeń.

Twierdzenie 6.19 (von Neumann 1923 [41])**.** *Załóżmy, że dla każdego β z faktu, że wszystkie liczby porządkowe mniejsze od β mają własność φ, wynika, że β ma własność φ. Wówczas φ zachodzi dla każdej liczby porządkowej.* ■

Podobnie jak przy aksjomatach ZFC, nie mamy tu do czynienia z twierdzeniem, ale ze schematem twierdzeń, ponieważ dla każdej formuły φ mamy osobne twierdzenie.

W NBG twierdzenia o indukcji pozaskończonej 6.19 oraz 6.20 można wyrazić ogólniej w postaci jednego twierdzenia.

Twierdzenie 6.20 (von Neumann 1923 [**?**], o indukcji pozaskończonej, NBG)**.** *Niech $A \subseteq \mathbb{ON}$. Załóżmy, że dla każdego $\beta \in \alpha$ z faktu, że wszystkie liczby porządkowe mniejsze od β należą do A, wynika, że β należy do A. Wówczas $A = \mathbb{ON}$.* ■

Zauważmy, że powyższe twierdzenie nie jest schematem twierdzeń, a pojedynczym twierdzeniem. NBG daje taką możliwość z powodu większej ekspresywności języka. Kwantyfikowanie po klasach w NBG daje nam podobną możliwość co kwantyfikowanie po formułach mówiących o zbiorach. Nadal nie możemy jednak kwantyfikować po formułach mówiących o klasach!

Indukcji pozaskończonej można użyć w celu zdefiniowania różnych ciągów pozaskończonych podobnie.

Definicja 6.21. *Ciągiem pozaskończonym nazywamy dowolną funkcję o dziedzinie α, gdzie $\alpha \in \mathbb{ON}$ lub o dziedzinie $\mathbb{ON}$.*

Poniższe twierdzenie o rekursji (rekurencji) pozaskończonej jest uogólnieniem analogicznego twierdzenia dla liczb naturalnych na liczby porządkowe.

Twierdzenie 6.22 (von Neumann 1923 [41], o rekursji pozaskończonej, NBG)**.** *Niech $\mathbb{X}$ będzie klasą taką, ze $\mathbb{ON} \subseteq \mathbb{X}$. Niech $f\colon \mathbb{X} \to \mathbb{X}$ będzie funkcją. Wówczas istnieje dokładnie jeden ciąg pozaskończony $(g_\alpha\colon \alpha \in \mathbb{ON})$, taki że $g_\alpha = f(g|_\alpha)$.* ■

Przykładem zastosowania indukcji pozaskończonej jest rozszerzenie działania dodawania na wszystkie liczby porządkowe. Zdefiniujemy jeszcze jedną przydatną własność ciągów:

Definicja 6.23. *Mówimy, że rosnąca funkcja $f\colon \mathbb{ON} \supseteq Dom(f) \to \mathbb{ON}$, jest ciągła albo też normalna, jeśli dla każdego $\alpha \in \mathbb{LIM}$ zachodzi $f(\alpha) = \sup\{f(\beta) : \beta < \alpha\}$.*

Nie będziemy definiować pojęcia funkcji ciągłej dla funkcji, które nie są rosnące. Możemy przejść do definicji dodawania.

Definicja 6.24. *Niech α i β będą liczbami porządkowymi. Sumą liczb α i β nazywamy liczbę $\alpha + \beta$ zdefiniowaną w następujący sposób:*
1) $\alpha + 0 = \alpha, \forall \alpha \in \mathbb{ON}$,
2) $\alpha + (\beta + 1) = (\alpha + \beta) + 1, \forall \alpha, \beta \in \mathbb{ON}$.
3) *Dodawanie prawostronne (tj. β jest argumentem przy funkcji $f(\beta) = \alpha + \beta$) jest funkcją ciągłą.*

Istotne jest to, że dodawanie liczb porządkowych nie jest przemienne, w szczególności $\omega + 1 \neq 1 + \omega = \omega$. Analogicznie do dodawania możemy zdefiniować mnożenie i potęgowanie liczb porządkowych.

Przy badaniu liczb porządkowych przydatne będą następujące własności:

Twierdzenie 6.25. *Zachodzą następujące fakty:*
$\mathbb{ON}$ 1 $0 = \emptyset$ *jest liczbą porządkową.*
$\mathbb{ON}$ 2 *Jeśli α jest liczbą porządkową, i $\beta \in \alpha$, to β jest liczbą porządkową.*
$\mathbb{ON}$ 3 *Dla każdej liczby porządkowej α mamy $\alpha = \{\beta \in \mathbb{ON}\colon \beta < \alpha\}$.*
$\mathbb{ON}$ 4 *Jeśli C jest niepustym zbiorem liczb porządkowych, to $\bigcap C$ jest liczbą porządkową i $\bigcap C = \inf C = \min C$.*
$\mathbb{ON}$ 5 *Jeśli C jest niepustym zbiorem liczb porządkowych, to $\bigcup C$ jest liczbą porządkową i $\bigcup C = \sup C$.*
$\mathbb{ON}$ 6 *Jeśli $\alpha \neq \beta$ są liczbami porządkowymi i $\alpha \subset \beta$, to $\alpha \in \beta$.*
$\mathbb{ON}$ 7 *Jeśli α, β są liczbami porządkowymi, to zachodzi $\alpha \subseteq \beta$ lub $\beta \subseteq \alpha$.*
$\mathbb{ON}$ 8 *Jeśli α i β są liczbami porządkowymi, to zachodzi $\alpha \leqslant \beta$ lub $\beta < \alpha$.*

Dowód. $\mathbb{ON}$ 1: Oczywiste. Oba warunki w definicji liczby porządkowej są pusto spełnione dla 0. Zbiór pusty jest dobrze uporządkowany, ponieważ nie ma niepustych podzbiorów, więc każdy niepusty podzbiór ma dowolną własność. Jest również tranzytywny, ponieważ każdy jego element ma dowolną własność. ■

$\mathbb{ON}$ 2: Jeśli $\beta \in \alpha$, to $\beta \subseteq \alpha$, bo α jest zbiorem tranzytywnym. Podzbiór zbioru dobrze uporządkowanego jest zbiorem dobrze uporządkowanym. Stąd β jest dobrze uporządkowane. Załóżmy teraz, nie wprost, że β nie jest zbiorem tranzytywnym. Zbiór wszystkich elementów α, które nie są tranzytywne, jest podzbiorem zbioru α. Załóżmy więc, że β jest jego najmniejszym elementem. Skoro β nie jest tranzytywne, to istnieje $\gamma \in \beta$ takie, że $\gamma \subsetneq \beta$, czyli zbiór $\gamma \setminus \beta$ jest niepusty, więc zawiera pewien element δ. Zbiór α jest tranzytywny, więc γ jest jego podzbiorem, czyli elementy zbioru γ są też elementami zbioru α. W szczególności $\delta \in \alpha$. Teraz zbiór $\{\delta, \beta\}$ jest podzbiorem zbioru α, czyli ma element najmniejszy. Nie może nim być β, bo $\delta \not< \beta$, i nie może nim być β, bo w przeciwnym przypadku mamy $\delta < \gamma < \beta < \delta$, co prowadzi do sprzeczności. ■

$\mathbb{ON}$ 3: Zbiór $\gamma = \{\beta \in \mathbb{ON} : \beta < \alpha\}$, czyli odcinek początkowy klasy $\mathbb{ON}$ zadany przez α, równoważnie możemy zapisać jako $\gamma = \{\beta \in \mathbb{ON} \colon \beta \in \alpha\}$, czyli $\gamma = \{\beta : \beta \in \alpha \wedge \beta \in \mathbb{ON}\} = \{\beta \colon \beta \in \alpha\} = \alpha$. Przy czym przedostatnia równość wynika z tego, że $\alpha \subseteq \mathbb{ON}$. ■

$\mathbb{ON}$ 4: Na ćwiczeniach. ■

$\mathbb{ON}$ 5: Na ćwiczeniach. ■

$\mathbb{ON}$ 6: Niech α będzie najmniejszą liczbą porządkową o tej własności, że jeśli $\alpha \subset \beta$, to $\alpha \notin \beta$. Niech γ będzie najmniejszym elementem zbioru $\alpha \setminus \beta$, który jest niepusty, ponieważ $\alpha \subsetneq \beta$. Wynika z tego, że α jest odcinkiem początkowym liczby β zadanym przez γ. Stąd $\alpha = \{\eta \in \beta \colon \eta < \gamma\} = \gamma \in \beta$. ■

$\mathbb{ON}$ 7: Załóżmy, że $\alpha \not\leqslant \beta$. W przeciwnym przypadku zachodzi bowiem $\mathbb{ON}$ 7 i $\mathbb{ON}$ 8 dla liczb α i β. Skoro β nie należy do α, to nie należy też do żadnego podzbioru zbioru α. Niech β będzie najmniejsze możliwe (dla danego α). Z założenia wynika, że $\alpha \not< \beta$. Oznacza to, że zbiór $\beta = \{\gamma : \gamma < \beta\}$ jest podzbiorem zbioru α. Z własności $\mathbb{ON}$ 3 wynika, że $\alpha \in \beta$, czyli $\alpha < \beta$, co daje sprzeczność. ■

$\mathbb{ON}$ 8: Wprost z $\mathbb{ON}$ 6 oraz $\mathbb{ON}$ 8. ■

Twierdzenie 6.26 (Von Neumann 1923 [41])**.** *Każdy zbiór dobrze uporządkowany jest izomorficzny z dokładnie jedną liczbą porządkową.*

Dowód. Niech W będzie zbiorem dobrze uporządkowanym. Zdefiniujmy funkcję $F(w) = \alpha$, jeśli $w \in X$ i odcinek początkowy zbioru W wyznaczony przez w jest izomorficzny z liczbą porządkową α. Zbiór F jest funkcją, a z aksjomatu zastępowania wynika, że $F(W)$ jest zbiorem. Jeśli γ jest najmniejszą liczbą porządkową, która nie należy do $F(W)$, to W jest izomorficzny z γ. ■

Definicja 6.27. *Niech $(X, <)$ będzie zbiorem dobrze uporządkowanym. Typem porządkowym zbioru $(X, <)$ nazywamy jedyną liczbę porządkową z nim izomorficzną. Oznaczamy ją przez $tp(X, <)$.*

Podobnie jak klasa wszystkich zbiorów $\mathbb{V}$ nie jest zbiorem tak i klasa $\mathbb{ON}$ wszystkich liczb porządkowych nim nie jest.

Twierdzenie 6.28 (Burali-Forti 1897 [7], Paradoks Buraliego-Fortiego)**.** *Nie istnieje zbiór wszystkich liczb porządkowych.* ■

Paradoks Buraliego-Fortiego został wprowadzony dla definicji liczb porządkowych wprowadzonej przez Cantora, ale jego dowód dla liczb porządkowych von Neumanna jest zupełnie analogiczny.

6.4. Aksjomat wyboru

W tym rozdziale omówimy pokrótce aksjomat wyboru oraz pewne jego równoważne i słabsze wersje. Przez formę aksjomatu wyboru rozumiemy każde zdanie A, takie że $Col(ZF + A) = Col(ZFC)$. Słabszą formą aksjomatu wyboru nazywamy każde zdanie A, takie że $Col(ZF) \subsetneq Col(ZF + A) \subsetneq Col(ZFC)$. Podstawowa forma aksjomatu wyboru mówi o tym, że dla każdej rodziny $\mathcal{A}$ zbiorów niepustych istnieje funkcja f, która każdemu zbiorowi $A \in \mathcal{A}$ przyporządkowuje element $a \in A$. Funkcję tę nazywamy funkcją wyboru. Najważniejszą formą aksjomatu wyboru (ważniejszą nawet od samego aksjomatu wyboru) jest twierdzenie Zermela mówiące o tym, że każdy zbiór da się dobrze uporządkować. Wynika ono bezpośrednio z następującego twierdzenia.

Twierdzenie 6.29 (ZF)(Zermelo 1904/1905 [42])**.** *Jeśli istnieje funkcja wyboru dla $P(X)$, to zbiór X da się dobrze uporządkować.*

Dowód. Niech X będzie dowolnym zbiorem i niech F będzie funkcją wyboru dla $P(X)$. Zdefiniujmy ciąg pozaskończony $c = (c_\beta \colon \beta < \alpha)$ dla pewnej liczby porządkowej α w sposób następujący:

Załóżmy, że dla pewnego $\beta < \alpha$ zdefiniowaliśmy wszystkie wyrazy ciągu pozaskończonego $(c_\gamma \colon \gamma < \beta)$. Wówczas definiujemy $c_\beta = F(X \setminus \{c_\gamma \colon \gamma < \beta\})$, o ile

nie jest to zbiór pusty. W przeciwnym wypadku kończymy konstrukcję ciągu pozaskończonego. Sprowadza się to do tego, że w każdym kroku β wybieramy element c_β spośród tych elementów zbioru X, które dotychczas nie zostały wybrane.

Pokażemy teraz, że istnieje takie α, że konstrukcja się zakończy. Załóżmy, że takie α nie istnieje. Wówczas funkcja $F : c_\beta \mapsto \beta$ jest funkcją, której obraz, na mocy aksjomatu zastępowania dla formuły definiującej funkcję F, jest zbiorem wszystkich liczb porządkowych. Z paradoksu Buraliego-Fortiego wiemy, że liczby porządkowe nie tworzą zbioru, co prowadzi do sprzeczności.

Teraz pokażemy, że do ciągu $(c_\beta \colon \beta < \alpha)$ należą wszystkie elementy zbioru X. Niech X' będzie zbiorem tych elementów X, które nie należą do tego ciągu. Wówczas $F(X \setminus \{c_\gamma \colon \gamma < \alpha\}) = F(X')$, a więc w kroku α mogliśmy wybrać pewien element $c_\alpha \in X'$, co prowadzi do sprzeczności.

Wystarczy teraz zauważyć, że zbiór X jest dobrze uporządkowany przez relację $<$ zdefiniowaną w następujący sposób: $c_\beta < c_\gamma$ wtedy i tylko wtedy, gdy zachodzi $\beta < \gamma$. ■

Podamy teraz kilka kolejnych równoważników aksjomatu wyboru. Pierwsze z nich jest bezpośrednim wnioskiem z twierdzenia 6.29.

Twierdzenie 6.30 (Zermelo 1904/1905 [42], Twierdzenie o zbiorach dobrze uporządkowanych)**.** *Każdy zbiór da się dobrze uporządkować.* ■

Twierdzenie 6.31 (Hausdorff 1914 [20])**.** *W każdym zbiorze częściowo uporządkowanym istnieje łańcuch maksymalny.* ■

Twierdzenie 6.32 (Kuratowski 1922 [26], Zorn 1935 [44], Lemat Kuratowskiego-Zorna)**.** *Niech X będzie zbiorem częściowo uporządkowanym, w którym każdy łańcuch ma ograniczenie górne. Wówczas w X istnieje element maksymalny.* ■

Twierdzenie 6.33 (Zermelo 1904/1905 [42])**.** *Każda przestrzeń liniowa ma bazę (Hamela).* ■

Twierdzenie 6.34 (Kőnig 1936 [25])**.** *Każdy (niekoniecznie skończony) graf spójny ma drzewo rozpinające.* ■

Równoważność twierdzenia o istnieniu bazy i aksjomatu wyboru udowodnił Tarski w 1924 roku [39]. Pozostałe równoważności są trywialne (może poza ostatnią, która jest jedynie bardzo łatwa) i pozostawiam je Państwu jako ćwiczenie.

Podamy teraz dwie słabsze, ale użyteczne wersje aksjomatu wyboru. Z poniższego twierdzenia, zwanego zwyczajowo lematem Kőniga, korzysta się (niekoniecznie wprost) w większości twierdzeń odnośnie grafów lokalnie skończonych, czyli takich, w których stopień każdego wierzchołka jest skończony.

Twierdzenie 6.35 (Kőnig 1936 [25], Lemat Kőniga)**.** *Niech T będzie nieskończonym grafem spójnym. Wówczas T ma wierzchołek stopnia nieskończonego lub w T istnieje promień.* ■

Mocniejsze od Lematu Kőniga, ale wciąż słabsze od pełnego aksjomatu wyboru, jest następujące twierdzenie:

Twierdzenie 6.36 (Bernays 1942 [4, 5], Zasada wyborów zależnych, DC)**.** *Niech R będzie relacją na niepustym zbiorze X taką, że dla każdego $a \in X$ istnieje $b \in X$, dla którego aRb. Wówczas istnieje ciąg $(X_i \colon i \in \omega)$, taki że $x_i R x i+1$ dla każdego $i \in \omega$.* ■

6.5. Liczby kardynalne

W tym rozdziale zajmiemy się teorią mocy, która jest uogólnieniem pojęcia liczby elementów zbioru skończonego na dowolne zbiory. Zbiór skończony X ma co najwyżej tyle elementów, co zbiór skończony Y, wtedy i tylko wtedy, gdy istnieje injekcja z X do Y. Podobnie, skończone zbiory X i Y mają tyle samo elementów wtedy i tylko wtedy, gdy istnieje bijekcja między nimi. Ta obserwacja posłuży nam do porównywania „liczby elementów" dowolnych, niekoniecznie skończonych, zbiorów. Podamy teraz trzy definicje oraz odpowiadające im oznaczenia.

Definicja 6.37. *Mówimy, że zbiory X oraz Y są równoliczne lub że mają taką samą moc, jeśli istnieje bijekcja między nimi. Piszemy wtedy $|X| = |Y|$.*

Definicja 6.38. *Mówimy, że zbiór X jest zbiorem nie większej mocy niż zbiór Y, jeśli istnieje injekcja z X do Y. Piszemy wtedy $|X| \leqslant |Y|$.*

Definicja 6.39. *Mówimy, że zbiór X jest zbiorem mniejszej mocy niż zbiór Y, jeśli $|X| \leqslant |Y|$ oraz $|X| \neq |Y|$. Piszemy wtedy $|X| < |Y|$.*

Oznaczenia te nie są przypadkowe. Zachodzi bowiem następujące twierdzenie:

Twierdzenie 6.40 (Dedekind 1888 [14], Bernstein 1898 [6], Twierdzenie Cantora-Bernsteina-Schrödera)**.** *Jeśli* $|X| \leqslant |Y|$ *oraz* $|Y| \leqslant |X|$*, to* $|X| = |Y|$*.* ■

Z kolei poniższe twierdzenie jest równoważne aksjomatowi wyboru:

Twierdzenie 6.41 (Cantor 1897 [9], von Neumann 1923 [41], Prawo trychotomii)**.** *Dla dowolnych zbiorów* X *oraz* Y *zachodzi* $|X| < |Y|$*,* $|Y| < |X|$ *lub* $|X| = |Y|$*.* ■

W teorii zbiorów dobrze uporządkowanych każdemu zbiorowi dobrze uporządkowanemu przypisywaliśmy w sposób jednoznaczny pewnego reprezentanta - liczbę porządkową, w taki sposób, aby zbiory nieizomorficzne miały przypisane inne liczby porządkowe, a zbiory izomorficzne tę samą liczbę porządkową.

W tym rozdziale postąpimy podobnie jak w poprzednim, gdzie do porównywania zbiorów dobrze uporządkowanych używaliśmy liczb porządkowych. Tutaj każdemu zbiorowi (niekoniecznie dobrze uporządkowanemu) będziemy przypisywać pewną unikalną dla niego liczbę porządkową, tj. jego moc, w taki sposób, aby dwa zbiory miały tę samą moc wtedy i tylko wtedy, gdy są równoliczne.

Definicja 6.42. *Mówimy, że liczba porządkowa jest liczbą kardynalną, jeśli nie jest równoliczna z żadną liczbą porządkową od siebie mniejszą.*

Ćwiczenie: Nie wszystkie liczby porządkowe są liczbami kardynalnymi. W szczególności proszę pokazać, że liczba porządkowa $\omega + 1$ jest równoliczna z liczbą porządkową ω.

Definicja 6.43. *Mocą zbioru* X *nazywamy jedyną liczbę kardynalną, która jest równoliczna z* X*. Oznaczamy ją przez* $|X|$ *lub czasami przez* $\#X$*.*

Definicja 6.44. *Niech* κ *będzie dowolną liczbą kardynalną. Następnikiem (kardynalnym liczby* κ *nazywamy najmniejszą liczbę kardynalną większą od* κ *i oznaczamy ją przez* κ^+*. Mówimy, że liczba kardynalna* κ *jest następnikowa, jeśli jest postaci* $\kappa = \gamma^+$ *dla pewnej liczby kardynalnej* γ*. W przeciwnym przypadku mówimy, że* κ *jest graniczna.*

Klasę liczb porządkowych granicznych oznaczamy przez $\mathbb{LIM}$.

Uwaga 6.45. *Liczby kardynalne następnikowe i graniczne nie są tym samym, co liczby porządkowe następnikowe i graniczne. Istotnie, w kolejnym podrozdziale okaże się, że wszystkie nieskończone liczby kardynalne są liczbami porządkowymi granicznymi.*

Wszystkie liczby kardynalne można poindeksować wszystkimi liczbami porządkowymi. Powstały ciąg pozaskończony nazywamy skalą alefów.

Definicja 6.46. *Skalą alefów lub hierarchią alefów nazywamy następujący ciąg:*

$$\begin{aligned}&\aleph_0 = \omega,\\ &\aleph_{\alpha+1} = \aleph_\alpha^+, \alpha \in \mathbb{ON},\\ &\aleph_\alpha = \sup\{\aleph_\beta \colon \beta < \alpha\}, \alpha \in \mathbb{LIM}.\end{aligned}$$

Twierdzenie 6.47 (Paradoks Cantora 1891 [8])**.** *Nie istnieje zbiór wszystkich liczb kardynalnych.* ■

6.5.1. Arytmetyka liczb kardynalnych

W tym podrozdziale zajmiemy się arytmetyką liczb kardynalnych, czyli mocą zbiorów powstałych za pomocą operacji teoriomnogościowych. Pierwszą z nich jest suma liczb kardynalnych, która odpowiada mocy sumy dwóch zbiorów rozłącznych.

Definicja 6.48. *Sumą liczb kardynalnych κ i λ nazywamy moc zbioru $\kappa \times \{0\} \cup \lambda \times \{1\}$. Oznaczamy ją przez $\kappa + \gamma$.*

Uwaga 6.49. *Suma liczb kardynalnych NIE JEST tym samym, co suma liczb porządkowych, pomimo tego, że stosujemy to samo oznaczenie! Służy to tylko zmyleniu studentów. Podobna uwaga będzie dotyczyć mnożenia oraz potęgowania liczb kardynalnych i porządkowych.*

Obserwacja 6.50. *Każda nieskończona liczba kardynalna κ jest równoliczna z liczbą porządkową $\kappa + 1$.* ■

Z powyższego stwierdzenia wynika, że liczba kardynalna $\kappa + 1$ (rozumiana jako suma liczb kardynalnych) nie jest tym samym, co liczba porządkowa $\kappa + 1$, jeśli κ jest nieskończone. Podobna sytuacja dotyczy innych operacji typu mnożenie czy potęgowanie!

Aby rozróżnić, kiedy mamy na myśli własności liczby kardynalnej $\kappa = \aleph_\alpha$ od sytuacji, w której mamy na myśli jej własności jako liczby kardynalnej, w tym pierwszym przypadku używamy zapisu ω_α zamiast $\aleph_\alpha$. Stąd $\aleph_\alpha + \aleph_\alpha$ będzie oznaczać sumę liczb kardynalnych. Natomiast $\omega_\alpha + \omega_\alpha$ będzie oznaczać sumę liczb porządkowych.

Z obserwacji 6.50 wynika następujące twierdzenie.

Wniosek 6.51. *Każda nieskończona liczba kardynalna jest liczbą porządkową graniczną.* ■

Zdefiniujemy teraz iloczyn liczb kardynalnych.

Definicja 6.52. *Iloczynem liczb kardynalnych κ i λ nazywamy moc zbioru $\kappa \times \lambda$. Oznaczamy ją przez $\kappa \cdot \gamma$.*

Jednym z dwóch (obok twierdzenia Kőniga, które pojawi się później) najważniejszych twierdzeń arytmetyki liczb kardynalnych jest następujące twierdzenie pochodzące od Hessenberga.

Twierdzenie 6.53 (Zermelo 1908 [43], twierdzenie Hessenberga)**.** *Niech $\alpha \in \mathbb{ON}$. Wówczas $\aleph_\alpha \cdot \aleph_\alpha = \aleph_\alpha$.*

Dowód. Zdefiniujmy dobry porządek na $\mathbb{ON} \times \mathbb{ON}$. Niech $(\beta, \gamma) < (\beta', \gamma')$, jeśli spełniony jest któryś z warunków:
1. $\max\{\beta, \gamma\} < \max\{\beta', \gamma'\}$,
2. $\max\{\beta, \gamma\} = \max\{\beta', \gamma'\}$ i $\beta < \beta'$,
3. $\max\{\beta, \gamma\} = \max\{\beta', \gamma'\}$, $\beta = \beta'$ i $\gamma < \gamma'$.

Porządek ten nazywamy kanonicznym dobrym porządkiem na $\mathbb{ON} \times \mathbb{ON}$.

Niech α będzie najmniejszą liczbą porządkową, taką że teza nie zachodzi. Dowód przeprowadzimy ze względu na indukcję na α. Dla $\alpha = 0$ twierdzenie jest prawdziwe. Możemy więc założyć, że α jest większe od zera. Dalsza część dowodu będzie polegać na pokazaniu, że typ porządkowy zbioru $\omega_\alpha \times \omega_\alpha$ jest równy ω_α. Z tego będzie wynikać, że moc zbioru $\omega_\alpha \times \omega_\alpha$ jest równa mocy zbioru ω_α, czyli $\aleph_\alpha$. Niech $\beta, \gamma < \omega_\alpha$ będą takie, że $tp(\beta \times \gamma) = tp(\{(\varepsilon, \delta) \colon \varepsilon < \beta, \delta < \gamma\}) \geqslant \omega_\alpha$. Weźmy $\varphi < \alpha$, takie że $\beta < \varphi$ i $\gamma < \varphi$. Wówczas zbiór $\omega_\varphi \times \omega_\varphi$ zawiera zbiór $\omega_\beta \times \omega_\gamma$, więc typ porządkowy tego zbioru jest silnie większy niż ω_α, ale ponieważ $\varphi < \alpha$, to z założenia indukcyjnego otrzymujemy, że $\aleph_\alpha \geqslant |\varphi| \cdot |\varphi| = |\varphi| < \aleph_\alpha$. Co daje sprzeczność. ■

Następująca silniejsza wersja twierdzenia Hessenberga jest prawdziwa w ZFC.

Twierdzenie 6.54 (Hessenberg 1906, Twierdzenie Hessenberga)**.** [ZFC] *Niech X będzie zbiorem nieskończonym. Wówczas $|X| = |X \times X|$.*

Dowód. Wystarczy zauważyć, że dla każdego zbioru nieskończonego X w ZFC mamy $|X| = \aleph_\alpha$ dla pewnego $\alpha \in \mathbb{ON}$. ■

Okazuje się, że ta wersja twierdzenia Hessenberga jest równoważna aksjomatowi wyboru na gruncie ZF.

Twierdzenie 6.55 (ZF)(Tarski 1924 [39])**.** *Twierdzenie Hessenberga jest równoważne aksjomatowi wyboru.* ■

Wniosek 6.56. *Dla każdej nieskończonej liczby kardynalnej κ zachodzi $\kappa \cdot \kappa = \kappa + \kappa = \kappa$.*

Wniosek 6.57. *Dla każdej pary niezerowych liczb kardynalnych $\{\kappa, \lambda\}$, z których przynajmniej jedna jest nieskończona, zachodzi $\kappa \cdot \lambda = \kappa + \lambda = \max\{\kappa, \lambda\}$.*

Definicja 6.58. *Mówimy, że liczba kardynalna jest regularna, jeśli dla każdego $A \subseteq \kappa$, jeśli $\sup A = \kappa$, to $|A| = \kappa$. W przeciwnym przypadku mówimy, że liczba kardynalna jest singularna.*

Definicja 6.59. *Niech α będzie liczbą porządkową graniczną. Kofinalnością lub Współkońcowością liczby porządkowej α nazywamy moc najmniejszego zbioru $A \subseteq \alpha$, takiego że $\sup A = \alpha$. Oznaczamy ją przez $cf(\kappa)$.*

Przykłady:
1. $cf(\aleph_0) = \aleph_0$,
2. $cf(\omega_0 + \omega_0) = \aleph_0$,
3. $cf(\omega_i + \omega_j) = cf(\omega_j)$,
4. Jeśli α jest liczbą porządkową graniczną większą od zera, to $cf(\aleph_\alpha) = cf(\alpha)$.
5. Jeśli α jest liczbą porządkową następnikową, to $cf(\aleph_\alpha) = \aleph_\alpha$.

Wniosek 6.60. *Każda regularna liczba kardynalna jest liczbą kardynalną graniczną.*

Problem: czy istnieje nieprzeliczalna liczba kardynalna, która jest jednocześnie regularna i graniczna?

Definicja 6.61. *Mówimy, że nieprzeliczalna liczba kardynalna jest słabo nieosiągalna, jeśli jest regularna i graniczna.*

Twierdzenie 6.62. *Istnieje model ZFC, w którym nie ma liczb słabo nieosiągalnych.*

Wniosek 6.63. *Nie można dowieść na gruncie ZFC, że istnieje regularna, nieprzeliczalna liczba kardynalna graniczna.*

Zdefiniujemy teraz potęgowanie oraz nieskończoną sumę i iloczyn.

Definicja 6.64. *Niech λ i κ będą dowolnymi liczbami kardynalnymi. Potęgą (kardynalną) κ^λ nazywamy moc zbioru funkcji z λ do κ.*

Definicja 6.65. *Niech I będzie indeksowaną rodziną liczb kardynalnych. Wówczas sumą po I nazywamy moc zbioru $\bigcup \kappa_i \times \{i\}$ i oznaczamy ją przez $\sum\{\kappa_i : i \in I\}$.*

Definicja 6.66. *Niech I będzie indeksowaną rodziną liczb kardynalnych. Wówczas iloczynem po I nazywamy moc zbioru $\prod \kappa_i \times \{i\}$ i oznaczamy ją przez $\prod\{\kappa_i : i \in I\}$.*

Uwaga: Pusty iloczyn i suma mnogościowa są z definicji takie, że iloczyn liczb kardynalnych po pustym zbiorze indeksów jest równy 1, a suma jest równa 0.

Poniższe twierdzenie, zwane czasami lematem Kőniga, jest najważniejszym twierdzeniem arytmetyki liczb kardynalnych. Podamy je jednak bez dowodu.

Twierdzenie 6.67 (Kőnig)**.** *Niech I będzie zbiorem indeksów i niech λ_i, κ_i będą liczbami kardynalnymi dla każdego $i \in I$. Jeśli $\lambda_i < \kappa_i$ dla każdego $i \in I$, to zachodzi:*

$$\sum_{i \in I} \lambda_i < \prod_{i \in I} \kappa_i.$$

Dla zobrazowania mocy twierdzenia Kőniga przedstawimy wynikające z niego w prosty sposób twierdzenia.

Wniosek 6.68. *Na gruncie ZF twierdzenie Kőniga pociąga aksjomat wyboru.*

Dowód. Wskazówka: rozpatrz $\lambda_i = 0$ oraz $\kappa_i = 1$ dla $I = X$, gdzie X jest zbiorem, dla którego chcemy znaleźć funkcję wyboru.

Wniosek 6.69 (Cantor 1891 [8], Twierdzenie Cantora o Przekątni)**.** *Dla każdej liczby kardynalnej κ mamy $\kappa < 2^\kappa$.*

Dowód. Wskazówka: rozpatrz $\lambda_i = 1$ oraz $\kappa_i = 2$ dla $I = \kappa$.

Wniosek 6.70. *Dla każdej liczby kardynalnej nieskończonej κ mamy $\kappa < \kappa^{cf(\kappa)}$.*

Dowód. Wskazówka: rozpatrzmy zbiór kofinalny w κ i na tej podstawie dobierzmy parametry λ_i w taki sposób, że $\sup\{\lambda_i \colon i < cf(\kappa)\} = \kappa$. Następnie weźmy $\kappa_i = \kappa$.

Wniosek 6.71. *Zachodzi $\kappa < cf(2^\kappa)$.*

Dowód. Wskazówka: Przypuśćmy dla dowodu nie wprost, że mamy nierówność w drugą stronę. Rozpatrzmy zbiór kofinalny w 2^κ i na tej podstawie dobierzmy parametry λ_i w taki sposób, że $\sup\{\lambda_i \colon i < cf(\kappa)\} = 2^\kappa$. Następnie weźmy $\kappa_i = \kappa$. Otrzymamy, że $2^\kappa < \kappa^\kappa \leqslant (2^\kappa)^\kappa \leqslant 2^\kappa$, co prowadzi do sprzeczności.

Wniosek 6.72. *Zachodzi $\aleph_0 < cf(2^{\aleph_0})$ oraz $2^{\aleph_0} \neq \aleph_\omega$.*

Pytanie: jakie może być położenie $2^{\aleph_0}$ w skali alefów? Cantor postawił Hipotezę Continuum, która mówi, że $2^{\aleph_0} = \aleph_1$.

Wiemy, że $cf(2^{\aleph_0}) > \aleph_0$. Czy są jakieś inne ograniczenia?

Okazuje się, że nie.

Twierdzenie 6.73 (Cohen 1963 [12,13], Solovay)**.** *Niech M będzie modelem ZFC takim, że $\kappa =]aleph_\alpha^M$ oraz $cf(\kappa) = \kappa$ w M. Wówczas istnieje model ZFC M', w którym $2^{\aleph_\alpha} = \aleph_\alpha$ oraz M i M' mają te same liczby porządkowe.*

Ogólna postać powyższego twierdzenia pochodzi od Solovaya. Wynik ten nie został przez niego opublikowany.

Uogólniona hipoteza continuum (GCH) mówi, że nie istnieje zbiór o mocy pośredniej między jakimkolwiek zbiorem mocy nieskończonej a mocą jego zbioru potęgowego.

Definicja 6.74. *Skalą betów lub hierarchią betów nazywamy następujący ciąg:*

$$\begin{aligned}
&\beth_0 = \aleph_0,\\
&\beth_{\alpha+1} = 2^{\beth_\alpha}, \alpha \in \mathbb{ON},\\
&\beth_\alpha = \sup\{\beth_\beta \colon \beta < \alpha\}, \alpha \in \mathbb{LIM}.
\end{aligned}$$

Równoważnie GCH mówi, że $\aleph_\alpha = \beth_\alpha$ dla każdego $\alpha \in \mathbb{ON}$. Podobnie jak CH jest ona niezależna od aksjomatów ZFC. Ponadto z Hipotezy Continuum nie wynika jej ogólniejsza wersja. Najmniejszą nieskończoną liczbą kardynalną, dla jakiej uogólniona hipoteza continuum nie zachodzi może być dowolna regularna liczba kardynalna. Okazuje się, że nie może nią być jednak singularna liczba kardynalna o nieprzeliczalnej kofinalności, co dowiódł Silver w 1975 roku [36].

6.6. Twierdzenia Löwenheima-Skolema-Tarskiego

W poniższych twierdzeniach dowolnie duża moc nie oznacza tego, co dowolna moc!

Twierdzenie 6.75 (Malcew 1936 [30], Górne twierdzenie Löwenheima-Skolema-Tarskiego)**.** *Jeśli Σ ma model nieskończony lub modele dowolnie dużej mocy skończonej, to Σ ma model dowolnie dużej mocy.*

Dowód. Tworzymy zbiór $\{C_\alpha \colon \alpha \in A\}$, gdzie A jest zbiorem dowolnie dużej mocy. Następnie tworzymy zbiór $\Sigma' = \Sigma \cup \{C_\alpha \neq C_\beta \colon \alpha \in A, \beta \in A, \alpha \neq \beta\}$. Z założenia każdy skończony podzbiór zbioru Σ' ma model. Zatem Σ' ma model $\mathcal{N}$. Ten model ma moc przynajmniej tak dużą jak moc zbioru A. Jeśli „zapomnimy" o nowo wprowadzonych symbolach C_α (taka operacja nazywa się redukcją), to dostaniemy szukany model $\mathcal{N}'$. ■

Twierdzenie 6.76 (Löwenheim 1915 [29], Skolem 1920 [37], Dolne twierdzenie Löwenheima-Skolema-Tarskiego)**.** *Jeśli* $|L| \leqslant \aleph_0$ *i* Σ *ma model mocy większej niż* $\aleph_0$*, to* Σ *ma model mocy* $\aleph_0$*.* ■

Twierdzenie 6.77 (Malcew 1936 [30], Dolne twierdzenie Löwenheima-Skolema-Tarskiego)**.** *Jeśli* Σ *ma model nieskończony, to* Σ *ma model każdej mocy nieskończonej większej lub równej* $|L|$*.* ■

Wniosek 6.78 (Skolem 1922 [38], Paradoks Skolema)**.** *Jeśli teoria mnogości ma model* M *, to istnieje przeliczalny model* M' *teorii mnogości, w którym to modelu istnieją zbiory nieprzeliczalne w* M'*.* ■

Część II
Teoria obliczalności

7. Maszyna Turinga

Definicja 7.1. *Częściową funkcją o dziedzinie D nazywamy funkcję, której dziedzina jest podzbiorem zbioru D.*

Częściową funkcję traktujemy jak funkcję, której nie wszystkie wartości są zdefiniowane. Dla odróżnienia funkcji częściowych od zwykle rozumianych funkcji czasami używa się nazwy funkcja totalna, która jest tożsama ze standardowym pojęciem funkcji.

Przejdziemy teraz do najważniejszej definicji tego przedmiotu.

Definicja 7.2. *Maszyną Turinga (lub w skrócie MT) nazywamy uporządkowaną siódemkę $M = (Q,\ \Sigma,\ \Gamma,\ \delta,\ q_0,\ B,\ F)$, gdzie:*
- *Q - skończony zbiór - zbiór stanów maszyny Turinga,*
- *Σ - skończony zbiór - zbiór symboli wejściowych,*
- *Γ - skończony zbiór taki, że $\Sigma \subset \Gamma$ - alfabet taśmy,*
- *q_0 - element zbioru Q - stan początkowy,*
- *B lub blank - wyróżniony element zbioru $\Gamma \setminus \Sigma$ - blank, znak pusty,*
- *F - podzbiór zbioru Q - zbiór stanów końcowych lub akceptujących,*
- *δ - częściowa funkcja $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, P\}$ - funkcja przejścia*

Jak działa Maszyna Turinga?
Maszyna Turinga operuje na dwustronnie nieskończonej, przeliczalnej taśmie, w której komórkach zapisane są symbole z Γ. Maszyna Turinga w danym momencie jest w jednym ze stanów ze zbioru Q, a jej głowica czyta jeden z symboli znajdujących się na taśmie. Maszyna Turinga wykonuje jednocześnie trzy operacje:
1. zmienia stan (może pozostawić ten sam),
2. zastępuje symbol znajdujący się na czytanej komórce (może pozostawić ten sam),
3. przechodzi do komórki znajdującej się na lewo lub na prawo czytanej komórki.

Uwaga! Często zamiast maszyn dwustronnie nieskończonych rozważa się maszyny jednostronnie nieskończone.

Ćwiczenie. Podać formalnie definicję maszyny Turinga z taśmą jednostronnie nieskończoną. Co należy zmodyfikować w podanych definicjach? Dodać warunek $(*)$ nie odwołując się do dodatkowego symbolu $\vdash$. Wskazówka: zmienić definicję konfiguracji, aby uwzględniała również pierwszy znak przed początkiem słowa. Musi on być pusty dla każdej konfiguracji, która powstaje w wyniku następstwa ze słowa w dla każdego $w \in \Sigma^*$.

Definicja 7.3. *Niech Γ będzie niepustym zbiorem zwanym alfabetem. Jego elementy nazywamy znakami, literami lub symbolami. Będziemy oznaczać przez Γ^* rodzinę wszystkich ciągów skończonych, których elementy należą do zbioru Γ. Elementy Γ^* nazywamy słowami. Słowo $(w_1, \ldots, w_n)$ oznaczamy przez $w_1 \ldots w_n$. Jeśli dla słowa $(w_1, \ldots, w_n)$, zachodzi $w_i = a$ dla każdego i, to oznaczamy je przez a^n. Będziemy zwykle milcząco zakładać, że alfabet, nad którym pracujemy ma przynajmniej dwa elementy. Zwykle (choć nie zawsze) będzie też skończony.*

Uwaga: można też rozważać słowa nieskończone, które nie będą dla nas przydatne na tym przedmiocie, ale mogą być ciekawe np. w teorii automatów, która będzie w przyszłym semestrze. Słowo $a \ldots a \ldots$ oznaczamy przez a^ω.

Definicja 7.4. *Konfiguracją maszyny Turinga lub opisem chwilowym nazywamy słowo $\alpha_1 q \alpha_2$, gdzie $\alpha_1,\ \alpha_2 \in \Gamma^*$ oraz q - stan bieżący.*

Konfigurację maszyny Turinga interpretujemy tak, że maszyna Turinga czyta pierwszy znak słowa α_2 lub blank, jeśli α_2 jest słowem pustym.

Definicja 7.5. *Konfiguracją początkową nazywamy konfigurację postaci $\alpha_1 q_0 w$, gdzie α_0 to słowo puste, a $w \in \Sigma^*$ to słowo wejściowe. Konfigurację $\alpha_1 q_0 w$ możemy też zapisać jako $q_0 w$.*

Definicja 7.6. *Niech c_1 i c_2 będą konfiguracjami pewnej maszyny Turinga. Mówimy, że c_2 powstaje w wyniku bezpośredniego następstwa z c_1 i piszemy $c_1 \vdash c_2$, jeśli konfiguracja c_2 jest otrzymywana po jednym ruchu maszyny Turinga znajdującej się w konfiguracji c_1. Mówimy, że c_2 powstaje w wyniku następstwa z c_1 i piszemy $c_1 \vdash^* c_2$, jeśli konfiguracja c_2 jest otrzymywana po skończonej liczbie ruchów maszyny Turinga z konfiguracji c_1.*

Ćwiczenie: czy $\vdash^*$ jest relacją częściowego porządku?

Definicja 7.7. *Mówimy, że maszyna Turinga akceptuje słowo $w \in \Sigma^*$, jeśli istnieje stan $q \in F$ oraz konfiguracja $\alpha_1 q \alpha_2$, taka że $q_0 w \vdash^* \alpha_1 q \alpha_2$.*

Definicja 7.8. *Niech $\alpha_2 \in \gamma^*$ będzie słowem i niech w_0 będzie pierwszą literą słowa α_2 lub blankiem, jeśli słowo α_2 jest puste. Mówimy, że maszyna Turinga zatrzymuje się w konfiguracji $\alpha_1 q \alpha_2$, jeśli $\delta(q, w)$ nie jest określone.*

Definicja 7.9. *Mówimy, że maszyna Turinga jest maszyną Turinga ze stopem lub, że ma własność stopu, jeżeli dla każdego $w \in \Sigma^*$ istnieje stan akceptujący q, taki że istnieją α_1, α_2, takie że $qw \vdash^* \alpha_1 q \alpha_2$ lub maszyna Turinga zatrzymuje się w konfiguracji $\alpha_1 q \alpha_2$.*

Definicja 7.10. *Niech q będzie stanem początkowym maszyny Turinga M. Mówimy, że M zapętla się na słowie w, jeśli nie istnieje konfiguracja, która powstaje w wyniku następstwa konfiguracji qW mająca stan akceptujący lub na której maszyna M się zatrzymuje.*

Definicja 7.11. *Językiem akceptowanym przez maszynę Turinga M lub językiem maszyny Turinga M nazywamy język $L(M) = \{w \in \Sigma^* : M \text{ akceptuje } w\}$.*

Definicja 7.12. *Mówimy, że język $L \subseteq \Sigma^*$ jest rekurencyjnie przeliczalny lub częściowo rozstrzygalny, jeśli istnieje maszyna Turinga M, taka że $L = L(M)$. Mówimy, że język $L \subseteq \Gamma^*$ jest rekurencyjny lub rozstrzygalny, jeśli istnieje maszyna Turinga ze stopem M, taka że $L = L(M)$.*

Teza Churcha 7.12.1 (Church 1936 [10], Turing 1936 [40], Kleene 1936 [24])**.** *Każdy problem obliczalny jest obliczalny na pewnej maszynie Turinga.*

Uwaga: teza Churcha nie jest twierdzeniem! Jest ona tylko nieformalną hipotezą odnośnie potencjalnych alternatyw dla maszyn Turinga, języków rekurencyjnych czy Λ-rachunku.

8. Modyfikacje maszyny Turinga

8.1. Maszyna Turinga z taśmą jednostronnie nieskończoną

Ćwiczenie: jak symulować działanie maszyny Turinga z taśmą dwustronnie nieskończoną na maszynie Turinga z taśmą jednostronnie nieskończoną?

8.2. Maszyna Turinga wielościeżkowa, ale z pojedynczą głowicą i taśmami dwustronnie nieskończonymi

Ćwiczenie: jak symulować działanie wielościeżkowej maszyny Turinga na maszynie Turinga z taśmą dwustronnie nieskończoną?

8.3. Wielotaśmowa Maszyna Turinga

Definicja 8.1. *Wielotaśmowa maszyna Turinga jest modyfikacją maszyny Turinga, która składa się z k głowic oraz k taśm dwustronnie nieskończonych. W pojedynczym ruchu wielotaśmowa maszyna Turinga wykonuje jednocześnie trzy operacje:*

1. *Zmienia stan.*
2. *Zastępuje symbol znajdujący się na każdej czytanej komórce.*
3. *Przemieszcza niezależnie każdą głowicę do komórki znajdującej się na lewo lub na prawo czytanej komórki.*

Wejście początkowo znajduje się na pierwszej taśmie, a pozostałe są puste (wypełnione blankami).

Twierdzenie 8.2. *Język L jest rozpoznawalny przez pewną maszynę Turinga wtedy i tylko wtedy, gdy jest rozpoznawalny przez pewną wielotaśmową maszynę Turinga.*

Dowód. Będziemy symulować działanie k-taśmowej maszyny Turinga M_1 za pomocą maszyny Turinga M_2 z $2k$ ścieżkami i jedną głowicą. Każdej taśmie M_1 odpowiadają dwie ścieżki M_2, z których jedna rejestruje zawartość odpowiedniej taśmy z M_1, a druga jest pusta poza znacznikiem w komórce zawierającej symbol obserwowany przez odpowiednią głowicę.

Ruch M_2 symulujący M_1:

1. Głowica M_2 przesuwa się w prawo, odwiedza komórki zawierające znacznik głowicy i zapamiętuje obserwowane symbole w sterowaniu skończonym.
2. Przesuwa głowicę w lewo aż osiągnie pierwszy znacznik głowicy uaktualniając symbol nad mijanymi znacznikami oraz odpowiednio je przesuwając.
3. Zmienia stan. ■

Wielotaśmowe maszyny Turinga można wykorzystać w ten sposób, aby na jednej z taśm znajdowało się wyjście inne niż „tak” lub „nie”, np. wynik dodawania liczby na pierwszej taśmie i liczby na drugiej taśmie. Wielotaśmowe maszyny Turinga można więc zastosować do modelowania maszyny obliczeniowej.

9. NDMT, czyli niedeterministyczna maszyna Turinga

Definicja 9.1. *Niedeterministyczna maszyna Turinga (w skrócie NDMT) składa się z jednej jednostronnie nieskończonej taśmy i z jednej głowicy. Dla danego stanu i symbolu obserwowanego przez głowicę taśmy niedeterministyczna maszyna Turinga ma skończoną liczbę opcji następnego ruchu (nie mamy tu do czynienia z funkcją przejścia, ale z pewną relacją przejścia).*

Definicja 9.2. *Mówimy, że niedeterministyczna maszyna Turinga M akceptuje słowo $w \in \Sigma^*$, jeśli istnieje ciąg możliwych wyborów kolejnych konfiguracji niedeterministycznej maszyny Turinga w obliczeniu prowadzący do stanu akceptującego.*

Twierdzenie 9.3. *Język L jest akceptowany przez pewną niedeterministyczną maszynę Turinga wtedy i tylko wtedy, gdy jest akceptowany przez pewną maszynę Turinga.*

Dowód. Niech M będzie niedeterministyczna maszyna Turinga. Dla dowolnego stanu i dowolnego symbolu taśmowego istnieje skończona liczba opcji następnego ruchu, a więc możemy ją oszacować z góry przez pewną liczbę naturalną r. Każdy skończony ciąg wybranych opcji można więc zaprezentować za pomocą ciągu liczb, którego elementy należą do zbioru $\{1, \ldots, r\}$. Konstruujemy maszynę Turinga M' o trzech taśmach. Pierwsza zawiera wejście. Na drugiej generowane będą wszystkie ciągi liczb ze zbioru $\{1, \ldots, r\}$. Dla każdego ciągu wygenerowanego na drugiej taśmie M' kopiuje wejście na trzecią taśmę, a następnie symuluje M. Jeśli M' nie może znaleźć słowa akceptującego dla danego ciągu, to zwiększa liczbę elementów ciągów i symuluje ponownie. Jeżeli słowo jest akceptowane na taśmie trzeciej dla pewnego ciągu na taśmie drugiej, to przechodzimy do stanu akceptującego. ■

Obserwacja 9.4. *Wszystkie opisane warianty maszyny Turinga mają tę samą moc obliczeniową, tj. są w stanie rozpoznać dokładnie te same języki.* ■

10. Funkcje rekurencyjne

10.1. Wprowadzenie

Ile jest wszystkich funkcji $\mathbb{N} \to \mathbb{N}$?

$$\aleph_0^{\aleph_0} = 2^{\aleph_0} = \mathfrak{c}.$$

Ograniczenie się do funkcji $\mathbb{N} \to \{0, 1\}$ o dwóch wartościach nie pomaga, bo:

$$2^{\aleph_0} = \mathfrak{c}.$$

Algorytm (także program komputerowy, dowód, wzór funkcji, etc.) jest opisany przez skończoną liczbę znaków z co najwyżej przeliczalnego alfabetu. Ile jest takich algorytmów (przy ustalonym alfabecie)?

$$\aleph_0 + \aleph_0 \cdot \aleph_0 + \cdots = \aleph_0 + \aleph_0^2 + \ldots \aleph_0^k + \cdots = (*).$$

Wiemy, że $\aleph_0 \cdot \aleph_0 = \aleph_0^2 = \aleph_0$, czyli moc kwadratu kartezjańskiego zbioru liczb naturalnych jest mocy $\aleph_0$. (Uwaga: ogólniej, moc kwadratu kartezjańskiego zbioru mocy κ jest mocy κ, o ile κ jest nieskończone.) Przez indukcję otrzymujemy, że $\aleph_0^k = \aleph_0$ dla każdego $k = 1,\ 2, \ldots$. Zatem:

$$(*) = \aleph_0 + \aleph_0 + \ldots.$$

Przeliczalna suma przeliczalnych mocy jest przeliczalna. Stąd otrzymujemy:

$$(*) = \aleph_0.$$

Mamy więc $\aleph_0$ algorytmów, ale aż $2^{\aleph_0}$ funkcji. Zatem prawie wszystkie muszą być nieobliczalne (w pewnym sensie). W dalszej części wykładu zajmiemy się sformalizowaniem pojęcia obliczalności funkcji.

10.2. Funkcje prymitywnie rekurencyjne

Definicja 10.1. *Mówimy, że funkcja $f : \mathbb{N}^{k+1} \to \mathbb{N}$ jest definiowana przez rekursję prostą z funkcji $h : \mathbb{N}^{k+2} \to \mathbb{N}$ i $g : \mathbb{N}^k \to \mathbb{N}$, jeśli*

$$\forall x\ \forall y : f(x,0) = g(x) \text{ oraz } f(x,y+1) = h(x,y,f(x,y)).$$

Definicja 10.2. *Mówimy, że klasa funkcji $\mathcal{C}$ jest zamknięta na złożenie, jeśli dla dowolnych funkcji $g_1 : \mathbb{N}^k \to \mathbb{N}, \ldots, g_n : \mathbb{N}^k \to \mathbb{N}$, $h : \mathbb{N}^n \to \mathbb{N}$ należących do klasy $\mathcal{C}$ funkcja $f(x) = h(g_1(x), \ldots, g_n(x))$ również należy do klasy $\mathcal{C}$.*

Definicja 10.3. *Klasą funkcji prymitywnie (pierwotnie) rekurencyjnych $\mathcal{PREC}$ nazywamy najmniejszą w sensie inkluzji klasę zamkniętą ze względu na złożenie funkcji i rekursję prostą oraz zawierającą następujące funkcje (zwane funkcjami inicjującymi):*
1. $0_n(x) = 0$, $0_n : \mathbb{N}^n \to \mathbb{N}$ - *zero,*
2. $S(x) = x + 1$ - *następnik,*
3. $I^n_k(x_1, \ldots, x_n) = x_k$, $I^k_n : \mathbb{N}^n \to \mathbb{N}$ - *rzutowanie na k-tą zmienną.*

Przykłady funkcji prymitywnie rekurencyjnych (dowody na ćwiczeniach):
1. funkcje stałe,
2. dodawanie,
3. mnożenie,
4. funkcja znaku $sgn(x)$,
5. funkcja „if $x = 0$, then y, else z”,
6. suma $\sum\{f(x,y) : y < z\}$, jeśli $f \in \mathcal{PREC}$,
7. iloczyn $\prod\{f(x,y) : y < z\}$, jeśli $f \in \mathcal{PREC}$.

Definicja 10.4. *Funkcją Ackermanna (w uproszczonej przez Pétera wersji) nazywamy funkcję $A\colon \mathbb{N}^2 \to \mathbb{N}$ zdefiniowaną w sposób następujący:*

$$A(x,y) = \begin{cases} y+1, & x = 0, \\ A(x-1,1), & x > 0,\ y = 0, \\ A(x-1, A(x,y-1)), & x > 0,\ y > 0. \end{cases}$$

Funkcja Ackermanna jest przykładem funkcji, która jest rekurencyjna, ale nie jest prymitywnie rekurencyjna. Aby tego dowieść, będziemy potrzebować kilku lematów.

Lemat 10.5. *Funkcja Ackermanna ma następujące własności:*
(A1) *funkcja A jest totalna, a więc dobrze zdefiniowana dla każdej pary (x,y),*
(A2) $A(1,y) = y + 2$,
(A3) $A(2,y) = 2y + 3$,
(A4) $A(x,y) > y$,
(A5) $A(x,y) < A(x,y+1)$,

(A6) $A(x, y+1) \leqslant A(x+1, y)$,
(A7) $A(r, A(s, y)) < A(r+s+2, y)$,
(A8) *dla każdego* r, s *istnieje* t *niezależne od* y *takie, że* $A(r, y)+A(s, y) < A(t, y)$.
Dowód. *Na ćwiczeniach.* ■

Lemat 10.6. *Niech* $A_n \colon \mathbb{N} \to \mathbb{N}$ *będzie funkcją daną jako* $A_n(m) = A(n, m)$. *Wówczas dla każdego* n *funkcja* $A_n(m)$ *jest funkcją prymitywnie rekurencyjną.*

Dowód. będziemy dowodzić twierdzenie przez indukcję za względu na n. Dla $n = 0$ mamy $A_0(m) = m + 1 = S(m)$, więc A_0 jest prymitywnie rekurencyjna. Załóżmy, że A_n jest prymitywnie rekurencyjna. Mamy

$$A_{n+1}(0) = A(n+1, 0) = A(n, 1) = A_n(1) = k$$

dla pewnej funkcji stałej k. Następnie

$$\begin{aligned} A_{n+1}(m+1) = A(n+1, m+1) = A(n, A(n+1, m)) &= A_n(A(n+1, m)) \\ &= A_n(A_{n+1}(m)). \end{aligned}$$

Otrzymaliśmy, że A_{n+1} jest zdefiniowana przez rekursję prostą z funkcji A_n, która jest prymitywnie rekurencyjna z założenia indukcyjnego. Funkcja A_{n+1} jest zatem prymitywnie rekurencyjna. ■

Definicja 10.7. *Mówimy, że funkcja* $g \colon \mathbb{N}^k \to \mathbb{N}$ *jest zmajoryzowana przez funkcję* $h \colon \mathbb{N}^2 \to \mathbb{N}$, *jeśli istnieje stała* b, *taka że dla każdego* $a_1, \ldots, a_k \in \mathbb{N}$ *zachodzi* $g(a_1, \ldots, a_k) < h(a, b)$, *gdzie* $a = \max\{a_1, \ldots, a_k\}$.

Lemat 10.8. *Nie istnieje funkcja* $h \colon \mathbb{N}^2 \to \mathbb{N}$, *która majoryzuje samą siebie.*

Dowód. Załóżmy, że taka funkcja istnieje. Zatem istnieje stała b, taka że dla każdego x, y mamy $h(x, y) < h(a, b)$, gdzie $a = \max\{x, y\}$. Niech $c = \max\{a, b\}$. Wówczas $h(c, b) < h(\max\{b, c\}, b) = h(c, b)$. ■

Lemat 10.9. *Każda funkcja pierwotnie rekurencyjna jest zmajoryzowana przez funkcję Ackermanna.*

Dowód. Oznaczmy przez $\mathcal{A}$ klasę funkcji, które są zmajoryzowane przez A. Dowód będzie polegał na pokazaniu, że klasa $\mathcal{A}$ zawiera wszystkie funkcje inicjujące, jest zamknięta ze względu na złożenie oraz na rekursję prostą. Jako że klasa $\mathcal{PREC}$ jest najmniejszą klasą spełniającą te warunki, to będzie z tego wynikać, że $\mathcal{PREC} \subseteq \mathcal{A}$.

W dalszej części dowodu niech $x = \{x_1, \ldots, x_n\}$, $y = \{y_1, \ldots, y_n\}$. Oznaczmy przez $\hat{x}$ największą z liczb $x_1, \ldots, x_n$. Podobnie niech $\hat{y}$ oznacza największą z liczb $y_1, \ldots, y_n$.

Najpierw pokażemy, że funkcje inicjujące należą do klasy $\mathcal{A}$.

- $0_n(x) = 0 < \hat{x} + 1 = A(0, \hat{x})$, więc $0_n(x) \in \mathcal{A}$,
- $S(n) = n + 1 < n + 2 = A(1, n)$, więc $S(n) \in \mathcal{A}$,
- $I_k^n(x_1, \ldots, x_n) = x_k \leqslant \hat{x} < \hat{x} + 1 = A(0, x)$, więc $I_n^k \in \mathcal{A}$.

Następnie pokażemy, że klasa $\mathcal{A}$ jest zamknięta ze względu na złożenie funkcji. załóżmy, że funkcje $g_1, \ldots, g_m$ są funkcjami k zmiennych, a h jest funkcją m zmiennych oraz że każda funkcja spośród g_i oraz h należy do klasy $\mathcal{A}$. Oznacza to, że istnieją stałe r_i oraz s, takie że $g_i(x) < A(r_i, x)$ oraz $h(y) < A(s, y)$ dla każdych x, y. Niech $f = h(g_1, \ldots, g_m)$ i $g_j(x) = \max\{g_i(x) \colon i = 1, \ldots, m\}$. Wówczas

$$f(x) < A(s, g_j(x)) < A(s, A(r_j, x)) < A(s + r_j + 2, x),$$

gdzie pierwsza nierówność wynika z ograniczenia na h. Druga nierówność wynika z ograniczenia na g, a ostatnia nierówność wynika z własności (A7). Stąd $f(x) \in \mathcal{A}$, a więc klasa $\mathcal{A}$ jest zamknięta ze względu na złożenie.

Pozostało nam do pokazania, że klasa $\mathcal{A}$ jest zamknięta ze względu na rekursję prostą. Niech g będzie funkcją k argumentów, h funkcją $k+2$ argumentów i niech g oraz h należą do $\mathcal{A}$. Analogicznie jak wcześniej mamy, że istnieją stałe r, s, takie że $g(x) < A(r, \hat{x})$ oraz $h(x) < A(s, \hat{x})$. Chcemy pokazać, że funkcja f zdefiniowana przez rekursję prostą z funkcji g oraz h należy do klasy $\mathcal{A}$.

W tym celu udowodnimy teraz następujące stwierdzenie: istnieje stała q, taka że $f(x, n) < A(q, n + \hat{x})$.

Niech $q = 1 + \max\{r, s\}$. Będziemy teraz przeprowadzać indukcję ze względu na n. Dla $n = 0$ mamy $f(x, 0) = g(x) < A(r, \hat{x}) < A(q, \hat{x})$. Załóżmy, że zachodzi $f(x, n) < A(q, n + \hat{x})$. Wówczas

$$f(x, n + 1) = h(x, n, f(x, n)) < A(s, z),$$

gdzie $z = \max\{\hat{x}, n, f(x, n)\}$. Korzystając z założenia indukcyjnego oraz faktu, że $\max\{x, n\} \leqslant n + \hat{x} < A(q, n + \hat{x})$ otrzymujemy $z < A(q, n + \hat{x})$. Stąd

$$f(x, n+1) < A(s, z) < A(s, A(q, n + \hat{x})) \leqslant A(q - 1, A(q, n + \hat{x})) = A(q, n + 1 + \hat{x}),$$

co kończy dowód stwierdzenia.

Aby zakończyć dowód twierdzenia, wystarczy wziąć $z = \max\{\hat{x}, \hat{y}\}$. Wówczas z udowodnionego stwierdzenia otrzymujemy, że

$$f(x, y) < A(q, \hat{x} + y) \leqslant 2z) < A(q, 2z + 3) = A(q, A(2, z)) = A(q + 4, z).$$

Wynika z tego, że f należy do klasy $\mathcal{A}$, co kończy dowód. ■

Twierdzenie 10.10 (Ackermann 1928 [1], Péter 1935 [32])**.** *Funkcja Ackermanna nie jest funkcją prymitywnie rekurencyjną.*

Dowód. Bezpośrednio z lematów 10.8 oraz 10.9. ■

10.3. Funkcje rekurencyjne

Definicja 10.11. *Mówimy, że funkcja $f : \mathbb{N}^k \to \mathbb{N}$ jest definiowana przez μ-rekursję z funkcji $g : \mathbb{N}^{k+1} \to \mathbb{N}$, jeśli*
1. $\forall x\ \exists y\colon g(x,y) = 1$,
2. $\forall x\ f(x) = \min\{y : g(x,y) = 1\}$.
Oznaczenie $f(x) = \mu_y\colon g(x,y)$.

Definicja 10.12. *Mówimy, że funkcja jest efektywnie obliczalna, jeśli jest obliczalna na pewnej maszynie Turinga.*

Twierdzenie 10.13 (Church 1936 [11], Turing 1936 [40], Kleene 1936 [24])**.** *Funkcja jest rekurencyjna wtedy i tylko wtedy, gdy jest rekurencyjna.*

Definicja 10.14. *Klasą funkcji rekurencyjnych $\mathcal{REC}$ nazywamy najmniejszą w sensie inkluzji klasę zawierającą funkcje inicjujące i zamkniętą ze względu na złożenie funkcji, rekursję prostą i μ-rekursję.*

Twierdzenie 10.15 (Ackermann 1928 [1])**.** *Funkcja Ackermanna jest funkcją rekurencyjną.* ■

Wniosek 10.16 (Ackermann 1928 [1])**.** *Zachodzi $\mathcal{PREC} \subsetneq \mathcal{REC}$.*

Definicja 10.17. *Mówimy, że relacja $R \subset \mathbb{N}^k$ jest (prymitywnie) rekurencyjna, jeśli funkcja charakterystyczna relacji R jest funkcją (prymitywnie) rekurencyjną.*

Twierdzenie 10.18 (Gödel 1934 [19], Kleene 1936 [24], o eliminacji rekursji prostej)**.** *Klasa funkcji rekurencyjnych jest najmniejszą w sensie inkluzji klasą zawierającą funkcje inicjujące, dodawanie, mnożenie, funkcję charakterystyczną równości oraz zamkniętą ze względu na złożenie funkcji i μ-rekursję.* ■

10.4. Twierdzenie Kleene'go o formie normalnej rekursji

Najpierw zajmiemy się numerowaniem wszystkich funkcji rekurencyjnych, w taki sposób, że każdej liczbie naturalnej m będzie odpowiadać dokładnie jedna funkcja φ_n^k, gdzie k oznacza liczbę argumentów funkcji φ_n^k.

Twierdzenie 10.19 (Ackermann 1928 [1], Kleene 1936 [24], o formie normalnej rekursji)**.** *Istnieje prymitywnie rekurencyjna funkcja $U : \mathbb{N} \to \mathbb{N}$ i relacje prymitywnie rekurencyjne $\mathcal{T}_n \in \mathbb{N}^{k+2}$, takie że dla dowolnej funkcji rekurencyjnej f istnieje stała e_f, taka że $f(n_1, \ldots, n_k) = U(\mu_y\colon \mathcal{T}_n(e_f, n_1, \ldots, n_k, y))$.*

Dowód. Każda funkcja rekurencyjna jest obliczalna na pewnej wielotaśmowej maszynie Turinga. Zatem istnieje pewna maszyna M_i, która oblicza funkcję f. Możemy założyć, że M_i jest o najmniejszym możliwym kodzie $< M_i >$. Niech e_f będzie jej kodem.

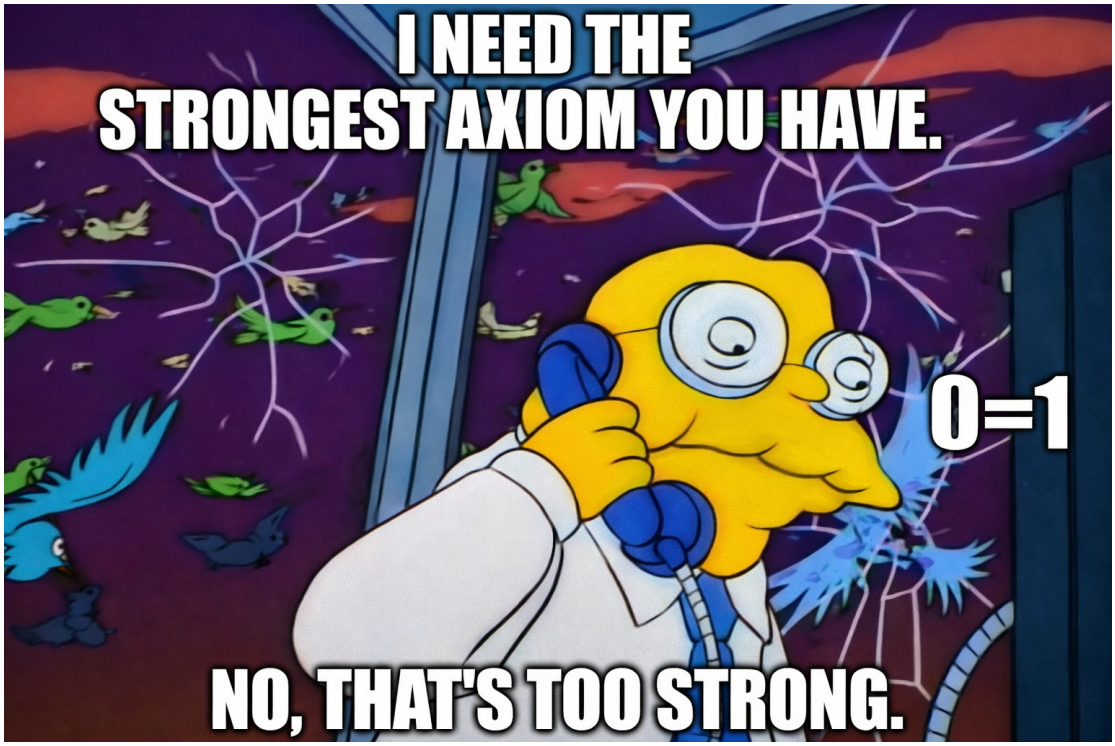
Niech M_i dla wejścia $(n_1, n_2, \ldots, n_k)$ zatrzymuje się (tj. kończy obliczenie funkcji rekurencyjnej f) w kroku t_0 z wyjściem q, czyli $f(n_1, n2, \ldots, n_k) = q$.

Niech $t > t_0$, czyli czekamy, aż M_i zakończy pracę.

Niech $(S_k(e_f, n_1, n_2, \ldots, n_k, t))_1 > length(e_k)$, gdzie $S_k(e_f, n_1, n_2, \ldots, n_k, t)$ jest kodem stanu w momencie t obliczenia na M_i z wejściem $(n_1, n_2, \ldots, n_k)$, $length(e_f)$ jest długością słowa e_f, liczba $(S_k(e_f, n_1, n_2, \ldots, n_k, t))_1$ jest wartością funkcji przejścia dla instrukcji, która ma być wykonana w kroku t. Potrzebna będzie nam jeszcze funkcja $(z)_j$, czyli funkcja pierwszej eksponenty. Połóżmy $z_0 = 2^q 3^{t_0}$. Wówczas $(z_0)_2 = t_0$ i $(z_0)_1 = q$. Stąd $(S_k(e_f, n_1, n_2, \ldots, n_k, (z_0)_2))_1 > length(e_k)$ i $(z_0)_1 = (S_k(e_f, n_1, n_2, \ldots, n_k, (z_0)_2))_2$. Co więcej, z_0 jest najmniejszą taką liczbą, ponieważ $(z_0)_2 = t_0$ jest momentem zatrzymania M_i. Mamy więc $f(n_1, n_2, \ldots, n_k) = q = (z_0)_1$. Weźmy $T_k(e_f, n_1, n_2, \ldots, n_k, z)$ jako następującą relację: $T_k(e_f, n_1, n_2, \ldots, n_k, z)$ wtedy i tylko wtedy, gdy zachodzi$(S_k(e_f, n_1, n_2, \ldots, n_k, (z_0)_2))_1 > length(e_k)$ oraz $(z)_1 = (S_k(e_f, n_1, n_2, \ldots, n_k, (z)_2))_2$. Wówczas T_k jest prymitywnie rekurencyjną relacją $k+1$ argumentową. Wówczas bierzemy $U(z) = (z)_1$. ■

Wniosek 10.20. *Do zdefiniowania dowolnej funkcji rekurencyjnej wystarczy użyć μ-rekursji co najwyżej raz.* ■

11. Twierdzenia Gödla



Definicja 11.1. *Mówimy, że funkcja jest efektywnie obliczalna, jeśli jest obliczalna na pewnej maszynie Turinga.*

Teza Churcha 11.1.1 (Church 1936, Turing 1936, Kleene 1943)**.** *Funkcja jest obliczalna wtedy i tylko wtedy, gdy jest efektywnie obliczalna.*

Definicja 11.2. *System formalny* $S = (L, A, B, \Sigma)$ *z rekurencyjnymi zbiorami*

L, A, B. Mówimy, że zbiór zdań Σ jest rekurencyjnie aksjomatyzowalny, jeśli istnieje rekurencyjny zbiór Σ', taki że $Col(\Sigma) = Col(\Sigma')$.

Zauważmy, że rekurencyjna aksjomatyzowalność zbioru Σ nie oznacza, że jest on rekurencyjny ani nawet przeliczalny, ale jest tak w przypadku zarówno arytmetyki Peana jak i ZFC. Podobnie możemy zdefiniować skończoną aksjomatyzowalność.

Definicja 11.3. *System formalny $S = (L, A, B, \Sigma)$ z rekurencyjnymi zbiorami L, A, B. Mówimy, że zbiór zdań Σ jest skończenie aksjomatyzowalny, jeśli istnieje rekurencyjny zbiór Σ', taki że $Col(\Sigma) = Col(\Sigma')$.*

Innymi słowy: moglibyśmy zastąpić zbiór aksjomatów Σ przez pewien skończony zbiór Σ^*. Przypomnijmy, że zbiór aksjomatów arytmetyki Peana nie jest skończony, a tylko rekurencyjnie przeliczalny.

Poniższe twierdzenie udowodnili niezależnie od siebie Andrzej Mostowski i Czesław Ryll-Nardzewski w 1952. Obie prace ukazały się w tym samym polskim czasopiśmie Fundamenta Mathematicae w tym samym roku, a nawet wydaniu.

Twierdzenie 11.4 (Mostowski 1952 [31], Ryll-Nardzewski 1952 [35])**.** *Dla każdego skończonego podzbioru Σ' zbioru aksjomatów Σ arytmetyki Peana mamy $\Sigma \vdash con(\Sigma')$.*

Podobne twierdzenie udowodnili także dla ZFC i ZF.

Twierdzenie 11.5 (Mostowski 1952 [31], Ryll-Nardzewski 1952 [35])**.** *Dla każdego skończonego podzbioru Σ' zbioru aksjomatów Σ teorii mnogości ZFC (lub ZF) mamy $\Sigma \vdash con(\Sigma')$.*

Nie oznacza to, że każde rozszerzenie arytmetyki Peana czy ZFC nie jest skończenie aksjomatyzowalne. Istotnie, teoria klas NBG (von Neumanna-Bernaysa-Gödla) jest rozszerzeniem ZFC i jest skończenie aksjomatyzowalna.

Definicja 11.6. *Mówimy, że zbiór zdań Σ jest ω-niesprzeczny, jeśli dla każdej formuły A, dla której z Σ można dowieść $A(0), A(1), \ldots$, z Σ nie można dowieść $\exists n\colon \neg A(n)$.*

Obserwacja 11.7. *Jeśli zbiór zdań Σ jest ω-niesprzeczny, to jest niesprzeczny.*

Uwaga! Implikacja w drugą stronę nie zachodzi!

Twierdzenie 11.8 (Gödel 1931 [18], pierwsze twierdzenie Gödla, o niezupełności)**.** *Niech $S = (A, B, T, \Sigma)$ będzie standardowym systemem formalnym z rekurencyjnie aksjomatyzowalnym zbiorem aksjomatów Σ. Jeśli arytmetyka Peana zawiera się w zbiorze konsekwencji logicznych zbioru Σ, to zbiór Σ nie może być jednocześnie ω-niesprzeczny i zupełny.*

Rosser w 1936 roku [34] uogólnił powyższe twierdzenie przez zastąpienie w nim ω-niesprzeczności przez niesprzeczność. Istnieją zbiory aksjomatów, które być może są niesprzeczne, ale nie są ω-niesprzeczne. Dla przykładu $PA \cup \{Con(PA)\}$ nie jest ω-niesprzeczny, ale jest niesprzeczny, jeśli Arytmetyka Peana jest niesprzeczna. Zatem twierdzenia Gödla w wersji Rossera są istotnie silniejsze niż oryginalne, o ile arytmetyka Peana faktycznie jest niesprzeczna (czego na mocy pierwszego twierdzenia Gödla nie można dowieść).

Twierdzenie 11.9 (von Neumann 1930, Gödel 1931 [18], drugie twierdzenie Gödla, o niedowodliwości niesprzeczności)**.** *Niech $S = (A, B, T, \Sigma)$ będzie standardowym systemem formalnym z rekurencyjnie aksjomatyzowalnym zbiorem aksjomatów Σ. Jeśli arytmetyka Peana zawiera się w zbiorze konsekwencji logicznych zbioru Σ i ω-niesprzeczna, to zbiór Σ nie może dowieść swojej niesprzeczności.*

Von Neumann w 1930 roku po przeczytaniu manuskryptu Gödla z pierwszym twierdzeniem zauważył, że można je wzmocnić do tego, co nazywamy dzisiaj drugim twierdzeniem. Nie opublikował on jednak tego wyniku, a Gödel włączył wynik do swojej pracy bez współautorstwa von Neumanna [16]. Założenie o standardowym systemie formalnym jest konieczne. W szczególności Gentzen w 1939 roku pokazał, że arytmetyka Peana jest niesprzeczna w silniejszym systemie formalnym, w którym poza składowymi standardowego systemu formalnego możliwa jest indukcja do liczby porządkowej ε_0. Co więcej, Gentzen w 1943 roku pokazał, że założenie o możliwej indukcji jest konieczne i wystarczające dla jego twierdzenia. O tym, że podany system formalny jest silniejszy, świadczy np. to, że prawdziwe jest w nim twierdzenie Goodsteina, które jest niedowodliwe w arytmetyce Peana (w standardowym systemie formalnym).

Dowiedziemy zaraz silniejszej wersji twierdzenia Gödla, która pochodzi od Kleene'go. Potrzebujemy jednak kilku definicji i jednego lematu.

Definicja 11.10. *Mówimy, że funkcja g jest uzupełnieniem częściowej funkcji jednoargumentowej f o dziedzinie $D_f \subseteq \mathcal{N}$, jeśli ma dziedzinę $\mathcal{N}$ i $f(x) = g(x)$, jeśli $f(x)$ jest zdefiniowane. Uzupełnienie funkcji f oznaczamy przez $\bar{f}$.*

Definicja 11.11. *Mówimy, że częściowa funkcja f jest potencjalnie rekurencyjna, jeśli istnieje jej uzupełnienie, które jest rekurencyjne.*

Lemat 11.12. *Istnieje częściowa funkcja rekurencyjna, która nie jest potencjalnie rekurencyjna.*

Dowód. Niech $f(n) = U(\mu_z \colon C(n, n, z)) + 1$, jeśli wartość po prawej stronie jest zdefiniowana. Funkcja $f(n)$ przyjmuje wartość $f_n(n)+1$, gdy ta jest zdefiniowana, i $f(n)$ jest niezdefiniowana w przeciwnym przypadku. Z konstrukcji $f(n)$ wynika, że jest ona rekurencyjna. Pokażemy, że nie jest ona uzupełnialna do totalnej funkcji rekurencyjnej.

Przypuśćmy nie wprost, że istnieje funkcja g, która jest rekurencyjnym uzupełnieniem funkcji f. Skoro g jest totalna, to $g(e) = f_e(e)$ jest zdefiniowana, a więc $f(e) = f_e(e) + 1$ jest zdefiniowana, ale $f(e) = g(e)$, więc $f_e(e) = g(e) = f(e) = f_e(e) + 1$, co daje sprzeczność i kończy dowód twierdzenia. ■

Definicja 11.13. *Mówimy, że system formalny $S = (A, B, T, \Sigma)$ pozwala na reprezentowanie k-argumentowej funkcji f, jeśli istnieje $k+1$ argumentowa formuła logiczna $\varphi(\bar{x}, y)$, taka że dla każdej k-krotki $\bar{m}$ oraz każdej liczby naturalnej n mamy:*
1) *Jeśli $f(\bar{m}) = 0$, to $\Sigma \vdash \varphi(\bar{m}, n)$.,*
2) *Jeśli $f(\bar{m}) \neq 0$, to $\Sigma \vdash \neg\varphi(\bar{m}, n)$.*

Twierdzenie 11.14 (Kleene, uogólnione pierwsze twierdzenie Gödla)**.** *Niech $S = (A, B, T, \Sigma)$ będzie standardowym systemem formalnym. Jeśli system formalny $\mathcal{S} = (L, A, B, \Sigma)$ pozwala na reprezentowanie każdej funkcji prymitywnie rekurencyjnej, zbiór Σ jest rekurencyjny i ω-niesprzeczny, to zbiór Σ nie jest zupełny.*

Dowód. Załóżmy, że Σ jest jednocześnie ω-niesprzeczny i zupełny. Skoro $\mathcal{S}$ pozwala na reprezentowanie każdej funkcji rekurencyjnej jednej zmiennej, to z twierdzenia Kleene'go o postaci normalnej istnieje formuła $\mathcal{C}$, która pozwala na reprezentowanie każdej funkcji prymitywnie rekurencyjnej. Niech $C = \mathcal{T}_1$ będzie funkcją z twierdzenia Kleene'go o postaci normalnej. Weźmy dowolne $e \in \omega$. Wówczas istnieje funkcja f, taka że $e = e_f$. Zdefiniujmy $\bar{f}_e$ w następujący sposób:
(1) $\bar{f}_e(n) = U(\mu_z \colon C(e, n, z))$, jeśli (1) $\exists z C(e, n, z) = 0$,
(2) $\bar{f}_e = 0$, jeśli (2) $\Sigma \vdash \forall z \neg\mathcal{C}(e, n, z, 0)$.
Dalsza część dowodu będzie polegać na pokazaniu, że $\bar{f}_e$ jest rekurencyjnym uzupełnieniem funkcji f_e. Pokażemy, że przy założeniach twierdzenia $\bar{f}_e$ jest rekurencyjną totalną funkcją dla każdego e.

Najpierw pokażemy, że jest to istotnie częściowa funkcja. Aby to pokazać, musimy uzasadnić, że warunki (1) i (2) są rozłączne. Załóżmy, że oba warunki zachodzą równocześnie. Wówczas istnieje z_0, takie że $C(e, n, z_0) = 0$ oraz $\Sigma \vdash \forall z \neg\mathcal{C}(e, n, z, 0)$, czyli w szczególności dla $z = z_0$ otrzymujemy $\Sigma \vdash \forall z \neg\mathcal{C}(e, n, z_0, 0)$. Skoro $\mathcal{C}$ reprezentuje C, to $C(e, n, z_0) = 0$ implikuje, że (2) $\Sigma \vdash \mathcal{C}(e, n, z_0, 0)$. To prowadzi do sprzeczności z tym, że T jest niesprzeczne.

Pokażemy teraz, że częściowa funkcja $\bar{f}_e$ jest funkcją totalną. Załóżmy, że warunek (1) nie zachodzi. Wówczas dla każdego k nie jest prawdą, że $C(e, n, k) = 0$ Skoro $\mathcal{S}$ reprezentuje C, to dla każdego z mamy $\Sigma \nvdash \mathcal{C}(e, n, z, 0)$. Skoro Σ jest zupełne, to oznacza to, że $\forall z \Sigma \vdash \neg\mathcal{C}(e, n, z, 0)$. Biorąc $z = k$ otrzymujemy sprzeczność.

Pokażemy teraz, że $\bar{f}_e$ jest (efektywnie) obliczalne. Zgodnie z wersją tezy Churcha, o równoważności modelu Turinga z modelem funkcji rekurencyjnych, jest to równoważne temu, że $\bar{f}_e$ jest rekurencyjne. Dokładniej: funkcja jest rekurencyjna

wtedy i tylko wtedy, gdy można ją obliczyć (przedstawić) na pewnej maszynie Turinga.

Dla wejścia n przeprowadzamy dwa przeszukiwania równoległe (tj. naprzemiennie krokami z pierwszego i drugiego przeszukiwania). Jedno przeszukiwanie przechodzi przez liczby $k = 0, 1, 2, \ldots$ i sprawdza, czy $C(e, n, k) = 0$. Jeśli tak, to dla pierwszego znalezionego takiego k przyjmujemy $\bar{f}_e(n) = U(\mu_z \colon C(e, n, z)) = k$. Może to zrobić ponieważ relacja $C(e, n, k)$ jest rekurencyjna. Drugie przeszukiwanie przebiega przez dowody w w $\mathcal{S}$ i sprawdza czy w dowodzi zdania $\varphi = (\forall z \neq \mathcal{C}(e, n, z, 0))$. Jedno z przeszukiwań musi się zakończyć, ponieważ T jest rekurencyjnie aksjomatyzowalne. Wynika z tego, że $\bar{f}_e$ jest obliczalna.

Wobec dowolności wyboru parametru e, wynika stąd, że każda rekurencyjna funkcja f_e jest potencjalnie rekurencyjna, co prowadzi do sprzeczności i kończy dowód twierdzenia. ■

Pytanie: co by było, gdybyśmy spróbowali dowieść niesprzeczności aksjomatyki Peana lub ZFC w odpowiednio „mocnym" systemie formalnym. Jaka stoi na drodze przeszkoda, aby w ten sposób dowieść, że systemy PA i ZFC są niesprzeczne? Porównaj lemat Lindenbauma z twierdzeniem Gödla.

Twierdzenie 11.15. *Niech Σ zbiorem aksjomatów pewnego systemu formalnego spełniającego założenia twierdzenia Gödla. Wówczas, jeśli zbiór $\Sigma^* \subset \Sigma$ jest skończony, to $\Sigma \vdash Con(\Sigma^*)$.* ■

Obserwacja 11.16. *Standardowa aksjomatyka teorii mnogości ZFC (a także ZF) spełniają założenia twierdzenia Gödla.* ■

Definicja 11.17. *Niech $\#\sigma$ będzie numerem Gödla zdania σ. Formułę T nazywamy definicją prawdy, jeśli dla każdego x mamy $T(x) \Rightarrow x \in \omega$ oraz $T(\#\sigma) \Leftrightarrow \sigma$.*

Twierdzenie 11.18 (Tarski)**.** *Definicja prawdy nie istnieje.*

Dowód. Niech $(\varphi_i \colon i \in \omega)$ będzie enumeracją wszystkich formul z jedną wolną zmienną. Niech $\psi(x)$ będzie formulą:

$$\psi(x) \Leftrightarrow x \in \omega \wedge \neg T(\#\varphi_x(x)).$$

Istnieje liczba naturalna k, taka że $\psi = \varphi_k$. Wówczas mamy

$$\sigma \Leftrightarrow \psi(k) \Leftrightarrow \neg T(\#\varphi_x(x)) \Leftrightarrow \neg T(\#\psi(k),$$

co prowadzi do sprzeczności. ■

12. Λ-rachunek *

Λ-rachunek jest alternatywnym (w stosunku do klasycznego, teoriomnogościowego) spojrzeniem na definicję funkcji. Funkcje w Λ-rachunku rozumiane są jako przepisy, pewne procedury, co podkreśla ich obliczalność. Rachunek Λ dzieli się na nietypowany oraz typowany. My jednak zajmiemy się tylko pierwszym z nich.

Napisy (poprawnie zbudowane wyrażenia) w Λ-rachunku nazywane są termami. Zdefiniujemy je formalnie:

Definicja 12.1. *Niech Var będzie nieskończonym zbiorem przeliczalnym nazywanym zbiorem zmiennych i niech symbole kropki i nawiasów otwartych oraz symbol λ nie należą do zbioru Var. Zbiorem termów Λ-rachunku nazywamy najmniejszy zbiór Λ skonstruowany w następujący sposób:*

1. *Każda zmienna ze zbioru Var jest termem Λ-rachunku,*
2. *Dla każdego termu M ze zbioru Λ oraz dowolnej zmiennej x ze zbioru Var term postaci $(\lambda x.M)$ jest termem rachunku lambda. Opisany sposób tworzenia termów nazywamy abstrakcją.*
3. *Jeżeli M i N są termami rachunku Λ, to (MN) również jest termem. Ten sposób tworzenia termów nazywamy aplikacją.*

Reguły opuszczania nawiasów:
Nawiasy są często pomijane według łączności lewostronnej, dla przykładu term $MNPQ$ jest tym samym, co term $(((MN)P)Q)$. Wieloargumentowe funkcje są reprezentowane przez funkcje jednoargumentowe, których argumentami też są funkcje. To, co standardowo zapisalibyśmy jako $F(N, M)$ zapisujemy jako $((FN)M)$ lub FNM. Dodatkowo powtórzenia symbolu λ są pomijane według zasady łączności prawostronnej, na przykład $(\lambda x.(\lambda y.(\lambda z.M)))$ jest tym samym co $\lambda xyz.M$.

Heurystycznie: Mówimy, że zmienna x w termie M jest związana, jeśli znajduje się w zasięgu operatora λx. W przeciwnym wypadku mówimy, że zmienna jest wolna. Bardziej formalnie:

Definicja 12.2. *Niech M będzie dowolnym termem. Zbiorem zmiennych wolnych termu M (oznaczenie $FV(M)$) nazywamy minimalny zbiór skonstruowany w następujący sposób:*
1. *Jeśli M jest pojedynczą zmienną, to M jest wolna,*
2. *Jeśli M jest postaci $\lambda x.N$ dla pewnego termu N, to operator λx wiąże wszystkie zmienne w termie N. Zatem $FV(M) = FV(N) \setminus \{x\}$.*
3. *Jeżeli M jest postaci PQ, to zmiennymi wolnymi termu M są wszystkie zmienne wolne występujące w termie P lub Q. Tj. $FV(M) = FV(P) \cup FV(Q)$.*

Zmienną występującą w termie M, która nie jest wolna, nazywamy związaną.

Definicja 12.3. *Term M nazywamy domkniętym, jeśli nie ma zmiennych wolnych.*

Przykłady:
- W termie $M = y(\lambda x.x)$ zmienna y jest zmienną wolną, a x zmienną związaną.
- W termie $M = x(\lambda x.x)$ pierwsze wystąpienie zmiennej x jest wolne, a więc $FV(M) = \{x\}$.

Na termy w Λ-rachunku patrzymy jako na przepisy, jak z zadanej wartości argumentu otrzymać wartość funkcji. Termy interpretujemy jako procedury, których parametry specyfikowane są poprzez abstrakcję.

Przedstawimy teraz dwie operacje, za pomocą których dokonywane są obliczenia w w Λ-rachunku: α-konwersję i β-redukcję.

α-konwersja polega na zamianie nazw zmiennych **związanych**. W przypadku, gdy poprzez stosowanie „podprocedur" lub parametrów pochodzących z wykonania innych procedur dochodzi do konfliktu nazw zmiennych. Aby zapewnić poprawne wykonanie, stosujemy α-konwersję, która pozwala zapewnić unikalność nazw.

$$\lambda x.M \to_\alpha \lambda y.M[x/y],$$

gdzie $y \notin FV(M)$, a $M[x/y]$ oznacza term powstały przez podstawienie y za każde związane wystąpienie zmiennej x w termie M.

Będziemy używać oznaczenia $T_1 =_\alpha T_2$, jeśli termy T_1 i T_2 są sobie równe modulo α-konwersja. Przykłady:
- $\lambda x.x =_\alpha \lambda y.y$,
- $\lambda x.xz =_\alpha \lambda y.yz$,
- $\lambda x.xy \neq_\alpha \lambda x.xz$.

β-redukcja polega na wykonaniu obliczenia przez procedurę. Aplikacja procedury $\lambda x.M$ do argumentu N:

$$(\lambda x.M)N \to_\beta M[x/N].$$

działa w taki sposób, że każda zmienna wolna termu N pozostaje wolna po podstawieniu. Jest to możliwe do osiągnięcia po ewentualnym wcześniejszym zastosowaniu α-konwersji. Na przykład

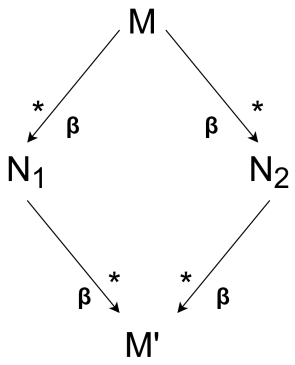
$$(\lambda xy.xy)y \not\twoheadrightarrow_\beta \lambda y.yy$$

jest nieprawidłowym użyciem β-konwersji, gdyż dochodzi do konfliktu oznaczeń i do utraty pierwotnego znaczenia termu (procedury). Możemy temu zaradzić przez wcześniejsze zastosowanie α-konwersji:

$$(\lambda xy.xy)y =_\alpha (\lambda xz.xz)y \to_\beta \lambda z.yz.$$

Podobnie będziemy używać oznaczenia $T_1 =_\beta T_2$, jeśli termy T_1 i T_2 są sobie równe modulo β-konwersja.

Twierdzenie 12.4 (Churcha-Rossera 1936, własność karo)**.** *Jeżeli term M poprzez pewne ciągi β-redukcji redukuje się do termów N_1 oraz N_2, to istnieje term M', taki że zarówno N_1 jak i N_2 można zredukować do termu M'.* ■



12.1. Numerale Churcha

Popatrzmy teraz na liczbę naturalną n jako na procedurę, która n-krotnie stosuje operację następnika do zera. Taka procedura w jednoznaczny sposób określa liczbę n i na odwrót. Ta idea prowadzi do pojęcia numerali Churcha.

Zdefiniujmy $0 = \lambda sx.x$, $1 = \lambda sx.sx$, $2 = \lambda sx.s(sx)$, $3 = \lambda sx.s(s(sx))$ i ogólnie $n = \lambda sx.s(s(\ldots(x)\ldots))$. Gdzie s występuje n razy. Numerale Churcha są zdefiniowane z dokładnością do α-konwersji.

Operacje na liczbach:
Term $Add = \lambda mnsx.(ms)(nsx)$ reprezentuje dodawanie.
Przykład: $Add\ 1\ 2 = \ldots.$
Term $Mult = \lambda mns.m(ns)$ reprezentuje mnożenie.
Przykład: $Mult\ 2\ 2 = \ldots.$

Twierdzenie 12.5 (Kleene 1936)**.** *Każda funkcja rekurencyjna jest reprezentowalna w* Λ*-rachunku.* ■

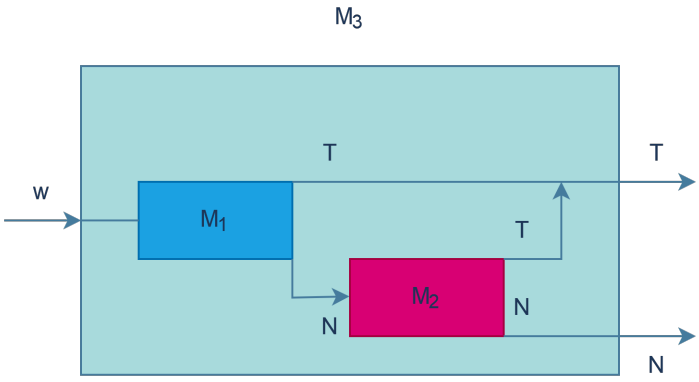
Twierdzenie 12.6 (Church 1936)**.** *Każda funkcja reprezentowalna w* Λ*-rachunku jest rekurencyjna.* ■

13. Obliczalność

Wrócimy teraz do wcześniej omawianych pojęć maszyn Turinga oraz języków rekurencyjnych i rekurencyjnie przeliczalnych. Przedstawimy teraz kilka prostych faktów na temat klas tych klas języków. Dla języka L oznaczmy przez $\overline{L}$ jego dopełnienie.

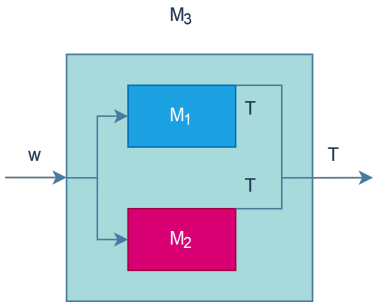
Twierdzenie 13.1. *Klasy języków rekurencyjnych jest zamknięta na sumę.*

Dowód. Niech M_1 będzie maszyną Turinga ze stopem rozpoznającą język L_1 i niech M_2 będzie maszyną Turinga ze stopem rozpoznającą język L_2. Skonstruujemy maszynę Turinga M_3 ze stopem, która rozpoznaje język $L_1 \cup L_2$. Maszyna M_3 najpierw symuluje działanie maszyny na wejściowym słowie. Jeśli M_1 da odpowiedź „tak", to M_3 daje odpowiedź „tak". Jeśli M_1 da odpowiedź „nie", to symulujemy działanie maszyny M_2 na wejściowym słowie. Jeśli M_2 da odpowiedź „tak", to M_3 daje odpowiedź „tak". Jeśli M_2 daje odpowiedź „nie", to M_3 daje odpowiedź „nie".



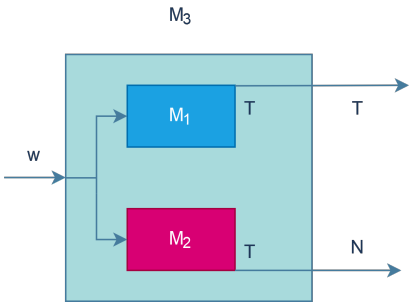
Twierdzenie 13.2. *Klasy języków rekurencyjnie przeliczalnych jest zamknięta na sumę.*

Dowód. Niech M_1 będzie maszyną Turinga rozpoznającą język L_1 i niech M_2 będzie maszyną Turinga rozpoznającą język L_2. Skonstruujemy dwutaśmową maszynę Turinga M_3, która rozpoznaje język $L_1 \cup L_2$. Maszyna M_3 jednocześnie symuluje działanie maszyn M_1 oraz M_2 na wejściowym słowie na oddzielnych taśmach. Jeśli któraś z maszyn M_1 i M_2 da odpowiedź „tak", to maszyna M_3 daje odpowiedź „tak".



Twierdzenie 13.3. *Jeżeli języki L i $\overline{L}$ są rekurencyjnie przeliczalne, to są też rekurencyjne.*

Dowód. Niech M_1 będzie maszyną Turinga rozpoznającą język L i niech M_2 będzie maszyną Turinga rozpoznającą język $\overline{L}$. Skonstruujemy dwutaśmową maszynę Turinga M_3 ze stopem, która rozpoznaje język L. Maszyna Turinga M_3 równocześnie na oddzielnych taśmach symuluje działania maszyn M_1 oraz M_2 na wejściowym słowie. Jeśli maszyna M_1 da odpowiedź „tak", to maszyna M_3 daje odpowiedź „tak". Jeśli maszyna M_2 da odpowiedź „tak", to maszyna M_3 daje odpowiedź „nie". Tezę dla języka $\overline{L}$ otrzymujemy przez zastąpienie L przez $\overline{L}$.



Twierdzenie 13.4. *Jeśli język L jest rekurencyjny, to $\overline{L}$ też jest rekurencyjny.*

Dowód. Ćwiczenia. ■

14. Arytmetyzacja maszyn Turinga

Arytmetyzacja Maszyn Turinga polega na następujących kodowaniach:
Kodowanie instrukcji $\delta(q_i, X_j) = (q_k, X_l, D_m)$, gdzie $D_1 = L$, $D_2 = R$ wygląda następująco:

$$code_1 = 0^i 10^j 10^k 10^l 10^m.$$

Następnie kodowanie maszyny Turinga:

$$111code_1 11code_2 11 \dots 111.$$

Jako że nie mamy zadanej kolejności instrukcji, to maszyna Turinga ma wiele kodów. Oznaczmy przez $< M >$ najmniejszy w porządku leksykograficznym kod maszyny M.

Twierdzenie 14.1. *Istnieje maszyna Turinga ze stopem, która rozpoznaje, czy dane słowo $w \in \{0,1\}^*$ jest kodem pewnej maszyny Turinga.*

Twierdzenie 14.2. *Istnieją maszyny P, Q, R ze stopem takie, że: maszyna P odpowiada na pytanie, czy wejście w jest kodem pewnej maszyny Turinga; dwutaśmowa maszyna Q mając na wejściu liczbę i generuje maszynę Turinga o kodzie i na drugiej taśmie, jeśli i jest kodem pewnej maszyny Turinga; dwutaśmowa maszyna R odpowiada na pytanie, czy wejście w jest równe pewnemu $< M_i >$ i jeśli tak, to generuje liczbę i na drugiej taśmie.*

Niech $obl(M, w, i) = w_0\$w_1\$ \dots w_i$, gdzie $w = w_0$ oraz w_k dla $k > 1$ oznacza stany maszyny M w momencie k, która wykonuje obliczenia na słowie w.

Definicja 14.3. *Językiem obliczeń wszystkich maszyn Turinga nazywamy język*

$$OBL = \{< M > obl(M, w, i) \colon M \text{ - maszyna Turinga}, w \in \{0,1\}^*, i \in \mathbb{N}\}.$$

Twierdzenie 14.4. *Język OBL jest językiem rekurencyjnym.*

Dowód. Maszyna rozpoznająca OBL czyta kod <M> i zaczyna symulować M do i-tego kroku na słowie w i porównuje wynik z wejściem. Maszyna OBL musi też sprawdzić, czy podany kod maszyny M jest najmniejszym w porządku leksykograficznym z tych kodów, co może zrobić za pomocą symulacji maszyny R. ■

Definicja 14.5. *Językiem L_d nazywamy język*

$$L_d = \{w_i \in \Sigma^* : w_i \text{ nie jest akceptowane przez } M_i\}.$$

Twierdzenie 14.6. *Język L_d nie jest rekurencyjnie przeliczalny.*

Dowód. Załóżmy nie wprost, że L_d jest rekurencyjnie przeliczalny. Niech i_0 będzie kodem maszyny Turinga rozpoznającej język L_d, czyli $L_d = L(M_{i_0})$. Rozważmy słowo w_{i_0}:

- Jeśli $w_{i_0} \in L(M_{i_0}) = L_d$, to w_{i_0} nie jest akceptowane przez maszynę M_{i_0}, czyli $w_{i_0} \notin L(M_{i_0})$.
- Jeśli $w_{i_0} \notin L(M_{i_0}) = L_d$, to w_{i_0} jest akceptowane przez maszynę M_{i_0}, czyli $w_{i_0} \in L(M_{i_0})$. ■

Wniosek 14.7. *Poniższy problem decyzyjny jest nierozstrzygalny.*
Instancja: słowo $w_i \in \Sigma^*$.
Pytanie: czy w_i nie jest akceptowane przez maszynę Turinga o numerze i? ■

Definicja 14.8. *Językiem uniwersalnym nazywamy język*

$$L_u = \{< M > w : M \text{ akceptuje } w\}.$$

Twierdzenie 14.9. *Język L_u jest językiem rekurencyjnie przeliczalnym.*

Dowód. Maszyna Turinga rozpoznająca L_u czyta kod maszyny M, a następnie symuluje jej działanie na słowie w. ■

Definicja 14.10. *Maszynę Turinga rozpoznającą język L_u (działającą jak w powyższym dowodzie) nazywamy maszyną uniwersalną i oznaczamy ją przez M_u.*

Twierdzenie 14.11. *Język L_u nie jest rekurencyjny.*

Dowód. Załóżmy nie wprost, że istnieje maszyna Turinga ze stopem M_u, która rozpoznaje język L_u. Dojdziemy do sprzeczności przez skonstruowanie maszyny Turinga rozpoznającej język L_d. ■

Wniosek 14.12. *Język $\overline{L}_u$ nie jest rekurencyjnie przeliczalny.*

Dowód. Jeśli L_u i $\overline{L}_u$ byłyby rekurencyjnie przeliczalne, to L byłby też rekurencyjny. ■

Definicja 14.13. *Językiem L_{HP} (od halting problem, problem stopu) nazywamy język*

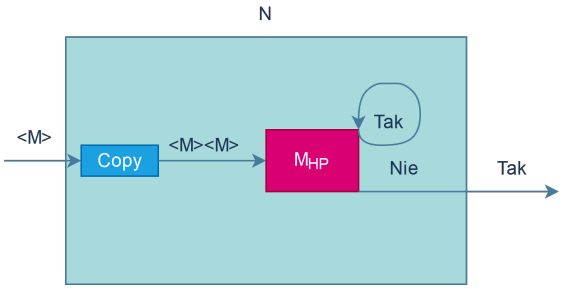
$$L_{HP} = \{< m > w\text{: maszyna } M \text{ zatrzymuje się na słowie } w\}.$$

Twierdzenie 14.14. *Język L_{HP} nie jest rekurencyjny.*

Dowód. Załóżmy, że istnieje maszyna ze stopem M_{HP}, która rozpoznaje język HP. Kodujemy maszynę N w sposób następujący. Wejściem maszyny N jest kod maszyny Turinga $< M >$. Najpierw N podwaja kod, to znaczy tworzy nowe słowo $< M >< M >$, a następnie przekazuje podwojony kod do maszyny M_{HP}. Teraz, jeśli M_{HP} da odpowiedź „tak", to zapętlamy maszynę M_{HP}, jeśli M_{HP} da odpowiedź „nie", to maszyna N daje odpowiedź „tak".

Zauważmy, że N da odpowiedź „tak" wtedy i tylko wtedy, gdy M zapętla się na swoim kodzie. Natomiast N zapętla się wtedy i tylko wtedy, gdy M zatrzymuje się na słowie $< M >$.

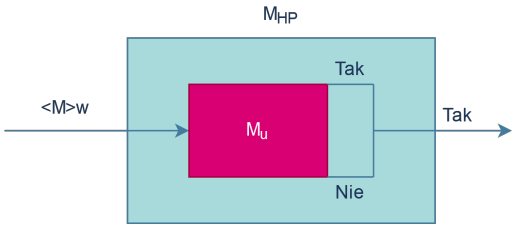
Rozważmy teraz maszynę N, które na wejściu dostaje swój kod $< N >$. Wówczas jeśli maszyna N odpowiedziałaby „tak", to oznaczałoby, że N zapętla się. Natomiast jeśli maszyna N zapętliłaby się, to oznaczałoby to, że N zatrzymuje się.



■

Twierdzenie 14.15. *Język L_{HP} jest rekurencyjnie przeliczalny.*

Dowód. Skonstruujemy maszynę Turinga M_{HP}, która rozpoznaje język L_{HP}. Maszyna M_{HP} dostaje na wejściu słowo $< M > w$. Następnie symuluje na uniwersalnej maszynie Turinga działanie maszyny $< M >$ na słowie w. Niezależnie od tego, czy dostanie odpowiedź „tak" czy „nie" daje odpowiedź „tak".



Wniosek 14.16. *Język $\overline{L}_{HP}$, mówiący o tym, czy maszyna się zapętli, nie jest rekurencyjnie przeliczalny.*

Dowód. Załóżmy, że $\overline{L}_{HP}$ jest rekurencyjnie przeliczalny. Wówczas zarówno L_{HP} jak i $\overline{L}_{HP}$ są rekurencyjnie przeliczalne. Wynika z tego, że L_{HP} jest rekurencyjnie przeliczalny, co daje sprzeczność.

15. Własności klas języków

Niech $\mathcal{S}$ będzie klasą pewnych języków rekurencyjnie przeliczalnych (nad zadanym alfabetem). Klasę $\mathcal{S}$ nazywamy też własnością języków. Przez $L_{\mathcal{S}}$ będziemy oznaczać zbiór $L_{\mathcal{S}} = \{< M >: L < M >\in \mathcal{S}\}$. Będziemy mówić, że klasa $\mathcal{S}$ jest rozstrzygalna, rekurencyjnie przeliczalna, etc., jeśli $L(S)$ jest rozstrzygalna, rekurencyjnie przeliczalna, etc..

Definicja 15.1. *Mówimy, że klasa $\mathcal{S}$ jest trywialna, jeśli $\mathcal{S}$ jest pusta lub $\mathcal{S}$ jest klasą wszystkich języków rekurencyjnie przeliczalnych.*

Twierdzenie 15.2 (Rice 1953)**.** *Klasa $\mathcal{S}$ jest rekurencyjna wtedy i tylko wtedy, gdy jest trywialna.* ■

Definicja 15.3. *Mówimy, że klasa języków rekurencyjnie przeliczalnych $\mathcal{S}$ jest monotoniczna, jeśli dla każdych dwóch języków rekurencyjnie przeliczalnych L_1 i L_2, jeśli $L_1 \in \mathcal{S}$ oraz $L_1 \subseteq L_2$, to $L_2 \in \mathcal{S}$.*

Definicja 15.4. *Mówimy, że język $L \in \mathcal{S}$ ma podjęzyk skończony w $\mathcal{S}$, jeśli istnieje skończony język $L' \subseteq L$, taki że $L' \in \mathcal{S}$.*

Nieformalnie:

Definicja 15.5. *Numeratorem języka L nazywamy maszynę Turinga z „drukarką", która drukuje listę (być może z powtórzeniami) wszystkich słów należących do L.*

Definicja 15.6. *Mówimy, że język L jest numerowalny, jeśli istnieje numerator dla języka L.*

Twierdzenie 15.7. *Język L jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy jest numerowalny.*

Dowód. Załóżmy, że istnieje numerator E, który rozpoznaje język A. Pokażemy, że istnieje maszyna Turinga M, która rozpoznaje język A. Maszyna Turinga M działa w następujący sposób:

1. wejście: słowo w,
2. uruchamiamy numerator E,
3. za każdym razem, gdy numerator drukuje słowo w' porównujemy je ze słowem w i akceptujemy, jeśli $w = w'$.

Skoro numerator E numeruje wszystkie elementy A (i tylko elementy A), to słowo w w końcu zostanie wydrukowane przez numerator E i w konsekwencji zaakceptowane przez maszynę M.

Załóżmy teraz, że istnieje maszyna Turinga M rozpoznająca język A. Skonstruujemy numerator E, który rozpoznaje język A. Niech $s_1, \ldots$ będzie listą wszystkich możliwych słów w Σ^*. Numerator E działa w następujący sposób:

1. ignorujemy słowo wejściowe, czyli numerator E działa tak samo niezależnie od wejścia,
2. uruchamiamy maszynę Turinga M przez i kroków kolejno dla każdego z wejść $s_1, \ldots, s_i$,
3. jeśli jakaś z konfiguracji zostanie zaakceptowana, to drukujemy odpowiadające wejście s_j.

Zauważmy, że jeśli maszyna Turinga M akceptuje pewne słowo w, to pojawi się ono na liście generowanej przez E. Co więcej, pojawi się na nim nieskończenie wiele razy. ■

Uwaga: stąd angielska nazwa języka rekurencyjnie przeliczalnego: recursively enumerable.

Twierdzenie 15.8 (Rice-Shapiro)**.** *K jeśli klasa $\mathcal{S}$ jest rekurencyjnie przeliczalna, to spełnione są wszystkie z następujących warunków:*

1. *klasa $\mathcal{S}$ jest monotoniczna,*
2. *każdy język z $\mathcal{S}$ ma podjęzyk skończony w $\mathcal{S}$.*

Część III
Złożoność obliczeniowa

16. Złożoność czasowa i pamięciowa maszyny Turinga

Klasyfikacja według złożoności obliczeniowej oparta jest na ilości czasu, miejsca czy innych zasobów (np. wyróżnionych operacji w obliczeniach kwantowych) potrzebnych do rozpoznania danego języka za pomocą urządzenia obliczeniowego, takiego jak maszyna Turinga.

Definicja 16.1. *Jeżeli dla dowolnego słowa $w \in \Sigma^*$ długości n maszyna Turinga M przegląda co najwyżej $S(n)$ klatek na taśmach roboczych przed przejściem do stanu akceptującego lub zatrzymania, to mówimy, że maszyna M ma złożoność pamięciową $S(n)$.*

Definicja 16.2. *Jeżeli dla dowolnego słowa $w \in \Sigma^*$ długości n maszyna Turinga M wykonuje co najwyżej $T(n)$ kroków przed przejściem do stanu akceptującego lub zatrzymania, to mówimy, że M ma złożoność czasową $T(n)$.*

Analogicznie możemy zdefiniować złożoność pamięciową i czasową dla niedeterministycznej maszyny Turinga, przy czym interesuje nas złożoność najdłuższej drogi prowadzącej do stanu akceptującego.

Definicja 16.3. *Niech $S(n)$ będzie pewną funkcją $S : \mathbb{N} \to \mathbb{N}$. Oznaczmy przez $DSPACE(S(n))$ rodzinę języków o złożoności pamięciowej $S(n)$.*

Definicja 16.4. *Niech $S(n)$ będzie pewną funkcją $S : \mathbb{N} \to \mathbb{N}$. Oznaczmy przez $DTIME(S(n))$ rodzinę języków o złożoności czasowej $S(n)$.*

Analogicznie definiujemy klasy $NSPACE(S(n))$ oraz $NTIME(S(n))$. Zwykle w przypadku złożoności ważne jest funkcyjne tempo wzrostu (liniowe, kwadratowe, wykładnicze, ...), a czynniki stałe pomijamy. Poniższe twierdzenia mówią o tym, że w większości przypadków te czynniki są istotnie pomijalne i nie możemy wybrać optymalnego.

Twierdzenie 16.5 (O przyspieszeniu liniowym)**.** *Jeśli język L jest rozpoznawalny przez maszynę Turinga o złożoności czasowej $T(n)$ oraz $\lim_{n\to\infty} \frac{T(n)}{n} = 0$, to jest rozpoznawalny przez maszynę Turinga o złożoności czasowej $c(T(n))$ dla dowolnego $c > 0$.* ■

Twierdzenie 16.6 (Blum 1967, o liniowej kompresji pamięci)**.** *Jeśli język L jest rozpoznawalny przez maszynę Turinga o złożoności pamięciowej $S(n)$ oraz $\lim_{n\to\infty} \frac{S(n)}{n} = 0$, to język L jest rozpoznawalny przez maszynę Turinga o złożoności pamięciowej $c(S(n))$ dla dowolnego $c > 0$.* ■

Pytanie: Czy istnieje ograniczenie czasowe lub pamięciowe $f(n)$, takie że każdy język rekurencyjny należy do $DSPACE(f(n))$ lub $DTIME(f(n))$?

Pytanie: Czy będziemy mogli rozpoznawać nowe języki, jeśli pomnożymy funkcję $f(n)$ przez jakąś funkcję wolno rosnącą?

Definicja 16.7. *Mówimy, że funkcja $S(n)$ jest konstruowalna pamięciowo, jeśli istnieje maszyna Turinga z ograniczeniem pamięci $S(n)$, która dla dowolnego n zużywa dokładnie $S(n)$ komórek przy pewnym wejściu długości n.*

Definicja 16.8. *Mówimy, że funkcja $S(n)$ jest w pełni konstruowalna pamięciowo, jeśli istnieje maszyna Turinga z ograniczeniem pamięci $S(n)$, która dla dowolnego n zużywa dokładnie $S(n)$ komórek przy każdym wejściu długości n.*

Przykłady funkcji w pełni konstruowalnych pamięciowo: $\log n$, n^k, 2^n, $n!$.

Twierdzenie 16.9 (O hierarchii pamięciowej)**.** *Niech $S_1(n)$ będzie dowolną funkcją i niech $S_2(n)$ będzie w pełni konstruowalna pamięciowo. Niech $S_1(n) \geqslant \log n$ oraz niech $S_2(n) \geqslant \log n$. Jeśli $\lim_{n\to\infty} \frac{S_1(n)}{S_2(n)} = 0$, to*

$$DSPACE(S_2(n)) \setminus DSPACE(S_1(n)) \neq \emptyset.$$

Definicja 16.10. *Klasą języków P (polynomial time) nazywamy klasę*

$$P = \bigcup_{k=1}^{\infty} DTIME(n^k).$$

Są to wszystkie języki rozpoznawalne w czasie wielomianowym przez maszynę Turinga, czyli problemy „łatwe”.

Definicja 16.11. *Klasą języków NP (non-deterministic polynomial time) nazywamy klasę*

$$NP = \bigcup_{k=1}^{\infty} NTIME(n^k).$$

Są to wszystkie problemy rozpoznawalne w czasie wielomianowym przez niedeterministyczną maszynę Turinga. Różnica pomiędzy klasą NP a P jest taka, jak pomiędzy znalezieniem dowodu twierdzenia a sprawdzeniem, czy dany dowód jest poprawny. Sprostujmy jeszcze, że klasa NP nie oznacza klasy „non-polynomial". Nie oznacza też klasy wykładniczej EXP. Zdefiniujmy jeszcze.

Definicja 16.12. *Klasą języków $PSPACE$ nazywamy klasę*

$$PSPACE = \bigcup_{k=1}^{\infty} DSPACE(n^k).$$

Definicja 16.13. *Klasą języków $NPSPACE$ nazywamy klasę*

$$NPSPACE = \bigcup_{k=1}^{\infty} NSPACE(n^k).$$

Hipoteza 16.14. $P \neq NP$.

Powyższa hipoteza jest jednym z najważniejszych nierozstrzygniętych problemów nie tylko w informatyce, ale też w całej matematyce. W szczególności znajduje się ona na liście siedmiu problemów milenijnych obok hipotezy Riemanna. Jest to bardzo istotne, ponieważ jeśli $P = NP$, to wiele z wyników w teorii złożoności obliczeniowej trywializuje się. Większość matematyków (chociaż nie wszyscy) uważa jednak, że prawdopodobnie $P \neq NP$.

Twierdzenie 16.15. *Niech $f : \mathbb{N} \to \mathbb{N}$. Wówczas:*

$$DTIME(f(n)) \subseteq DSPACE(f(n)).$$

Dowód. Jeżeli Maszyna Turinga wykonuje nie więcej niż $f(n)$ ruchów, to nie może przejrzeć więcej niż $f(n) + 1$ komórek pamięci. ■

Twierdzenie 16.16. *Niech $f : \mathbb{N} \to \mathbb{N}$. Jeżeli $f(n) \geqslant \log_2 n$, to*

$$L \in DSPACE(f(n)) \Rightarrow \exists c(L) > 0 \colon L \in DTIME(c^{f(n)}).$$

Dowód. Liczba różnych konfiguracji maszyny Turinga o ograniczeniu pamięci $f(n)$, s stanach i t symbolach wejściowych przy wejściu długości n jest ograniczona. Po liczbie ruchów $c^{f(n)}$ dla pewnej stałej c maszyna Turinga musi powtórzyć jakąś konfigurację, a więc się zapętli. ■

Twierdzenie 16.17. *Niech $f : \mathbb{N} \to \mathbb{N}$.*

$$L \in NTIME(f(n)) \Rightarrow \exists c(L) > 0 \colon L \in DTIME(c^{f(n)}).$$

Dowód. Przejście z modelu niedeterministycznego do deterministycznego wymaga wykładniczego nakładu czasu. Konstruujemy listę wszystkich konfiguracji możliwych dla niedeterministycznej maszyny Turinga przy danym wejściu. Jest ich wykładnicza liczba. Następnie dla każdej konfiguracji wykonujemy deterministyczny algorytm, który ma złożoność czasową $f(n)$. ■

Twierdzenie 16.18 (Savitch 1970)**.** *Niech $S(n)$ będzie funkcją w pełni konstruowalną pamięciowo i niech $S(n) \geqslant \log n$. Zachodzi wówczas:*
$NSPACE(S(n)) \subseteq DSPACE(S^2(n))$. ■

17. Transformacja wielomianowa

Definicja 17.1. *Niech języki L i L' należą do Σ^*. Mówimy, że L transformuje się wielomianowo do L' (ozn. $L \leqslant_P L'$), jeśli istnieje funkcja $f\colon \Sigma^* \to \Sigma^*$ realizowana w czasie wielomianowym na maszynie Turinga taka, że dla dowolnego $x \in \Sigma^*$ zachodzi $x \in L$ wtedy i tylko wtedy, gdy $f(x) \in L'$.*

Definicja 17.2. *Problemem decyzyjnym Q nazywamy funkcję, która przyporządkowuje danym wejściowym (instancjom) liczbę 0 lub 1, gdzie 0 odpowiada słowu „nie”, a 1 odpowiada słowu „tak”. Dziedzinę problemu Q będziemy oznaczać przez D_Q, a zbiór tych instancji, które dają odpowiedź „tak”, będziemy oznaczać przez Y_Q.*

Podanie problemu decyzyjnego Q (dla ustalonej dziedziny) jest równoważne podaniu zbioru Y_Q. Dlatego często utożsamiamy problem Q ze zbiorem Y_Q. Również dowolny zbiór $Y \in \Sigma^*$ możemy utożsamić ze zbiorem Y_Q pewnego problemu decyzyjnego Q. Dlatego też wszystko, co mówiliśmy o językach możemy przenieść na problemy decyzyjne. Będziemy milcząco zakładać, że instancje są skończone.

Definicja 17.3. *Podproblemem problemu decyzyjnego Q nazywamy problem Q', taki że $D_{Q'} \subseteq D_Q$ oraz $Y_{Q'} = Y_Q \cap D_{Q'}$. Mówimy wtedy, że problem Q' powstaje przez zacieśnienie problemu Q do zbioru instancji $D_{Q'}$.*

Definicja 17.4. *Niech Q i Q' będą problemami decyzyjnymi. Mówimy, że Q transformuje się wielomianowo do Q', jeśli istnieje funkcja $f\colon D_Q \to D_{Q'}$ realizowana w czasie wielomianowym na maszynie Turinga taka, że dla dowolnego $x \in D_Q$ zachodzi $x \in Y_Q$ wtedy i tylko wtedy, gdy $f(x) \in Y_{Q'}$. Piszemy wtedy, że $Q \leqslant_P Q'$.*

Obserwacja 17.5. *Jeśli $Q \leqslant_P Q'$ oraz $Q' \in P$, to $Q \in P$.* ■

Obserwacja 17.6. *Relacja $\leqslant_P$ jest relacją przechodnią.* ■

Definicja 17.7. *Jeżeli $Q \leqslant_P Q'$ oraz $Q' \leqslant_P Q$, to mówimy, że problemy Q i Q' są równoważne.*

Definicja 17.8. *Niech C będzie klasą problemów. Mówimy, że problem L jest jest trudny w klasie C (lub jest C-trudny), jeśli dla każdego problemu L' w klasie C zachodzi $L' \leqslant_P L$. Problem L, który należy do C i jest C-trudny nazywamy zupełnym w klasie C lub C-zupełnym.*

Obserwacja 17.9. *Jeśli dla pewnego problemu NP-zupełnego istniałby algorytm wielomianowy, to wówczas $P = NP$. Zatem, jeśli $P \neq NP$, to każdy problem NP-zupełny nie należy do klasy P.*

W celu udowodnienia, że problem decyzyjny jest zupełny w klasie NP, wykonujemy zwykle dwa kroki:

1. pokazujemy, że problem Q należy do klasy NP, czyli że jest obliczalny w czasie wielomianowym na niedeterministycznej maszynie Turinga,
2. dowodzimy NP-silności Q przez skonstruowanie transformacji wielomianowej znanego nam wcześniej problemu NP-zupełnego do Q.

Jako że na początku nie mamy do dyspozycji żadnego problemu, o którym wiemy, że jest NP-silny, to NP-silność pierwszego problemu NP-zupełnego Q musimy udowodnić z z definicji i pokazać, że każdy problem z klasy NP transformuje się wielomianowo do problemu Q.

Historycznie pierwszym znanym problemem NP-zupełnym jest problem spełnialności SAT. W celu postawienia tego problemu będziemy potrzebować kilku definicji.

Definicja 17.10. *Literałem nazywamy zmienną logiczną x_i lub jej zaprzeczenie $\overline{x}_i$. Klauzulą nazywamy (być może wielokrotną) alternatywę literałów. Alternatywa ta może być trywialna, tj. składać się tylko z jednego literału.*

Definicja 17.11. *Mówimy, że formuła φ jest w koniunkcyjnej postaci normalnej, jeśli φ jest (być może wielokrotną) koniunkcją literałów. Koniunkcja ta może być trywialna, tj. składać się tylko z jednego literału.*

Twierdzenie 17.12. *Każda formuła zdaniowa bez kwantyfikatorów jest równoważna formule w koniunkcyjnej postaci normalnej.*

Dowód. Na ćwiczeniach. Wskazówka: każda formuła zdaniowa jest równoważna formule w alternatywnej postaci normalnej (oczywiste) i przejść do zaprzeczenia. ■

Definicja 17.13. *Problemem spełnialności w skrócie SAT (od satisfiability) nazywamy następujący problem:*
Instancja: formuła zdaniowa φ w koniunkcyjnej postaci normalnej.

Pytanie: czy φ jest spełnialna, czyli czy istnieje takie przypisanie wartości logicznych 0 i 1 zmiennych w φ, dla których φ przyjmuje wartość 1, a więc formuła jest prawdziwa dla tego wartościowania.

Obserwacja 17.14. *Podstawianie wszystkich możliwych wartości dla wejścia długości n i m zmiennych ma złożoność $\Theta(2^m \cdot n) \leqslant O(2^n)$. Nie jest więc to szybki (choć poprawny) algorytm.* ■

Twierdzenie 17.15 (Cook 1971)**.** *Problem SAT jest NP-zupełny.*

Dowód. Zaczniemy od pokazania, że SAT należy do klasy NP. Niedeterministyczna maszyna Turinga może rozstrzygnąć problem spełnialności w następujący sposób:

1. Niedeterministyczna maszyna Turinga „zgaduje" rozwiązanie poprzez generowanie wszystkich możliwych wartościowań ciągu $x_1, \ldots, x_n$.
2. Niedeterministyczna maszyna Turinga sprawdza, czy dla danego wartościowania formuła jest prawdziwa. Może to zrobić w czasie liniowym.

Wynika z tego, że problem SAT należy do klasy NP.

W dalszej części dowodu pokażemy, że SAT jest NP-silny, czyli, że każdy problem Q z klasy NP transformuje się wielomianowo do problemu SAT, czyli $Q \leqslant_P SAT$.

Niech $Q \in NP$. Szukamy funkcji

$$f\colon D_Q \mapsto \{\varphi\colon \varphi\text{-formuła w koniunkcyjnej postaci normalnej}\},$$

taka że f jest obliczalna w czasie wielomianowym i dla dowolnego elementu $q \in D_Q$ zachodzi następujący warunek: Q jest spełnialna wtedy i tylko wtedy, gdy $q \in Y_Q$.

Skoro $Q \in NP$, to istnieje niedeterministyczna maszyna Turinga M, która rozpoznaje Q w czasie wielomianowym. Przyjmijmy oznaczenia:

- $q_1, \ldots, q_s$ - stany maszyny M,
- $x_1, \ldots, x_m$ - symbole maszyny M,
- $p(n)$ - złożoność czasowa M (pewien wielomian),
- δ - funkcja następnego ruchu.
- w - słowo wejściowe,
- $|w| = n$ - długość słowa wejściowego.

Definiujemy zbiór zmiennych zero-jedynkowych dla formuły $\varphi(w)$:

- $C(i,j,t)$ przyjmuje wartość 1, jeśli w klatce i w chwili t jest symbol x_i dla $1 \leqslant t \leqslant p(n)$, $1 \leqslant i \leqslant p(n)$, $1 \leqslant j \leqslant m$ (tych zmiennych jest $O(p^2(n))$),
- $S(k,t)$ przyjmuje wartość 1, jeśli w chwili t maszyna jest w stanie q_k (tych zmiennych jest $O(p(n))$,
- $H(i,t)$ przyjmuje wartość 1, jeśli w chwili t głowica czyta klatkę i (tych zmiennych jest rzędu $O(p^2(n))$.

Dla uproszczenia załóżmy, że każde obliczenie akceptujące w M składa się z dokładnie $p(n)$ konfiguracji. (W razie potrzeby powtarzamy stan akceptujący).

Ciąg konfiguracji $k_1, \ldots, k_{p(n)}$ jest akceptujący, jeśli spełnione są wszystkie z następujących warunków:

1. w każdej konfiguracji głowica czyta dokładnie jedną klatkę,
2. w każdej konfiguracji każda klatka zawiera dokładnie jeden symbol,
3. w każdej konfiguracji jest dokładnie jeden stan,
4. przy przejściu z konfiguracji k_i do k_{i+1} tylko klatka czytana może zmienić zawartość,
5. przy przejściu z konfiguracji k_i do k_{i+1} położenie głowicy i zawartość klatek zmienia się zgodnie z funkcją przejścia δ,
6. konfiguracja k_1 jest początkowa,
7. konfiguracja $k_{p(n)}$ jest akceptująca.

Formuła $\varphi(w)$ będzie więc koniunkcją zdań odpowiadających tym siedmiu częściom.

Niech

$$U(x_1, \ldots, x_n) = (x_1 \vee \cdots \vee x_m) \prod_{j \leqslant i} \prod_{i \neq j} (\overline{x}_i \vee \overline{x}_j).$$

Zauważmy, że U ma wartość 1 wtedy i tylko wtedy, gdy dokładnie jedna ze zmiennych x_i ma wartość 1. Skonstruujemy teraz siedem wyrażeń $A - G$:

1. Wyrażenie A stwierdza, że maszyna M czyta dokładnie jedną klatkę w każdej chwili. Niech A_i stwierdza, że że w chwili t czytana jest dokładnie jedna klatka. Wtedy $A = A_0 \cdot \cdots \cdot A_p(n)$, gdzie

$$A_t = U(H(1,t), \ldots, H(p(n)(t)).$$

   Wyrażenie A ma długość $O(p^3(n))$.
2. Wyrażenie B stwierdza, że każda klatka zawiera dokładnie jeden symbol. Niech $B_{i,t}$ stwierdza, ze i-ta komórka zawiera dokładnie jeden symbol w chwili t. Wtedy

$$B = \prod_{i,t} B_{i,t},$$

   gdzie $B_{i,t} = U(C(i,1,t), \ldots, C(i,m,t))$. Wyrażenie B ma długość $p(n)$.
3. Wyrażenie C stwierdza, że w każdej chwili t maszyna M jest tylko w jednym stanie:

$$C = \prod_{0 \leqslant i \leqslant p(n)} U(S(1,t), \ldots, S(s,t)).$$

   Wyrażenie C ma długość $p(n)$.

4. Wyrażenie D stwierdza, że przy przejściu z konfiguracji k_i do k_{i+1} tylko klatka czytana może zmienić zawartość.

$$D = \prod_{i,j,k} [(\neg C(i,j,t) \vee C(i,j,t+1)) \wedge (\neg C(i,j,t+1) \vee C(i,j,t)) \vee H(i,t)].$$

Długość D wynosi $O(p^2(n))$. Możemy przekształcić to wyrażenie do koniunkcyjnej postaci normalnej.

Podobnie musimy zdefiniować wyrażenia E, F oraz G (pozostawiamy to do samodzielnej pracy). Ostatecznie otrzymujemy $\varphi = A \wedge B \wedge C \wedge D \wedge E \wedge E \wedge F \wedge G$.

Zauważmy, że φ możemy wygenerować w czasie wielomianowym oraz że $\varphi(w)$ jest spełnialna wtedy i tylko wtedy, gdy M akceptuje słowo w. ■

18. Hierarchia wielomianowa

Przez większą część wykładu rozważamy jedynie klasy P oraz NP. Nie są to jedyne klasy złożoności czasowej. W szczególności wspomnieliśmy o klasie EXP, czyli klasie wszystkich problemów realizowalnych w czasie wykładniczym. Pomiędzy P, NP a EXP istnieją jednak inne klasy, co do których nie wiemy, czy są równe NP, EXP czy żadnej z nich. Najważniejszą z takich klas jest zapewne klasa coNP.

Definicja 18.1. *Niech C będzie klasą problemów decyzyjnych. Klasą coC nazywamy klasę tych problemów, których dopełnienie należy do klasy C.*

O problemach klasy NP mogliśmy myśleć jako o problemach, dla których łatwo (czyli w czasie wielomianowym) sprawdzić, czy dany przykład (ciąg generowany przez niedeterministyczną maszynę Turinga) daje nam rozwiązanie pozytywne.
Jak moglibyśmy w podobny sposób interpretować klasę $coNP$?

Otóż problemy klasy coNP, to te, dla których możemy łatwo sprawdzić, czy dany przykład jest kontrprzykładem.

Hipoteza 18.2. *Zachodzi $coNP = NP$.*

Zauważmy, że jeśli $P = NP$, to odpowiedź na to pytanie jest oczywista. Powyższa hipoteza nie jest jednak w oczywisty sposób rozstrzygnięta, jeśli wiemy, że $P \neq NP$.

Podamy teraz definicję ogólniejszej niż NP czy coNP klasy problemów, które tworzą tak zwaną hierarchię wielomianową.

Definicja 18.3. *Niech $i \geqslant 1$. Mówimy, że język L należy do klasy $\mathbf{\Sigma_i^P}$ (boldface $\mathbf{\Sigma_i^P}$), jeśli istnieje maszyna Turinga M działająca w czasie wielomianowym oraz wielomian q, taki że:*

$$x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q_x} \forall u_2 \in \{0,1\}^{q_x} \ldots Q_i u_i \in \{0,1\}^{q_x} M(x, u_1, \ldots, u_i) = 1,$$

gdzie q_x oznacza wartość $q(|x|)$, a Q_i jest kwantyfikatorem ogólnym lub szczególnym w zależności od parzystości liczby i.

Definicja 18.4. *Niech $i \geqslant 1$. Mówimy, że język L należy do klasy $\mathbf{\Pi_i^p}$ (boldface $\mathbf{\Pi_i^p}$), jeśli istnieje maszyna Turinga M działająca w czasie wielomianowym oraz wielomian q, taki że:*

$$x \in L \Leftrightarrow \forall u_1 \in \{0,1\}^{q_x} \exists u_2 \in \{0,1\}^{q_x} \ldots Q_i u_i \in \{0,1\}^{q_x} M(x, u_1, \ldots, u_i) = 1,$$

gdzie q_x oznacza wartość $q(|x|)$, a Q_i jest kwantyfikatorem szczególnym lub ogólnym w zależności od parzystości liczby i.

Definicja 18.5. *Hierarchią wielomianową nazywamy klasę* $\mathbf{PH} = \bigcup_{\mathbf{i}} \mathbf{\Sigma_i^p}$.

Zwróćmy uwagę, że $\mathbf{\Sigma_1^p}$ jest klasą NP, a $\mathbf{\Pi_1^p}$ jest klasą $coNP$. Pozwala to nam w inny sposób spojrzeć na klasę NP nie odwołując się do pojęcia niedeterministycznej maszyny Turinga. Możemy również inaczej popatrzeć na niedeterministyczną maszynę Turinga. Możemy zdefiniować $\mathbf{\Sigma_0^p} = \mathbf{\Pi_0^p} = \mathbf{P}$.

W ogólności dla każdej liczby naturalnej i mamy $\mathbf{\Pi_i^p} = \mathbf{co\Sigma_i^p}$. Zauważmy również, że $\mathbf{\Sigma_i^p} \subseteq \mathbf{co\Pi_{i+1}^p}$, więc moglibyśmy zdefiniować hierarchię wielomianową przez $\mathbf{PH} = \bigcup_{\mathbf{i}} \mathbf{\Pi_i^p}$. Nietrudno dowieść, że podobnie jak NP zawiera się w klasie EXP, hierarchia wielomianowa zawiera się w klasie EXP.

Pytanie: W definicjach składowych hierarchii wielomianowej kwantyfikatory ogólne i szczególne pojawiają się naprzemiennie. Co by się zmieniło, jakbyśmy umieścili dwa (lub więcej) kwantyfikatorów tego samego typu obok siebie?

Hipoteza 18.6. *Dla każdego $i \in \mathbb{N}$ zachodzi* $\mathbf{\Sigma_i^p} \subsetneq \mathbf{\Sigma_{i+1}^p}$.

Zwróćmy uwagę, że powyższa hipoteza jest uogólnieniem hipotezy $P \neq NP$. Jeśli dla pewnego i mamy $\mathbf{\Sigma_i^p} \subseteq \mathbf{\Sigma_{i+1}^p}$, to oznaczmy przez i_0 najmniejsze takie i. Wówczas mówimy, że hierarchia wielomianowa zapada się na poziomie i_0. Mówimy wtedy też o kolapsie hierarchii wielomianowej.

19. Klasyczne problemy NP-zupełne

W 1972 roku Karp udowodnił NP-zupełność 21 problemów. Wśród nich znajduje się sześć „najsławniejszych":
1. Problem 3-spełnialności (3-SAT),
2. Problem pokrycia wierzchołkowego (VC),
3. Problem kliki (CLIQUE),
4. Problem cyklu Hamiltona (HC),
5. Problem pokrycia 3-wymiarowego (3DM),
6. Problem podziału (Partition).

Omówimy powyższe problemy i pokażemy NP-zupełność części z nich.

19.1. Problem 3-spełnialności (3-SAT)

Definicja 19.1. *Mówimy, że formuła φ jest w k-koniunkcyjnej postaci normalnej, jeśli $\varphi = C_1 \wedge \ldots C_m$, gdzie C_j dla $j \leqslant m$ jest klauzulą dokładnie k literałów.*

Definicja 19.2. *Problemem 3-SAT nazywamy podproblem problemu SAT powstały przez zacieśnienie instancji do formuł w 3-koniunkcyjnej postaci normalnej.*

Twierdzenie 19.3. *Problem 3-SAT jest NP-zupełny.*

Dowód. Problem 3-SAT należy do klasy NP, ponieważ jest podproblemem problemu SAT, który należy do klasy NP.

Pokażemy, że 3-SAT jest NP-zupełny przez udowodnienie, że SAT $\leqslant_p$ 3-SAT.
- Wejście: φ dla SAT,
- Wyjście: φ' dla 3-SAT,

gdzie formuła φ jest spełnialna wtedy i tylko wtedy, gdy φ' jest spełnialna. Niech $\varphi = C_1 \wedge \cdots \wedge C_m$ i $\varphi' = C'_1 \wedge \cdots \wedge C'_m$.
1. Jeżeli C_i jest alternatywą trzech literałów, to $C'_i = C_i$.

2. Jeżeli C_i jest literałem x_1, to wprowadzamy nowe zmienne z_1 oraz z_2. Wówczas

$$C_i' = (x_1 \vee z_1 \vee z_2) \wedge (x_1 \vee \overline{z}_1 \vee z_2) \wedge (x_1 \vee z_1 \vee \overline{z}_2) \wedge (x_1 \vee \overline{z}_1 \vee \overline{z}_2).$$

3. Jeżeli $C_i = x_1 \vee x_2$ jest alternatywą dwóch literałów, to wprowadzamy nowe zmienną z_1. Wówczas

$$C_i' = (x_1 \vee x_2 \vee z_1) \wedge (x_1 \vee x_2 \vee \overline{z}_1).$$

4. Jeżeli $C_i = x_\vee \cdots \vee x_k$ jest alternatywą co najmniej czterech literałów, to wprowadzamy zmienne $z_1, \ldots, z_{k-2}$. Wówczas

$$C_i' = (x_1 \vee x_2 \vee z_1) \wedge (x_3 \vee \overline{z}_1 \vee z_2) \wedge (x_4 \vee \overline{z}_2 \vee z_3) \wedge \cdots \wedge (x_{k-1} \vee x_{k-2} \vee \overline{z}_{k-2}).$$

Zauważmy, że wyrażenie C_i' jest spełnialne wtedy i tylko wtedy, gdy C_i jest spełnialne. Długość wyjścia jest liniowa w stosunku do długości wejścia. ■

Problem 3-SAT jest problemem, który wygodnie transformować do innych problemów w celu dowodzenia ich NP-zupełności.

Uwaga: analogicznie zdefiniowany problem 2-SAT należy do klasy P. Zacieśnienie do 3-SATu jest więc w pewnym sensie optymalne.

19.2. Problem pokrycia wierzchołkowego

Definicja 19.4. *Problemem pokrycia wierzchołkowego (vertex cover, VC) nazywamy następujący problem:*
Instancja: dany jest graf $G = (V, E)$ oraz liczba naturalna dodatnia $k \leqslant |V|$.
Pytanie: czy dla danego grafu G i danej liczby naturalnej k graf G ma pokrycie wierzchołkowe o liczności nie większej niż k?

Twierdzenie 19.5 (Karp 1972)**.** *Problem pokrycia wierzchołkowego jest problemem NP-zupełnym.*

Dowód. Najpierw uzasadnimy, że problem VC należy do klasy NP. Niedeterministyczna maszyna Turinga generuje wszystkie możliwe podzbiory zbioru wierzchołków, a następnie sprawdza czy dany podzbiór jest pokryciem oraz czy jest mocy co najwyżej k. Może to zrobić w czasie wielomianowym.

Pokażemy, że 3-SAT redukuje się wielomianowo do VC. Niech φ będzie formuła w 3-koniunkcyjnej formule normalnej. Niech $x_1, \ldots, x_n$ będą zmiennymi w formule φ oraz niech $C_1, \ldots, C_m$ będą klauzulami w formule φ. Skonstruujemy graf $G = (V, E)$, taki że graf G ma pokrycie mocy co najwyżej k wtedy i tylko wtedy, gdy formuła φ jest spełnialna.

Graf G konstruujemy w następujący sposób: Dla każdej zmiennej x_i tworzymy dwa wierzchołki x_i oraz $\overline{x}_i$, a dla każdej klauzuli C_j tworzymy trzy wierzchołki C_j^1, C_j^2 oraz C_j^3. Krawędzią łączymy każdą zmienną x_i z jej zaprzeczeniem $\overline{x}_i$. Dla każdego j wierzchołki C_j^1, C_j^2 oraz C_j^3 łączymy ze sobą w klikę K_3. Krawędź między literałem x_i lub $\overline{x_i}$ a wierzchołkiem C_j^l tworzymy wtedy, gdy literał jest l-tą zmienną w klauzuli C_j.

Niech $k = n + 2m$. Pozostało pokazać, że graf G ma pokrycie wierzchołkowe wtedy i tylko wtedy, gdy graf G ma pokrycie wierzchołkowe co najwyżej k wierzchołkami. Najpierw uzasadnimy, że jeśli istnieje $k-$pokrycie w grafie G, to zawiera ono dokładnie k wierzchołków. Pokrycie w grafie G o mocy k musi zawierać dwa wierzchołki z każdego trójkąta oraz po jednym wierzchołku z każdej zmiennej i jej negacji. Dla literałów wybranych do pokrycia kładziemy wartość 1 w formule φ i w ten sposób w każdej alternatywie mamy co najmniej jedną 1.

Teraz pokażemy, że jeśli formuła φ jest spełnialna, to istnieje pokrycie wierzchołkowe grafu G mocy k. Dla wartościowania zmiennych x_1, $ldots$, x_n, dla których φ jest spełnione wybierzmy literały mające wartość 1. Pokrywają one krawędzie między literałami i co najmniej jedną krawędź między literałami a wierzchołkami C_m^l dla każdego m. Następnie do pokrycia dokładamy po dwa wierzchołki z klik odpowiadającym klauzulom w taki sposób, aby wybrane zostały wszystkie wierzchołki, których odpowiadające zmienne mają wartość 0. Ponieważ formuła φ jest spełniona, to w każdej klice odpowiadającej klauzuli znajdują się co najwyżej dwa takie wierzchołki. ■

19.3. Problem kliki

Definicja 19.6. *Problemem kliki (CLIQUE) nazywamy następujący problem:*
Instancja: graf $G = (V, E)$ oraz liczba naturalna dodatnia $j \leqslant |V|$.
Pytanie: czy dla podanego grafu G i liczby naturalnej j graf G zawiera klikę rzędu co najmniej j?

Twierdzenie 19.7 (Karp 1972)**.** *Problem kliki jest problemem NP-zupełnym.*

Dowód. Na ćwiczeniach. ■

19.4. Problem cyklu Hamiltona

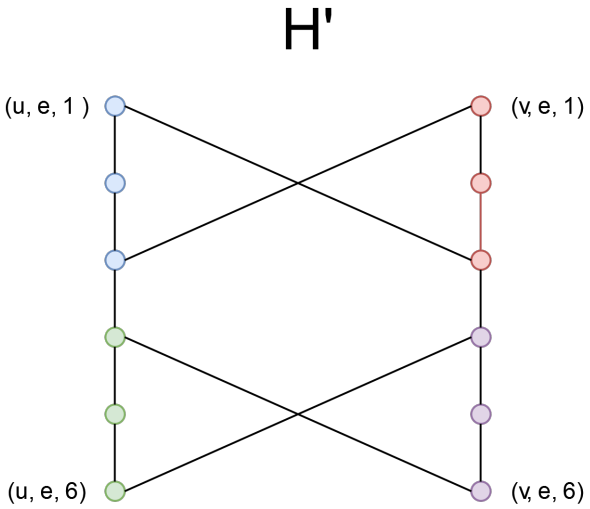
Definicja 19.8. *Problemem cyklu Hamiltona (Hamilton cycle, HC) nazywamy następujący problem:*
Instancja: graf $G = (V, E)$.
Pytanie: czy dla podanego grafu G istnieje cykl Hamiltona?

Twierdzenie 19.9 (Karp 1972)**.** *Problem cyklu Hamiltona jest NP-zupełny.*

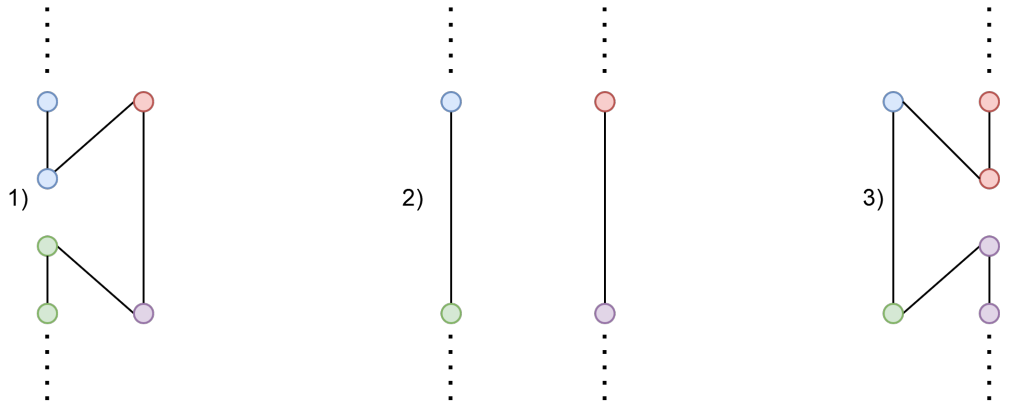
Dowód. Najpierw uzasadnimy, że problem cyklu Hamiltona należy do klasy NP. Niedeterministyczna maszyna Turinga, która rozpoznaje problem cyklu Hamiltona, generuje wszystkie ciągi wierzchołków i sprawdza, czy dany ciąg tworzy cykl Hamiltona. Może to zrobić w czasie wielomianowym.

Pokażemy teraz, że problem cyklu Hamiltona jest NP-zupełny poprzez transformację problemu pokrycia wierzchołkowego do problemu cyklu Hamiltona. Konstruujemy funkcję $f\colon D_{VC} \to D_{HC}$, taką że:
Wejście: graf $G = (V, E)$ oraz dodatnia liczba naturalna $k \leqslant |V|$,
Wyjście: graf $G' = (V, E')$, taki że graf G ma pokrycie mocy k wtedy i tylko wtedy, gdy graf G' ma cykl Hamiltona.

Niech $e = uv$ będzie krawędzią w grafie G. Funkcja g zastępuje tą krawędź składową H' o 12 wierzchołkach tworzących zbiór V'_e i 14 krawędziach tworzących zbiór E'_e.



Istnieją trzy możliwe przejścia cyklu Hamiltona przez taką składową:



Odpowiadają one sytuacjom, gdy:
1. wierzchołek u należy do pokrycia, a wierzchołek v nie,
2. oba wierzchołki u, v należą do pokrycia,
3. wierzchołek v należy do pokrycia, a wierzchołek u nie.

Pozostałe krawędzie w grafie G' będą łączyć te składowe ze sobą oraz łączyć składowe z dodatkowymi wierzchołkami, które będziemy nazywać selektorami. Dla dowolnego wierzchołka v w F oznaczmy krawędzie incydentne z v przez $e_1, \ldots, e_{d(v)}$. Wszystkie składowe odpowiadające tym krawędziom łączymy w G' za pomocą krawędzi $(v, e_i, 6)(v, e_{i+1}, 1)$.

Wprowadzimy teraz selektory $s_1, \ldots, s_k$. Wierzchołki postaci $(v, e_i, 6)$ oraz $(v, e_{i+1}, 1)$ łączymy z każdym z selektorów.

Wykażemy teraz, że graf G ma pokrycie wierzchołkowe mocy k wtedy i tylko wtedy, gdy graf G' ma cykl Hamiltona. Załóżmy najpierw, że istnieje cykl Hamiltona w grafie G. Rozważmy fragment tego cyklu zaczynający się i kończący wierzchołkiem ze zbioru selektorów i nie zawierający poza tym żadnego innego selektora. Ze względu na sposób łączenia składowych ten fragment cyklu zawiera ścieżkę przyporządkowaną pewnemu wierzchołkowi v_1 w grafie G. Po przejściu tej ścieżki cykl idzie do kolejnego selektora, a następnie znowu przechodzi przez ścieżkę odpowiadającą pewnemu wierzchołkowi v_2. Rozumując w ten sposób otrzymujemy zbiór wierzchołków $v_1, \ldots, v_k$, które tworzą pokrycie w grafie G, ponieważ przejście z selektora s_i do składowej odpowiadającej krawędzi e jest możliwe tylko wtedy, gdy $v_i \in e$.

Załóżmy teraz, że graf G ma pokrycie $\{v_1, \ldots, v_k\}$. Skonstruujemy cykl Hamiltona w G'. Startujemy w jednym z selektorów s_1 i następnie przechodzimy przez kolejne składowe po ścieżce dla wierzchołka v_1, idziemy do s_2, a następnie ścieżką dla v_2, itd.. Przez składowe przechodzimy zgodnie z opisanymi wcześniej zasadami. Zauważmy, że powstaje nam cykl, który przechodzi przez wszystkie wierzchołki grafu G', a więc cykl Hamiltona. ■

Definicja 19.10. *Problemem ścieżki Hamiltona (Hamilton path, HP) nazywamy następujący problem:*
Instancja: graf $G = (V, E)$.
Pytanie: czy dla podanego grafu G istnieje ścieżka Hamiltona?

Twierdzenie 19.11 (Karp 1972)**.** *Problem ścieżki Hamiltona jest NP-zupełny.*

Dowód. Problem ścieżki Hamiltona należy do klasy NP, ponieważ niedeterministyczna maszyna Turinga rozpoznająca ten problem generuje wszystkie możliwe ciągi wierzchołków, a następnie sprawdza, czy tworzą one ścieżkę Hamiltona. Może to zrobić w wielomianowym czasie.

Dowód NP-zupełności możemy przeprowadzić na dwa sposoby
1. zmodyfikować transformację $VC \leqslant_P HC$,
2. skonstruować transformację $HC \leqslant_P HP$.

Szczegóły pozostawiam do samodzielnej pracy. ■

19.5. Problem pokrycia 3-wymiarowego

Definicja 19.12. *Problemem pokrycia 3-wymiarowego (3-dimensional matching, 3DM) nazywamy następujący problem:*
Instancja: parami rozłączne zbiory W, X, Y, takie że $|X| = |W| = q$ oraz podzbiór $M \subseteq W \times X \times Y$.
Pytanie: czy istnieje zbiór $M' \subseteq M$, taki że $|M'| = q$ i żadne dwa różne elementy ze zbioru M' nie mają wspólnych współrzędnych?

Twierdzenie 19.13 (Karp 1972)**.** *Problem pokrycia 3-wymiarowego jest problemem NP-zupełnym.* ■

19.6. Problem podziału

Definicja 19.14. *Problemem podziału (PARTITION) nazywamy następujący problem:*
Instancja: zbiór C pewnych liczb naturalnych dodatnich.
Pytanie: czy istnieje zbiór $C' \subseteq C$, taki że suma elementów należących do C' jest równa sumie elementów należących do $C \setminus C'$?

Twierdzenie 19.15 (Karp 1972)**.** *Problem podziału jest NP-zupełny.* ■

Definicja 19.16. *Problemem sumy podzbiorów (Subset-sum) nazywamy następujący problem:*
Instancja: zbiór C pewnych liczb naturalnych dodatnich oraz liczba naturalna dodatnia B.
Pytanie: czy istnieje zbiór $C' \subseteq C$, taki że suma elementów należących do C' jest równa B?

Twierdzenie 19.17 (Karp 1972)**.** *Problem sumy podzbiorów jest NP-zupełny.* ■

Problem podziału jest szczególnym przypadkiem problemu sumy podzbiorów. Jak uzasadnić to twierdzenie?

20. Analiza złożoności problemów

Spotykając się z nowym problemem obliczalnym podstawowym pytaniem, jakie można sobie zadać, jest: czy jest on rozwiązywalny w czasie wielomianowym? W przypadku, gdy potrafimy odpowiedzieć twierdząco na to pytanie, pozostaje znalezienie możliwie jak najefektywniejszego algorytmu. W innym przypadku pozostaje pytanie, czy problem jest NP-zupełny?

Definicja 20.1. *Niech Q będzie problemem decyzyjnym. Podproblemem Q' problemu Q nazywamy problem Q', taki że $D_{Q'} \subseteq D_Q$ oraz $Y_{Q'} = Y_Q \cap D_{Q'}$. Mówimy wtedy o zacieśnieniu problemu Q do zbioru $D_{Q'}$.*

Jeśli mamy problem NP-zupełny, to może się okazać, że dany podproblem jest łatwy (np. problem cyklu Hamiltona dla grafów o maksymalnym stopniu 2). Podobnie problem 2-kolorowania wierzchołkowego grafu jest wielomianowy, ponieważ polega na sprawdzeniu, czy dany graf jest dwudzielny. Pokażemy później, że problem 3-kolorowania jest NP-zupełny. W przypadku problemu cyklu Hamiltona czy 3-kolorowania problem w ogólności był NP-zupełny, ale pewne jego podproblemy były wielomianowe. Często nie jest jedna tak łatwo sprawdzić, czy dane zacieśnienie powoduje, że problem staje się łatwy. Ponadto, zakładając $P \neq NP$, mogą istnieć problemy pośrednie. Stawiamy więc granicę pomiędzy problemami łatwymi i trudnymi. Analizując dany podproblem chcemy się dowiedzieć, po której stronie granicy jesteśmy. Omówimy to dokładniej na przykładzie problemu 3-kolorowania wierzchołkowego. Potrzebna nam będzie jeszcze definicja problemu NAESAT.

Definicja 20.2. *Problemem NAESAT (Not all equal SAT) nazywamy następujący problem:*
Instancja: formuła zdaniowa φ w 3-koniunkcyjnej postaci normalnej.
Pytanie: czy φ jest spełnialna w taki sposób, aby w każdej klauzuli przynajmniej jeden literał miał wartość 0.

Twierdzenie 20.3. *Problem NAESAT jest NP-zupełny.* ■

Definicja 20.4. *Problemem k-kolorowania wierzchołkowego (k-COLORING) nazywamy następujący problem:*
Instancja: graf $G = (V, E)$, liczba naturalna dodatnia $k \leqslant |V|$.
Pytanie: czy dany graf jest k-kolorowalny.

Twierdzenie 20.5 (Karp 1972)**.** *Problem 3-kolorowania jest NP-zupełny.*

Dowód. Najpierw uzasadnimy, że problem 3-kolorowania należy do klasy NP. Niedeterministyczna maszyna Turinga rozpoznająca problem 3-kolorowania generuje wszystkie (niekoniecznie właściwe) kolorowania grafu trzema kolorami. Następnie sprawdza, czy dane kolorowanie jest właściwe. Może to zrobić w czasie wielomianowym.

Pokażemy teraz, że problem NAESAT transformuje się wielomianowo do problemu 3-kolorowania.

Niech $C_1, \ldots, C_m$ będą klauzulami formuły φ w 3-koniunkcyjnej postaci normalnej i niech $x_1, \ldots, x_n$ będą zmiennymi w φ. Skonstruujemy graf G w następujący sposób:
1. klauzulę C_i reprezentuje klika K_3 o wierzchołkach C_i^1, C_i^2, C_i^3,
2. dodajemy nowy wierzchołek a,
3. każdą zmienną x_i przedstawiamy za pomocą kliki K_3 o wierzchołkach $a, x_i, \overline{x}_i$,
4. każde C_i^j łączymy z literałem, który reprezentuje.

Pokażemy, że graf G można pokolorować właściwie kolorami ze zbioru $\{0, 1, 2\}$ wtedy i tylko wtedy, formuła φ jest spełnialna w taki sposób, że każda klauzula ma fałszywy literał. Załóżmy, że graf G można pokolorować właściwie kolorami ze zbioru $\{0, 1, 2\}$. Bez straty dla ogólności możemy przyjąć, że wierzchołek a ma kolor 2. Wtedy pozostałe wierzchołki w klice $\{a, x_i, \overline{x}_i\}$ mają kolor 1 lub 0. Jeśli klika odpowiadająca danej klauzuli jest połączona krawędzią z trzema wierzchołkami koloru 1, to żaden wierzchołek kliki nie może dostać koloru 1, więc nie może być pokolorowany właściwie. Analogicznie dla koloru 0. Wynika z tego, że każda z klik odpowiadających klauzulą jest połączona z przynajmniej jednym wierzchołkiem koloru 1 oraz z przynajmniej jednym wierzchołkiem koloru 0. Wartościowanie literałów odpowiadające kolorowi 1 daje nam spełnialność formuły φ w sensie NAESAT.

Załóżmy teraz, że formuła φ jest spełniona w sensie NAESAT. Wówczas kolorujemy wierzchołek a kolorem 2, a literały ich wartościami logicznymi. Dla każdego C_i wybieramy dwa literały o różnych wartościach. Kolorujemy wierzchołek sąsiedni z wierzchołkiem koloru 1 za pomocą 0, a wierzchołek sąsiedni z wierzchołkiem koloru 0 za pomocą 1. Pozostałe wierzchołki w klauzulach kolorujemy kolorem 2. Wystarczy zauważyć, że skonstruowane kolorowanie grafu G jest właściwe. ■

Być może wprowadzając pewne ograniczenia na stopień wierzchołków w grafie otrzymamy problem wielomianowy. Przytoczymy znane z teorii grafów twierdzenie Brooksa.

Twierdzenie 20.6 (Brooks 1941)**.** *Niech G będzie grafem spójnym o stopniu maksymalnym nie większym niż 3. Wtedy G jest 3-kolorowalny wtedy i tylko wtedy, gdy nie zawiera kliki K_4.*

Dowód. Twierdzenie było udowodnione na teorii grafów. ■

Postawmy więc pytanie: czy spójny graf o stopniu maksymalnym co najwyżej 3 jest 3-kolorowalny?

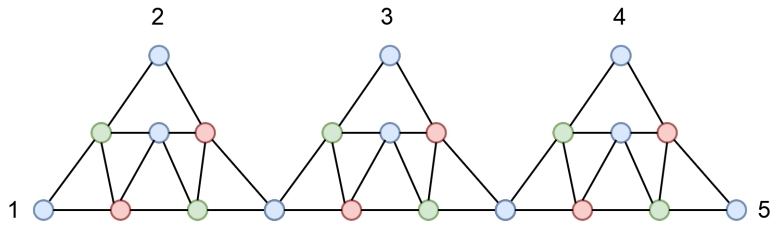
Odpowiedź: tak, ten problem należy do klasy P, ponieważ wystarczy sprawdzić, czy zawiera graf pełny na czterech wierzchołkach, co możemy zrobić w czasie wielomianowym.

Pytanie: czy taką samą odpowiedź otrzymamy, gdy zmienimy ograniczenie na stopień maksymalny z 3 na 4?

Twierdzenie 20.7. *Problem 3-kolorowania grafu o stopniu maksymalnym nie większym niż 4 jest problem NP-zupełnym.*

Dowód. Problem należy do klasy NP, bo ogólny problem 3-kolorowania należy do klasy NP.

Pokażemy teraz, że ogólny problem 3-kolorowania transformuje się wielomianowo do problemu 3-kolorowania, w którym wprowadzimy ograniczenie na stopień maksymalny nie większy niż 4. Niech $G = (V, E)$ będzie wejściem dla ogólnego problemu. Skonstruujemy odpowiadający mu graf $G' = (V, E)$ o stopniu maksymalnym nie większym niż 4 taki, że G jest 3-kolorowalny wtedy i tylko wtedy, gdy G' jest 3-kolorowalny. Konstruujemy pewne grafy H_k dla $k \geqslant 3$, które będziemy nazywać składowymi. Graf H_5:



Każda z tych składowych ma wierzchołki stopnia dwa, nazwijmy je wierzchołkami zewnętrznymi. Ponadto żaden z wierzchołków grafu H_k nie zawiera wierzchołka o stopniu maksymalnym większym niż 4. Zauważmy, że graf H_k nie jest 2-kolorowalny, ale jest 2-kolorowalny.

Graf G' konstruujemy w sposób następujący: niech $v_1, \ldots, v_n$ oznacza wierzchołki stopnia co najmniej 5 w grafie G. Konstruujemy rekurencyjnie ciąg grafów $G = G_0, \ldots, G_r = G'$ w taki sposób, że graf G_i powstaje z grafu G_{i-1} zastępując wierzchołek v_i stopnia $d(v_i)$ składową $H_{d(v_i)}$, a wierzchołki połączone w G_{i-1} z wierzchołkiem v_i łączymy z odpowiednimi wierzchołkami wewnętrznymi, których jest dokładnie $d(v_i)$. Pozostaje udowodnić, że graf G jest 3-kolorowalny wtedy i tylko wtedy, gdy graf G' jest 3-kolorowalny.

Niech dane będzie 3-kolorowanie grafu G. W grafie G' wierzchołki zewnętrzne danej składowej H_k są tego samego koloru, co odpowiadający jej wierzchołek v w grafie G. Pozostałą część grafu składowej możemy dokolorować.

Niech dane będzie 3-kolorowanie grafu G. Zauważmy, że oznacza to, że wszystkie wierzchołki zewnętrzne w danej składowej H_k są tego samego koloru. Ten kolor nadajemy wierzchołkowi w grafie G, który odpowiada składowej H_k. Natomiast kolory pozostałych wierzchołków pozostawiamy takie same. Zauważmy, że otrzymane kolorowanie grafu G jest właściwe. ■

Twierdzenie 20.8. *Problem 3-kolorowania spójnego grafu planarnego jest problemem NP-zupełnym.* ■

Twierdzenie 20.9. *Problem 4-kolorowania spójnego grafu planarnego jest problemem wielomianowym.*

Dowód. Na mocy twierdzenia o czterech barwach każdy graf planarny jest 4-kolorowalny. Wynika z tego, że odpowiedź na to pytanie jest zawsze twierdząca, a więc możemy jej udzielić w czasie wielomianowym. ■

21. Problemy liczbowe

Rozważmy problem podziału: niech $s_1, \ldots, s_n$ będą liczbami naturalnymi dodatnimi. Sprawdzamy, czy istnieje zbiór $I \subseteq \{1, \ldots, n\}$, taki że $\sum_{i \in I} s_i = \sum_{i \notin I} s_i = B$.

Rozważmy następujący algorytm. Najpierw zauważmy, że jeśli suma $\sum_{i=1}^{n} s_i$ jest nieparzysta, to odpowiedź jest w oczywisty sposób negatywna. W przeciwnym przypadku zdefiniujmy liczbę $t_{i,j}$ dla $i \in \{1, \ldots, n\}$ oraz $j \in \{0, \ldots, B\}$, której przypiszemy wartość 1 lub 0 w zależności od tego, czy istnieje podzbiór zbioru $\{s_1, \ldots, s_i\}$, w którym suma elementów jest równa j.

Przykład: Niech $\{s_1, s_2, s_3\} = \{5, 2, 4\}$.

$t_{i,j}$	0	1	2	3	4	5	6	7	8	9	...	B
1	1					1						
2	1		1			1		1				
3	1		1		1	1	1	1		1		
...												
n												1

Zauważmy, że w pierwszym wierszu stawiamy 1 tylko dla $j = 0$ i $j = s_1$. Każdy kolejny wiersz uzupełniamy korzystając z poprzedniego. Problem posiada odpowiedź twierdzącą, gdy w którymś wierszu kolumny B wystąpi 1. Zauważmy, że złożoność tego algorytmu wynosi $O(nB)$. Mogłoby się więc wydawać, że mamy do czynienia z algorytmem wielomianowym, co implikowałoby, że P=NP. Jednak każda wartość s_i jest zapisana na wejściu ciągiem o długości $\log s_i$. Stąd długość zapisu B wynosi $\log B$, czyli wejście dla problemu podziału ma długość $O(n \log B)$ i nB nie jest

ograniczone funkcją wielomianową takiej zmiennej. Omówiony algorytm jest tak zwanym algorytmem pseudowielomianowym.

NP-zupełność problemu podziału związana jest więc z faktem, że liczby na wejściu mogą być dowolnie duże. Gdyby założyć pewne ograniczenie na rozmiar wejścia, to algorytm stałby się algorytmem wielomianowym.

Niech I będzie instancją pewnego problemu. Oznaczmy przez $Lenght(I)$ rozmiar instancji, czyli długość zapisu oraz przez $Max(I)$ wartość największej liczby występującej w I.

Wówczas algorytm A jest algorytmem wielomianowym, jeśli jego złożoność jest wielomianową funkcją od $Lenght(I)$.

Definicja 21.1. *Mówimy, że algorytm A jest algorytmem pseudowielomianowym, jeśli jego złożoność jest wielomianową funkcją od $(Lenght(I), Max(I))$.*

Zauważmy, że dla pewnych problemów $Max(I)$ jest ograniczona przez wielomianową funkcję od $Lenght(I)$, np. jedyną liczbą występującą w wejściu dla problemu kliki jest liczba J ograniczająca rozmiar kliki, która nie może być większa niż liczba wierzchołków rozważanego grafu. Przy wejściu dla problemu spełnialności w ogóle nie występują żadne liczby, pomijając ewentualnie liczbę zmiennych i literałów. co jednak możemy zignorować, ponieważ tego typu liczby są zawsze ograniczone przez rozmiar instancji.

Definicja 21.2. *Mówimy, że problem jest nieliczbowy, jeśli istnieje wielomian p, taki, że $Max(I) \leqslant p(Lenght(I))$, gdzie I oznacza instancję problemu. W przeciwnym przypadku mówimy, że problem jest liczbowy.*

Dla problemów nieliczbowych algorytmy pseudowielomianowe oraz wielomianowe są tożsame. Sposób sześciu podstawowych problem NP-zupełnych jedynie problem podziału jest problemem liczbowym. Innym przykładem problemu nieliczbowego jest problem sumy podzbiorów.

Obserwacja 21.3. *Jeśli $P \neq NP$, to każdy NP-zupełny problem nieliczbowy nie jest rozwiązywalny za pomocą algorytmu pseudowielomianowego.* ■

Definicja 21.4. *Mówimy, że problem Q jest silnie NP-zupełny, jeśli istnieje wielomian p, taki że problem Q' powstały z zacieśnienia instancji I do takich instancji, że $Max(I) \leqslant p(Lenght(I))$ jest problemem NP-zupełnym.*

Problemy silnie NP-zupełne to problemy, które przy wielomianowym ograniczeniu liczb występujących w ich opisie pozostają NP-zupełne.

Obserwacja 21.5. *Jeśli NP-zupełny problem jest nieliczbowy, to jest też silnie NP-zupełny.* ■

Obserwacja 21.6. *Jeśli $P \neq NP$, to każdy problem silnie NP-zupełny nie jest rozwiązywalny za pomocą algorytmu pseudowielomianowego.* ■

Definicja 21.7. *Problemem komiwojażera (TSP, travelling salesman problem) nazywamy następujący problem:*
Instancja: liczby naturalne dodatnie (odległości) $d(i,j)$ dla $i,j \in \{1,\ldots,n\}$ oraz liczba naturalna dodatnia B.
Pytanie: Czy istnieje ciąg $(\pi(1),\ldots,\pi(n))$, taki że:

$$\sum_{i=1}^{n-1}(d(\pi(i),\pi(i+1)) + d(\pi(n),\pi(1)) \leqslant B?$$

Definicja 21.8. *Niech Q oraz Q' będą problemami decyzyjnymi. Mówimy, że problem Q transformuje się pseudowielomianowo do problemu Q' (ozn. $Q \leqslant_{ps} Q'$), jeżeli istnieje funkcja $f\colon D_Q \to D_{Q'}$, taka że:*

1. *dla dowolnej instancji I problemu Q instancja I należy do Y_Q wtedy i tylko wtedy, gdy $f(I)$ należy do $Y_{Q'}$,*
2. *funkcja f jest realizowalna w czasie pseudowielomianowym, tzn. w czasie $p(Max(I), Length(I))$,*
3. *istnieje wielomian q_1, taki że dla dowolnego $I \in D_Q$ mamy*

$$a_1(Length'(f(I)) \geqslant Length(I),$$

   czyli funkcja f nie kurczy ponadwielomianowo instancji,
4. *istnieje wielomian dwóch zmiennych q_2, taki że dla dowolnego $I \in D_Q$ mamy*

$$Max'(f(I)) \leqslant q_2(Max(I), Length(I)),$$

   czyli funkcja f nie powiększa ponadwielomianowo $Max(I)$.

Przykład:
Wejście: n, $Length(n) = log(n)$.
Wyjście: $\log n$, $Length(\log n) = \log\log n$.
W tym przypadku warunek 3. nie jest spełniony.

Twierdzenie 21.9. *Niech problem Q będzie silnie NP-zupełny i niech problem Q' będzie problemem klasy NP. Jeśli $Q \leqslant_{ps} Q'$, to Q' jest silnie NP-zupełny.*

Definicja 21.10. *Problemem 3-podziału nazywamy następujący problem:*
Instancja: skończony zbiór A mocy $3m$, liczba naturalna dodatnia B, dla każdego $a \in A$ liczby naturalne dodatnie $s(a)$, takie że $\frac{B}{4} < s(a) < \frac{B}{2}$ oraz suma elementów $s(a)$ jest równa mB. Pytanie: czy istnieje podział zbioru A na rozłączne podzbiory $S_1, \ldots, S_m$, takie że dla dowolnego $j \in \{1,\ldots,m\}$ mamy $\sum\{s(a) : a \in S_j\} = B$?

Zauważmy, że ze względu na założenia dotyczące rozmiarów każdy ze zbiorów S_j musi być trójelementowy.

Twierdzenie 21.11. *Problem 3-podziału jest problemem silnie NP-zupełnym.* ■

Definicja 21.12. *Problemem zanurzenia grafów nazywamy następujący problem:*
Instancja: grafy G_1, G_2.
Pytanie: Czy G_2 jest podgrafem G_1?

Twierdzenie 21.13. *Problem zanurzenia grafów jest silnie NP-zupełny.*

Dowód. Dowód będzie wynikał z twierdzenia, które udowodnimy później. ■

Twierdzenie 21.14 (Edmonds, Matula 1976)**.** *Niech Q' będzie podproblemem problemu zanurzenia grafów powstałym z zacieśnienia instancji G_1 i G_2 do drzew. Wówczas problem Q' jest wielomianowy.* ■

Zastanówmy się nad problemami między problemem zanurzenia a jego powyższym podproblemem.

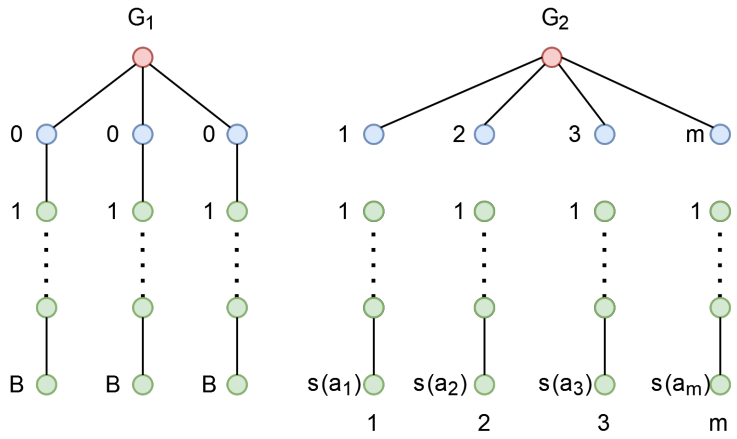
Twierdzenie 21.15. *Niech dany będzie podproblem Q problemu zanurzenia grafów powstały przez zacieśnienie instancji G_2 do drzew. Wówczas problem Q jest NP-zupełny.*

Dowód. Do zastanowienia się. Na ćwiczeniach. (Zauważmy, że ten problem zawiera jako podproblem problem ścieżki Hamiltona.) ■

Zastanówmy się, co będzie, jeśli G_1 jest drzewem. Wówczas jeśli G_2 ma cykl (co można sprawdzić w wielomianowym czasie), to odpowiedź na pytanie jest negatywna. Przyjmijmy więc, że G_2 jest lasem. Otrzymamy więc następujący problem:
Instancja: drzewo G_1 oraz las G_2.
Pytanie: czy graf G_2 jest podgrafem grafu G_1?

Twierdzenie 21.16. *Powyższy problem jest silnie NP-zupełny.*

Dowód. Pokażemy, że problem 3-podziału transformuje się pseudowielomianowo do powyższego problemu.
Wejście: instancja 3-podziału. Wyjście: drzewo G_1 oraz las G_2, takie że G_2 jest podgrafem G_1 wtedy i tylko wtedy, gdy istnieje odpowiedni 3-podział.



Drzewo G_1 składa się z m ścieżek po $B+1$ wierzchołków każda połączonych w dodatkowym wierzchołku. Las G_2 składa się z gwiazdy $K_{1,m}$ oraz $3m$ ścieżek po $s(a_i)$ wierzchołków każda.

Zauważmy, że przy zanurzeniu lasu G_2 w drzewo G_1 środek gwiazdy musi przejść w wierzchołek grafu G_1 o maksymalnym stopniu. Łatwo zauważyć, że 3-podział istnieje wtedy i tylko wtedy, gdy istnieje zanurzenie grafu G_2 w graf G_1. Oszacujemy teraz czas transformacji.

$$Length(I) = \sum_{i=1}^{3m} \{\lceil \log(s(a_i)) \rceil\},$$

$$Length(f(I)) \geqslant \sum_{i=1}^{3m} \{\lceil (s(a_i)) \rceil\},$$

co oznacza że transformacja nie działa w czasie wielomianowym. Działa natomiast w czasie pseudowielomianowym, bo zależy wielomianowo od m i B, a więc spełniony jest warunek 2.; łączna liczba wierzchołków w grafach G_1 i G_2 wynosi $2(mB+1)$, czyli zachodzi warunek 3.; warunek 4. również jest spełniony, ponieważ liczby w konstruowanej instancji są ograniczone, bo właściwie nie ma tam liczb. ■

Wniosek 21.17. *Problem zanurzenia grafów jest silnie NP-zupełny.* ■

22. Problemy optymalizacyjne

Dotychczas mieliśmy do czynienia z problemami decyzyjnymi, czyli takimi, na które odpowiedź jest postaci „tak" lub „nie". Szerszą klasą problemów są problemy optymalizacyjne, czyli takie, w których spośród możliwych rozwiązań musimy znaleźć takie, które jest optymalne w jakimś sensie. Zwykle dotyczy to maksymalizacji lub minimalizacji pewnej funkcji. Żeby zastosować dotychczasową teorię do problemów optymalizacyjnych musimy przerobić je na problemy decyzyjne. Zwykle polega to na narzuceniu ograniczenia na optymalizowaną wartość. Nie szukamy już wtedy wartości optymalnej, a odpowiadamy na pytanie, czy istnieje co najmniej tak dobre rozwiązanie, jak podaliśmy w instancji problemu decyzyjnego. Każdy problem optymalizacyjny możemy przerobić na odpowiadający mu problem decyzyjny.

Definicja 22.1. *Maszyną Turinga z wyrocznią nazywamy maszynę Turinga z dodatkowym podprogramem, który dla ustalonego problemu rozwiązuje go w czasie stałym.*

Definicja 22.2. *Mówimy, że problem Q_1 transformuje się wielomianowo w sensie Turinga do problemu Q_2 (ozn. $Q_1 \leqslant_T Q_2$), jeśli istnieje maszyna Turinga z wyrocznią dla Q_2, która rozwiązuje problem Q_1 w czasie wielomianowym.*

Jaki jest związek między transformacją w sensie Turinga, a transformacją wielomianową?
Ćwiczenie: sformułować alternatywną definicję klasy NP oraz klasy problemów NP-zupełnych.

Definicja 22.3. *Mówimy, że problem Q jest problemem NP-trudnym, jeśli istnieje NP-zupełny problem Q', taki że $Q' \leqslant_T Q$.*

Definicja 22.4. *Mówimy, że problem Q jest problemem NP-łatwym, jeśli zachodzi $Q \leqslant_T Q'$*

Problemy NP-łatwe są podobną klasą problemów, co problemy NP z tą różnicą, że nie muszą być problemami decyzyjnymi.

Transformacja wielomianowa w sensie Turinga służy do porównywania trudności obliczeniowych problemów optymalizacyjnych.

Przykład. Problem pokrycia wierzchołkowego. Wersja decyzyjna: szukamy pokrycia mocy co najwyżej k. Wersja optymalizacyjna: szukamy pokrycia minimalnej mocy.

Do domu: porównaj problem pokrycia wierzchołkowego w obu wersjach.

Jeśli problem optymalizacyjny jest NP-trudny np. dlatego, że jest decyzyjna wersja jest NP-zupełna, to sensownym z praktycznego punktu widzenia podejściem dla rozwiązywania problemu dla dużych instancji jest szukanie rozwiązań przybliżonych i stosowanie algorytmów aproksymacyjnych. Interesują nas algorytmy szybkie, to znaczy działające w czasie wielomianowym, o jak najlepszym stopniu przybliżenia rozwiązania znajdowanego do rozwiązywania optymalnego. Podstawową metodą mierzenia jakości rozwiązania przybliżonego jest tak zwany współczynnik efektywności przybliżenia.

Definicja 22.5. *Niech n będzie kosztem instancji problemu optymalizacyjnego. Niech C będzie kosztem obliczonym przez algorytm aproksymacyjny i niech C^* będzie rozwiązaniem optymalnym. Mówimy, że współczynnik efektywności przybliżenia (stała aproksymacji) jest ograniczony przez funkcję $r(n)$, jeśli zachodzi* $\max\{\frac{C}{C^*}, \frac{C^*}{C}\} \leqslant r(n)$.

Zauważmy, że $\max\{\frac{C}{C^*}, \frac{C^*}{C}\} = \frac{C}{C^*}$ dla problemów minimalizacyjnych oraz że $\max\{\frac{C}{C^*}, \frac{C^*}{C}\} = \frac{C}{C^*}$ dla problemów maksymalizacyjnych. Zauważmy, że algorytm optymalny osiąga zawsze $r(n) = 1$. Chcemy, aby $r(n)$ było niewielkie. Zwykle chcemy, aby $r(n)$ było stałą niezależną od n i bliską 1. Dla niewielu problemów NP-trudnych udało się znaleźć dobre w tym sensie strategie aproksymacyjne.

Przykład. Problem komiwojażera. W problemie komiwojażera kosztem jest długość drogi d. Załóżmy dodatkowo, że $d(u, v) = d(v, u)$ oraz że zachodzi nierówność trójkąta: $d(u, v) \leqslant d(u, w) + d(w, v)$. Odnotujmy, że przy tych założeniach problem komiwojażera nadal jest problemem NP-zupełnym, bo problem cyklu Hamiltona nadal jest podproblemem tej wersji problemu komiwojażera.

Algorytm aproksymacyjny dla problemu komiwojażera:
Wejście: $G = (V, E)$ graf niezorientowany z wagami (kosztami) krawędzi $d(u, v)$.
Wyjście: permutacja wierzchołków (droga komiwojażera).

1. r - dowolny wierzchołek (korzeń).
2. T - zbiór krawędzi tworzących minimalne drzewo rozpinające o korzeniu r (np. algorytm Prima-Jarníka, to jest to nieskomplikowany, wielomianowy algorytm zachłanny, ale nie będziemy go tutaj podawać).
3. L - lista wierzchołków w kolejności przeglądu preorder drzewa T, co oznacza, że każda krawędź pojawia się dwukrotnie.

4. P - lista L po usunięciu wierzchołków powtarzających się, czyli na ścieżce L wykonaj skróty omijające wierzchołki już odwiedzone.
5. Zwróć P.

Złożoność powyższego algorytmu jest wielomianowa i wynosi $\Theta(n^2)$.

Twierdzenie 22.6. *Dla podanego powyżej algorytmu stała aproksymacji jest nie większa niż* 2.

Dowód. Oznaczmy przez $d(A)$ sumę kosztów krawędzi należących do zbioru $A \leqslant E$, a przez H optymalną drogę komiwojażera. Zauważmy, że zachodzi $d(T) \leqslant d(H)$, bo przykładowe drzewo rozpinające powstaje z usunięcia jednej krawędzi z drogi komiwojażera. Zatem minimalne drzewo rozpinające nie może być dłuższe. Zauważmy, że $d(L) = 2d(T)$, co z nierówności trójkąta daje nam $d(P) \leqslant d(L)$. Ostatecznie otrzymujemy $c(P) \leqslant 2c(H)$. ■

Bibliografia

[1] W. Ackermann. Zum hilbertschen aufbau der reellen zahlen. *Mathematische Annalen*, 99:118–133, 1928.

[2] W. Ackermann. Untersuchungen über das eliminationsproblem der mathematischen logik. *Mathematische Annalen*, 110:390–413, 1937.

[3] J. Ax. On the undecidability of power series fields. *Proceedings of the American Mathematical Society*, 16(4):846–846, 1965.

[4] P. Bernays. A system of axiomatic set theory, part ii. *Journal of Symbolic Logic*, 6(1):1–17, 1941.

[5] P. Bernays. A system of axiomatic set theory, part iii. *Journal of Symbolic Logic*, 7(2):65–89, 1942.

[6] F. Bernstein. Untersuchungen aus der mengenlehre. *Mathematische Annalen*, 61:117–155, 1898.

[7] C. Burali-Forti. Una questione sui numeri transfiniti. *Rendiconti del Circolo Matematico di Palermo*, 11:154–164, 1897.

[8] G. Cantor. Über eine elementare frage der mannigfaltigkeitslehre. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 1:75–78, 1891.

[9] G. Cantor. Beiträge zur begründung der transfiniten mengenlehre. ii. *Mathematische Annalen*, 49:207–246, 1897.

[10] A. Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41, 1936.

[11] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936.

[12] P. J. Cohen. The independence of the continuum hypothesis. *Proceedings of the National Academy of Sciences*, 50:1143–1148, 1963.

[13] P. J. Cohen. The independence of the continuum hypothesis ii. *Proceedings of the National Academy of Sciences*, 51:105–110, 1964.

[14] R. Dedekind. *Was sind und was sollen die Zahlen?* Friedrich Vieweg und Sohn, Braunschweig, 1888.

[15] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41(1):50–58, 1976.

[16] S. Feferman et al., editors. *Kurt Gödel: Collected Works. Volume III: Unpublished Essays and Lectures*. Oxford University Press, 1995.

[17] K. Gödel. Die vollständigkeit der axiome des logischen funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37:349–360, 1930.

[18] K. Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[19] K. Gödel. On undecidable propositions of formal mathematical systems. 1934. Lecture notes, Institute for Advanced Study; published in Collected Works, Vol. I, Oxford University Press, 1986.

[20] F. Hausdorff. *Grundzüge der Mengenlehre*. Veit & Comp., Leipzig, 1914.

[21] G. Hessenberg. Grundbegriffe der mengenlehre. *Abhandlungen der Fries'schen Schule*, 1:479–706, 1906.

[22] D. Hilbert and W. Ackermann. *Grundzüge der theoretischen Logik*. Springer, Berlin, 1928.

[23] L. Kirby and J. Paris. Accessible independence results for peano arithmetic. *Bulletin of the London Mathematical Society*, 14(4):285–293, 1982.

[24] S. C. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112:727–742, 1936.

[25] D. Kőnig. *Theorie der endlichen und unendlichen Graphen*. Akademische Verlagsgesellschaft, Leipzig, 1936.

[26] K. Kuratowski. Sur l'opération $\overline{A}$ de l'analysis situs. *Fundamenta Mathematicae*, 3:182–199, 1922.

[27] A. Lindenbaum and A. Tarski. Untersuchungen über den aussagenkalkül. *Comptes Rendus des Séances de la Société des Sciences et des Lettres de Varsovie*, 23:30–50, 1930.

[28] A. Lindenbaum and A. Tarski. Untersuchungen über den aussagenkalkül. *Fundamenta Mathematicae*, 25:305–326, 1935.

[29] L. Löwenheim. Über möglichkeiten im relativkalkül. *Mathematische Annalen*, 76:447–470, 1915.

[30] A. I. Mal'cev. Untersuchungen aus dem gebiete der mathematischen logik. *Matematicheskii Sbornik*, 1:323–336, 1936.

[31] A. Mostowski. On the independence of axioms of arithmetic. *Fundamenta Mathematicae*, 39:201–208, 1952.

[32] R. Péter. Über eine nicht primitiv-rekursive funktion. *Mathematische Annalen*, 111:42–60, 1935.

[33] R. Rado. Universal graphs and universal functions. *Acta Arithmetica*, 9:331–340, 1964.

[34] J. B. Rosser. Extensions of some theorems of gödel and church. *Journal of Symbolic Logic*, 1(3):87–91, 1936.

[35] C. Ryll-Nardzewski. On the consistency of certain fragments of arithmetic. *Fundamenta Mathematicae*, 39:125–131, 1952.

[36] J. Silver. On the singular cardinals problem. *Proceedings of the International Congress of Mathematicians (Vancouver, 1974)*, 1:265–268, 1975.

[37] T. Skolem. Logisch-kombinatorische untersuchungen über die erfüllbarkeit oder beweisbarkeit mathematischer sätze. *Videnskapsselskapets Skrifter I. Matematisk-naturvidenskapelig Klasse*, (4):1–36, 1920.

[38] T. Skolem. Einige bemerkungen zur axiomatischen begründung der mengenlehre. *Mathematische Zeitschrift*, 13:118–125, 1922.

[39] A. Tarski. Sur quelques théorèmes qui équivalent à l'axiome du choix. *Fundamenta Mathematicae*, 5:147–154, 1924.

[40] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936.

[41] J. von Neumann. Zur einführung der transfiniten zahlen. *Acta Scientiarum Mathematicarum (Szeged)*, 1:199–208, 1923.

[42] E. Zermelo. Beweis, dass jede menge wohlgeordnet werden kann. *Mathematische Annalen*, 59:514–516, 1904.

[43] E. Zermelo. Untersuchungen über die grundlagen der mengenlehre i. *Mathematische Annalen*, 65:261–281, 1908.

[44] M. Zorn. A remark on method in transfinite algebra. *Bulletin of the American Mathematical Society*, 41(10):667–670, 1935.