ZDZISŁAW PAWLAK

AUTOMATYCZNE DOWODZENIE TWIERDZEŃ

WARSZAWA
PAŃSTWOWE ZAKŁADY WYDAWNICTW SZKOLNYCH

Biblioteczka Matematyczna PZWS
i czasopisma „Matematyka"

Kolegium redakcyjne:
B. Iwaszkiewicz, Z. Krygowska, S. Olczak, J. Podemski,
W. Stojda, S. Straszewicz.

Okładkę projektował
Alojzy Balcerzak

WSTĘP

Współczesne maszyny matematyczne są stosowane głównie do wykonywania obliczeń. Dzięki olbrzymiej szybkości liczenia — dochodzącej w najszybszych maszynach do miliona operacji arytmetycznych w ciągu sekundy — można je stosować do rozwiązywania zadań, wymagających wykonania miliardów działań arytmetycznych. Ręczne wykonanie takich obliczeń byłoby oczywiście praktycznie niemożliwe, trwałoby bowiem setki, jeżeli nie tysiące lat. Nic więc dziwnego, że inżynierowie, konstruktorzy, fizycy, którzy w swej pracy napotykają na trudne problemy obliczeniowe, widzą we współczesnych maszynach matematycznych narzędzie otwierające całkiem nowe perspektywy.

Okazało się, że maszyny matematyczne, budowane pierwotnie do celów wyłącznie rachunkowych, mogą być również stosowane w wielu innych dziedzinach. Rozległe możliwości zastosowań tych maszyn były nawet pewnym zaskoczeniem dla ich konstruktorów.

Do jednych z najciekawszych zastosowań maszyn tego rodzaju należą niewątpliwie próby przeprowadzenia z ich pomocą rozumowań matematycznych. Jakkolwiek poważniejsze badania w tej dziedzinie trwają dopiero zaledwie kilka lat [1]), to sam problem mechanizacji rozumowania jest już bardzo stary i początki jego sięgają starożytnej Grecji. Zanim przystąpimy do przedstawienia roli maszyn matematycznych w mechanizacji rozumowania, warto nieco uwagi poświęcić logice, nauce o poprawnych formach rozumowania.

Rozumowanie polega na wyciąganiu wniosków z przesłanek. Logika zajmuje się takimi postaciami rozumowania, w których postępowanie nie zależy od treści występujących w nich pojęć. Rozumowanie takie jest nazywane **formalnym.** Ponieważ przesłanki jak i wnioski

[1]) Inicjatorem ich jest matematyk amerykański pochodzenia chińskiego, Hao Wang.

są zapisywane w jakimś języku, rozumowanie formalne sprowadza się do czysto mechanicznego przekształcania jednych napisów (przesłanek) w inne napisy (wnioski). Wnioskowanie formalne przypomina więc nieco rachunek, polegający na przekształcaniu napisów w określony sposób. Idea zastąpienia myślenia rachunkiem jest niezwykle atrakcyjna, nic więc dziwnego, że prześladuje ona ludzkość z różnym nasileniem od niepamiętnych czasów.

Pierwszy system logiki został stworzony przez Arystotelesa w czwartym wieku przed naszą erą. System ten znany jest pod nazwą **sylogistyki.** Jest zadziwiające, że przez przeszło 2000 lat po Arystotelesie nie odnotowano w rozwoju logiki żadnego istotnego postępu. Dopiero w połowie XIX wieku matematyk angielski George Boole, stwarzając matematyczne podstawy tzw. rachunku zdań, zapoczątkował dalszy intensywny rozwój logiki. Rachunek Boole'a pozwalał na czysto mechaniczne wykonywanie wielu rozumowań matematycznych. Jednakże w trakcie dalszych badań okazało się, że prawa myślenia odkryte przez Boole'a są w wielu przypadkach niewystarczające. Dopiero stworzenie tzw. **rachunku kwantyfikatorów** pozwoliło na pełną formalizację wszystkich znanych do tej pory rozumowań matematycznych. Za pomocą praw algebry Boole'a oraz rachunku kwantyfikatorów można już, wychodząc z niewielkiej liczby aksjomatów, wyprowadzić wszystkie znane aktualnie wnioski matematyczne.

Nasunęło to myśl, że działalność matematyków właściwie może być zastąpiona przez odpowiednie maszyny, które stosując na wszelkie możliwe sposoby prawa algebry Boole'a oraz rachunku kwantyfikatorów do wyjściowych aksjomatów, będą produkowały wszelkie możliwe twierdzenia matematyczne. Tak jednak nie jest. Wykazał to przed około 30 laty wybitny matematyk austriacki Kurt Gödel. Wszystkich prawdziwych twierdzeń matematycznych nie da się wyprowadzić w ten sposób. W matematyce są twierdzenia, których nie da się wywieść z żadnych aksjomatów według dowolnych reguł wnioskowania. W pracy matematyka występują więc istotnie twórcze elementy. Na czym polega istota twórczości matematycznej, podobnie zresztą jak i każdej innej twórczości, do tej pory nie wiadomo.

Zdawałoby się, że Gödel raz na zawsze zdewaluował rolę maszyn matematycznych w matematyce i skazał matematyków na stosowanie najprymitywniejszych narzędzi: papieru i ołówka bądź kredy

i tablicy. Praca matematyków jest bowiem działalnością twórczą i zastosowanie tu nawet najwymyślniejszych maszyn nie może mieć istotnego znaczenia. Trudno oprzeć się jednakże uwadze, że zarówno pojęcie twórczości jak i pojęcie maszyny nie są zbyt jasne i ścisłe — dlatego nie wszyscy zgadzają się na taką interpretację wyników Gödla i mają cichą nadzieję, że może z tych więzów da się jakoś wyplątać. Są to jednak, jak do tej pory, marzenia fantastyczne. Nie będziemy się więc nimi zajmować.

Z twierdzenia Gödla nie wynika, że cała praca matematyków ma charakter twórczy; są oczywiście w niej fragmenty, które z powodzeniem można zmechanizować. Powstaje więc pytanie jakie są proporcje elementów twórczych i mechanicznych w działalności matematycznej. Istnieje wśród matematyków przekonanie, że główną rolę odgrywają tu elementy pracy intelektualnej, natomiast Hao Wang postawił hipotezę, że rzecz przedstawia się wręcz odwrotnie: wprawdzie w pracy matematyka występują elementy twórcze, to jednak znakomita większość tej pracy polega na zupełnie mechanicznych czynnościach, które z powodzeniem mogą być wykonane za pomocą odpowiedniej maszyny. (Hipoteza ta nie jest oczywiście w żadnej sprzeczności z rezultatami Gödla).

Dla uzasadnienia tej hipotezy, po raz pierwszy chyba w historii matematyki odwołano się do doświadczenia. Maszyna matematyczna została użyta nie jako pojęcie abstrakcyjne, a jako konkretne narzędzie. W dotychczasowych bowiem badaniach dotyczących podstaw matematyki studia nad możliwościami maszyn miały charakter czysto werbalny. Wang natomiast zadał sobie trud praktycznego sprawdzenia czego rzeczywiście można dowieść za pomocą współczesnych maszyn elektronicznych. Jako obiekt swoich doświadczeń wziął on znane dzieło B. Russel'a i A. N. Whitehead'a *Principia mathematica.* W książce tej jest kilkaset dowodów twierdzeń matematycznych. Znalezienie tych dowodów zajęło Autorom na pewno kilka miesięcy, jeżeli nie kilka lat pracy. Tymczasem Wang dowiódł większą część występujących tam twierdzeń, za pomocą szybkiej maszyny matematycznej, w czasie ... około 9 MINUT [1]).

[1]) Podany czas dotyczy tylko czasu działania maszyny. Należy do tego doliczyć czas przygotowania programów pracy maszyny. Do spraw tych wrócimy jeszcze w dalszym ciągu.

Fakt ten zmusza do zastanowienia. Jeżeli bowiem wieloletnią pracę można wykonać w czasie kilku godzin, to konsekwencje tego faktu mogą mieć duże znaczenie dla dalszego rozwoju matematyki.

Nie wszyscy zgadzają się z tym poglądem, uważając, że sprawa takiej czy innej szybkości wykonania dowodu nie ma tutaj istotnego znaczenia. Tak chyba jednak nie jest. Wiadomo, że idąc pieszo można dojść do dowolnie odległej miejscowości. Wprowadzenie komunikacji kolejowej czy lotniczej skraca znacznie czas podróży, jednakże konsekwencje lotnictwa nie sprowadzają się tylko do szybkości podróży, wpłynęły one pośrednio na cały szereg innych spraw. Nie wykluczone, że maszyny matematyczne mogą pośrednio odegrać dużą rolę w matematyce.

Ilustracją powyższej myśli może być np. sprawa zrozumienia dowodu matematycznego. Dowód matematyczny polega na rozbiciu całego rozumowania matematycznego na takie elementarne kroki, z których żaden nie może być kwestionowany przez osobę badającą poprawność dowodu — o ile oczywiście jest on poprawny i jest przez sprawdzającego rozumiany. Zrozumienie dowodu polega więc na zrozumieniu poszczególnych jego elementów i zdaniu sobie sprawy z ich wzajemnego powiązania.

Wyobraźmy sobie, że matematyk chce sprawdzić czy jakieś wyrażenie jest twierdzeniem badanej przez niego teorii. Dowód tego twierdzenia wymaga jednak milionów bądź miliardów operacji, tak że wykonanie ich przez człowieka jest praktycznie niemożliwe. A więc o twierdzeniu tym nie można orzec czy jest ono prawdziwe czy nie. Zastosowanie w tym przypadku maszyny pozwoli przeprowadzić dowód; powstaje jednak pytanie, czy dowód ten może być przez człowieka rozumiany? W dotychczasowym sensie — chyba nie. Jeżeli nie, to za pomocą maszyn matematycznych można dowodzić twierdzeń, których nie można zrozumieć, ewentualnie pojęcie zrozumienia wymaga innej interpretacji.

Zastosowanie maszyn do dowodzenia twierdzeń ma jeszcze jeden aspekt. Matematyk dowodząc jakiegoś twierdzenia, na ogół nie przeprowadza dowodu z całą drobiazgowością, robi tylko jego szkic tak, aby były widoczne zasadnicze elementy dowodu, dbając jednakże o to, aby w razie potrzeby dowolny fragment można było uściślić. Szczegółowe przeprowadzenie dowodu wymagałoby bowiem zazwyczaj wykonania olbrzymiej ilości operacji, przez co dowód straciłby

na przejrzystości; z drugiej strony, taka drobiazgowa praca jest nieciekawa, nie odpowiada wysokim kwalifikacjom matematyków.

Zastosowanie maszyny do dowodzenia zmusza natomiast do bardzo szczegółowej analizy dowodu i spojrzenia na niego z zupełnie innej strony, od strony organizacji pracy: w jaki sposób najsprawniej wykonać tak dużą liczbę operacji. Prowadzi to w konsekwencji do nowych zagadnień, które w przypadku pracy ręcznej nie występowały, jak np. dobór odpowiednich reguł wnioskowania, czy sposobu zapisywania dowodów. Zastosowanie w tym przypadku maszyn może być więc interesujące nie tylko dlatego, że można za ich pomocą dowieść szybciej pewnej liczby twierdzeń, lecz dlatego, że na proces dowodzenia można spojrzeć z nieco innego punktu widzenia. Przypomina to sytuację w dziedzinie innych odkryć, np. podróży Magellana. Wartość odkrycia może leżeć zupełnie gdzie indziej, aniżeli się spodziewano.

Tak więc po 30 latach niełaski maszyny matematyczne znalazły się ponownie w kręgu zainteresowań logików i matematyków i to nie tylko jako obiekt filozoficznych spekulacji, ale również jako ewentualne narzędzie pracy. Dziś trudno jeszcze przewidzieć czy i w jakim stopniu spełnią one pokładane w nich nadzieje. Nie ulega natomiast wątpliwości, że badania nad mechanicznym rozumowaniem znajdą zastosowania pozamatematyczne, np. w technice, pozwalając na bardzo szybkie podejmowanie decyzji w sytuacjach, w których konieczne jest uprzednie przeanalizowanie olbrzymiej liczby konsekwencji przyjętych założeń. Bez użycia maszyn elektronicznych rozwiązanie takiego zadania, jakkolwiek teoretycznie możliwe, w praktyce jest zupełnie niewykonalne.

W Polsce prace nad automatycznym dowodzeniem twierdzeń są prowadzone w Instytucie Matematycznym PAN pod kierunkiem dra Andrzeja Ehrenfeuchta. Wydaje się, że kierunek badań reprezentowany przez Ehrenfeuchta może być bardziej przydatny do celów maszynowych aniżeli inne dotychczasowe badania. Jednakże prace te nie zostały jeszcze zakończone i dlatego nie będziemy ich tutaj omawiali.

Celem niniejszej książeczki jest pokazanie zasady dowodzenia twierdzeń za pomocą maszyn matematycznych, a także pokazanie

zasady działania maszyn cyfrowych, elementów programowania tych maszyn a także częściowo naszkicowanie niektórych zagadnień i trudności związanych z konstrukcją maszyn matematycznych. Zagadnienia te wydają się o tyle ważne, że dla matematyków maszyny stworzyły nową interesującą dziedzinę pracy. We wszystkich krajach, gdzie są stosowane bądź konstruowane maszyny matematyczne, istnieje bardzo duże zapotrzebowanie na matematyków tej specjalności. Wydaje się więc, że wszelkie informacje na ten temat dla tych, którzy chcieliby w przyszłości studiować matematykę, mogą się okazać przydatne. Książeczka ta jest więc głównie przeznaczona dla uczniów szkół średnich, którzy interesują się maszynami matematycznymi, oraz dla nauczycieli. Mam nadzieję, że przeczytają ją również wszyscy, których interesują nowe perspektywy zastosowań maszyn matematycznych. Dla tych ostatnich, w spisie literatury dotyczącej automatycznego dowodzenia twierdzeń, zamieściłem również niektóre dostępne w Polsce prace źródłowe dotyczące poruszanego tematu.

Materiał zawarty w tej książce został podzielony na dwie części. W części pierwszej podano pojęcie twierdzenia, dowodu, zasadę działania maszyny cyfrowej oraz przedstawiono ogólną strukturę programu dowodzącego twierdzenia.

Część druga zawiera omówienie zastosowania maszyn matematycznych do dowodzenia twierdzeń w prostej teorii matematycznej — rachunku zdań. Jako punkt wyjścia przyjęto koncepcje Hao Wanga. Próby zastosowania maszyn do dowodzenia twierdzeń w rachunku zdań i w rachunku kwantyfikatorów były robione również przed Wangiem, jednakże nie miały one większego znaczenia. Dopiero Wang postawił to zagadnienie w należyty, jasny i ścisły sposób. Ogólna koncepcja Wanga wydaje się być dobrym punktem wyjścia również do badania innych teorii matematycznych pod kątem zastosowania w nich maszyn matematycznych.

Do rozumienia książki wystarczają wiadomości matematyczne w zakresie szkoły średniej, jednakże Czytelnikowi zainteresowanemu głębiej tym tematem radzimy gruntowniejsze zapoznanie się z rachunkiem zdań z książki A. Grzegorczyka, *Logika popularna* oraz z teorią dowodu z książki W. Pogorzelskiego i J. Słupeckiego, *O dowodzie matematycznym.*

Część druga książki, jakkolwiek nie wymaga do zrozumienia szerszych wiadomości niż część pierwsza, jest od części pierwszej nieco

trudniejsza, wymaga bowiem drobiazgowego prześledzenia dość długich programów. Czytelnik nie posiadający biegłości w manipulowaniu dużą ilością symboli może tu napotkać trudności.

Jednakże do rozumienia zasadniczych idei przedstawionych w tej książce szczegółowe czytanie części drugiej nie jest potrzebne. Długie, nieczytelne i nieprzejrzyste programy są w gruncie rzeczy nudne; mogą być one interesujące jedynie dla kogoś, kto zajmuje się zawodowo programowaniem. Dlaczego więc je tutaj umieściłem? Są one bowiem bardzo charakterystyczne dla wszelkich zastosowań maszyn matematycznych. Maszyna matematyczna nie zrobi najprostszych nawet wnioskowań samodzielnie; postępowanie maszyny musi być nawet w najbanalniejszych sytuacjach szczegółowo i drobiazgowo opisane w postaci programów. Aby cokolwiek rozwiązać za pomocą maszyny matematycznej, musimy najpierw sami znać metodę rozwiązania zagadnienia, które nas interesuje, a dopiero wtedy można mówić o zastąpieniu pracy ludzkiej przez maszynę. Nie jest tak, jak to sobie niektórzy wyobrażają, że maszyna rozwiąże jakiekolwiek zagadnienie samodzielnie. Przytoczone programy są tego najlepszym dowodem. Nie potrzeba ich szczegółowo analizować, aby zrozumieć, że maszyna matematyczna nie jest mniej czy więcej udaną kopią mózgu ludzkiego.

Na zakończenie chciałbym wyrazić podziękowanie drowi Andrzejowi W. Mostowskiemu za udzielenie mi wielu cennych rad i wskazówek przy redagowaniu niniejszej książki.

CZĘŚĆ I

Rozdział I

POJĘCIE DOWODU MATEMATYCZNEGO

§ 1. Wnioskowanie

Jak powiedzieliśmy we wstępie, wnioskowanie jest to wyprowadzanie, na podstawie odpowiednich reguł, wniosków z przesłanek. Oczywiście interesują nas tylko takie formy wnioskowania, które od prawdziwych przesłanek prowadzą do prawdziwych wniosków. Inaczej wnioskowanie byłoby pozbawione jakiejkolwiek użyteczności. Jednakże zagadnieniem prawdziwości nie będziemy się tutaj interesowali. Nie jest to bowiem konieczne do zrozumienia poruszanego przez nas tematu.

Ponieważ zdania występujące we wnioskowaniu są zapisywane w jakimś języku, więc jeżeli pominiemy treść tych zdań, to wnioskowanie możemy uważać za przekształcanie symboli według zadanych reguł. Takie podejście bardzo zawęża sens pojęcia wnioskowania — jednakże do niektórych celów jest ono wygodne. W szczególności założenie takie jest zawsze przyjmowane w maszynach matematycznych, realizujących dowolny proces przekształcania symboli, albowiem **znaczenie symboli przekształcanych przez maszynę nie ma żadnego wpływu na jej konstrukcję ani zasadę działania.** Dlatego też w dalszym ciągu zamiast konkretnych rozumowań matematycznych będziemy rozpatrywali przykłady przekształcania symboli według zadanych reguł, nie wiążąc z rozpatrywanymi symbolami ani regułami żadnej treści matematycznej. Będziemy mieli tylko na uwadze, aby charakter tego przekształcania był podobny do rzeczywistych rozumowań matematycznych.

Podkreślamy raz jeszcze, że zastosowanie maszyny do przeprowadzania wnioskowania możliwe jest tylko wtedy, gdy wnioskowanie można sprowadzić do czysto mechanicznych manipulacji na symbolach maszyna bowiem nie rozumie treści wykonywanych czynności, nie może więc posługiwać się regułami wnioskowania odnoszącymi się do sensu przesłanek. Widać stąd wyraźnie, że maszyny mogą tu odgrywać jedynie rolę pomocniczych narzędzi.

Jeżeli wnioskowanie nie zależy od treści rozpatrywanych zdań, a tylko od ich postaci, to wnioskowanie takie nazywamy **formalnym.** Ponieważ w niniejszej książeczce będziemy się zajmowali tylko wnioskowaniem formalnym, dla uzupełnienia proponujemy Czytelnikowi zapoznanie się z treścią matematyczną wnioskowań formalnych z książek: A. Grzegorczyk, *Logika popularna*, PWN, Biblioteka Problemów, 1960 oraz W. A. Pogorzelski i J. Słupecki, *O dowodzie matematycznym*, PZWS, 1962.

§ 2. Teorie dedukcyjne

Teorie matematyczne są teoriami dedukcyjnymi. Każda teoria dedukcyjna jest zespołem zdań — zwanych **tezami** albo **twierdzeniami** tej teorii — takim, że każde zdanie jest aksjomatem lub wnioskiem z aksjomatów. Jeżeli zdanie Z jest twierdzeniem teorii **T**, to zapiszemy $\vdash_{\mathbf{T}} Z$ i przeczytamy: „Zdanie Z jest twierdzeniem teorii **T**".

Ponieważ nie będziemy się interesować treścią teorii matematycznych, możemy więc uważać je za zespół napisów takich, że każdy z nich jest napisem wyjściowym, bądź też napisem otrzymanym z napisów wyjściowych za pomocą odpowiednich reguł przekształcania napisów. Napisy te będziemy nazywali **wyrażeniami** lub **zdaniami** teorii.

Dla określenia teorii dedukcyjnej konieczne jest, oprócz podania aksjomatów oraz reguł wnioskowania, podanie symboli dopuszczalnych w tej teorii oraz reguł pozwalających tworzyć z symboli wyjściowych wyrażenia należące do tej teorii. Zbiór symboli wyjściowych [1]) oraz reguł tworzenia wyrażeń nazywany jest **językiem teorii.**

Tak więc teorię dedukcyjną charakteryzujemy przez podanie

[1]) Zbiór symboli wyjściowych języka nazywany jest **alfabetem** lub **słownikiem** języka.

1. Zbioru symboli wyjściowych,
2. reguł tworzenia wyrażeń,
3. zbioru aksjomatów,
4. reguł wnioskowania.

Na teorię dedukcyjną możemy również patrzeć jako na metodę produkowania wniosków z przesłanek. Dlatego też czasem zamiast terminu wnioskowanie, będziemy używali terminu produkcja.

Jeżeli zdanie Z jest otrzymane z przesłanek $P_1, P_2, \ldots, P_n$ za pomocą jednej dopuszczalnej reguły wnioskowania, to zdanie Z nazwiemy **wnioskiem bezpośrednim** z przesłanek $P_1, P_2, \ldots, P_n$.

Jeżeli zdanie Z jest otrzymane z przesłanek $P_1, P_2, \ldots, P_k$ za pomocą więcej niż jednokrotnego stosowania dowolnych reguł wnioskowania dopuszczalnych w teorii oraz zdanie Z nie jest przesłanką dla żadnego wniosku w danym wnioskowaniu, to zdanie Z nazwiemy **wnioskiem końcowym.**

Rozpatrzymy teraz kilka przykładów prostych teorii dedukcyjnych, ilustrujących powyżej wprowadzone pojęcia. Z przykładami takich teorii Czytelnik spotkał się na lekcjach matematyki. Najbardziej znanym przykładem teorii dedukcyjnej jest geometria. Nie będziemy tutaj jednakże rozpatrywali żadnej rzeczywistej teorii matematycznej, zajęłoby to bowiem zbyt wiele miejsca. Z drugiej zaś strony strukturę teorii dedukcyjnych łatwiej jest prześledzić na specjalnie w tym celu skonstruowanych przykładach, nie posiadających określonego sensu matematycznego. Taki sposób postępowania jest często praktykowany, nie tylko w celach popularyzacyjnych, gdyż pozwala na łatwe uchwycenie niektórych istotnych własności teorii dedukcyjnych.

Przykład 1. Rozpatrzymy najpierw przykład bardzo prostej teorii dedukcyjnej, którą oznaczymy przez $\mathbf{T}_1$. Celem określenia teorii musimy najpierw podać jej język, a następnie aksjomaty i reguły wnioskowania obowiązujące w tej teorii. Przyjmiemy, że alfabet języka teorii $\mathbf{T}_1$ składa się tylko z dwu symboli, a mianowicie liter a i b [1]).

[1]) Zamiast liter a i b można by przyjąć dowolne inne dwa symbole, np. + i 0, bądź 0 i 1, itp. Rodzaj przyjętych symboli jest tu nie istotny. Ważne jest tylko, aby były to dwa różne symbole.

Język teorii $\mathbf{T}_1$ posiada jedną regułę tworzenia wyrażeń z symboli alfabetu.

$$\mathbf{Q}_1\colon\ \alpha, \beta \rightarrow \alpha\beta$$

Reguła ta nosi nazwę **konkatenacji** (stykania) i oznacza, że jeżeli dwa wyrażenia α i β należące do języka teorii $\mathbf{T}_1$ napiszemy obok siebie, to tak otrzymane wyrażenie należy również do języka teorii $\mathbf{T}_1$. Np. wyrażenia *a* oraz *a* należą do języka teorii $\mathbf{T}_1$, a więc wyrażenie *aa* należy również do tego języka. Również wyrażenie *aab* jest wyrażeniem języka teorii $\mathbf{T}_1$, gdyż można je otrzymać za pomocą reguły $\mathbf{Q}_1$ z symboli alfabetu języka. Mówiąc prościej, wyrażeniami języka teorii $\mathbf{T}_1$ są dowolne skończone ciągi liter *a* i *b*.

A więc każdy ciąg składający się z liter *a* i *b* jest wyrażeniem języka teorii $\mathbf{T}_1$. W rzeczywistych teoriach matematycznych język jest oczywiście znacznie bardziej skomplikowany aniżeli w teorii $\mathbf{T}_1$. Alfabet zawiera znacznie większą liczbę symboli a także wyrażenia są tworzone w bardziej skomplikowany sposób aniżeli w naszym przykładzie. Tym nie mniej ogólny schemat postępowania jest dla każdego języka jednakowy: z ustalonego zbioru symboli, alfabetu, tworzymy ciągi symboli według odpowiednich reguł. Dla ilustracji tego faktu podany przez nas przykład jest wystarczający.

Pominęliśmy tutaj całkowicie fakt, że język teorii matematycznych jest tworzony celem wyrażania pewnych myśli, a więc wyrażenia takiego języka posiadają określone znaczenie. Język teorii $\mathbf{T}_1$ natomiast żadnego znaczenia nie posiada. Ciągi symboli alfabetu tego języka nie wyrażają żadnych myśli, nie posiadają żadnej interpretacji matematycznej. Z punktu widzenia zastosowania maszyn matematycznych do dowodzenia twierdzeń jest rzeczą całkowicie obojętną czy wyrażenia przekształcane przez maszynę posiadają jakiekolwiek znaczenie, czy też nie. Maszyna postępuje tutaj w sposób całkowicie mechaniczny, przekształcając zadane wyrażenia w określony sposób, bez rozumienia ich sensu. Dlatego też dla zrozumienia idei mechanicznego dowodzenia twierdzeń nie musimy koniecznie rozpatrywać żadnej konkretnej teorii matematycznej, a wystarczą nam do tego celu bardzo proste „teorie" zbudowane podobnie do teorii spotykanych w matematyce.

Jako jedyny aksjomat teorii $\mathbf{T}_1$ przyjmiemy symbol *a* [1]).

$$A_1\colon\ a$$

Dla zupełnego określenia teorii $\mathbf{T}_1$ należy podać jeszcze reguły wnioskowania obowiązujące w tej teorii. Są one następujące:

$$\mathbf{R}_1\colon\ A \rightarrow Ab$$
$$\mathbf{R}_2\colon\ A \rightarrow aAa$$

Reguła $\mathbf{R}_1$ mówi, że jeżeli do dowolnego twierdzenia *A* należącego do teorii $\mathbf{T}_1$ dopiszemy na końcu literę *b*, to w wyniku otrzymamy twierdzenie należące do teorii $\mathbf{T}_1$.

Reguła $\mathbf{R}_2$ mówi, że do dowolnego twierdzenia *A* teorii $\mathbf{T}_1$ można dopisać z obu stron literę *a* i tak otrzymane wyrażenie jest również twierdzeniem teorii $\mathbf{T}_1$.

Reguły $\mathbf{R}_1$ i $\mathbf{R}_2$ pozwalają więc na produkowanie twierdzeń teorii $\mathbf{T}_1$, wychodząc z jedynego aksjomatu $\mathbf{A}_1$ tej teorii.

Produkowanie twierdzeń w teorii $\mathbf{T}_1$ przebiega następująco:

Stosując do aksjomatu A_1 regułę $\mathbf{R}_1$, otrzymamy twierdzenie *ab*.

Stosując do twierdzenia *ab* jeszcze raz regułę $\mathbf{R}_1$, otrzymamy twierdzenie *abb*.

Z twierdzenia *abb* na podstawie reguły $\mathbf{R}_2$ otrzymamy twierdzenie *aabba*.

W dalszym ciągu proces produkowania twierdzeń będziemy zapisywali w postaci tabelki, jak to podano niżej.

1.	*a*	A_1
2.	*ab*	$\mathbf{R}_1$, 1
3.	*abb*	$\mathbf{R}_1$, 2
4.	*aabba*	$\mathbf{R}_2$, 3

Każdy krok wnioskowania jest zapisany w jednym wierszu tabelki. Wszystkie wiersze są ponumerowane z lewej strony liczbami 1, 2, itd. Z prawej strony zapisane są użyte reguły oraz numer wiersza, do którego je zastosowano. Np. w wierszu 3 podano $\mathbf{R}_1$, 2, co oznacza,

[1]) Dla prostoty przyjęto, że aksjomat jest identyczny z symbolem alfabetu. Ogólnie biorąc, aksjomaty są wyrażeniami składającymi się z większej niż jeden liczby symboli wyjściowych.

że twierdzenie *abb* otrzymano przez zastosowanie reguły $\mathbf{R}_1$ do przesłanki zapisanej w wierszu 2.

Jak widzimy każde twierdzenie teorii $\mathbf{T}_1$ jest pewnym zdaniem języka tej teorii ale nie na odwrót: nie każde zdanie języka teorii jest twierdzeniem tej teorii. Twierdzenia są to więc pewne zdania zapisane w języku teorii.

W dalszych przykładach przyjmiemy język identyczny jak w teorii $\mathbf{T}_1$, dlatego w określeniu teorii nie będziemy go podawać.

Przykład 2. Teoria $\mathbf{T}_2$ jest określona następująco:

A_1: a
A_2: ab
$\mathbf{R}_1$: $Aa \rightarrow Aab$
$\mathbf{R}_2$: $Ab \rightarrow Aba$

Teoria $\mathbf{T}_2$ posiada dwa aksjomaty, a oraz ab, i dwie reguły wnioskowania.

Pierwsza reguła $\mathbf{R}_1$ pozwala na dopisanie do twierdzenia zakończonego literą a, litery b.

Jeżeli ostatnim symbolem twierdzenia jest litera b, to zgodnie z regułą $\mathbf{R}_2$ możemy dopisać do niego literę a i tak otrzymane wyrażenie jest również twierdzeniem teorii $\mathbf{T}_2$. Np.

1.	ab	A_2
2.	aba	$\mathbf{R}_2$, 1
3.	$abab$	$\mathbf{R}_1$, 2
4.	$ababa$	$\mathbf{R}_2$, 3

W teorii $\mathbf{T}_2$ otrzymywanie twierdzeń polega na dopisywaniu do otrzymanych już twierdzeń litery a lub b, zależnie od tego jaką literą jest zakończone rozpatrywane twierdzenie.

Przykład 3. Teoria $\mathbf{T}_3$ posiada jeden aksjomat i dwie reguły wnioskowania jak niżej,

A_1: a
$\mathbf{R}_1$: $A \rightarrow Ab$
$\mathbf{R}_2$: $A, B \rightarrow AB$

Pierwsza reguła wnioskowania $\mathbf{R}_1$ mówi, że dopisując do dowolnego twierdzenia teorii $\mathbf{T}_3$ literę b na końcu, otrzymamy nowe twierdzenie teorii $\mathbf{T}_3$.

Reguła druga $\mathbf{R}_2$ mówi, że jeżeli dwa dowolne twierdzenia tej teorii A i B napiszemy obok siebie, to w rezultacie otrzymamy nowe twierdzenie $\mathbf{T}_3$. Np.

1.	a	A_1
2.	ab	$\mathbf{R}_1$, 1
3.	abb	$\mathbf{R}_1$, 2
4.	$ababb$	$\mathbf{R}_2$, 2, 3

Przykład 4. Teoria $\mathbf{T}_4$.

A_1: a
$\mathbf{R}_1$: $A \rightarrow aAa$
$\mathbf{R}_2$: $aA, Aa \rightarrow bAb$

Reguła $\mathbf{R}_1$ mówi, że jeżeli do dowolnego twierdzenia teorii $\mathbf{T}_4$ dopiszemy z obu stron literę a, to otrzymamy nowe twierdzenie teorii $\mathbf{T}_4$.

Reguła $\mathbf{R}_2$ jest nieco bardziej skomplikowana. Jeżeli jedno twierdzenie teorii $\mathbf{T}_4$ zaczyna się na literę a, natomiast drugie twierdzenie tej teorii kończy się na literę a i pozostałe symbole obu twierdzeń są identyczne, to możemy utworzyć nowe twierdzenie tej teorii, pisząc po obu stronach identycznych symboli litery b. Np.

1.	a	A_1
2.	aaa	$\mathbf{R}_1$, 1
3.	$baab$	$\mathbf{R}_2$, 2, 2
4.	$abaaba$	$\mathbf{R}_1$, 3
5.	$bbaabb$	$\mathbf{R}_2$, 4, 4

Wyjaśnienia wymagają tu wiersze 3 i 5. W wierszu 3 zastosowaliśmy regułę $\mathbf{R}_2$ do wiersza 2. Liczba 2 jest napisana dwukrotnie, gdyż twierdzenie *baab* jest wnioskiem bezpośrednim z dwóch przesłanek: *aaa* i *aaa*. Pierwsza przesłanka zaczyna się od litery a, druga natomiast kończy się na literę a. Pozostałe dwa symbole są oczywiście jednakowe, a więc zgodnie z regułą $\mathbf{R}_2$ możemy do nich z obu stron dopisać literę b i otrzymamy wtedy *baab*. Podobnie przebiega konstruowanie twierdzenia *bbaabb*, podanego w wierszu 5 [1]).

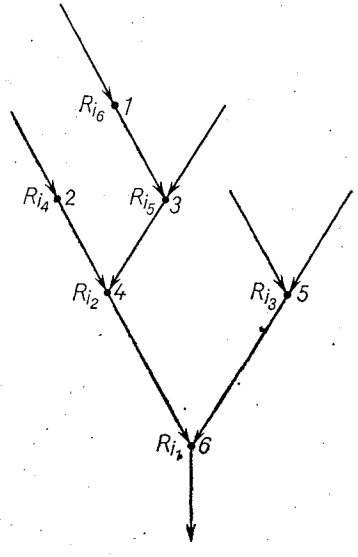
[1]) Podane przykłady teorii dedukcyjnych zostały zaczerpnięte z książki P. Lorenzena, *Einführung in die operative Logik und Mathematik*, Berlin, Göttingen-Heidelberg, 1955.

Rozpatrywane przykłady teorii dedukcyjnych były bardzo proste. Teorie te zawierały po jednym, bądź dwa aksjomaty i kilka prostych reguł wnioskowania.

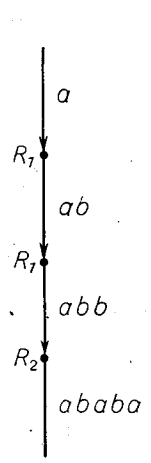
W rzeczywistych teoriach matematycznych liczba aksjomatów może być znacznie większa, często nieskończona. Ilość reguł wnioskowania może być również większa i mają one znacznie bardziej skomplikowany charakter. Nie zmienia to jednak istotnie ważnego dla nas faktu, że otrzymywanie twierdzeń w takich teoriach polega na odpowiednim przekształcaniu napisów wyjściowych, chociaż napisy te mogą być mniej czy bardziej skomplikowane i postać reguł wnioskowania może również znacznie odbiegać od podanych tu przykładów.

§ 3. Wnioskowanie i drzewa

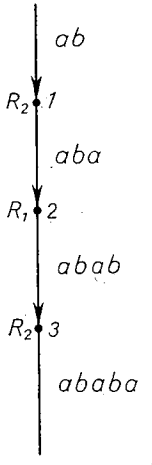
Przebieg procesu wnioskowania wygodnie jest przedstawić graficznie w sposób następujący: każdej operacji przekształcania przesłanek we wnioski przypiszemy punkt, natomiast przesłankom i wnioskom przypiszemy odcinki zorientowane w ten sposób, że przesłankom odpowiadają odcinki wchodzące do punktu reprezentującego operacje wnioskowania, a wniosek bezpośredni jest przedstawiony przez odcinek wychodzący z tego punktu. Rysunek taki przypomina drzewo (rys. 1). Najwyżej położony odcinek przedstawia wniosek końcowy. Odcinki nie zaczynające się od punktu przedstawiają aksjomaty, lub twierdzenia już wyprodukowane, przyjęte jako punkty wyjściowe wnioskowania. Numery obok punktów wskazują kolejność stosowania operacji. Oczywiście kolejność ta może być inna niż podana tutaj. Z naszego punktu widzenia nie gra ona specjalnej roli i dlatego nie będziemy jej bliżej omawiać. Ważne jest tylko by operacje, których przesłanki nie zostały jeszcze otrzymane, były wykonywane po operacjach, których wynikami są potrzebne właśnie przesłanki. Np. operacja 5 może być wykonana jedynie jako ostatnia operacja wnioskowania, gdyż do jej zrealizowania potrzebne są przesłanki będące wynikami operacji 4 i 3.



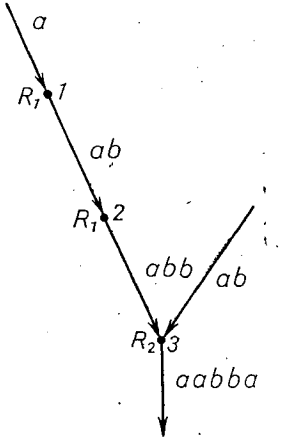
Rys. 1



Rys. 2



Rys. 3

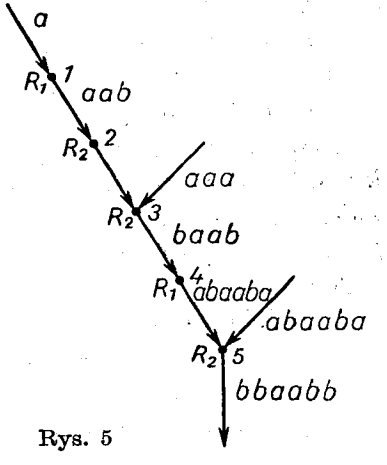


Rys. 4

Podane w poprzednim paragrafie przykłady wnioskowań będą miały odpowiednią postać graficzną, jak to pokazano na rysunkach 2, 3, 4 i 5.

Ponieważ w teoriach $\mathbf{T_1}$ i $\mathbf{T_2}$ reguły wnioskowania odnosiły się tylko do jednej przesłanki, więc na rysunkach 1 i 2 przykłady wnioskowania mają postać nie drzewa, a ciągu nie rozgałęziających się odcinków.

W teoriach $\mathbf{T_3}$ i $\mathbf{T_4}$ reguły $\mathbf{R_2}$ odnoszą się do dwu przesłanek. Należy zwrócić uwagę, że z naszego punktu widzenia dwie nawet identyczne przesłanki są dwoma napisami i dlatego na drzewie będą one występowały jako dwie różne gałęzie. Np. na rys. 4 przesłanka *ab* występuje raz jako przesłanka operacji nr 2, a raz jako przesłanka operacji nr 3. Podobnie na rys. 5 prze-



Rys. 5

słanka *aaa* jest reprezentowana przez dwie gałęzie wchodzące do punktu 3, wnioskowanie polega tu bowiem na otrzymaniu z dwu napisów: *aaa* oraz *aaa* nowego napisu *baaab*. Podobna jest sytuacja w przypadku przesłanki *abaaba* i reguły $\mathbf{R}_2$, zrealizowanej w punkcie 5.

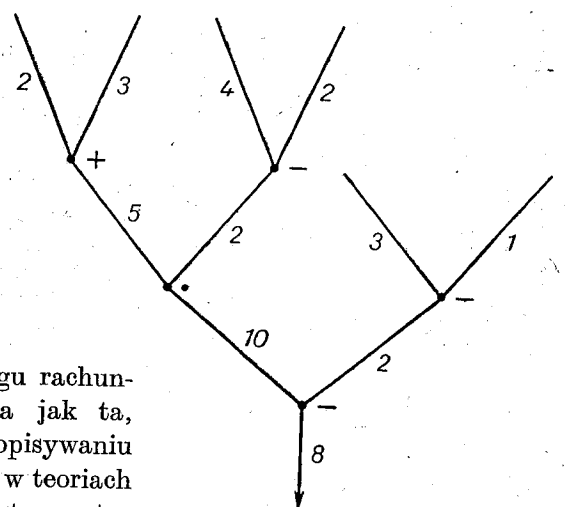
§ 4. Wnioskowanie i obliczanie

Proces wnioskowania przypomina pod wieloma względami wykonywanie rachunku, który również polega na przekształcaniu symboli w myśl reguł rachunkowych. Dane liczby odpowiadają aksjomatom, wyniki częściowe obliczenia — wnioskom pośrednim, wynik końcowy obliczenia — wnioskowi końcowemu, a działania arytmetyczne — regułom wnioskowania.

Przyjmujemy tutaj, że liczby to nie twory abstrakcyjne, jak to jest powszechnie przyjęte w matematyce, lecz napisy i rachunek jest przekształcaniem napisów.

Np. rachunek przedstawiony na rys. 6 możemy zapisać:

1. 2 dana
2. 3 dana
3. 5 +, 1, 2
4. 4 dana
5. 2 dana
6. 2 —, 4, 5
7. 10 ·, 3, 6
8. 3 dana
9. 1 dana
10. 2 —, 8, 9
11. 8 —, 7, 10



Rys. 6

Zasada opisu przebiegu rachunku jest tutaj identyczna jak ta, którą przyjęliśmy w opisywaniu produkowania twierdzeń w teoriach dedukcyjnych. Z lewej strony tabeli podany jest numer wiersza, następna kolumna zawiera symbole, które są przekształcane, w ostatniej zaś kolumnie podano rodzaj zastosowanej reguły przekształcenia oraz numery wierszy, do których daną operację arytmetyczną

zastosowano. Np. wiersz 3 w ostatniej kolumnie podano +, 1, 2, co oznacza, że wykonano dodawanie na liczbach zapisanych w wierszach 1 i 2.

Taki sposób zapisu przebiegu obliczenia nie jest powszechnie przyjęty w rachunkach wykonywanych za pomocą papieru i ołówka, jednakże zasada ta jest stosowana w maszynach matematycznych. W zwykłych obliczeniach podany rachunek zapisalibyśmy konwencjonalnie w powszechnie używanej symbolice matematycznej następująco:

$$(((2 + 3) \cdot (4 - 2)) - (3 - 1))$$

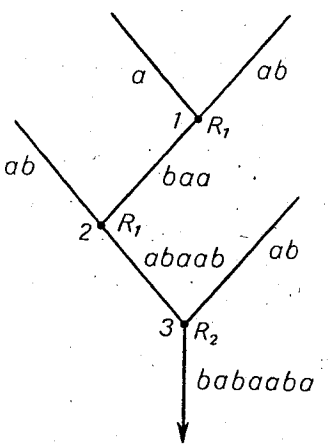
Podobnie możemy zapisać również przebieg produkowania twierdzeń teorii, jednakże ten sposób nie jest przyjęty w tej dziedzinie. Zapisanie poprzednio podanych przykładów produkowania twierdzeń w postaci wzoru wymagałoby dodatkowych wyjaśnień, dlatego ograniczymy się chwilowo do przykładu nie związanego z poprzednio dyskutowanymi teoriami, a podamy specjalny przykład teorii, gdzie to podobieństwo jest widoczne od razu.

Rozpatrzmy przykład produkowania twierdzenia w teorii $\mathbf{T}_5$ określonej następująco:

A_1: a
A_2: ab
$\mathbf{R}_1$: $Ab, B \rightarrow ABb$
$\mathbf{R}_2$: $A, Bb \rightarrow bAB$

Reguła $\mathbf{R}_1$ pozwala napisać obok siebie dwa twierdzenia przez dopisanie ostatniej litery pierwszego twierdzenia na końcu, jeżeli pierwsze twierdzenie jest zakończone na b.

Reguła $\mathbf{R}_2$ pozwala na napisanie obu twierdzeń obok siebie, jeżeli ostatnia litera drugiego twierdzenia kończy się na b, przy czym litera ta pisana jest na początku.



Rys. 7

Przykład wnioskowania w teorii $\mathbf{T}_5$ przedstawiony jest na rys. 7. Wnioskowanie to w postaci tabelki zapiszemy jak niżej:

1.	a	A_1
2.	ab	A_2
3.	baa	$\mathbf{R}_1$, 1, 2
4.	ab	A_2
5.	$abaab$	$\mathbf{R}_1$, 4, 3
6.	ab	A_2
7.	$babaaba$	$\mathbf{R}_2$, 5, 6

Natomiast w postaci wzoru możemy to wnioskowanie przedstawić następująco:

$$((ab\,\mathbf{R}_1\,(a\,\mathbf{R}_1\,ab))\,\mathbf{R}_2\,ab)$$

W jaki sposób drzewo wyrazić w postaci wzoru, może być z łatwością odczytane przez Czytelnika na podstawie analizy rysunku przedstawiającego przebieg obliczenia.

Tak więc produkcję twierdzenia, jak i przebieg rachunku możemy przedstawić zarówno w postaci tabeli jak i formuły matematycznej, powszechnie przyjętej w algebrze, czy arytmetyce.

W dalszym ciągu tej książeczki będziemy stosować zarówno jedną jak i drugą formę przekształcania symboli.

§ 5. Dowód twierdzenia

Jednym z najważniejszych pojęć matematycznych jest pojęcie dowodu.

Aby się przekonać, czy jakieś wyrażenie jest twierdzeniem teorii, wystarczy podać, w jaki sposób może być ono wyprodukowane z aksjomatów. Przyjęliśmy, że produkcję taką przedstawiamy w postaci tabelki, opisującej kolejność stosowania reguł wnioskowania. Tablicę opisującą przebieg produkcji twierdzenia będziemy nazywali **dowodem** tego twierdzenia. Np. dowód twierdzenia *aabbab* w teorii $\mathbf{T}_3$ ma postać:

1.	a	A_1
2.	ab	$\mathbf{R}_1$, 1
3.	abb	$\mathbf{R}_1$, 2
4.	$aabb$	$\mathbf{R}_2$, 1, 3
5.	$aabbab$	$\mathbf{R}_2$, 4, 2

Ściślej biorąc, przez dowód rozumie się w matematyce nie całą tablicę, a tylko ciąg występujących w niej zdań, należących do roz-

patrywanej teorii. Dla twierdzenia *aabbab* dowodem będzie więc ciąg zdań:

a
ab
abb
$aabb$
$aabbab$

Symbole numerujące wiersze dowodu oraz symbole opisujące, jakie reguły zostały zastosowane w każdym kroku dowodu, mają charakter objaśniający, pomocniczy i do dowodu nie są zaliczane.

Ponieważ dowód bez tych dodatkowych objaśnień byłby często zupełnie nieczytelny, więc przez dowód będziemy rozumieli ciąg zdań wraz z objaśnieniami, w jaki sposób zdania te są wzajemnie ze sobą powiązane logicznie — nawet wtedy, gdy o objaśnieniach tych nie będziemy wyraźnie mówili.

A więc dowód twierdzenia, to ciąg zdań teorii takich, że ostatnie zdanie tego ciągu jest twierdzeniem, którego dowód dotyczy, a każde inne zdanie dowodu jest aksjomatem [1]) teorii lub jest wnioskiem bezpośrednim ze zdań go poprzedzających.

Mówiąc nieco dokładniej, **ciąg wyrażeń**

$$\alpha_1, \alpha_2, \ldots, \alpha_n$$

nazwiemy dowodem twierdzenia α w teorii T, jeżeli $\alpha = \alpha_n$ oraz dla każdego $(1 \leqslant i \leqslant n)$ wyrażenie α_i jest aksjomatem teorii T lub też jest wnioskiem bezpośrednim z wyrażeń α_j, gdzie $j < i$, poprzedzających α_i.

Twierdzenie jest jednym zdaniem teorii. Dowód tego twierdzenia to ciąg zdań, jak gdyby rodowód twierdzenia, jego drzewo genealogiczne, mówiące o wszystkich zdaniach potrzebnych do wyprodukowania tego twierdzenia i o związkach zachodzących między tymi zdaniami.

Oczywiście dla danego twierdzenia może istnieć wiele dowodów. Inaczej mówiąc, to samo twierdzenie może być wyprodukowane

[1]) Zamiast aksjomatów możemy w dowodzie przyjąć twierdzenia już udowodnione. Gdybyśmy bowiem każde twierdzenie chcieli wywodzić z aksjomatów, większość dowodów byłaby niepotrzebnie długa i nieczytelna.

z aksjomatów na różne sposoby. Np. twierdzenie *aabbab* w teorii $\mathbf{T}_3$ może mieć również dowód następujący:

1.	*a*	A_1
2.	*ab*	$\mathbf{R}_1$, 1
3.	*aab*	$\mathbf{R}_2$, 1, 2
4.	*aabb*	$\mathbf{R}_1$, 3
5.	*aabbab*	$\mathbf{R}_2$, 4, 2

Obie tabelki będziemy uważali za dwa różne dowody tego samego twierdzenia. Przebieg produkcji w obu przypadkach bowiem był inny. A więc dowód w naszym rozumieniu jest szczegółowym opisem całego przebiegu produkcji twierdzenia. Używając innego porównania, dowód to ślad po wszystkich czynnościach, które doprowadziły do wyprodukowania twierdzenia.

W dalszych rozważaniach będziemy przez dowód rozumieli jakiś konkretny jeden sposób produkcji twierdzenia [1]).

Przez dowód rozumiemy więc opis produkcji twierdzenia, inaczej mówiąc opis wnioskowania.

W następnym rozdziale zajmiemy się zasadniczym zagadnieniem dla maszynowego dowodzenia twierdzeń, a mianowicie poszukiwaniem dowodów.

---

[1]) Dla głębszego zrozumienia pojęcia dowodu odsyłamy Czytelnika do książki W. Pogorzelskiego i J. Słupeckiego *O dowodzie matematycznym*, PZWS, 1962, s. 51.

Rozdział II

POSZUKIWANIE DOWODÓW

§ 1. Budowanie teorii matematycznych

W każdej teorii dedukcyjnej możemy wyprodukować nieskończenie wiele twierdzeń. Do twierdzeń już wyprodukowanych można ponownie zastosować odpowiednie reguły wnioskowania, otrzymując dalsze twierdzenia teorii. Jednakże w każdej teorii liczba aktualnie wyprowadzonych twierdzeń jest oczywiście skończona. Z rozwojem teorii liczba ta może wzrastać. Jednakże wzrost ten nie jest w praktyce nieograniczony. Powód tego jest następujący: Jakkolwiek w każdej teorii dedukcyjnej można wyprodukować nieskończenie wiele twierdzeń, nie zapominajmy o tym, że teoria to nie formalna zabawa symbolami. Tworzenie teorii przez matematyka nie sprowadza się do kolejnego wypisywania twierdzeń i ich dowodów; teorie te są budowane w celach poznawczych. A więc twierdzenia teorii muszą być zrozumiałe, muszą się dać czytać przez człowieka ze zrozumieniem. Wiadomo zaś, że zdolności recepcyjne człowieka są ograniczone. Zbyt długie ciągi symboli nie mogą być przez człowieka rozpoznawane i czytane ze zrozumieniem [1]).

Gdybyśmy więc rozpatrzyli wszystkie znane twierdzenia w dowolnej teorii matematycznej okazałoby się, że liczba występujących

---

[1]) Podobna sytuacja występuje przy zapisywaniu liczb. Wprawdzie możemy zapisywać liczby zawierające dowolną liczbę cyfr, np. 6345297354678198734567386529187538736459873000175368827118723657894 32, jednakże posługiwanie się tak zapisanymi liczbami jest w praktyce niemożliwe. Nie rozumiemy zupełnie o jaki rząd wielkości tu chodzi. Z tych samych względów słowa dowolnego języka mają ograniczoną liczbę liter. W żadnym istniejącym języku nie ma na pewno słów składających się ze stu liter. Były by one bowiem nieczytelne.

w nich symboli jest ograniczona i nie przekracza np. stu. Liczba możliwych twierdzeń jest więc również ograniczona[1]).

Drugim czynnikiem ograniczającym liczbę twierdzeń w teorii jest fakt, że nawet wśród twierdzeń o określonej ilości symboli, nie wszystkie są jednakowo dla matematyka interesujące. Matematyk nie wyprowadza bezmyślnie wszystkich twierdzeń po kolei, lecz stara się uzyskać tylko takie twierdzenia, które stanowią jakieś interesujące go fakty w badanej teorii. Oczywiście, co jest interesujące a co nie, jest sprawą trudną do formalnego rozstrzygnięcia i zależy od intuicji badacza. W każdym razie budowanie i badanie teorii nie sprowadza się do mniej czy więcej bezmyślnego produkowania kolejnych jej twierdzeń. Zazwyczaj nowe twierdzenia są stawiane jako hipotezy i aby się przekonać, czy stawiana hipoteza jest twierdzeniem teorii, należy podać jej dowód. A więc zadane jest jakieś wyrażenie i pytamy, czy można je wyprowadzić z aksjomatów rozważanej teorii. Chodzi tu przy tym nie tylko o odpowiedź „tak" bądź „nie", a o podanie dowodu, o ile taki istnieje.

§ 2. Poszukiwanie dowodów w teorii dedukcyjnej

Zgodnie z uwagami w poprzednim paragrafie, matematyk badający teorię matematyczną stoi zazwyczaj przed następującym zagadnieniem: czy zadane wyrażenie jest twierdzeniem rozpatrywanej teorii.

Np. możemy zapytać, czy wyrażenie *aabbab* jest twierdzeniem teorii T_3. Jak się o tym przekonać? Należy spróbować wyprodukować to wyrażenie z aksjomatów teorii T_3. Dla rozpatrywanego przykładu przebieg produkcji będzie następujący:

[1]) Liczbę twierdzeń, którymi da się praktycznie manipulować, można znacznie zwiększyć w każdej teorii, przez wprowadzenie do języka teorii, tzw. **definiowania.** Definiowanie pozwala na zastępowanie długich, nieczytelnych ciągów symboli — nowymi, krótkimi symbolami. Zbyt długie ciągi tych nowych symboli zastępujemy znów nowymi symbolami itp. Oczywiście procesu definiowania nie możemy praktycznie kontynuować w nieskończoność, gdyż stoją tu na przeszkodzie czynniki podobne jak przy czytaniu zbyt długich ciągów symboli: ilość definicji, które możemy wprowadzić jest również ograniczona naszą zdolnością percypowania. Tym niemniej definiowanie pozwala na znaczne zwiększenie liczby twierdzeń praktycznie wyprowadzonych w teorii.

1. a	A_1
2. ab	$\mathbf{R}_1$, 1
3. abb	$\mathbf{R}_1$, 2
4. $aabb$	$\mathbf{R}_2$, 1, 3
5. $aabbab$	$\mathbf{R}_2$, 4, 2

Ponieważ produkcja taka jest możliwa, więc $\vdash_{T_3} aabbab$.

Warto zwrócić uwagę, że poszukiwanie metody wyprodukowania twierdzenia jest tutaj oparte na intuicji, a nie na żadnych ustalonych zasadach. Poszukiwanie takie może się udać lub nie. Jeżeli zostało ono zakończone pozytywnym rezultatem, badane wyrażenie jest twierdzeniem teorii; w przypadku przeciwnym nie możemy nic orzec. Produkcja taka może bowiem istnieć, jednakże przez nas nie została znaleziona.

W prostych przypadkach, podobnych do przytoczonego, na podstawie struktury wyrażenia, którego dowodu szukamy, można się domyśleć, jak dowód taki może prawdopodobnie przebiegać. Jednakże w bardziej skomplikowanych sytuacjach odgadnięcie dowodu nie jest takie proste i może nawet się nie udać po wielu próbach, mimo że dowód taki istnieje.

Nasuwa się więc konieczność podania jednoznacznej i wykonalnej realnie — przy użyciu środków, którymi rozporządzamy — metody postępowania, pozwalającej dla zadanego wyrażenia znaleźć jego dowód, o ile taki dowód istnieje? Gdyby taka metoda istniała, poszukiwanie dowodów byłoby czynnością równie prostą jak np. wykonywanie rachunków. Kilka uwag ogólniejszych na ten temat uczynimy w ostatnim rozdziale tej książki, tutaj natomiast tylko stwierdzamy, że czasem podanie takiej metody jest możliwe. Znajdowanie dowodu nie jest wtedy dziełem przypadku czy szczęścia, a rezultatem systematycznego postępowania, według określonego jednoznacznie przepisu. W dalszych paragrafach tego rozdziału zajmiemy się tym zagadnieniem, jednakże nim przystąpimy do tego tematu, uczynimy kilka uwag na temat dowodzenia i wnioskowania.

§ 3. Dowodzenie i wnioskowanie

Obie czynności wymienione w tytule tego paragrafu są w pewnym sensie czynnościami wzajemnie odwrotnymi.

W pierwszym przypadku mamy zadane aksjomaty oraz kolejność i rodzaj reguł wnioskowania, a szukamy wniosku końcowego [1]).

W dowodzeniu natomiast zadane mamy aksjomaty, reguły wnioskowania teorii oraz wyrażenie, które chcemy zbadać, czy jest twierdzeniem — szukamy natomiast dowodu, tj. ciągu wyrażeń, który jest logicznie powiązanym łańcuchem wniosków i przesłanek, zaczynającym się od aksjomatów a kończącym się na badanym twierdzeniu.

Szukanie dowodu możemy więc przyrównać do następującego zadania arytmetycznego: zadany mamy wynik końcowy jakiegoś nieznanego nam obliczenia, rodzaj możliwych wykonywanych działań arytmetycznych oraz zbiór liczb, na których działania te są określone, szukamy natomiast obliczenia, które do tego wyniku prowadzi, tj. ciągu operacji arytmetycznych i danych początkowych takich, że po ich wykonaniu otrzymamy żądany wynik. Np. możemy zapytać, w wyniku jakich działań arytmetycznych na liczbach naturalnych otrzymamy liczbę 5? Odpowiedź może być np., że w wyniku dodania liczb 2 i 3. Oczywicie nie jest to jedyne możliwe rozwiązanie.

Natomiast w przypadku wnioskowania pytanie brzmi: jaką liczbę otrzymamy w wyniku zastosowania do liczb 2 i 3 operacji dodawania?

Używając jeszcze innego sformułowania, możemy powiedzieć, że we wnioskowaniu podobnie jak i w obliczeniu zadane mamy drzewo opisujące przebieg produkcji, natomiast przy dowodzeniu mamy wynik końcowy i szukamy drzewa, które opisuje produkcję tego wyniku. Szukamy więc opisu procesu, który prowadzi do z góry zadanego rezultatu.

Zdanie sobie jasno sprawy z tych dwu rodzajów czynności ma zasadnicze znaczenie dla rozumienia dalszych rozważań i dlatego proponujemy Czytelnikom uważne przemyślenie tego problemu.

§ 4. Wnioskowanie redukcyjne

Stwierdziliśmy, że poszukiwanie dowodu za pomocą dotychczasowych środków jest do celów maszynowych nieprzydatne. Nasuwa się więc pytanie, czy można podać taką metodę szukania dowodu, która

[1]) Podobna sytuacja jest w rachunku, gdzie zadane są liczby początkowe oraz kolejność i rodzaj operacji arytmetycznych, które należy wykonać, aby otrzymać szukany wynik końcowy obliczenia.

by była jednoznacznie określana przez strukturę wyrażenia, którego dowodu szukamy [1]).

W tym celu wprowadzimy **reguły redukowania.** Reguły wnioskowania pozwalały na wyprowadzanie wniosków z przesłanek. Reguły redukowania pozwalają na czynności odwrotne — na znalezienie przesłanek dla zadanego wniosku [2]).

Jeżeli więc szukamy dowodu zadanego wyrażenia, to najpierw za pomocą odpowiedniej reguły redukowania szukamy przesłanek, z których to wyrażenie wynika. Tak otrzymane przesłanki stanowią punkt wyjściowy dalszej analizy, tzn. dla każdej przesłanki szukamy dalszych przesłanek, z których one wynikają. Czynność tę powtarzamy tak długo aż otrzymamy wyrażenia, których dalej rozłożyć już nie można. Wyrażenia takie będziemy nazywać **atomami.** Jeżeli otrzymane w wyniku takiego postępowania atomy wszystkie są aksjomatami teorii, to badane wyrażenie jest twierdzeniem. Jeżeli choć jeden atom redukowanego wyrażenia nie jest aksjomatem, wtedy wyrażenie to twierdzeniem nie jest.

Przykłady teorii z regułami redukowania rozpatrzymy w następnym paragrafie. Teorię z regułami redukowania będziemy nazywali **teorią redukcyjną.**

§ 5. Teorie redukcyjne

Rozpatrzmy szukanie dowodów w teorii redukcyjnej T_1' z następującymi aksjomatami:

$$A_1: \quad aa$$
$$A_2: \quad ab$$
$$A_3: \quad ba$$

oraz z jedną regułą redukowania

$$\mathbf{R_1}: \quad \alpha A \beta \rightarrow \alpha A, A\beta$$

Małe litery greckie α i β oznaczają symbole a lub b, natomiast litera A oznacza dowolny ciąg liter a i b, np. aab. Reguła redukowania

[1]) Problem ten jest podobny do następującego zadania: Podać opis przebiegu składania przedmiotu na podstawie analizy jego struktury.

[2]) Reguły redukcyjne można więc przyrównać do reguł rozkładania przedmiotu na jego części składowe.

mówi, że dla dowolnego wyrażenia przesłankami są wyrażenia otrzymane przez opuszczenie pierwszego i ostatniego symbolu w tym wyrażeniu. Np. przesłankami wyrażenia *aabba* będą odpowiednio wyrażenia *aabb* oraz *abba*. Pierwszą przesłankę otrzymaliśmy przez opuszczenie w wyrażeniu *aabba* ostatniej litery tego wyrażenia, natomiast drugą przesłankę otrzymaliśmy, opuszczając w wyrażeniu *aabba* pierwszy symbol.

Szukanie dowodu za pomocą reguł redukowania jest bardzo proste i jednoznaczne i nie wymaga domyślności. Każdy krok postępowania jest tutaj ściśle określony.

Poszukajmy np. dowodu twierdzenia *aaba*, pisząc poszczególne kroki w postaci tabelki, jak to czyniliśmy poprzednio:

1.	*aaba*		
2.	*aab*	$\mathbf{R}_1$, 1	
3.	*aba*	$\mathbf{R}_1$, 1	
4.	*aa*	$\mathbf{R}_1$, 2	A_1
5.	*ab*	$\mathbf{R}_1$, 2	A_2
6.	*ab*	$\mathbf{R}_1$, 3	A_2
7.	*ba*	$\mathbf{R}_1$, 3	A_3

Stosując do wyrażenia 1 regułę redukowania, otrzymaliśmy przesłanki podane w wierszach 2 i 3.

Dla wyrażenia 2 przesłankami są wyrażenia podane w wierszach 4 i 5.

Przesłankami wyrażenia 3 są wyrażenia 6 i 7.

Wyrażenia 4, 5, 6 i 7 są atomami, dalej nie możemy ich bowiem rozkładać według reguły $\mathbf{R}_1$. Ponieważ wszystkie atomy są aksjomatami, więc $\vdash_{T_1} aaba$.

Przykład innego dowodu w teorii T_1'.

1.	*baaba*		
2.	*baab*	$\mathbf{R}_1$, 1	
3.	*aaba*	$\mathbf{R}_1$, 1	
4.	*baa*	$\mathbf{R}_1$, 2	
5.	*aab*	$\mathbf{R}_1$, 2	
6.	*aab*	$\mathbf{R}_1$, 3	
7.	*aba*	$\mathbf{R}_1$, 3	
8.	*ba*	$\mathbf{R}_1$, 4	A_3
9.	*aa*	$\mathbf{R}_1$, 4	A_1

10.	*aa*	$\mathbf{R}_1$, 5	A_1
11.	*ab*	$\mathbf{R}_1$, 5	A_2
12.	*aa*	$\mathbf{R}_1$, 6	A_1
13.	*ab*	$\mathbf{R}_1$, 6	A_2
14.	*ab*	$\mathbf{R}_1$, 7	A_2
15.	*ba*	$\mathbf{R}_1$, 7	A_3

Ponieważ wszystkie atomy są aksjomatami, więc $\vdash_{T_1'} baaba$. Spróbujmy znaleźć dowód wyrażenia *abbb*.

1.	*abbb*		
2.	*abb*	$\mathbf{R}_1$, 1	
3.	*bbb*	$\mathbf{R}_1$, 1	
4.	*ab*	$\mathbf{R}_1$, 2	A_2
5.	*bb*	$\mathbf{R}_1$, 2	—
6.	*bb*	$\mathbf{R}_1$, 3	—
7.	*bb*	$\mathbf{R}_1$, 3	—

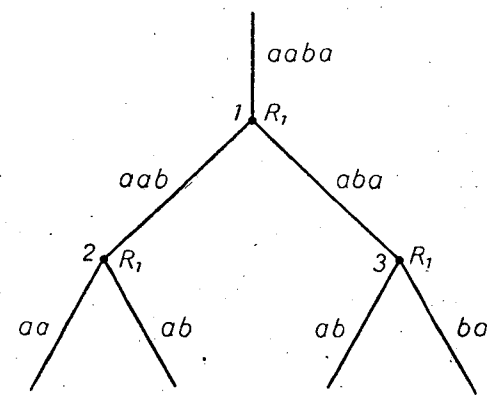
Atomami wyrażenia *abbb* są wiersze 4, 5, 6 i 7, jednakże tylko w wierszu 4 znajduje się aksjomat; w pozostałych wierszach 5, 6 i 7 atomy nie są aksjomatami. Wyrażenie *abbb* nie jest więc twierdzeniem teorii T_1'.

Widzimy, że szukanie dowodu za pomocą reguł redukowania nie jest dziełem przypadku. Dowód każdego twierdzenia teorii T_1' może być zawsze znaleziony w prosty sposób.

Łatwo można podać przykłady innych podobnych teorii redukcyjnych.

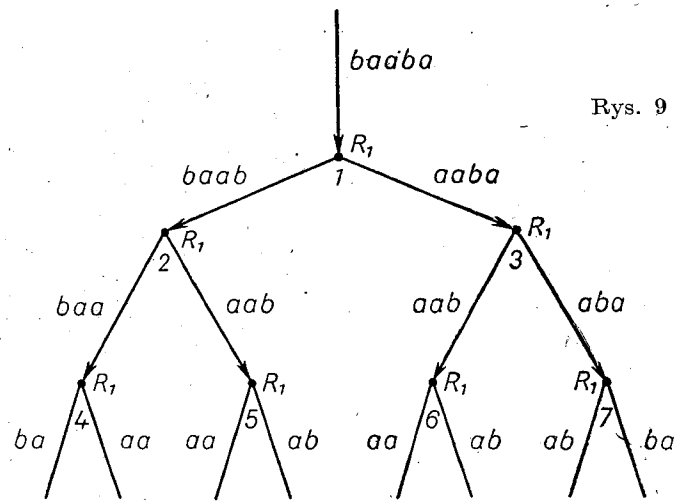
§ 6. Redukowanie i drzewa

Podobnie jak wnioskowanie dedukcyjne, wnioskowanie redukcyjne możemy również przedstawić w postaci drzewa. Dowody twierdzenia *aaba* oraz *baaba* są przedstawione odpowiednio na rys. 8 i 9. Znaczenie ga-



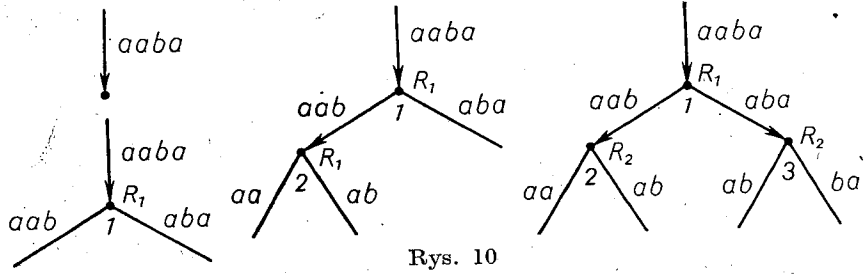
Rys. 8

łęzi i punktów jest identyczne jak poprzednio. Jednakże, o ile w przypadku wnioskowania dedukcyjnego drzewo było zadane przed rozpoczęciem wnioskowania, to dla wnioskowania redukcyjnego



Rys. 9

drzewo opisujące przebieg wnioskowania możemy narysować dopiero po znalezieniu dowodu. Ściślej mówiąc, drzewo to możemy rysować w trakcie wykonywania dowodu. Proces rysowania drzewa zaczynamy od „pnia" odpowiadającemu zadanemu wyrażeniu.



Rys. 10

Po każdym zastosowaniu reguły redukowania dorysowujemy dwie nowe gałęzie, odpowiadające znalezionym przesłankom.

Drzewo jest tutaj narysowane „do góry nogami" co oczywiście nie ma znaczenia. Zakładamy przy tym, że w każdej chwili wykonywana jest co najwyżej jedna operacja redukowania.

Proces szukania dowodu najlepiej ilustrowałoby więc drzewo „rosnące", w którym po każdym kroku redukcji wyrastają dwie nowe gałęzie z odpowiedniego rozgałęzienia, jak to pokazano na rysunku 10.

§ 7. Symetria reguł

Przyjęliśmy, że reguła $\mathbf{R}_1$ w teorii $\mathbf{T}_1'$ pozwala na znalezienie przesłanek dla każdego wyrażenia języka teorii $\mathbf{T}_1'$.

Regułę $\mathbf{R}_1$ można również stosować w drugą stronę, tj. do znalezienia wniosków dla dowolnych przesłanek. Regułę R_1 należy wtedy rozumieć następująco: Jeżeli dwa wyrażenia αA oraz $A\beta$, są twierdzeniami, gdzie α i β są literami a lub b, natomiast A oznacza dowolny ciąg symboli a i b, to możemy je zastąpić wyrażeniem postaci $\alpha A\beta$, i tak otrzymane wyrażenie jest również twierdzeniem. Np. jeżeli ab i ba są twierdzeniami, to również wyrażenie aba jest twierdzeniem.

Produkcję twierdzenia $aaba$ możemy — za pomocą reguły $\mathbf{R}_1$ — przedstawić następująco:

1. aa	A_1
2. ab	A_2
3. aab	$\mathbf{R}_1$, 1, 2
4. ab	A_2
5. ba	A_3
6. aba	$\mathbf{R}_1$, 4, 5
7. $aaba$	$\mathbf{R}_1$, 3, 6

Reguły wnioskowania, produkcji pozwalają więc złożyć wyrażenie z aksjomatów, reguły redukowania pozwalają natomiast rozłożyć wyrażenie złożone na jego składniki elementarne.

Drzewo tego procesu jest oczywiście takie samo, jak na rys. 8.

W przypadkach, w których jest możliwe stosowanie reguł od przesłanek do wniosków, jak i odwrotnie, tj. od wniosków do przesłanek, proces szukania dowodu jest jednoznacznie wyznaczony przez strukturę badanego wyrażenia. A więc każde twierdzenie takiej teorii zawiera w sobie informację o tym, w jaki sposób zostało ono zbudowane z aksjomatów. W strukturze twierdzenia jest wtedy zawarta historia jego powstania.

Rozdział III

PROCES REDUKOWANIA

§ 1. Pojęcie procesu

Wspominaliśmy w poprzednich paragrafach, że szukanie dowodu za pomocą reguł redukowania przypomina rozkładanie przedmiotu złożonego na jego części składowe. Wygodnie będzie dla dalszych rozważań wprowadzić ogólne pojęcie **procesu redukowania**, które zależnie od interpretacji może oznaczać szukanie dowodu, bądź jakąkolwiek inną czynność o podobnym charakterze. Pozwoli to nam lepiej zrozumieć strukturę maszyny, która taki proces będzie realizowała. Struktura takiej maszyny bowiem nie zależy od tego, czy ma ono zrealizować proces dowodzenia, czy też jakikolwiek inny proces do niego podobny, a zależy od pewnych ogólnych własności procesu.

Z pojęciem procesu spotykamy się na każdym kroku. Mówimy np. o procesie uczenia się, starzenia, rozwoju itp. Ogólnie mówiąc, w pojęciu procesu zawarty jest element zmiany. Zmiana ta może dotyczyć struktury lub stanu jakiegoś obiektu materialnego.

Tak szerokie zrozumienie pojęcia procesu byłoby dla naszych celów nieprzydatne i dlatego je zawęzimy.

Przez proces będziemy rozumieli ciąg operacji, w wyniku których z jednych przedmiotów tworzymy nowe przedmioty. W szczególności przedmiotami procesu mogą być napisy, a operacje będą wtedy polegały na przekształcaniu napisów w myśl określonych reguł.

Proces jest więc określony przez zespół przedmiotów X, zespół operacji Ω oraz porządek wykonywania operacji Π.

Proces będziemy oznaczać symbolicznie przez

$$P = (X, \Omega, \Pi)$$

Dalsze ograniczanie polegać będzie na założeniu, że rozpatrujemy

tylko takie procesy, w których wykonywana jest w każdej chwili tylko jedna operacja. Procesy takie nazwiemy **sekwencyjnymi** [1]).

W przypadku procesu dowodzenia elementami X są wyrażenia teorii, w której szukamy dowodu. Ale oczywiście nie wszystkie wyrażenia tej teorii a tylko te, które wchodzą w skład dowodu. Podobnie elementami Ω są wszystkie operacje wykonane w trakcie dowodzenia. Każda operacja jest tutaj realizacją pewnej reguły redukowania. Oczywiście ta sama reguła może być zrealizowana wielokrotnie. A więc w Ω mogą występować jednakowe operacje. Podobnie w X mogą występować wielokrotnie te same wyrażenia.

§ 2. Definicja procesu redukowania

Powiemy, że proces $P = (X, \Omega, \Pi)$ jest **procesem redukcji**, jeżeli P spełnia następujące warunki:

1. W wyniku każdej operacji ω, należącej do Ω, zastosowanej do obiektu α, należącego do X, otrzymamy dwa obiekty β i γ należące do X. Obiekt α nazwiemy argumentem operacji ω, a obiekty β i γ nazwiemy odpowiednio lewym i prawym wynikiem operacji ω.

2. Dla każdego obiektu α należącego do X istnieje co najmniej jedna taka operacja ω, że α jest argumentem tej operacji lub też jej lewym, lub prawym wynikiem.

3. Istnieje dokładnie jeden taki obiekt α należący do X, że α nie jest wynikiem żadnej operacji należącej do Ω. Takie α nazwiemy **argumentem początkowym** procesu.

4. Dla każdego obiektu α należącego do X istnieje dokładnie jeden ciąg

$$\alpha_1, \omega_1, \alpha_2, \omega_2, \ldots, \alpha_n, \omega_n$$

taki, że α_i i ω_i są obiektami i operacjami procesu P, oraz dla każdego i obiekt α_i jest argumentem ω_i, obiekt α_{i+1} jest wynikiem operacji ω_i oraz $\alpha_n = \alpha$.

Podaną definicję prostego procesu redukowania łatwiej zrozumieć,

[1]) Można sobie wyobrazić, że w procesie wykonywanych jest jednocześnie wiele operacji. Np. jeden dowód przeprowadzany jest nie przez jednego matematyka, a przez kilku jednocześnie. Każdy przy tym wykonuje inną część dowodu. Takie postępowanie nie byłoby procesem sekwencyjnym.

jeżeli przyjmiemy, że mówimy nie o obiektach i operacjach, lecz o „gałęziach" i „rozgałęzieniach"[1]).

Proces redukowania jest wtedy interpretowany jako rysunek przypominający drzewo. Definicja procesu redukowania jest właściwie opisem takiego rysunku.

Moglibyśmy przyjąć, że z każdego rozgałęzienia wychodzą nie dwie, lecz dowolna ilość gałęzi od 1 do n. Jednakże pozostaniemy przy dotychczasowym założeniu.

Zamiast proces będziemy też często mówili drzewo.

Jak to już wspominaliśmy, w wyniku każdej operacji z jednego rozgałęzienia drzewa wyrastają dwie nowe gałęzie. Proces redukowania można by więc przyrównać do procesu wzrostu drzewa, takiego jednak, że w każdej chwili wyrasta co najwyżej jedna para nowych gałęzi[2]).

§ 3. Porządek operacji

W procesie redukowania operacje można wykonywać według różnych kolejności. Okazuje się, że z punktu widzenia konstrukcji maszyny nie jest rzeczą obojętną, w jakim porządku wykonywane są operacje w procesie redukowania. Cztery z nich są dla zastosowania w maszynach szczególnie wygodne. Zapoznamy się tylko z dwiema z nich; dwa dalsze są do nich podobne, nie będziemy ich więc tutaj omawiać. Porządki te nazwiemy porządkiem **poprzecznym** oraz porządkiem **wzdłużnym** i będziemy je odpowiednio oznaczać literami $\overline{P}$ oraz $\overline{W}$.

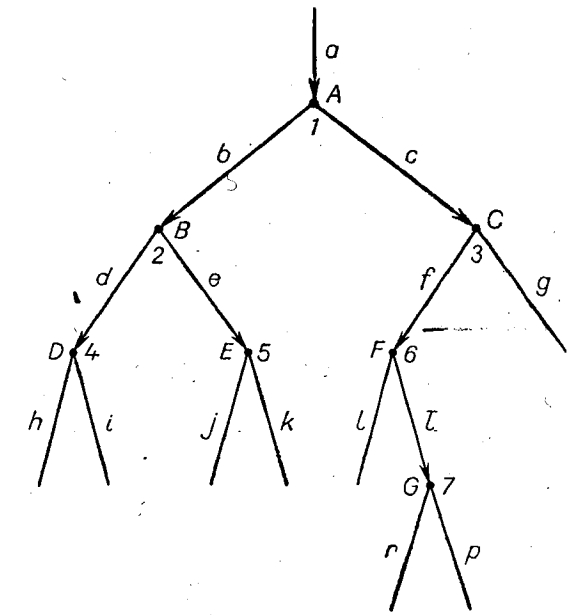
Porządek wykonywania działań wyznaczymy przez podanie numeracji punktów drzewa. Zasadę numerowania działań w obu porządkach wyjaśnimy na podstawie przykładów.

Porządek poprzeczny ($\overline{P}$). Zasadę numeracji poprzecznej wyjaśnia rys. 11. Operacja położona najwyżej ma numer 1. Pozostałe operacje numerujemy według następującej zasady: drzewo dzielimy na „piętra". Na każdym piętrze numerujemy operacje kolejnymi liczbami naturalnymi od strony lewej do prawej. Na każdym piętrze

[1]) lub odcinkach i punktach.

[2]) W rzeczywistym drzewie wyrasta jednocześnie wiele gałęzi z różnych rozgałęzień, a więc nie jest to proces sekwencyjny.

najmniejszy numer operacji jest o jedność większy od największego numeru operacji na piętrze wyższym. A więc operacje numerujemy z lewa na prawo i od góry w dół.



Rys. 11

Małe litery na rysunku oznaczają kolejne obiekty otrzymywane w wyniku każdej operacji, duże litery natomiast oznaczają wykonywane operacje.

W wyniku pierwszej operacji A z obiektu a otrzymamy obiekty b i c. Następnie wykonujemy drugą operację B i otrzymamy w wyniku obiekty d i e itd.

Dla porządku $\overline{P}$ proces będzie więc przebiegał następująco:

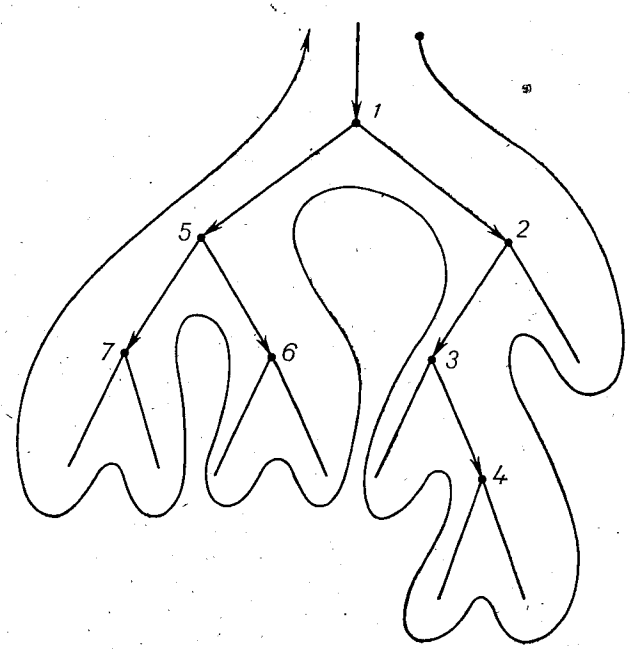
1. a	A, 2, 3	1
2. b	B, 4, 5	2
3. c	C, 6, 7	3
4. d	D, 8, 9	4
5. e	E, 10, 11	5
6. f	F, 12, 13	6
7. g	—	
8. h	—	

9. i	—	
10. j	—	
11. k	—	
12. l	—	
13. $ł$	G, 14, 15	7
14 m	—	
15. n	—	

Przyjęliśmy tu nieco inną zasadę opisu procesu, niż to uczyniliśmy poprzednio, a mianowicie podajemy przy każdym obiekcie operację, którą do niego zastosowano, oraz numery wierszy, gdzie zapisano wyniki tej operacji. Kreska w trzeciej kolumnie tablicy oznacza, że obiekty w tym wierszu nie są już dalej rozkładane.

Poprzednio zaś pisaliśmy operację przy jej wyniku, podając numery wierszy, z których wynik ten został otrzymany. Oczywiście obie metody są równoważne. Będziemy stosowali obie zależnie od okoliczności. Ponadto w ostatniej kolumnie po prawej stronie podano numer wykonywanej operacji.

Porządek wzdłużny. ($\overline{W}$). Zasadę numerowania operacji w porządku $\overline{W}$ pokazano na rys. 12. Na rysunku tym dla uproszczenia



Rys. 12

nie podano oznaczeń gałęzi i rozgałęzień literami, które są takie same, jak na rys. 11.

Zasada numeracji w porządku $\overline{W}$ jest następująca: obchodzimy drzewo wzdłuż obwiedni, poczynając od punktu A i jeżeli idziemy w dół linii AB, piszemy kolejny numer mijanego rozgałęzienia.

Przebieg procesu przy porządku $\overline{W}$ wykonywania operacji będzie wyglądał następująco:

1. a	A, 2, 3	1
2. b	B, 10, 11	5
3. c	C, 4, 5	2
4. f	F, 6, 7	3
5. g	—	
6. e	—	
7. $ł$	G, 8, 9	4
8. m	—	
9. n	—	
10. d	D, 14, 15	7
11. e	E, 12, 13	6
12. j	—	
13. k	—	
14. h	—	
15. i	—	

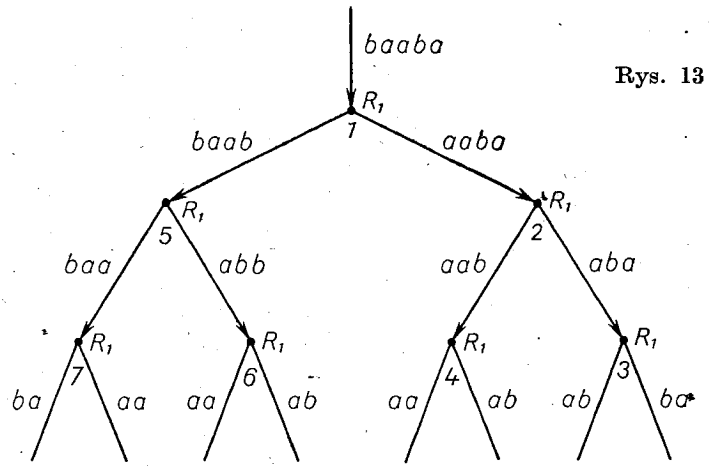
Widzimy, że sposób pisania tablicy, opisującej przebieg procesu, jest dla porządku $\overline{W}$ całkiem inny niż dla porządku $\overline{P}$.

W poprzedniej tablicy operacje procesu były wykonywane w takim samym porządku, w jakim były wpisywane do tablicy. Dla porządku $\overline{W}$ jest odwrotnie: wykonujemy zawsze ostatnią wpisaną do tablicy operację.

Dla dokładniejszego zrozumienia różnicy między oboma porządkami proponujemy Czytelnikowi wykonanie kilku prostych przykładów.

Np. dowód twierdzenia *baaba* w teorii T_1, podany w rozdziale II, § 5, był wykonany w porządku $\overline{P}$. Przebieg tego procesu dla porządku $\overline{W}$ będzie miał natomiast postać: (numeracja operacji w porządku $\overline{W}$ dla tego dowodu podana jest na rys. 13)

Rys. 13



1. *baaba*	$\mathbf{R}_1$, 2, 3	1
2. *baab*	$\mathbf{R}_1$, 10, 11	5
3. *aaba*	$\mathbf{R}_1$, 4, 5	2
4. *aab*	$\mathbf{R}_1$, 8, 9	4
5. *aba*	$\mathbf{R}_1$, 6, 7	3
6. *ab*	A_2	
7. *ba*	A_3	
8. *aa*	A_1	
9. *ab*	A_2	
10. *baa*	$\mathbf{R}_1$, 14, 15	7
11. *aab*	$\mathbf{R}_1$, 12, 13	6
12. *aa*	A_1	
13. *ab*	A_2	
14. *ba*	A_3	
15. *aa*	A_1	

Porównanie tablicy z rysunkiem 13 pozwoli na dokładne zorientowanie się w kolejności wykonywania operacji oraz sposobie zapisywania wyników operacji w tablicy.

Podane obie kolejności wykonywania operacji mają zasadnicze znaczenie dla dalszych rozważań, dlatego dokładne ich opanowanie jest nieodzowne dla zrozumienia dalszych rozważań.

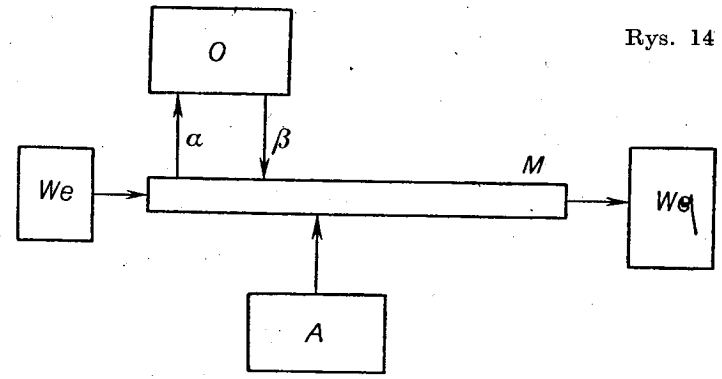
Rozdział IV

REALIZACJA PROCESU WNIOSKOWANIA Z PORZĄDKIEM $\overline{P}$ ZA POMOCĄ MASZYNY BEZADRESOWEJ

§ 1. Ogólny schemat maszyny

W rozdziale tym zajmiemy się zagadnieniem, jak powinna być zbudowana maszyna matematyczna realizująca proces wnioskowania redukcyjnego w porządku $\overline{P}$.

Rys. 14



Ogólny schemat maszyny pokazany jest na rys. 14. Maszyna składa się z:

O — operatora, realizującego operacje redukowania,
M — magazynu liniowego przesłanek,
A — magazynu aksjomatów,
S — sterowania, nie pokazanego na rysunku,
We — wejścia,
Wy — wyjścia.

Urządzenia zwane **wejściem i wyjściem** pozwalają na wprowadzanie i wyprowadzanie do i z maszyny wszelkich informacji. Tak więc np. za pośrednictwem wejścia wprowadzane są do maszyny aksjomaty teorii oraz wyrażenie, którego dowodu szukamy. Wyrażenie to będzie argumentem pierwszej operacji. Będziemy je w dalszym ciągu nazywać **hipotezą początkową.**

Za pomocą wyjścia wyprowadzane są na zewnątrz maszyny wyniki wnioskowania i dodatkowe objaśnienia.

Wyjściem maszyny może być elektryczna maszyna do pisania, która automatycznie drukuje po znalezieniu dowód twierdzenia. Wejściem natomiast może być np. klawiatura, za pomocą której wpisujemy aksjomaty i hipotezy do odpowiednich magazynów.

Wejścia i wyjścia dla prostoty również na rysunku nie pokazano.

Operator, magazyny i sterowanie są skomplikowanymi, szybko działającymi urządzeniami elektronowymi. Np. można przyjąć, że operator może znaleźć 100 000 przesłanek w ciągu jednej sekundy. Zagadnieniem szybkości działania nie będziemy się jednak tutaj zajmowali i dlatego nie musimy znać realizacji technicznej elementów maszyny. Będzie nas natomiast interesować logiczna struktura maszyny. Dlatego możemy przyjąć szereg założeń upraszczających, nie wnikając w szczegóły techniczne.

§ 2. Magazyn liniowy

Jak wspomnieliśmy w poprzednim paragrafie, magazyn maszyny jest skomplikowanym szybko działającym urządzeniem elektronowym. Jednakże dla naszych celów wystarczy, jeżeli przyjmiemy, że magazyn to pokratkowana taśma papieru. Każdą kratkę będziemy nazywali **miejscem** magazynu.

W każdym miejscu magazynu może być zapisane jedno wyrażenie badanej teorii.

Miejsca magazynu będziemy oznaczali $m_1, m_2, \ldots, m_k$.

Określimy teraz działanie magazynu liniowego. W magazynie liniowym mogą być wykonywane następujące operacje:

1. Zapis,
2. odczyt,
3. przygotowanie miejsca do zapisu,

4. przygotowanie miejsca do odczytu,
5. wymazanie zawartości miejsca w magazynie.

W magazynie liniowym w każdej chwili może być przygotowane tylko jedno miejsce do odczytu oraz jedno miejsce do zapisu.

W magazynie liniowym można zapisać tylko w miejscu przygotowanym do zapisu.

W magazynie liniowym można odczytać tylko komórkę przygotowaną do odczytu.

Przy zapisywaniu zawartość poprzednia miejsca jest wymazywana. Niech m_i oznacza miejsce przygotowane do odczytu oraz m_j — miejsce przygotowane do zapisu.

Do opisu działania magazynu liniowego wygodnie jest przyjąć następujące oznaczenia:

$(\vec{M})$ — Odczytaj zawartość miejsca m_i oraz przygotuj do odczytu miejsce m_{i+1}.

$[\vec{M}]$ — Zapisz w miejscu m_j oraz przygotuj do zapisu miejsce m_{j+1}[1]).

Wtedy magazyn liniowy możemy określić jako magazyn w którym są wykonywane operacje $(\vec{M})$ i $[\vec{M}]$.

A więc w magazynie liniowym wpisywanie odbywa się w kolejnych miejscach magazynu $m_1, m_2, \ldots, m_k$ i odczytywanie magazynu odbywa się w tej samej kolejności co zapis, tj. odczytywane są kolejno miejsca $m_1, m_2, \ldots, m_k$.

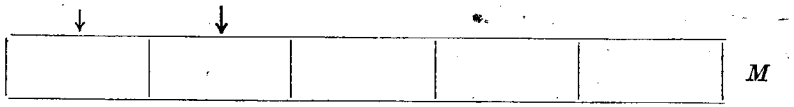
Magazyn liniowy możemy sobie wyobrazić również jako plik kartek papieru. Każda kartka jest miejscem magazynu. W każdym miejscu może być zapisane jedno wyrażenie teorii. Wszystkie zapisywane kartki są układane kolejno jedna na drugiej. Odczytywanie natomiast następuje od kartki położonej najniżej i polega na czytaniu kolejnych kartek od dołu począwszy.

Magazyn liniowy można by przyrównać również do kolejki przed sklepem. Osoby stojące w kolejce są obsługiwane przez ekspedientkę w takim samym porządku, w jakim przybyły do sklepu. Nie jest tutaj ważne, w jakim miejscu stoi dowolna osoba, lecz przed kim i za kim.

---

[1]) Przyjmujemy, że operacje zapisu i odczytu $(\vec{M})$, $[\vec{M}]$ nie mogą być w magazynie wykonywane jednocześnie, lecz są wykonywane kolejno.

Magazyn liniowy będziemy w dalszym ciągu rysować w postaci



zaznaczając miejsce przygotowane do odczytu za pomocą strzałki ↓, a miejsce przygotowane do zapisu — za pomocą strzałki ↓

§ 3. Cykl pracy maszyny

Maszyna realizując proces wnioskowania wykonuje cyklicznie szereg powtarzających się czynności. Dlatego dla zrozumienia działania maszyny wystarczy podać nie wszystkie czynności, a tylko te, które są związane z wykonaniem jednego kroku dowodu, tj. z realizowaniem jednej operacji redukowania. Dla każdej innej operacji czynności maszyny są identyczne. Zbiór wszystkich czynności maszyny, potrzebny do zrealizowania jednej operacji redukowania, będziemy nazywali **cyklem pracy maszyny.**

Będziemy zawsze rozpatrywać działanie maszyny od momentu, w którym badana hipoteza została umieszczona w pierwszym miejscu magazynu przesłanek M. Przyjmiemy ponadto, że aksjomaty są również już umieszczone w magazynie aksjomatów A.

Zakładamy ponadto, że przed rozpoczęciem działania maszyny pierwsze miejsce m_1 magazynu przesłanek M przygotowane jest do odczytu oraz drugie miejsce m_2 magazynu M przygotowane jest do zapisu.

Działaniem operatora O chwilowo się nie zajmujemy. Zakładamy tylko, że operator ten chwilowo, w bliżej nam nie znany sposób, oblicza przesłanki zadanej hipotezy, podobnie jak np. arytmometr oblicza wynik działania arytmetycznego.

Cykl pracy przedstawia się teraz następująco:

1. Wydrukuj na wyjściu hipotezę z miejsca przygotowanego do odczytu.
2. Odczytaj hipotezę z miejsca przygotowanego do odczytu i prześlij do operatora O.
3. Przygotuj następne miejsce do odczytu.

4. Sprawdź czy odczytana hipoteza jest atomem. Jeżeli TAK, to wykonaj czynność nr 5.
   Jeżeli NIE, to wykonaj czynność nr 6.
5. Sprawdź czy atom jest aksjomatem.
   Jeżeli TAK, to wykonaj czynność nr 15.
   Jeżeli NIE, wykonaj czynność nr 13.
6. Oblicz pierwszą przesłankę.
7. Zapisz pierwszą przesłankę w miejscu przygotowanym do zapisu.
8. Przygotuj następne miejsce do zapisu.
9. Oblicz drugą przesłankę.
10. Zapisz drugą przesłankę w miejscu przygotowanym do zapisu.
11. Przygotuj następne miejsce do zapisu.
12. Wykonaj czynność nr 1.
13. Wydrukuj na wyjściu maszyny: HIPOTEZA POCZĄTKOWA FAŁSZYWA.
14. Wykonaj czynność nr 17.
15. Sprawdź czy $i = j$.
    Jeżeli NIE, to wykonaj czynność nr 1.
    Jeżeli TAK, to wykonaj czynność nr 16.
16. Wydrukuj na wyjściu maszyny objaśnienie: HIPOTEZA POCZĄTKOWA PRAWDZIWA.
17. Zakończ wnioskowanie.

Cykl pracy składa się z 17 czynności. Wykonywaniem tych czynności rządzi sterowanie maszyny.

W każdym cyklu pracy maszyna najpierw drukuje badaną hipotezę na wyjściu maszyny, tj. na arkuszu elektrycznej maszyny do pisania. W ten sposób po zakończeniu pracy — o ile hipoteza wyjściowa okazała się prawdziwa — mamy wydrukowany na wyjściu cały dowód.

Następnie maszyna sprawdza, czy badana hipoteza jest atomem.

Jeżeli nie jest atomem, obliczane są obie przesłanki tej hipotezy i zapisywane w magazynie M, po czym analizowana jest następna hipoteza w magazynie M.

Jeżeli badana hipoteza jest atomem, maszyna sprawdza czy atom ten jest aksjomatem. Jeżeli atom nie jest aksjomatem, to znaczy,

że hipoteza początkowa jest fałszywa. Maszyna drukuje więc odpowiednie objaśnienie i na tym kończy się działanie.

Jeżeli atom jest aksjomatem, maszyna sprawdza, czy wnioskowanie jest już zakończone. Łatwo sprawdzić, że wnioskowanie jest zakończone wtedy i tylko wtedy, jeżeli $i = j$, tzn. miejsce przygotowane do odczytu i miejsce przygotowane do zapisu są tymi samymi miejscami magazynu M. Maszyna nie ma wtedy więcej hipotez do sprawdzania i w trakcie wnioskowania nie znalazła żadnego atomu, który by nie był aksjomatem. A więc hipoteza początkowa jest prawdziwa. Maszyna drukuje więc objaśnienie: HIPOTEZA POCZĄTKOWA PRAWDZIWA i kończy działanie.

Tak więc, jeżeli hipoteza okazała się prawdziwa, na wyjściu maszyny mamy wydrukowany dowód oraz na końcu dowodu uwagę HIPOTEZA PRAWDZIWA [1]).

Jeżeli hipoteza była fałszywa, maszyna przy pierwszym napotkanym atomie, który nie jest aksjomatem, drukuje objaśnienie: HIPOTEZA FAŁSZYWA i przerywa działanie.

Opis cyklu pracy wygodnie jest przedstawić w postaci rysunku, jak to pokazano w tablicy I.

W poszczególnych kratkach napisane są czynności wykonywane przez maszynę oraz numery tych czynności. Czynności napisane jedna pod drugą wykonywane są kolejno.

W przypadku czynności sprawdzających, gdzie wynik może być dwojaki: TAK lub NIE, narysowane są rozgałęzienia i zależnie od odpowiedzi wykonywane są następnie czynności wskazane przez strzałki.

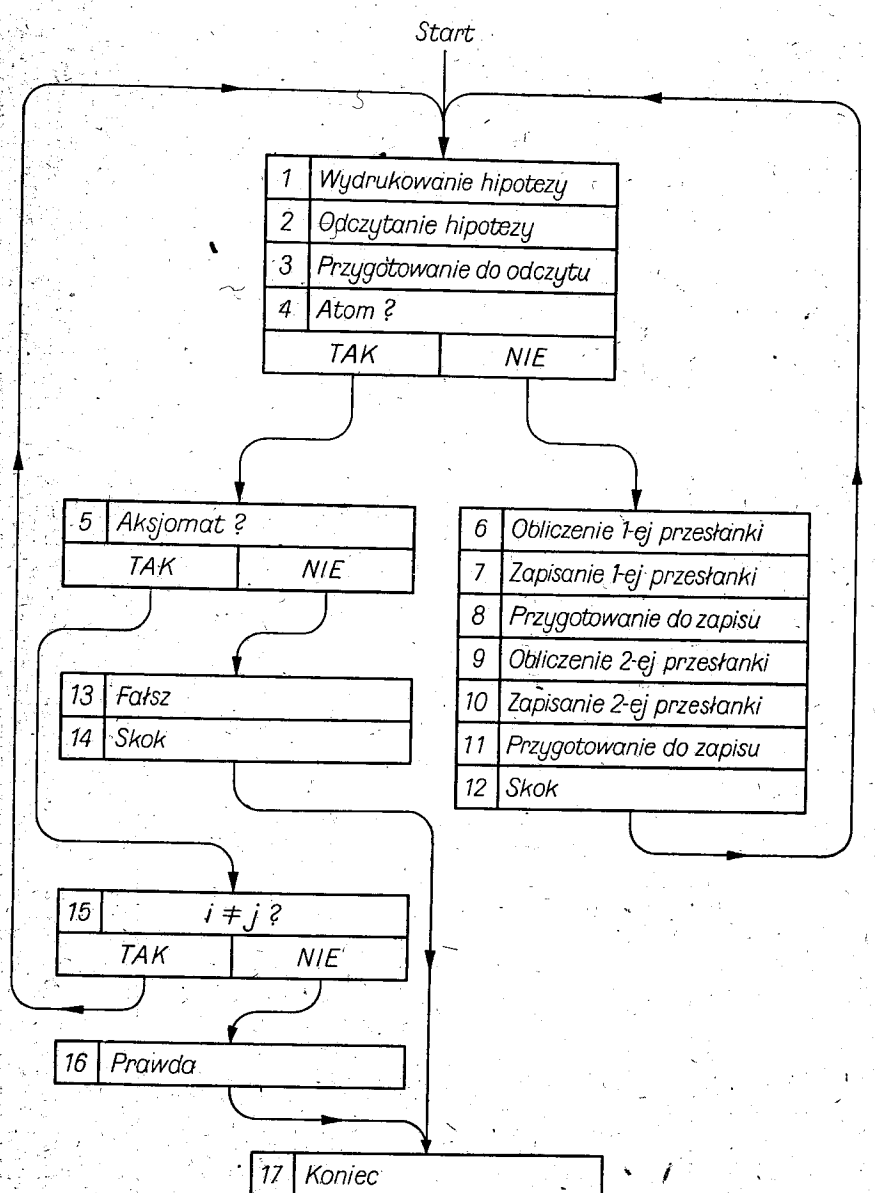
Taki sposób opisu pracy maszyny pozwala na szybkie zorientowanie się w całej strukturze działania maszyny.

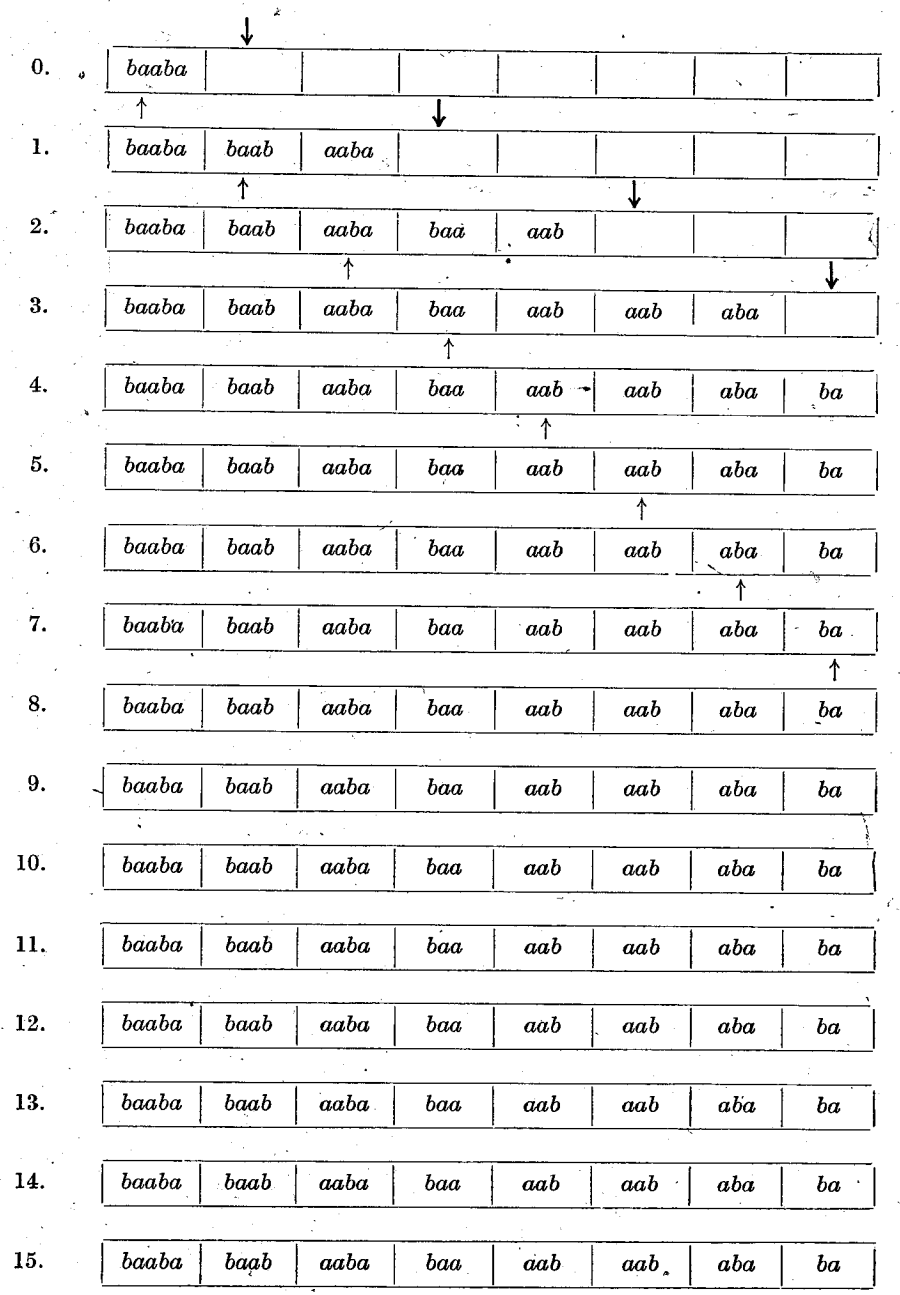
Podany schemat działania maszyny nie zależy od teorii, której twierdzeń dowodzimy. Dla różnych teorii różnie działa tylko operator O, obliczający przesłanki oraz fragment maszyny, w którym następuje badanie, czy atomy są aksjomatami. Ponieważ działania tych obu fragmentów nie rozpatrywaliśmy, podany schemat jest całkowicie ogólny.

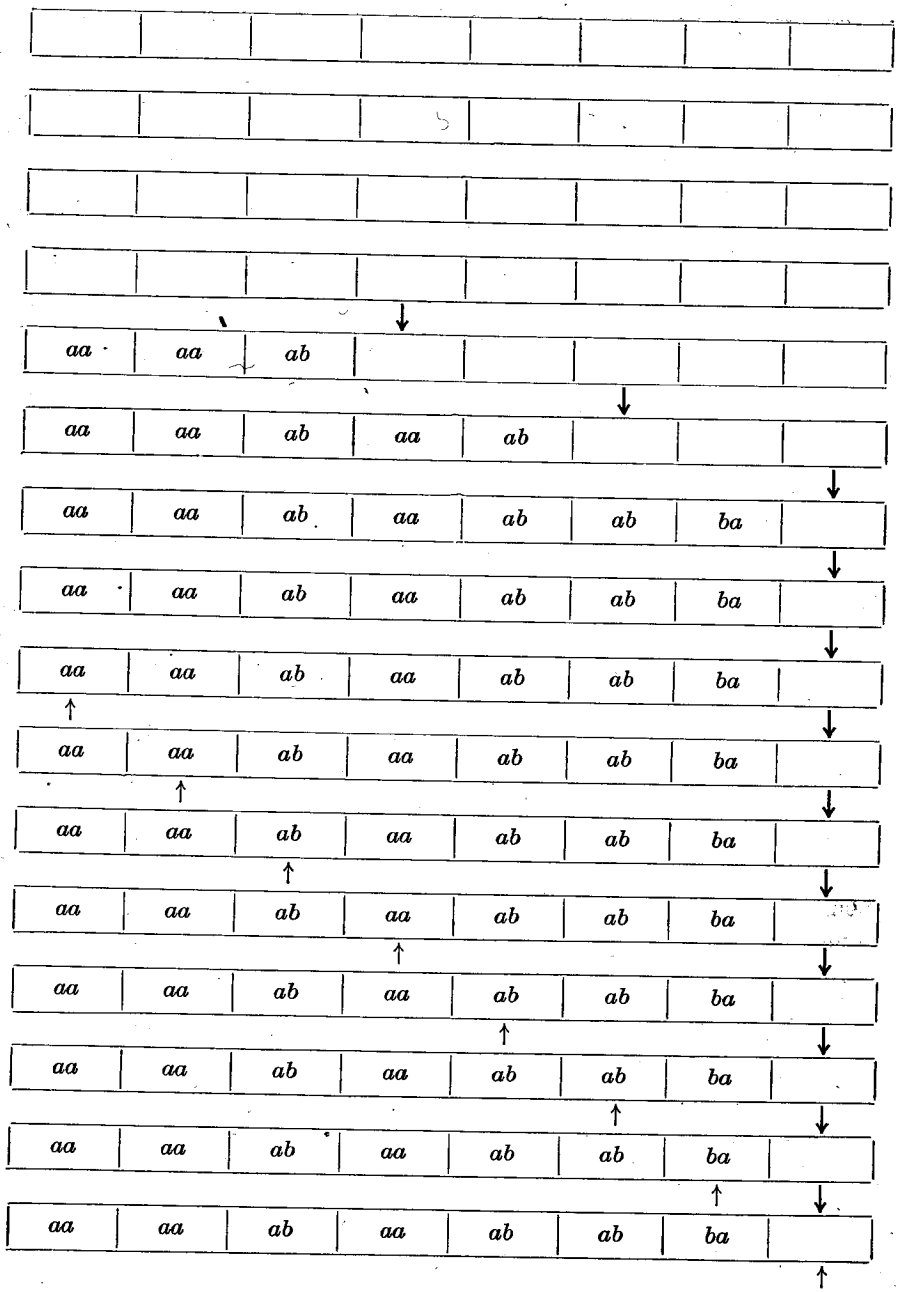
---

[1]) Przyjęliśmy dla uproszczenia, że na wyjściu nie są drukowane objaśnienia informujące o tym, za pomocą jakiej reguły i której hipotezy została obliczona dana przesłanka.

TABLICA I




0. baaba
1. baaba baab aaba
2. baaba baab aaba baa aab
3. baaba baab aaba baa aab aab aba
4. baaba baab aaba baa aab aab aba ba
5. baaba baab aaba baa aab aab aba ba
6. baaba baab aaba baa aab aab aba ba
7. baaba baab aaba baa aab aab aba ba
8. baaba baab aaba baa aab aab aba ba
9. baaba baab aaba baa aab aab aba ba
10. baaba baab aaba baa aab aab aba ba
11. baaba baab aaba baa aab aab aba ba
12. baaba baab aaba baa aab aab aba ba
13. baaba baab aaba baa aab aab aba ba
14. baaba baab aaba baa aab aab aba ba
15. baaba baab aaba baa aab aab aba ba


aa aa ab
aa aa ab aa ab
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba
aa aa ab aa ab ab ba

§ 4. Przykład szukania dowodu

Dla ilustracji rozpatrzymy kolejne etapy pracy maszyny przy szukaniu dowodu twierdzenia *baaba* w teorii $\mathbf{T}_1'$.

Będziemy podawali tylko stany magazynu M dla kolejnych cyklów pracy po ich wykonaniu, zaznaczając strzałkami, które miejsca magazynu są przygotowane do zapisu i odczytu.

W podanej tablicy (s. 48—49) możemy zobaczyć, że przesłanki są zapisywane w kolejnych miejscach magazynu i że pobieranie hipotez następuje w kolejności ich otrzymywania.

Dla innych, bardziej skomplikowanych teorii cykl pracy jest identyczny.

§ 5. Obliczanie przesłanek

O ile ogólny schemat maszyny nie zależał od konkretnej teorii matematycznej, to samo obliczanie przesłanek jest ściśle związane z rozważaną teorią. Dla różnych bowiem teorii stosowane są różne reguły wnioskowania redukcyjnego. Operator O musi właśnie te reguły realizować. Dlatego ograniczymy się do rozważania działania operatora dla teorii $\mathbf{T}_1'$, tj. omówimy realizowanie jedynej reguły redukowania $\mathbf{R}_1$, obowiązującej w tej teorii.

Nie będziemy szczegółowo analizowali działania operatora, a podamy tylko zasady ogólne.

Przyjmiemy, że w skład operatora wchodzą dwa magazyny, które będziemy nazywali **rejestrami.** Jeden rejestr nazwiemy **rejestrem hipotezy** i oznaczymy literą H, drugi zaś rejestr nazwiemy **rejestrem przesłanek** i oznaczymy literą P.

Możemy przyjąć, że oba rejestry są również pokratkowanymi taśmami papieru. Poszczególne kratki rejestrów będziemy nazywali **pozycjami.**

W każdej pozycji rejestru może być zapisany jeden symbol języka teorii [1]).

[1]) Pozycje odgrywają w rejestrze podobną rolę, jak miejsca w magazynie. Różnica polega na tym, że w miejscu magazynowane jest wyrażenie, natomiast w pozycji — jeden symbol.

W rejestrach mogą być wykonywane czynności podobne jak w magazynie, tj.

1. Odczytywanie,
2. zapisywanie,
3. wymazywanie,
4. przygotowanie do zapisu,
5. przygotowanie do odczytu.

Czynności te dotyczą obecnie nie wyrażeń a pojedynczych symboli.

Przesłanie hipotezy z magazynu M do operatora O polega na przepisaniu jej z przygotowanego do odczytu miejsca magazynu M do rejestru H w ten sposób, że pierwszy symbol hipotezy znajduje się w pierwszej pozycji, drugi w drugiej itd.[1]).

Przyjmiemy ponadto, że operator może wykonywać następujące operacje:

1. Przepisanie symbolu z pozycji przygotowanej do odczytu w rejestrze H do pozycji przygotowanej do zapisu w rejestrze P.
2. Sprawdzenie, czy w pozycji przygotowanej do odczytu w rejestrze H jest zapisany jakiś symbol.

Obliczenie pierwszej przesłanki będzie przebiegało następująco:

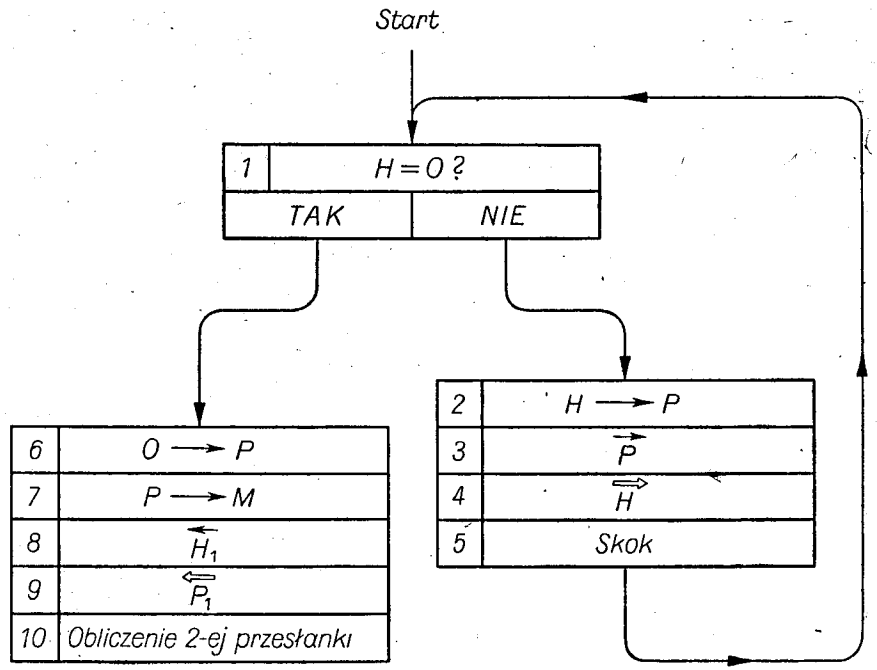
1. Sprawdź czy pozycja przygotowana do odczytu jest pusta. Jeżeli TAK, to wykonaj czynność nr 6. Jeżeli NIE, to wykonaj czynność nr 2.
2. Przepisz symbol z pozycji przygotowanej do odczytu w rejestrze H, do pozycji przygotowanej do zapisu w rejestrze P.
3. Przygotuj następne miejsce do odczytu w rejestrze H.
4. Przygotuj następne miejsce do zapisu w rejestrze P.
5. Wykonaj czynność nr 1.
6. Wymaż ostatni wpisany symbol w rejestrze P.
7. Przepisz przesłankę z rejestru P do magazynu M na miejsce przygotowane do zapisu.
8. Przygotuj do odczytu pierwszą pozycję w rejestrze H.
9. Przygotuj do zapisu pierwszą pozycję w rejestrze P.
10. Oblicz drugą przesłankę.

Wykres tych czynności przedstawiony jest w tablicy II.

Przyjęliśmy milcząco, że przed obliczeniem pierwszej przesłanki

[1]) Oczywiście zakładamy, że rejestr posiada dostateczną ilość pozycji, aby zapisać dowolne wyrażenie występujące w czasie wnioskowania.

TABLICA II



→ – oznacza przepisanie
$\overleftarrow{H_1}$ – przygotowanie do odczytu pierwszej pozycji rejestru H
$\Leftarrow H_{p_1}$ – przygotowanie do zapisu pierwszej pozycji rejestru P

w rejestrach H i P były przygotowane do zapisu i do odczytu pierwsze pozycje.

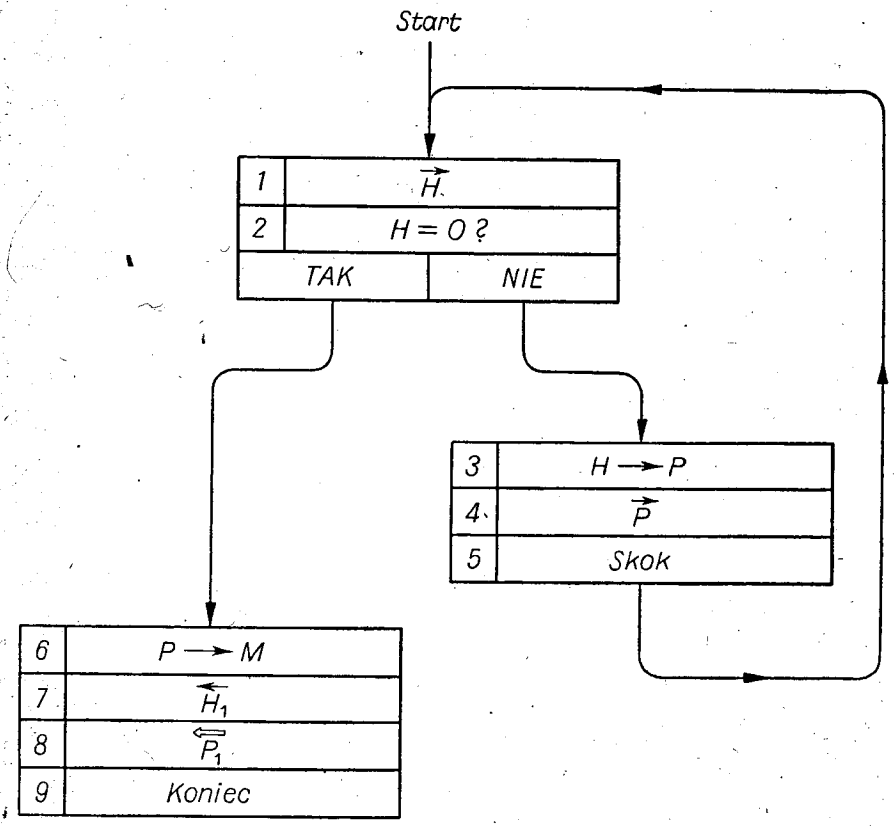
Sam przebieg obliczenia przesłanki jest tutaj bardzo prosty i nie będziemy go bliżej omawiać.

Obliczenie drugiej przesłanki przebiega podobnie, z tą jedynie różnicą, że pierwszy symbol z rejestru H jest omijany, a przepisywanie do rejestru P zaczynamy od drugiego symbolu.

Przebieg obliczenia drugiej przesłanki jest więc następujący:

1. Przygotuj do odczytu następną pozycję w rejestrze H.
2. Sprawdź czy pozycja przygotowana do odczytu w rejestrze H

TABLICA III



jest pusta. Jeżeli TAK, to wykonaj czynność nr 6. Jeżeli NIE, to wykonaj czynność nr 3.

3. Przepisz symbol z pozycji przygotowanej do odczytu w rejestrze H, do pozycji przygotowanej do zapisu w rejestrze P.
4. Przygotuj do zapisu następne pozycje w rejestrze P.
5. Wykonaj czynność nr 1.
6. Przepisz przesłankę z rejestru P do miejsca magazynu M, przygotowanego do zapisu.
7. Przygotuj do odczytu pierwszą pozycję rejestru H.
8. Przygotuj do zapisu pierwszą pozycję w rejestrze P.
9. Koniec obliczania przesłanek.

Przebieg tych czynności pokazany jest graficznie w tablicy III. Ponieważ pierwszą czynnością przy obliczaniu drugiej przesłanki było przygotowanie następnego miejsca do odczytu w rejestrze H, więc pierwszy symbol zapisany w rejestrze H nie został przepisany do rejestru P, a więc zgodnie z regułą $\mathbf{R}_1$ w rejestrze P znajdzie się, po wykonaniu tych czynności, druga przesłanka.

W podobny sposób można opisać dowolne manipulacje na symbolach. A więc dla teorii, gdzie obliczanie przesłanek jest bardziej skomplikowane, operator będzie wykazywał więcej różnorodnych czynności, jednakże istota jego działania będzie podobna do pokazanego na tym przykładzie. Operator jest jak gdyby drugą mniejszą maszyną, która w istocie działa dość podobnie do ogólnego schematu maszyny, gdzie również zachodzi przepisywanie symboli, porównywanie itp.

§ 6. Badanie atomów

Do omówienia pozostało jeszcze, w jaki sposób maszyna sprawdza czy analizowany atom jest aksjomatem.

Przyjmijmy, że wszystkie aksjomaty badanej aktualnie teorii znajdują się w specjalnym magazynie aksjomatów A. Sprawdzanie, czy atom jest aksjomatem można by wykonać przez porównanie atomu, symbol po symbolu, z każdym aksjomatem po kolei. Takie rozwiązanie byłoby dość niewygodne i nie będziemy go tutaj bliżej omawiać. Dla ćwiczenia proponujemy Czytelnikowi rozwiązać to zadanie samodzielnie.

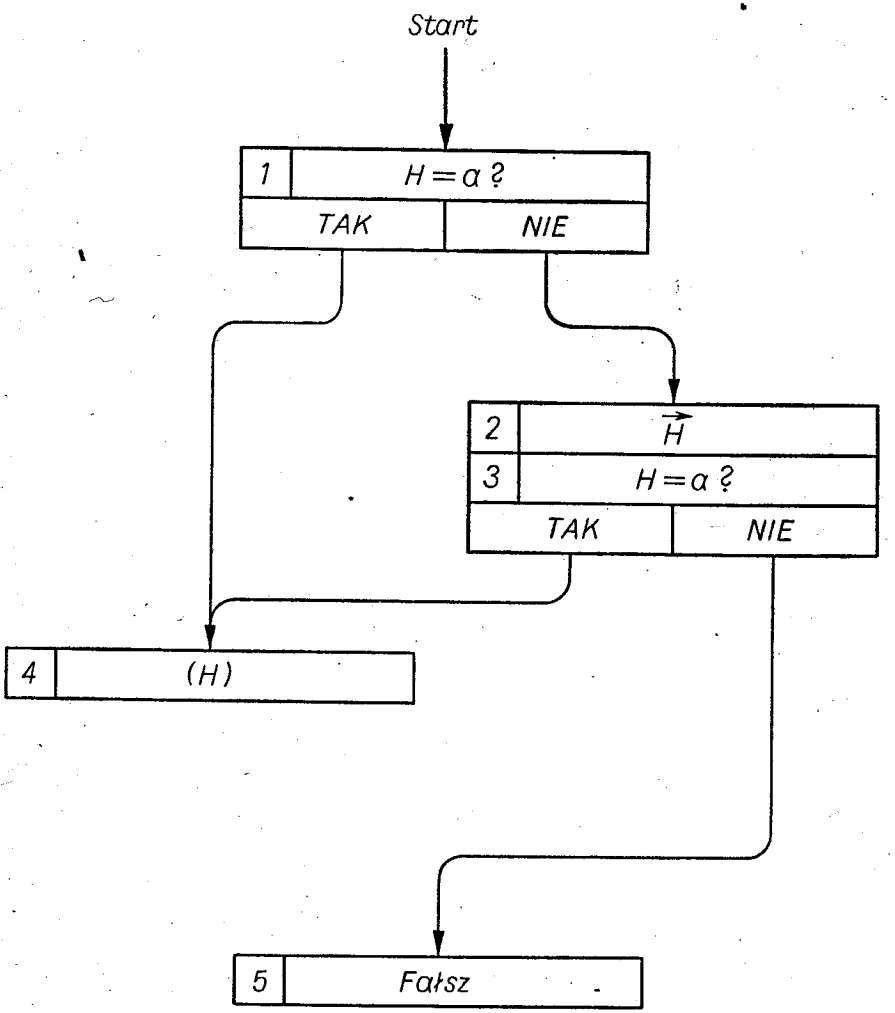
Najwygodniej byłoby dla nas, abyśmy o wyrażeniu mogli powiedzieć, czy jest aksjomatem, na podstawie badania struktury tego wyrażenia. W rozpatrywanej przez nas przykładowo teorii, takie badanie jest możliwe. Mianowicie każde wyrażenie, składające się z dwu symboli a i b i nie będące wyrażeniem bb, jest aksjomatem.

Załóżymy, że badane wyrażenie zostało przepisane z magazynu do rejestru H i że maszyna sprawdziła, że jest ono atomem. Wobec tego wiemy, że składa się ono z dwu symboli.

Sprawdzenie, czy atom jest aksjomatem będzie więc przebiegać następująco:

1. Sprawdź, czy symbol zapisany w przygotowanej do odczytu pozycji rejestru H jest literą a.

TABLICA IV



Jeżeli TAK, to wykonaj czynność nr 4.
Jeżeli NIE, to wykonaj czynność nr 2.

2. Przygotuj do odczytu następną pozycję rejestru H.
3. Sprawdź, czy symbol zapisany w pozycji przygotowanej do odczytu rejestru H jest literą a.

Jeżeli TAK, to wykonaj czynność nr 4.
Jeżeli NIE, to wykonaj czynność nr 5.
4. Odczytaj z magazynu następną hipotezę.
5. Wydrukuj: HIPOTEZA POCZĄTKOWA FAŁSZYWA.

Wykres tych czynności podano w tablicy IV.

W opisie działania maszyny podawaliśmy bardzo szczegółowe opisy czynności, które maszyna powinna wykonywać dla zrealizowania określonego zadania. Cechą charakterystyczną tych opisów było to, że operowały one następującymi elementarnymi czynnościami:

1. Odczytywanie symboli,
2. zapisywanie symboli,
3. przygotowanie odczytu,
4. przygotowanie zapisu,
5. odróżnianie symboli,
6. wybieranie jednej z dwu możliwych dróg postępowania.

Wymienione wyżej czynności pozwalają na opisanie znacznie bardziej skomplikowanych operacji na symbolach niż te, które przygotowaliśmy tutaj. Mając zaś opis wszystkich standardowych czynności elementarnych, które ma wykonać maszyna, nietrudno już opracować na tej podstawie szczegółowy schemat maszyny. Zagadnieniem tym nie będziemy się jednak zajmowali i przyjmiemy, że schemat na rys. 14 plus opis działania, dostatecznie wyjaśniają przebieg pracy maszyny przy dowodzeniu twierdzeń.

Rozdział V

REALIZACJA WNIOSKOWANIA Z PORZĄDKIEM $\overline{W}$ ZA POMOCĄ MASZYNY BEZADRESOWEJ

§ 1. Ogólny schemat maszyny

Jeżeli przyjmiemy, że proces wnioskowania redukcyjnego jest wykonywany w porządku wzdłużnym, to ogólny schemat maszyny nie ulega zmianie. Maszyna taka zawiera identyczne elementy jak maszyna opisana w poprzednim rozdziale. Inaczej działa tylko magazyn M, który dla porządku $\overline{W}$ jest tzw. **magazynem rewersyjnym.** Natomiast pozostałe elementy w obu maszynach są identyczne.

§ 2. Magazyn rewersyjny

Magazyn rewersyjny możemy uważać również za pokratkowaną taśmę papierową. Jednakże sposób zapisywania i odczytywania jest tutaj nieco inny, niż w magazynie liniowym. Magazyn rewersyjny realizuje następujące operacje:

1. Zapis,
2. odczyt,
3. przygotowanie (do odczytu lub zapisu),
4. wymazywanie zawartości miejsca w magazynie.

W magazynie rewersyjnym jest tylko jeden rodzaj przygotowania, niezależnie od tego, czy ma być w magazynie wykonana operacja odczytu, czy też zapisu. Przygotowanie miejsca następnego oznaczymy przez $\rightarrow$, przygotowanie miejsca poprzedniego oznaczymy przez $\leftarrow$. Ponadto wprowadzimy oznaczenie:

$\left[\overrightarrow{M}\right]$ — Przygotuj do zapisu miejsce m_{j+1} oraz zapisz w miejscu m_{j+1}.

Znaczy to, że jeżeli w magazynie jest aktualnie przygotowane miejsce m_j, to w wyniku operacji $\left[\vec{M}\right]$ przygotowane jest w magazynie następne miejsce m_{j+1} i w miejscu tym jest zapisany jakiś symbol. Warto zwrócić uwagę na różnicę między operacją $[\vec{M}]$ a $\left[\vec{M}\right]$. Operacja $[\vec{M}]$ powoduje zapisanie w miejscu przygotowanym i po wykonaniu zapisu przygotowanie następnego miejsca, natomiast w wyniku operacji $\left[\vec{M}\right]$ czynności przygotowania i zapisu są wykonywane w odwrotnym porządku niż poprzednio, tj. najpierw przygotowane jest następne miejsce i dopiero wtedy następuje zapis w magazynie.

Wyrażenie $(\overleftarrow{M})$ będzie oznaczało operacje odczytania miejsca przygotowanego m_j i po odczytaniu przygotowanie miejsca m_{j-1}.

Magazyn rewersyjny jest to taki magazyn, w którym są realizowane operacje $(\overleftarrow{M})$ oraz $\left[\vec{M}\right]$ [1]).

Magazyn rewersyjny działa więc następująco:

1. Najpierw zapisywane jest miejsce m_1.
2. Następnie zapisywane jest każde kolejne miejsce.
3. Zawsze następuje odczytywanie ostatniego zapisanego miejsca.
4. Po odczytaniu zawartość miejsca jest wymazywana.

Działanie magazynu rewersyjnego można też wyjaśnić na następującym przykładzie: załóżmy, że miejscami magazynu są oddzielne kartki. Wtedy zapisywanie do magazynu polegało na położeniu zapisanej kartki na stosie kart, odczytanie zaś na pobraniu kartki z wierzchu stosu i jej usunięciu. A więc czynności są tu w pewnym sensie odwrotne do czynności magazynu liniowego.

Przykładem magazynu rewersyjnego jest stóg siana. Warstwy siana są układane jedna na drugiej i pobierane w kolejności odwrotnej do ich układania.

[1]) Jednoczesne wykonywanie operacji $(\overleftarrow{M})$ oraz $\left[\vec{M}\right]$ jest niedozwolone. W magazynie może być wykonana jedna, bądź druga operacja.

§ 3. Przykład stanów magazynu rewersyjnego w trakcie szukania dowodu

W rozdziale IV, § 4 podaliśmy kolejne stany magazynu M przy realizowaniu procesu wnioskowania w porządku $\overline{P}$. Obecnie rozpatrzymy ten sam przykład dowodu dla porządku $\overline{W}$.

Cykl pracy maszyny z magazynem rewersyjnym jest identyczny, jak i w maszynie z magazynem liniowym, z tą jedynie różnicą, że badanie końca procesu dowodzenia odbywa się teraz w nieco inny sposób niż poprzednio. Łatwo sprawdzić, że proces wnioskowania jest zakończony, jeżeli $i = 0$, gdzie i jest miejscem przygotowanym w magazynie M.

Mówiąc inaczej, jeżeli zawartość całego magazynu została wymazana, proces wnioskowania jest zakończony.

Kolejne stany magazynu M podane są niżej:

0.	*baaba*			
1.	*baab*	*aaba*		
2.	*baab*	*aab*	*aba*	
3.	*baab*	*aab*	*ab*	*ba*
4.	*baab*	*aab*	*ab*	
5.	*baab*	*aa*	*ab*	
6.	*baab*	*aa*		
7.	*baab*			
8.	*baa*	*aab*		
9.	*baa*	*aa*	*ab*	

10.	*baa*	*aa*		
11.	*baa*			
12.	*ba*	*aa*		
13.	*ba*			
14.				

W wierszu nr 1 podano stan magazynu po wykonaniu operacji nr 1 przy numeracji $\overline{W}$.

Zgodnie z zasadą działania magazynu rewersyjnego, przy odczytywaniu aksjomatów z magazynu M następuje wymazanie aksjomatu z magazynu, bez wpisywania jego przesłanek [1]).

Jak widzimy w podanym przykładzie, magazyn rewersyjny wymaga mniejszej ilości miejsc dla rozwiązania danego problemu, niż magazyn liniowy.

Oczywiście zamiast wpisywać aksjomaty do magazynu M oraz przy ich odczytywaniu je wymazywać, można by od razu przy obliczeniu nie wpisywać ich do magazynu, unikając w ten sposób zbędnych czynności. Jednakże dla przejrzystości przyjęto, że każdy aksjomat jest również wpisywany do magazynu M.

Maszyny, o których była mowa w rozdziałach IV i V nie istnieją. Obecnie nie opłacałoby się bowiem do celów dowodzenia budować specjalne aparaty. Prace nad maszynowym dowodzeniem twierdzeń są bowiem dopiero w stadium początkowym i nie wiadomo na pewno czy znajdą zastosowanie i w jakim ewentualnie zakresie.

Dlatego dotychczasowe próby dowodzenia twierdzeń były robione nie za pomocą specjalnie w tym celu zbudowanych urządzeń, lecz za pomocą istniejących maszyn do celów rachunkowych.

W dalszym ciągu zajmiemy się zastosowaniem uniwersalnych maszyn cyfrowych do dowodzenia twierdzeń. Omówione dwa przykłady maszyn pozwolą nam lepiej zrozumieć dalsze rozważania.

[1]) Aksjomat można uważać za wniosek z pustego zbioru przesłanek.

Rozdział VI

ADRESOWA MASZYNA CYFROWA

§ 1. Schemat ogólny maszyny

W rozdziale tym podamy zasadę działania adresowej maszyny cyfrowej, zwanej dalej krótko maszyną cyfrową, aby w następnym rozdziale wyjaśnić zastosowanie maszyn cyfrowych do dowodzenia twierdzeń. Omawiając zasadę działania maszyny cyfrowej, nie będziemy — podobnie jak poprzednio — wchodzili w szczegóły techniczne, a postaramy się podać ogólną myśl przewodnią jej konstrukcji. Maszyny cyfrowe są powszechnie stosowanymi urządzeniami do szybkiego wykonywania obliczeń. Jednakże za ich pomocą można również, jak się niebawem przekonamy, dokonywać różnorodnych manipulacji na symbolach.

Maszyna cyfrowa składa się z następujących elementów:

1. Operator,
2. magazyn wewnętrzny,
3. wejście,
4. wyjście,
5. sterowanie.

Rola operatora jest podobna jak w maszynach bezadresowych: wykonuje on operacje przekształcania symboli. W szczególności operator wykonuje również cztery podstawowe działania arytmetyczne: dodawanie, odejmowanie, mnożenie i dzielenie. Działaniem operatora nie będziemy się bliżej zajmować, przyjmiemy tylko, że wykonuje on wszystkie potrzebne operacje na symbolach.

Wejście i wyjście jest tutaj identyczne jak w maszynach bezadresowych. Wejście jest urządzeniem, gdzie zapisane są wszystkie potrzebne dane i mogą być one przez maszynę w miarę potrzeby automatycznie czytane.

Wyjście np. pozwala na drukowanie na papierze interesujących nas wyników. Możemy przyjąć, że wejściem jest sterowana elektrycznie maszyna do pisania [1]).

Rola sterowania jest identyczna jak w maszynie bezadresowej; rządzi ono całym przebiegiem pracy maszyn.

Inaczej jest natomiast zbudowany magazyn tej maszyny. Działa on na zasadzie adresowej. Pojęcie magazynu wewnętrznego omówimy w następnym paragrafie [2]).

§ 2. Magazyn wewnętrzny. Lista

Magazyn wewnętrzny jest podobnie jak i inne elementy maszyny cyfrowej szybko działającym urządzeniem elektronicznym.

Dla zrozumienia działania maszyny cyfrowej niepotrzebne jest zapoznawanie się szczegółowe z konstrukcją magazynu wewnętrznego, a wystarczy przyjąć, że magazyn adresowy to taśma papieru podzielona na kratki. Wszystkie kratki są ponumerowane kolejnymi liczbami naturalnymi 0, 1, ..., n. Poszczególne kratki będziemy nazywali **miejscami** magazynu. Numery miejsc nazwiemy ich **adresami.** Magazyn wewnętrzny będziemy przedstawiać w postaci **listy.**

0	
1	
2	
...	
n	

[1]) Wejście i wyjście maszyny można by uważać za magazyny liniowe. Wejście to taśma papieru, podzielona na wiersze i w każdym wierszu może być zapisany dowolny ciąg symboli ustalonego alfabetu. Maszyna cyfrowa może w miarę potrzeby odczytywać kolejne wiersze wejścia do magazynu wewnętrznego.

Wyjście również możemy uważać za magazyn liniowy. W kolejnych miejscach tego magazynu wpisywane są automatycznie kolejne interesujące nas wyniki.

[2]) Magazyn liniowy i rewersyjny są magazynami bezadresowymi. Stąd nazwa: maszyny bezadresowe.

Dlatego w dalszym ciągu zamiast magazyn wewnętrzny, będziemy też używać termin: lista.

W każdym miejscu listy może być zapisany ciąg symboli ustalonego alfabetu.

Miejsce o adresie zero na liście spełnia specjalną, wyróżnioną rolę, z którą zapoznamy się w dalszym ciągu. Miejsce to będziemy nazywali **akumulatorem.**

W magazynie wewnętrznym mogą być wykonywane następujące operacje:

1. Zapisanie,
2. odczytanie,
3. wymazanie.

Bliżej z operacjami tymi zapoznamy się w dalszym ciągu.

Wyrażenia znajdujące się w magazynie mogą być przepisywane, zmieniane, wymazywane. Czynności te są w maszynie cyfrowej opisywane za pomocą tzw. **instrukcji.**

§ 3. Instrukcje. Spis instrukcji

Maszyna cyfrowa może wykonywać szereg instrukcji, które opisują manipulacje na symbolach. Instrukcje maszyny są zapisywane w specjalnym języku. Każda instrukcja składa się z dwóch symboli:

1. Symbolu operacji,
2. adresu.

Np. $\rightarrow$ 5; strzałka $\rightarrow$ oznacza operację przepisywania. Liczba 5 jest adresem na liście. Instrukcja $\rightarrow$ 5 oznacza:

Przepisz zawartość akumulatora (miejsca 0) do miejsca o adresie 5.

Ponieważ w instrukcji występuje jeden adres, nazwiemy ją **instrukcją jednoadresową,** a maszynę, która realizuje instrukcje jednoadresowe — **maszyną jednoadresową** [1]).

Każda maszyna może wykonywać pewną liczbę z góry ustalanych instrukcji. Dla różnych maszyn ilość i rodzaj instrukcji są różne. Przyjmiemy tutaj, że rozważana maszyna wykonuje w zasadzie tylko te instrukcje, które będą nam potrzebne do dowodzenia twierdzeń.

[1]) Istnieją również maszyny wieloadresowe, jednakże nimi nie będziemy się zajmowali.

Spis instrukcji

Nr kolejny	Symbol instrukcji	Nazwa instrukcji
1	$+a$	Dodaj
2	$-a$	Odejmij
3	$\cdot a$	Pomnóż
4	$:a$	Podziel
5	$\rightarrow a$	Zapisz
6	$\leftarrow a$	Odczytaj
7	$!a$	Skocz
8	$?a$	Skocz warunkowo
9	$\rightarrow\!\!\rightarrow a$	Wprowadź
10	$\leftarrow\!\!\leftarrow a$	Wyprowadź
11	$0a$	Wyzeruj
12	Na	Następnik
13	Pa	Poprzednik
14	$=a$	Zero
15	stop	Stop

Instrukcje $\delta(a)$ będziemy nazywali **instrukcjami pośrednimi,** a instrukcje δa — **instrukcjami bezpośrednimi.** Podobnie nazwiemy adresy a i (a). Pierwszy z nich nazwiemy bezpośrednim, a drugi pośrednim.

Instrukcje pośrednie nie są konieczne do dalszych rozważań, wprowadzą one tylko szereg uproszczeń.

Przyjęliśmy, że rozważana przez nas maszyna może wykonać w sumie 27 różnych instrukcji. Każda czynność, którą będziemy chcieli wykonać za pomocą maszyny, musimy najpierw opisać za pomocą tych 27 instrukcji. Opis taki nazywamy **programem** procesu

realizowanego przez maszynę. Aby więc np. opisać proces dowodzenia twierdzeń przez maszynę adresową, musimy najpierw przebieg tego procesu opisać za pomocą przyjętych instrukcji. Zanim przystąpimy do rozwiązania tego zadania w następnym rozdziale, zapoznamy się najpierw dokładniej z działaniem maszyny adresowej.

§ 4. Cykl pracy maszyny adresowej

Przyjmiemy, że w stanie początkowym w magazynie wewnętrznym maszyny znajduje się program procesu, który ma być przez maszynę zrealizowany. Instrukcje programu są umieszczone w kolejnych miejscach magazynu, poczynając od miejsca o adresie m. W stanie początkowym miejsce m jest przygotowane do odczytu.

Cykl pracy maszyny jest wtedy następujący:

1. Odczytanie z magazynu wewnętrznego instrukcji zapisanej w miejscu przygotowanym do odczytu.
2. Wykonanie odczytanej instrukcji.
3. Przygotowanie w magazynie wewnętrznym następnego miejsca do odczytu [1]).
4. Sprawdzenie czy odczytana instrukcja jest instrukcją STOP.

Jeżeli TAK, to zakończyć działanie maszyny.

Jeżeli NIE, to wykonać czynność nr 1.

A więc w maszynie adresowej sterowanie na przemian odczytuje instrukcje z magazynu wewnętrznego i wykonuje odczytane instrukcje. Czynności te są powtarzane tak długo, aż zostanie odczytana instrukcja STOP.

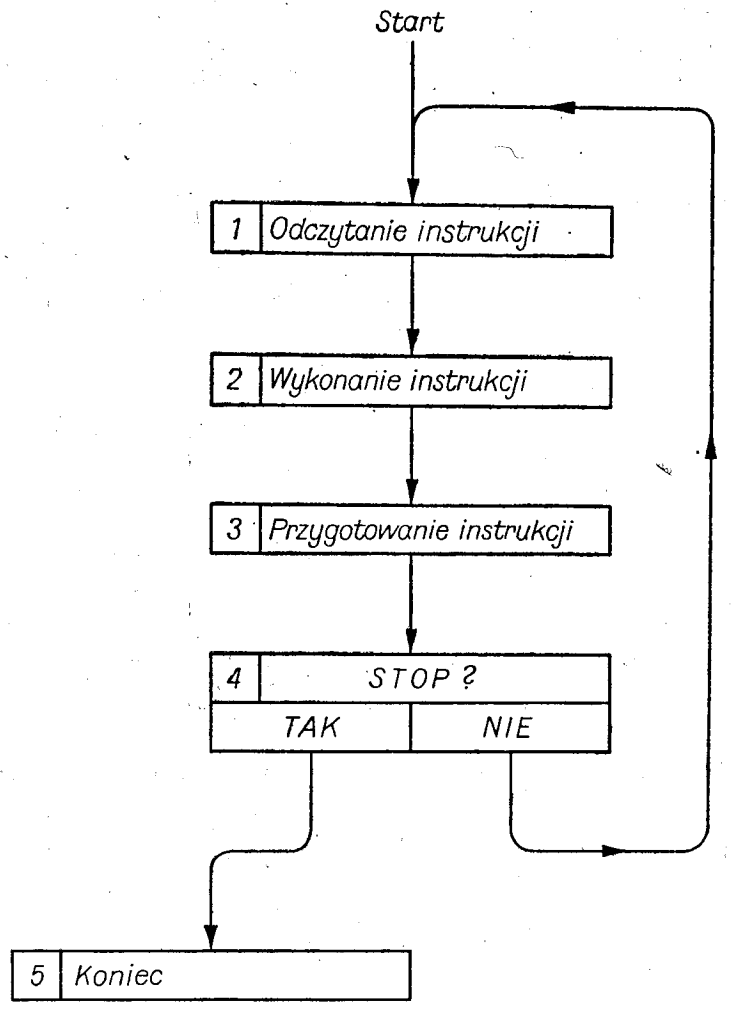
Wykres cyklu pracy podany jest w tablicy V.

Teraz możemy już przystąpić do omówienia zasad układania programów dla maszyny cyfrowej.

---

[1]) Przygotowanie następnego miejsca można zaliczyć do czynności nr 2, jak to czyniliśmy przy opisie instrukcji. Czynność tę tutaj wyróżniamy dla podkreślenia jej ważności.

TABLICA V



§ 5. Przykłady programów

Pokażemy obecnie jak za pomocą instrukcji należy opisywać przebieg wykonania przez maszynę różnych prostych czynności.

1. Obliczenia arytmetyczne

Zadanie. Obliczyć $m + n$ i wynik umieścić pod adresem 22. Pierwszą instrukcję programu umieścić pod adresem 10.

Założenia

$C0 = 0$
$C20 = m$
$C21 = n$

Program

10.	← 20	Odczytaj
11.	+ 21	Dodaj
12.	→ 22	Zapisz
13.	Stop	

Pierwsza instrukcja ← 20 przesyła liczbę m do akumulatora. Instrukcja + 21 dodaje liczbę n do liczby znajdującej się w akumulatorze. Instrukcja → 22 przesyła wynik pod adres 22.

Dla ilustracji wygodnie jest przedstawić zapis tego zadania w magazynie wewnętrznym maszyny.

	0	0
	.....	.....
→	10	→ 20
	11	+ 21
	12	← 22
	13	stop
	.....	.....
	20	m
	21	n
	22	
	.....	.....

Stan magazynu wewnętrznego podano przed wykonaniem programu.

Strzałka przy adresie 10 oznacza początek programu, tj. w stanie początkowym miejsce 10 jest przygotowane do odczytu.

Opisanie za pomocą instrukcji wykonania bardziej skomplikowanego obliczenia arytmetycznego jest podobne. Np.

Zadanie. Wykonać obliczenie

$$\frac{(a + b) \cdot c}{d - e}$$

Wynik obliczenia umieścić w miejscu 200. Pierwszą instrukcję programu umieścić w miejscu 10.

Założenie

$C0 = 0$
$C100 = a$
$C101 = b$
$C102 = c$
$C103 = d$
$C104 = e$

Program

10. ← 100 Odczytaj
11. + 101 Dodaj
12. · 102 Pomnóż
13. → 105 Zapisz
14. ← 103 Odczytaj
15. — 104 Odejmij
16. → 106 Zapisz
17. ← 105 Odczytaj
18. : 106 Podziel
19. → 200 Zapisz
20. Stop

Stan początkowy magazynu wewnętrznego dla tego rachunku podany jest na s. 73.

Odczytanie tego programu nie przedstawia trudności, nie będziemy go więc omawiali.

	0	0
	.....	.....
→	10	← 100
	11	+ 101
	12	· 102
	13	→ 105
	14	← 103
	15	— 104
	16	→ 106
	17	← 105
	18	: 106
	19	→ 200
	20	stop
	.....	.....
	100	a
	101	b
	102	c
	103	d
	104	e

2. Zastosowanie instrukcji warunkowej do obliczeń arytmetycznych

Zadanie. Obliczyć sumę $a_1 + \ldots + a_{100}$ i wynik umieścić w miejscu 400. Pierwszą instrukcję programu umieścić w miejscu 100.

Założenie

$C0 = 0$
$C10 = 99$
$C11 = 201$
$C201 = a_1$
$C202 = a_2$
$C300 = a_{100}$

Program. Bez użycia instrukcji warunkowej ułożenie programu nie sprawia trudności. Uczynilibyśmy to podobnie jak dla pokazanych poprzednio wzorów arytmetycznych. Program taki zawierałby 99 instrukcji. Posługując się instrukcją warunkową, można znacznie zmniejszyć ilość instrukcji w programie. Z użyciem instrukcji warunkowej program sumowania będzie miał postać.

100.	+ (11)	Dodaj
101.	*N* 11	Następnik
102.	*P* 10	Poprzednik
103.	= 10	Zero
104.	? 100	Skok warunkowy
105.	← 400	Zapisz
106.	Stop	

Pierwsza instrukcja powoduje dodanie do akumulatora liczby z miejsca o adresie podanym pod adresem 11. Pod adresem 11 mamy zapisaną liczbę 201, która jest adresem pierwszego składnika a_1. Ponieważ na początku w akumulatorze znajdowało się zero, więc w wyniku rozkazu + (11) w akumulatorze znajdzie się pierwszy składnik.

Instrukcja *N* 11 powoduje dodanie jedynki do liczby znajdującej się w miejscu 11. Po wykonaniu tego rozkazu znajdzie się więc w miejscu 11 liczba 202.

Instrukcja *P* 10 powoduje odjęcie od liczby znajdującej się w miejscu 10 jedynki. W miejscu 10 znajduje się liczba 99 — liczba wszystkich dodawań, które należy wykonać, aby otrzymać sumę stu liczb. Po wykonaniu rozkazu, w miejscu 10 mamy więc liczbę 98, co oznacza, że jedno dodawanie zostało już wykonane.

Następna instrukcja = 10 bada, czy wszystkie dodawania zostały już wykonane, tzn. czy w miejscu 10 znajduje się liczba 0.

Instrukcja ? 100 jest skokiem warunkowym. Jej znaczenie za-

0	0
.....	.....
10	99
11	201
.....	.....
100	+ (11)
101	*N* 11
102	*P* 10
103	= 10
104	? 100
105	← 300
106	stop
.....	.....
201	a_1
202	a_2
.....	.....
300	a_{100}

leży od ostatnio wykonanej instrukcji, badania zawartości miejsca, = 10. Ponieważ liczba ta jest różna od zera, więc po instrukcji ? 100 zostanie wykonana instrukcja + (11).

Jednakże obecnie instrukcja + (11) ma inne znaczenie, niż miała przed jej pierwszym wykonaniem, gdyż pod adresem 11 znajduje

się teraz liczba 202. Wykonanie tej instrukcji powoduje więc dodanie do akumulatora drugiego składnika sumy a_2.

Ciąg tych instrukcji jest powtarzany tak długo, aż w miejscu 10 pojawi się liczba zero.

Po 99-krotnym wykonaniu instrukcji 100—104 w miejscu 10 znajduje się liczba zero. Wtedy po instrukcji 104 zostanie wykonana instrukcja 105: → 400, która spowoduje zapisanie obliczonej sumy w miejscu 400.

Stan magazynu dla tego obliczenia przedstawiony jest na s. 75.

3. Wprowadzanie i wyprowadzanie

Zadanie. Ułożyć program wprowadzania liczb $a_1, a_2, \ldots, a_n$ do magazynu wewnętrznego na kolejne miejsca, poczynając od miejsca m. Program umieścić poczynając od miejsca k.

Założenia. W magazynie wejściowym znajduje się ciąg liczb (lub innych symboli) $a_1, a_2, \ldots, a_n$. Liczby te są umieszczone w kolejnych miejscach magazynu. Ponadto

$$C10 = m$$
$$C11 = n$$

Program

$k.$	→ (10)	Wprowadzenie
$k+1.$	N 10	Następnik
$k+2.$	P 11	Poprzednik
$k+3.$	= 11	Zero
$k+4.$	? k	Skok warunkowy
$k+5.$	Stop	

Instrukcja → (10) powoduje przesłanie pierwszej liczby a_1 z wejścia do magazynu wewnętrznego pod adres podany pod adresem 10, czyli pod adres m.

Instrukcja N 10 dodaje do m jedność.

Instrukcja P 11 odejmuje od liczby — wskazującej ile przesłań zostało jeszcze do wykonania — jedność.

Następna instrukcja powoduje sprawdzenie, czy w miejscu 11 znajduje się zero. Jeżeli nie, to znaczy, że należy powtórzyć wszystkie

czynności do instrukcji → (10). Jeżeli tak, to przesyłanie jest zakończone.

Stan początkowy dla tego programu przedstawia się następująco:

10	m
11	n
k	→ (10)
$k+1$	N 10
$k+2$	P 11
$k+3$	= 11
$k+4$	? k
$k+5$	stop

Program przepisania zawartości miejsc magazynu wewnętrznego do magazynu wyjściowego będzie identyczny, z wyjątkiem pierwszej instrukcji. Zamiast instrukcji → (10) znajdzie się wtedy instrukcja ← (10).

4. Program podstawiania

Zadanie. Dany jest ciąg liczb $a_1, a_2, \ldots, a_n$. Liczbę $a_i = k$ w tym ciągu zastąpić liczbą m. Program umieścić, poczynając od miejsca p.

Założenia

$$C\,10 = k$$
$$C\,11 = m$$
$$C\,12 = r$$
$$C\,r+1 = a_1$$
$$C\,r+2 = a_2$$
$$\ldots\ldots\ldots$$
$$C\,r+n = a_n$$

Program

$p.$	$N\ 12$	Następnik
$p+1.$	$\leftarrow(12)$	Odczytaj
$p+2.$	$-\ 10$	Odejmij
$p+3.$	$=0$	Zero
$p+4.$	$?\ p$	Skok warunkowy
$p+5.$	$\leftarrow(11)$	Odczytaj
$p+6$	$\rightarrow(12)$	Zapisz
$p+7.$	stop	

Pierwsza instrukcja $N\ 12$ oblicza adres pierwszej liczby a_1.

Instrukcja $\leftarrow(12)$ powoduje przesłanie a_1 do akumulatora.

Instrukcja $-\ 10$ wykonuje działanie $a_1 = k$.

Następna instrukcja $=0$ oraz instrukcja $?\ p$ sprawdzają, czy w akumulatorze znajduje się zero.

Jeżeli nie, to cykl instrukcji jest powtarzany od początku. Jeżeli tak, to pobierana jest następna instrukcja $\leftarrow(11)$, powodująca przesłanie do akumulatora liczby m.

Dalsza instrukcja $\rightarrow(12)$ powoduje wpisanie liczby znajdującej się w akumulatorze na miejsce o adresie podanym pod adresem 12. Pod adresem 12 znajduje się właśnie adres liczby k.

Stan magazynu dla tego postępowania przedstawiony jest na s. 79.

Przedstawione przykłady dotyczyły wykonywania różnych operacji na liczbach. W podobny sposób można układać programy wykonujące dowolne operacje na jakichkolwiek symbolach.

W następnym rozdziale zapoznamy się ze strukturą programu do dowodzenia twierdzeń.

10	k
11	m
12	r
p	$N\ 12$
$p+1$	$\leftarrow(12)$
$p+2$	$-\ 10$
$p+3$	$=0$
$p+4$	$?\ p$
$p+5$	$\leftarrow(11)$
$p+6$	$\rightarrow(12)$
$p+7$	stop
r + 1	a_1
r + 2	a_2
r + n	a_n

Rozdział VII

PROGRAM DOWODZENIA TWIERDZEŃ

§ 1. Ogólna struktura programu dowodzącego twierdzenia

Ułożenie programu dowodzącego twierdzenia polega na opisaniu czynności podanych w tablicy 1 za pomocą instrukcji maszyny adresowej. Dla ułatwienia wygodnie jest cały program podzielić na mniejsze odcinki takie, że każdy z tych odcinków opisuje jeden z fragmentów całego postępowania przy dowodzeniu twierdzenia.

Program dowodzenia twierdzeń powinien zawierać fragmenty pozwalające na:

1. Badanie, czy rozważana hipoteza jest atomem.
2. Badanie, czy atom jest aksjomatem.
3. Obliczanie przesłanek.
4. Sprawdzenie końca dowodzenia.

Oczywiście program ten musi również automatycznie umieszczać obliczone przesłanki w magazynie wewnętrznym maszyny [1]).

Wygodnie jest tak ułożyć program, żeby mógł on służyć do badania nie tylko jednej hipotezy, a kolejnego ciągu hipotez jakiejś teorii, zapisanych w magazynie wejściowym maszyny. Założenie to nie komplikuje zbytnio programu, a natomiast zwiększa jego użyteczność.

Badanie, czy wyrażenie jest atomem czy aksjomatem, obliczanie

[1]) Umieszczanie przesłanek może następować w porządku $\overline{W}$ lub $\overline{P}$. Porządek $\overline{W}$ wydaje się tu korzystniejszy, ze względu na mniejszą zajętość miejsc w magazynie, co przy dowodach zawierających wielką ilość operacji może mieć duże znaczenie.

przesłanek zależy od rozważanej teorii matematycznej. Ponieważ do tej pory nie rozpatrywaliśmy żadnej konkretnej teorii matematycznej, fragmenty programu dotyczące tych czynności nie mogą być szczegółowo napisane. Ułożenie takiego programu nie przedstawia trudności. Należy tylko dokładnie sprecyzować, co to jest atom, aksjomat teorii i podać dokładnie reguły wnioskowania. Dla niektórych teorii programy te mogą być dość skomplikowane, jakkolwiek ich ogólna zasada jest zawsze prosta.

§ 2. Podział magazynu wewnętrznego na bloki

Dla łatwiejszego ułożenia programu dowodzącego twierdzenia podzieliliśmy cały magazyn na bloki, z których każdy będzie zawierał jeden fragment programu lub też spełniał inne funkcje, podane w poniższym spisie.

B_0 — Blok pomocniczy,
B_1 — czynności przygotowawcze,
B_2 — wprowadzenie hipotezy wyjściowej do maszyny,
B_3 — wydrukowanie badanej hipotezy na wyjściu,
B_4 — badanie, czy hipoteza jest atomem,
B_5 — obliczanie pierwszej przesłanki,
B_6 — obliczanie drugiej przesłanki,
B_7 — badanie, czy atom jest aksjomatem,
B_8 — sprawdzenie końca wnioskowania,
B_9 — blok roboczy.

Rolę bloków omówimy w następnym paragrafie. W tym paragrafie natomiast podamy rozmieszczenie adresów w blokach. Ponieważ z góry nie wiemy, ile miejsc powinien mieć każdy blok, wygodnie będzie przyjąć następujący sposób oznaczania adresów w blokach: Pierwszy adres bloku B_i oznaczymy przez a_i. Pierwszy adres bloku B_2 będzie a_2. Drugi adres oznaczymy $a_i + 1$, trzeci $a_i + 2$ itd. Np. adres $a_3 + 2$ oznacza trzecie miejsce w bloku B_3. Ostatni adres bloku B_i oznaczymy przez $a_{i+1} - 1$, co należy rozumieć jako miejsce poprzedzające pierwsze miejsce w bloku B_{i+1}. Ten sposób oznaczania adresów pozwoli nam na pisanie programów, mimo że nie znamy dokładnie adresów poszczególnych bloków oraz liczby miejsc w bloku.

Tak więc magazyn przy podziale na bloki będzie miał postać:

	.....	.....
	a_0	
B_0	$a_0 + 1$	
	.....	.....
	$a_1 - 1$	
B_1	a_1	
	$a_1 + 1$	
	.....	.....
	$a_2 - 1$	
B_2	a_2	
	$a_2 + 1$	
	.....	.....
	.....	.....
	$a_9 - 1$	
B_9	a_9	
	$a_9 + 1$	
	.....	.....
B_{10}	$a_{10} - 1$	

§ 3. Zadania bloków

Omówimy teraz rolę bloków, podając tam gdzie to będzie możliwe szczegółowe programy. Przyjmijmy, że wnioskowanie jest wykonywane w porządku $\bar{P}$.

Blok B_0

	0	0
	1	Hipoteza początkowa fałszywa
	2	Hipoteza początkowa prawdziwa
	3	a_9
B_0	4	i
	5	j
	.....	.....
	$a_1 - 1$	

Napis znajdujący się w miejscu 1 będzie — w przypadku stwierdzenia, że hipoteza jest fałszywa — wydrukowany na wyjściu maszyny. Podobnie, objaśnienie wpisane w miejscu 2 zostanie wydrukowane w przypadku, jeżeli hipoteza okaże się prawdziwa [1]).

W miejscu 3 podany jest adres pierwszego miejsca w bloku roboczym, który będzie grał rolę magazynu liniowego, służącego do magazynowania kolejnych hipotez i przesłanek.

Miejsce 4 przewidziane jest do magazynowania w bloku roboczym aktualnego adresu miejsca, z którego wysyłamy hipotezę. Inaczej mówiąc, adres w miejscu 4 wskazuje miejsce przygotowane do odczytu w bloku roboczym.

W miejscu 5 znajduje się adres miejsca, w którym należy umieścić

[1]) W miejscach magazynu mogą znajdować się dowolne teksty objaśniające, które zależnie od okoliczności są przez odpowiednio ułożony program wykorzystywane do drukowania przez maszynę objaśnień odnośnie wyników pracy.

w bloku roboczym obliczoną przesłankę. Jest to więc adres miejsca przygotowanego do zapisu w bloku roboczym.

Blok B_1

B_1	a_1	$\leftarrow 3$
	$a_1 + 1$	$\rightarrow 4$
	$a_1 + 2$	$\rightarrow 5$

Blok B_1 służy do ustawienia wartości początkowych adresów miejsc, przygotowanych do odczytu i do zapisu w bloku roboczym B_9.

Instrukcja $\leftarrow 3$ przesyła adres a_9 do akumulatora. Instrukcje $\rightarrow 4$, $\rightarrow 5$ powodują przesłanie adresu a_9 pod adresy 4 i 5, tak że w chwili początkowej przed rozpoczęciem wnioskowania miejsce a_9 jest przygotowane do zapisu i odczytu.

Blok B_2

B_2	a_2	$\rightarrow a_9$

Blok B_2 składa się w naszym przypadku tylko z jednej instrukcji, która powoduje przesłanie hipotezy początkowej z wejścia do pierwszego miejsca bloku roboczego B_9.

Blok B_3

B_3	a_3	$\leftarrow (4)$

Blok B_3 zawiera również tylko jedną instrukcję $\leftarrow (4)$, która powoduje wydrukowane na wyjściu zawartości miejsca o adresie podanym pod adresem 4. Ponieważ pod adresem 4 znajduje się adres miejsca przygotowanego do odczytu, więc instrukcja ta drukuje aktualnie badaną przez maszynę hipotezę. Przy pierwszym wykonywaniu programu w miejscu 4 znajduje się adres a_9, a więc zostanie wtedy wydrukowana hipoteza początkowa, znajdująca się w pierwszym miejscu bloku roboczego.

Blok B_4

B_4	a_4	$\leftarrow (4)$
	.....	.....
	$a_5 - 2$	$= 0$
	$a_6 - 1$	$? a_7$

W bloku tym znajduje się program badający, czy hipoteza przygotowana do odczytu w bloku roboczym jest atomem. Jak wspominaliśmy, ponieważ nie rozpatrujemy żadnej konkretnej teorii matematycznej, nie możemy podać programu tego badania. Jednakże każdy taki program będzie się zaczynał od instrukcji $\leftarrow (4)$, powodującej przesłanie wyrażenia z miejsca przygotowanego do odczytu w bloku roboczym — do akumulatora [1]).

Następne instrukcje programu powodują badanie wyrażenia, znajdującego się w akumulatorze. Program ten można ułożyć w ten sposób, że jeżeli badane wyrażenie jest atomem, to w akumulatorze znajdzie się liczba zero. W przypadku przeciwnym, w akumulatorze powinna być liczba różna od zera. Wtedy instrukcja $= 0$ zbada jaka liczba jest w akumulatorze i zależnie od wyniku badania przejdzie do bloku następnego B_5 lub do bloku B_7.

Blok B_5

B_5	a_5	$\leftarrow (4)$
	.....	.....
	$a_6 - 2$	$N\,5$
	$a_6 - 1$	$\rightarrow (5)$

[1]) Przypominamy, że w miejscu przygotowanym do odczytu znajduje się aktualnie badana hipoteza.

Blok ten powoduje obliczenie pierwszej przesłanki. Rola instrukcji $\leftarrow (4)$ jest identyczna jak w poprzednim bloku. Powoduje ona pobieranie hipotezy do akumulatora, której pierwszą przesłankę mamy obliczyć. Następne instrukcje wykonują odpowiednie czynności na symbolach i przyjmiemy, że w wyniku tych czynności pierwsza przesłanka znajdzie się w akumulatorze.

Instrukcja $N\,5$ dodaje do adresu w miejscu 5 jedynkę tak, że w miejscu 5 otrzymujemy adres miejsca, gdzie należy umieścić obliczoną przesłankę.

Ostatnia instrukcja $\rightarrow (5)$ powoduje przesłanie obliczonej przesłanki z akumulatora pod adres podany pod adresem 5. W rezultacie obliczona przesłanka znajdzie się w odpowiednim miejscu bloku roboczego.

Blok B_6

	a_6	$\leftarrow (4)$
	.....	.....
B_5	$a_7 - 4$	$N\,5$
	$a_7 - 3$	$\rightarrow (5)$
	$a_7 - 2$	$N\,(4)$
	$a_7 - 1$	$!\,a_3$

W bloku B_6 odbywa się obliczenie drugiej przesłanki. Obliczenie to zaczyna się również od instrukcji $\leftarrow (4)$, pobierającej badaną hipotezę z bloku roboczego do akumulatora [1]).

Zakładamy, że obliczona druga przesłanka znalazła się w akumulatorze. Instrukcja $N\,5$ powoduje przygotowanie adresu, pod który należy przesłać drugą przesłankę, a instrukcja $\rightarrow (5)$ przesyła obliczoną przesłankę pod ten adres.

[1]) Przy obliczaniu drugiej przesłanki konieczne jest pobieranie ponowne hipotezy, gdyż w trakcie obliczania pierwszej przesłanki był używany akumulator i hipoteza, pobrana przez pierwszą instrukcję bloku B_5, musiała zostać z akumulatora usunięta.

Następna instrukcja $N\,4$ powoduje przygotowanie do odczytu następnego miejsca w bloku roboczym.

W wyniku ostatniej instrukcji $!\,a_3$ następuje przejście do bloku B_3, odczytującego nową hipotezę z bloku roboczego.

Blok B_7

	a_7	$\leftarrow (4)$
	.....	.....
B_7	$a_8 - 4$	$= 0$
	$a_8 - 3$	$?\,a_8$
	$a_8 - 2$	$\leftarrow 11$
	$a_8 - 1$	stop

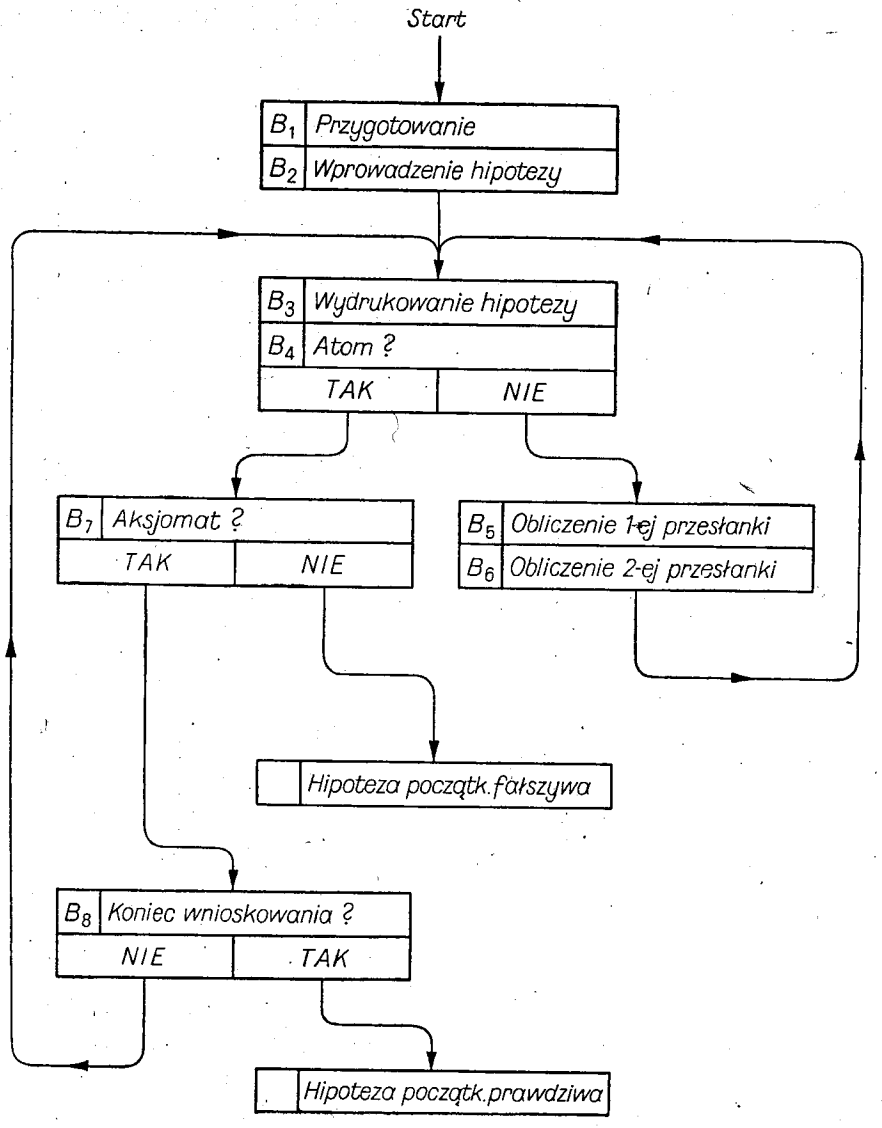
Blok B_8

	a_8	$\leftarrow 4$
	$a_8 + 1$	$- 5$
B_8	$a_8 + 2$	$= 0$
	$a_8 + 3$	$?\,a_3$
	$a_8 + 4$	$\leftarrow 2$
	$a_8 + 5$	stop

Badanie, czy rozpatrywana hipoteza jest aksjomatem, zaczyna się również od pobrania hipotezy z bloku roboczego do akumulatora (instrukcja $\leftarrow 4$).

Program badania, tak jak to miało miejsce w poprzednich przypadkach, jest tak ułożony, że jeżeli hipoteza jest aksjomatem, to w akumulatorze ma być liczba różna od zera. Powoduje to za pośrednictwem instrukcji $= 0$ oraz $?\,a_8$ przejście do bloku B_8 badają-

TABLICA VI



cego koniec wnioskowania. Jeżeli hipoteza nie jest aksjomatem, to w akumulatorze znajdzie się liczba zero, co spowoduje przejście do instrukcji ← 1 drukującej na wyjściu objaśnienie zawarte w miejscu 1 bloku B_1: HIPOTEZA POCZĄTKOWA FAŁSZYWA, po czym następuje zatrzymanie maszyny.

W bloku B_8 następuje badanie końca wnioskowania przez porównanie adresów miejsc, przygotowanych do odczytu i zapisu, znajdujących się w miejscach 4 i 5. Instrukcje ← 4 oraz — 5 powodują przesłanie adresu miejsca przygotowanego do zapisu do akumulatora i odjęcie od niego adresu miejsca, przygotowanego aktualnie do odczytu. Jeżeli oba te adresy są jednakowe, to znaczy że wnioskowanie zostało zakończone. W wyniku instrukcji = 0, ? a_3 zostaje wykonana instrukcja ← 2, powodująca wydrukowanie tekstu objaśniającego, zawartego w miejscu 2 bloku B_0: HIPOTEZA POCZĄTKOWA PRAWDZIWA, po czym maszyna kończy działanie.

Jeżeli oba badane adresy nie są równe, to znaczy że wnioskowanie nie zostało zakończone i za pośrednictwem instrukcji ? a_3 maszyna przechodzi do wykonania instrukcji znajdującej się pod adresem a_3 w bloku B_3.

Zależność między blokami można przedstawić graficznie, jak w tablicy VI.

§ 4. Wnioskowanie w porządku $\overline{W}$

Przy wykonywaniu operacji wnioskowania w porządku $\overline{W}$ ogólny schemat programu nie ulegnie zmianie. Zmieniają się nieco tylko niektóre bloki.

W bloku B_0 będziemy teraz mieli następujące informacje:

B_0

0	0
1	Jak poprzednio
2	Jak poprzednio
3	a_9
4	i
5	

Blok roboczy gra teraz rolę magazynu rewersyjnego, a więc potrzebny jest tylko jeden adres wskazujący miejsce w bloku roboczym, przygotowane do odczytu lub do zapisu. Adres tego miejsca znajduje się w miejscu 4 bloku B_0. W związku z powyższym również blok B_1 zawiera tylko instrukcje powodujące przesłanie adresu początkowego bloku B_9 do miejsca 4.

B_1

a_1	$\leftarrow 3$
$a_1 + 1$	$\rightarrow 4$

Bloki B_2, B_3 i B_4 nie ulegają zmianie. W bloku B_5 należy zmienić ostatnie dwie instrukcje, powodujące przesłanie obliczonej przesłanki do bloku roboczego. Zgodnie z zasadą działania bloku roboczego, przesłankę pierwszą należy umieścić na miejscu badanej hipotezy.

Jednakże hipoteza, znajdująca się pod adresem i będzie również potrzebna w następnym bloku do obliczenia drugiej przesłanki, a więc przed zapisaniem na jej miejscu pierwszej przesłanki, hipotezę z tego przejścia przepiszemy do bloku B_0 na miejsce 5. Ponieważ przesyłanie może odbywać się tylko za pośrednictwem akumulatora, musimy tymczasowo umieścić np. w miejscu 6 bloku B_0 znajdującą się w akumulatorze pierwszą przesłankę. A więc blok B_5 ostatecznie będzie zawierał następujące instrukcje:

B_5

a_5	$\leftarrow (4)$
.....	.....
$a_6 - 5$	$\rightarrow 6$
$a_6 - 4$	$\leftarrow (4)$
$a_6 - 3$	$\rightarrow 5$
$a_6 - 2$	$\leftarrow 6$
$a_6 - 1$	$\rightarrow (4)$

Przy obliczaniu drugiej przesłanki pobranie hipotezy nie nastąpi z bloku roboczego, gdyż tam na jej miejscu znajduje się już pierwsza

przesłanka, a z miejsca 5, gdzie została ona umieszczona na czasowe przechowanie. Umieszczenie drugiej przesłanki nastąpi w następnym miejscu bloku roboczego. A więc blok B_6 będzie miał teraz postać:

B_6

a_6	$\leftarrow (4)$
.....	.....
$a_7 - 3$	$N\,4$
$a_7 - 2$	$\rightarrow (4)$
$a_7 - 1$	$!\,a_3$

Blok B_7 nie ulega zmianie.

Badanie końca wnioskowania polega — w przypadku magazynu rewersyjnego — na zbadaniu, czy cały magazyn jest pusty. A więc w naszym przypadku oznacza to, że koniec wnioskowania jest wtedy, gdy $C\,4 = a_9 - 1$. Badanie końca wnioskowania będzie więc przebiegać następująco:

a_8	$\leftarrow 3$
$a_8 + 1$	$P\,0$
$a_8 + 2$	$-\,4$
$a_8 + 3$	$= 0$
$a_8 + 4$	$?\,a_3$
$a_8 + 5$	$\leftarrow 2$
$a_8 + 6$	stop

§ 5. Obliczanie przesłanek

Na zakończenie tego rozdziału podamy jeszcze program obliczania przesłanek w teorii T_1'. Potraktujemy ten program jako oddzielną całość, nie wnikając w jaki sposób powinien on współpracować z całym programem wnioskowania. Chodzi nam bowiem o zilustrowanie

zasady, w jaki sposób można za pomocą instrukcji maszyny adresowej opisać znajdowanie przesłanek dla zadanej hipotezy.

Przypominamy, że w teorii T_1' była jedna reguła redukowania

$$R_1\colon\ \alpha A\beta \to \alpha A,\ A\beta$$

Znalezienie pierwszej przesłanki polegało na usunięciu z hipotezy ostatniego symbolu, znalezienie zaś drugiej przesłanki sprowadzało się do usunięcia pierwszego symbolu hipotezy.

Przy „ręcznym" realizowaniu tej reguły R_1, znalezienie przesłanek dowolnej hipotezy jest więc banalne. Program tego procesu dla maszyny cyfrowej jest już jednak niewspółmiernie skomplikowany [1]).

Przyjmiemy, że każdy symbol hipotezy znajduje się w jednym miejscu magazynu. Ponadto przyjmiemy podział magazynu na następujące bloki:

A_0 — Blok pomocniczy,
A_1 — blok programu,
A_2 — blok hipotezy,
A_3 — blok pierwszej przesłanki,
A_4 — blok drugiej przesłanki.

Oznaczenia adresów w blokach są takie jak poprzednio, tzn. pierwszy adres bloku A_i oznaczymy przez a_i, a ostatni adres tego bloku — przez $a_{i+1} - 1$.

Program obliczający przesłanki będzie więc wpisywał symbole z bloku A_2 do bloku A_3, z pominięciem ostatniego symbolu, oraz do bloku A_4, z pominięciem pierwszego symbolu.

Blok B_0

0	Akumulator
1	a_2
2	a_3
3	a_4
4	i
5	j

[1]) Jest to typowa sytuacja w maszynach matematycznych. Stosunkowo proste zadania wymagają programów składających się z wielu instrukcji.

Miejsca 1, 2, 3 w bloku B_0 zawierają odpowiednio adresy początku hipotezy, początku pierwszej przesłanki i początku drugiej przesłanki. W miejscu 4 znajduje się adres aktualnie odczytywanego symbolu hipotezy z bloku A_2. Miejsce 5 zawiera adres aktualnie zapisywanego symbolu przesłanki w bloku A_3 lub A_4 [1]).

Program obliczania przesłanek podzielimy na dwie części:

1. P — Obliczanie pierwszej przesłanki,
2. Q — obliczanie drugiej przesłanki.

Ponieważ P i Q są programami całkowicie niezależnymi, więc omówimy je oddzielnie.

Program P

p	$\leftarrow 1$
$p+1$	$\rightarrow 4$
$p+2$	$\leftarrow 2$
$p+3$	$\rightarrow 5$
$p+4$	$\leftarrow (4)$
$p+5$	$\rightarrow (5)$
$p+6$	$N\,4$
$p+7$	$N\,5$
$p+8$	$= (4)$
$p+9$	$?\,p+4$
$p+10$	$P\,5$
$p+11$	$0\ (5)$

Pierwsze cztery instrukcje powodują przesłanie adresów początkowych do miejsc 4 i 5. Następne dwie, $\leftarrow (4)$ i $\rightarrow (5)$, instrukcje przesyłają pierwszy symbol hipotezy z bloku A_2 na pierwsze miejsce w bloku A_3. Dalsze dwie instrukcje $N4$ i $N5$ zwiększają adresy

[1]) Jeżeli obliczana jest pierwsza przesłanka, to j jest adresem w bloku A_3; jeżeli obliczana jest druga przesłanka, to j jest adresem w bloku A_4.

miejsc 4 i 5 o 1, przygotowując w ten sposób następne miejsce w blokach A_2 i A_3 do odczytu i do zapisu. Instrukcje $= (4)$ i $? p + 4$ badają, czy następny symbol w hipotezie jest zerem [1]).

Jeżeli następny symbol nie jest zerem, to program przepisywania jest powtarzany. W przypadku przeciwnym następne obie instrukcje $0(5)$ i $P5$ powodują wymazanie ostatniego symbolu w przesłance.

Program Q

q	$\leftarrow 1$
$q+1$	$\rightarrow 4$
$q+2$	$\leftarrow 3$
$q+3$	$\rightarrow 5$
$q+4$	$N4$
$q+5$	$\leftarrow (4)$
$q+6$	$\rightarrow (5)$
$q+7$	$N4$
$q+8$	$N5$
$q+9$	$= (4)$
$q+10$	$? q+5$
$q+11$	stop

Rola pierwszych czterech instrukcji jest identyczna jak w programie P z tą różnicą, że obecnie na miejsce 5 wprowadzany jest adres pierwszego miejsca w bloku A_4. Instrukcja $N4$ powoduje opuszczenie pierwszego symbolu w hipotezie. Następne dwie instrukcje $\leftarrow (4)$ oraz $\rightarrow (5)$ przepisują drugi symbol hipotezy na pierwsze miejsce w bloku A_4. Instrukcje $N4$ i $N5$ przygotowują następne miejsca do odczytu i zapisu w blokach A_3 oraz A_5. Instrukcje $= (4)$ i $? q+5$ badają, czy następne miejsce w bloku hipotezy A_3 jest puste. Jeżeli

[1]) Przyjęliśmy, że miejsce puste jest oznaczone symbolem zero.

nie, to obliczanie przesłanki jest kontynuowane przez powtórzenie programu od instrukcji $\leftarrow (4)$. Jeżeli następny symbol jest zerem, obliczanie przesłanek jest zakończone.

W ten sposób program P powoduje obliczenie pierwszej przesłanki i umieszczenie jej w bloku A_4, a program Q — obliczenie drugiej przesłanki i umieszczenie jej w bloku A_5.

Widzimy, że program dowodzenia twierdzeń — nawet tak prostej teorii jak teoria T_1' — jest dość skomplikowany[1]). Dla rzeczywistych teorii matematycznych stopień skomplikowania jest znacznie większy, może więc powstać pytanie, czy zamiast układać program dowodzenia twierdzeń nie jest prościej po prostu twierdzenia te dowieść bez użycia maszyny. Tak oczywiście nie jest. Dla zadanej teorii program dowodzenia układany jest bowiem jednokrotnie i trud włożony w jego ułożenie jest na pewno wielokrotnie mniejszy, aniżeli dowiedzenie wszystkich twierdzeń, które za pomocą tego programu można dowieść.

Tutaj rozpatrywaliśmy zbyt proste przykłady, aby różnica ta była widoczna. Jeżeli jednak do wykonania dowodu jakiegoś twierdzenia należy zastosować kilka tysięcy razy regułę wnioskowania okaże się, że nawet skomplikowany program będzie prostszy niż sam dowód.

Z drugiej strony należy uwzględnić szybkość działania maszyn matematycznych. Maszyna może wykonywać kilkadziesiąt tysięcy, czy nawet kilkaset tysięcy instrukcji w ciągu sekundy. Znalezienie dowodu nawet bardzo skomplikowanego twierdzenia jest kwestią ułamka sekundy. Tak więc, gdybyśmy chcieli porównać czas potrzebny na dowiedzenie zadanej ilości (ilości dużej) twierdzeń jakiejś

[1]) Jak wykazały badania w Instytucie Matematycznym PAN oraz na Uniwersytecie Warszawskim, zamiast pisać bezpośrednio programy realizujące zadane reguły wnioskowania, wygodniej jest podać najpierw realizację reguł wnioskowania w tzw. maszynie Turinga a dopiero potem ułożyć programy dla maszyny cyfrowej naśladujące działanie maszyny Turinga. (Z pojęciem maszyny Turinga można zapoznać się z książki: B. A. Trachtenbrot, *Algorytmy i automatyczne rozwiązywanie zadań*, Warszawa 1961). Jednakże, aby nie wprowadzać nowych pojęć, podaliśmy zasadę układania programów bezpośrednio w języku maszyn cyfrowych, co przy okazji pozwoliło nam na podanie zasady działania maszyn tego rodzaju.

teorii „ręcznie", z czasem użytym na wykonanie tych samych dowodów za pomocą maszyny — uwzględniając w tym również czas potrzebny na ułożenie programu — ten ostatni okaże się wielokrotnie mniejszy niż pierwszy. Zastosowanie maszyny, jeżeli w ogóle jest możliwe, jest na pewno opłacalne.

Na marginesie programowania powstaje jeszcze jeden problem. Mając na uwadze zastosowanie maszyn w tej dziedzinie, byłoby rzeczą pożądaną, aby teorie matematyczne były budowane w sposób umożliwiający zastosowanie w nich maszyn matematycznych. Mówiąc ściśle, chodzi tu o takie dobieranie reguł wnioskowania, żeby ich realizacja nie komplikowała zbytnio programów. Reguły bowiem przystosowane do manipulowania ręcznego nie zawsze są wygodne do zastosowania w maszynie. Byłoby również pożyteczne, gdyby udało się określić na podstawie analizy struktury danego wyrażenia, czy jest ono aksjomatem. Nie byłoby wtedy konieczne magazynowanie wszystkich aksjomatów i porównywanie badanego wyrażenia z każdym z nich[1]). Z maszynowego punktu widzenia jest to bowiem dość niewygodne. Nie jest natomiat kłopotliwa duża ilość reguł wnioskowania, gdyż maszyna może je łatwo realizować.

Tak więc wymagania odnośnie teorii matematycznych są w przypadku maszynowego dowodzenia twierdzeń inne niż te, które się stawia w matematyce obecnie. A więc sformułowanie większości istniejących teorii matematycznych jest dla celów maszynowych nieprzydatne. Zastosowanie więc maszyny w jakiejś teorii wymaga uprzedniego przeformułowania teorii, odpowiedniego doboru aksjomatów i reguł wnioskowania. Nie są to sprawy całkiem proste. Przy tej okazji powstaje szereg nowych trudnych do rozwiązania zagadnień. Dlatego też zastosowanie maszyn do dowodzenia twierdzeń natrafia na dość duże trudności.

W następnej części książki podamy zastosowanie dotychczasowych rozważań do bardzo prostej, ale już „rzeczywistej" teorii matematycznej, tzw. **rachunku zdań.**

[1]) Gdybyśmy do badania aksjomatów zastosowali również pojęcie maszyny Turinga, sprawa by się znacznie uprościła i nie musielibyśmy porównywać badanych wyrażeń z aksjomatami zmagazynowanymi w odpowiednich magazynach.

CZĘŚĆ II

Rozdział VIII

JĘZYK RACHUNKU ZDAŃ

§ 1. Uwagi wstępne

Rachunek zdań jest teorią matematyczną, stanowiącą podstawę rozumowań matematycznych. Podaje on zasady poprawnych wnioskowań, pozwalających z jednych zdań prawdziwych uzyskiwać nowe zdania prawdziwe. Zdaniami tego rachunku są w zasadzie zdania języka potocznego, ale nie wszystkie, lecz tylko zdania oznajmujące, jak np. „Dzisiaj jest niedziela", „Dwa razy trzy jest sześć" itp. Rachunek zdań nie zajmuje się więc zdaniami takimi jak np. „Czy pójdziesz dzisiaj do kina" lub „Idź do domu" itp.

Zdania dotyczące pewnych faktów matematycznych zapisuje się dziś nie w języku potocznym, a w specjalnym języku symbolicznym. Np. zamiast „Dwa razy trzy jest sześć" piszemy $2 \cdot 3 = 6$. Tak więc formuły matematyczne możemy uważać również za zdania oznajmujące. Rachunek zdań stosujemy z korzyścią do zdań, których treścią są pewne fakty matematyczne, gdyż pozwala on na przekształcenie jednych formuł prawdziwych na inne formuły prawdziwe.

Rachunek zdań jest fragmentem logiki matematycznej — nauki, zajmującej się poprawnymi formami wnioskowania w matematyce. Jest to fragment najprostszy, tym niemniej na jego podstawie można już dobrze zilustrować szereg problemów występujących w związku z zastosowaniem maszyn matematycznych do dowodzenia twierdzeń. Pierwsze prace na temat automatycznego dowodzenia twierdzeń dotyczyły właśnie rachunku zdań. Do dzisiaj ten problem nie został

zadowalająco rozwiązany. Obecne metody są bowiem zbyt skomplikowane i mają dość ograniczony zasięg zastosowań.

Zreferowana w dalszym ciągu metoda dowodzenia oparta będzie w zasadzie na rezultatach uzyskanych przez, cytowanego we wstępie, Wanga. Schemat postępowania zaproponowany przez Wanga wydaje się na tyle ogólny, że w podobny sposób można postępować również w innych dziedzinach.

Czytelnik, który chciałby dokładniej zrozumieć dalsze rozważania, a nie znający rachunku zdań, powinien bliżej zapoznać się z tym działem logiki. Jeszcze raz przypominamy, że rachunek zdań jest bardzo jasno opisany w książce A. Grzegorczyka *Logika popularna* PWN, 1960. Ujęcie rachunku w książce Grzegorczyka jest inne niż to, które nam będzie potrzebne tutaj, nie stanowi to jednak przeszkody.

W dalszym ciągu podamy kilka wiadomości o rachunku zdań, potrzebnych nam do dalszych rozważań. Rachunek zdań podamy w sformułowaniu Gentzena, w postaci nieco zmienionej przez Wanga[1]).

§ 2. Zdania proste i złożone

Wiemy z gramatyki, że zdania dzielimy na **zdania proste** i **zdania złożone.** Np. zdanie „Dzisiaj pójdę do kina" jest zdaniem prostym, natomiast zdanie „Dzisiaj pójdę do kina i jutro pójdę do teatru" jest zdaniem złożonym. Składa się ono z dwu zdań prostych „Dzisiaj pójdę do kina", „Jutro pójdę do teatru" oraz spójnika zdaniowego „i".

Zdania złożone są to zdania utworzone ze zdań prostych oraz spójników zdaniowych. W języku istnieje wiele spójników zdaniowych, ale tylko niektóre z nich odgrywają rolę w logice. Są to następujące spójniki:

1. nieprawda że ...
2. ... lub ...
3. ... i ...

[1]) Istnieje jeszcze inne sformułowanie rachunku zdań, podane przez prof. Rasiową oraz prof. Sikorskiego które wydaje się szczególnie wygodne do maszynowego dowodzenia twierdzeń w rachunku zdań; jednakże zbadanie tego sformułowania pod kątem przydatności do celów maszynowych nie zostało jeszcze zakończone, dlatego nie przytaczamy go tutaj.

4. ... jeżeli ..., to ...
5. ... wtedy i tylko wtedy, gdy ...

W logice spójniki te noszą specjalne nazwy, a mianowicie:

1. Negacja,
2. Alternatywa (lub suma logiczna),
3. Koniunkcja (lub iloczyn logiczny),
4. Implikacja,
5. Równoważność.

Może się wydawać dziwne, że negację również nazywamy spójnikiem, choć dotyczy ona tylko jednego zdania, a nie dwu zdań, jak wszystkie spójniki pozostałe. Negacja pozwala jednak z jednego zdania tworzyć nowe zdanie, rola jej więc jest podobna do roli pozostałych spójników, które z dwu danych zdań tworzą nowe zdanie. Wygodnie więc negację również nazywać spójnikiem. Jeżeli zastosujemy negację do zdania „Dzisiaj świeci słońce", to otrzymamy nowe zdanie „Nieprawda, że dzisiaj świeci słońce".

Alternatywa pozwala ze zdań „Dzisiaj świeci słońce", „Dzisiaj pada deszcz" utworzyć nowe zdanie „Dzisiaj świeci słońce lub dzisiaj pada deszcz".

Podobnie stosując do dwu zdań prostych „Ala ma książkę", „Ola ma książkę" spójnik koniunkcji, otrzymamy zdanie „Ala ma książkę i Ola ma książkę".

Spójnik implikacji pozwala ze zdań „Jutro będzie pogoda", „Jutro pójdę na wycieczkę" — utworzyć zdanie „Jeżeli jutro będzie pogoda, to jutro pójdę na wycieczkę".

I wreszcie spójnik równoważności tworzy np. ze zdań „Pójdę do kina", „Dostanę pieniądze" zdanie „Pójdę do kina wtedy i tylko wtedy, gdy dostanę pieniądze".

§ 3. Funkcje zdaniowe

Spójniki zdaniowe pozwalają łączyć dowolne zdania. Zamiast więc pisać konkretne zdania, możemy, w rozważaniach dotyczących zdań, posługiwać się literami oznaczającymi dowolne zdania. Np. możemy pisać

1. nieprawda że p
2. p lub q

3. p i q
4. Jeżeli p, to q
5. p wtedy i tylko wtedy, gdy q.

Jeżeli zamiast liter p, q wpiszemy w powyższe wyrażenia dowolne zdania, to z podanych schematów otrzymamy nowe zdania. Litery p, q grają tu podobną rolę, jak zmienne w matematyce. Np. w wyrażeniu $x + y$, x i y nie są liczbami, lecz na miejsce tych liter możemy wpisywać liczby. Przez analogię litery p, q itd. będziemy nazywać **zmiennymi zdaniowymi.** Oczywiście wyrażenia p lub q, czy p i q nie są zdaniami. Przez analogię z funkcjami liczbowymi, będziemy je nazywać **funkcjami zdaniowymi.** Z funkcji zdaniowych otrzymamy zdania, jeżeli za zmienne zdaniowe podstawimy konkretne zdania. Funkcje zdaniowe są to więc jak gdyby schematy zdań. Aby analogia z pojęciem funkcji w matematyce była zupełna, wprowadzimy jeszcze symboliczne oznaczenia spójników logicznych, według poniższego klucza:

1. $\neg$ — nieprawda że ...
2. $\vee$ — ... lub ...
3. $\&$ — ... i ...
4. $\supset$ — ... jeżeli, to ...
5. $\equiv$ — ... wtedy i tylko wtedy ...

A więc poprzednie przykłady możemy napisać:

1. $\neg p$
2. $p \vee q$
3. $p \,\&\, q$
4. $p \supset q$
5. $p \equiv q$

Podobieństwo do funkcji arytmetycznych jest teraz już zupełne. Oczywiście dowolne funkcje zdaniowe możemy połączyć nowym spójnikiem i otrzymamy znowu funkcję zdaniową, np. $\neg (p \vee q)$, $(p \vee q) \,\&\, (p \vee r)$ itp. Dla zaznaczenia jakie funkcje połączono spójnikiem wygodnie jest posługiwać się nawiasami w sposób identyczny, jak to czynimy w arytmetyce czy algebrze. Funkcje zdaniowe będziemy w dalszym ciągu oznaczać dużymi literami greckimi.

§ 4. Język rachunku zdań

Zgodnie z zasadami podanymi w pierwszych rozdziałach tej książki, określenie jakiejś teorii rozpoczynamy od podania języka tej teorii. Ponieważ rachunek zdań dotyczy funkcji zdaniowych, więc język tego rachunku musi być tak określony, aby można było w nim zapisywać dowolne funkcje zdaniowe. Przypominamy, że język teorii określamy podając:

1. Symbole początkowe języka,
2. reguły tworzenia wyrażeń poprawnych w języku.

Dla rachunku zdań język określimy następująco:

1. Symbole pierwotne
   - I. małe litery alfabetu $p, q, r, s, t, u, w, x, y, z$ (zmienne zdaniowe),
   - II. $\neg, \vee, \&, \supset, \equiv$ (spójniki logiczne),
   - III. nawiasy (,).
2. Wyrażenia poprawne
   - I. Zmienne zdaniowe są wyrażeniami poprawnymi,
   - II. jeżeli α i β są wyrażeniami poprawnymi, a Δ jest spójnikiem logicznym $\vee, \&, \supset, \equiv$, to wyrażenie $(\alpha \,\Delta\, \beta)$ jest również wyrażeniem poprawnym,
   - III. jeżeli α jest wyrażeniem poprawnym, to $\neg\, \alpha$ jest również wyrażeniem poprawnym.

Powyższy sposób określenia jest nazywany indukcyjnym. Na mocy pierwszego punktu definicji, wyrażenia $p, q, r, \ldots$ są wyrażeniami poprawnymi. Na mocy punktu trzeciego $\neg\, p$ jest wyrażeniem poprawnym, na mocy zaś punktu drugiego $(\neg\, p \vee q)$ jest wyrażeniem poprawnym. Również poprawnym jest wyrażenie $\neg\, (\neg\, p \vee q)$, natomiast wyrażenie $(p \vee \&)$ nie jest poprawne. Reguła ta pozwala więc na tworzenie wszystkich wyrażeń poprawnych w rachunku zdań. Wyrażenia te zawierają wszystkie nawiasy.

A więc np. poprzednio podany przykład funkcji zdaniowej $(p \vee q) \,\&\, (p \supset r)$ w przyjętym języku rachunku zdań będzie miał postać $((p \vee q) \,\&\, (p \supset r))$. Można by tu wprowadzić zasady, pozwalające opuszczać zbędne nawiasy, jak to się czyni w algebrze, jednakże do celów maszynowych wygodniejsza jest postać taka, w której występują wszystkie nawiasy. Wyrażenia poprawne będziemy też nazywać **formułami.**

§ 5. Spójnik główny

Do dalszych rozważań będzie nam potrzebne określenie spójnika głównego w funkcji zdaniowej. Zanim podamy dokładne określenie, rozpatrzymy najpierw kilka przykładów. Oczywiście, jeżeli funkcja zdaniowa zawiera tylko jeden spójnik, to jest on spójnikiem głównym. Np. w $\neg p$, negacja jest spójnikiem głównym, a w funkcji $(p \vee q)$ alternatywa jest spójnikiem głównym. W funkcji $((p \vee p) \& (p \supset r))$ spójnikiem głównym jest koniunkcja.

Jeżeli wyrażenie rozpoczyna się od symbolu negacji, to symbol negacji jest symbolem głównym. Np. w wyrażeniu $\neg (p \vee q)$ symbolem głównym jest pierwszy od lewej symbol, spójnik $\neg$. Podobnie w wyrażeniu $\neg ((p \& q) \vee r)$ spójnikiem głównym jest negacja, stojąca na początku wyrażenia. Dla wyrażeń nie rozpoczynających się od negacji, spójnik główny określimy za pomocą funkcji $F(\sigma_i)$, przypisującej kolejnym symbolom $\sigma_1, \sigma_2, \ldots, \sigma_n$ formuły F liczbę naturalną według poniższej zasady:

1. $F(\sigma_1) = 0$,
2. $F(\sigma_{i+1}) = F(\sigma_i) + 1$, jeżeli σ_{i+1} jest lewostronnym nawiasem lub zmienną,
3. $F(\sigma_{i+1}) = F(\sigma_i) - 1$, jeżeli σ_{i+1} jest prawostronnym nawiasem lub spójnikiem zdaniowym,
4. $F(\sigma_{i+1}) = F(\sigma_i)$, jeżeli σ_{i+1} jest negacją.

Spójnik, dla którego $F(\sigma_i) = 0$, nazwiemy **spójnikiem głównym.** Np.

	(	.(	¬	(	p	∨	q	)	∨	q	)	&	(	p	&	¬	r	)	)
Nr symbolu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$F(\sigma_i)$	0	1	1	2	3	2	3	2	1	2	1	0							

A więc w powyższym przykładzie symbol 12 jest spójnikiem głównym.

Wprowadzimy jeszcze pojęcie **argumentów** spójnika głównego. Jeżeli główny spójnik jest negacją, to argumentem tego spójnika jest wyrażenie po nim stojące. Np. argumentem $\neg$ w formule

$\neg ((p \vee q) \& r)$ jest formuła $((p \vee q) \& r)$. Dla pozostałych spójników, jeżeli Δ jest spójnikiem głównym w formule $(\Phi_1 \Delta \Phi_2)$, to argumentami Δ są formuły Φ_1 i Φ_2. Formułę Φ_1 nazwiemy **lewym argumentem** spójnika Δ, formułę zaś Φ_2 — **prawym argumentem** Δ. Np. lewym i prawym argumentem głównego spójnika w formule

$$((\neg (p \vee q) \vee q) \& (p \& \neg r))$$

są formuły $(\neg (p \vee q) \vee q)$ oraz $(p \& \neg r)$.

§ 6. Zdania zawsze prawdziwe

Z pojęciem zdania wiąże się zagadnienie jego prawdziwości. Zdanie prawdziwe, to tyle co zdanie zgodne z rzeczywistością. Np. zdanie „Mam tysiąc złotych" jest prawdziwe, jeżeli rzeczywiście tysiąc złotych posiadam. W przypadku przeciwnym, jeżeli nie mam tysiąca złotych, zdanie to jest fałszywe. Są jednak takie zdania, których prawdziwość, czy fałszywość nie zależy od konfrontacji z rzeczywistością. Jest ona wynikiem samej struktury tych zdań. Np. „Mam tysiąc złotych lub nie mam tysiąca złotych" jest zdaniem zawsze prawdziwym. Do stwierdzenia jego prawdziwości nie potrzeba uciekać się do sprawdzania tego co głosi zdanie ze stanem mojego posiadania. Niezależnie od stanu mojej kasy jest tak, jak właśnie głosi owo zdanie.

Zdanie „Mam tysiąc złotych i nie mam tysiąca złotych" jest natomiast zdaniem zawsze fałszywym. Każdy się zgodzi, że fałszywość tego zdania wynika z samej jego struktury, z rodzaju i rozmieszczenia występujących w nim spójników. Każde zdanie, które ma taką samą budowę, jak zdanie pierwsze, jest zawsze prawdziwe, a każde zdanie o strukturze zdania drugiego jest zawsze fałszywe. Możemy więc napisać schematy obu tych zdań w postaci:

$$(p \vee \neg p), \quad (p \& \neg p)$$

gdzie za p możemy podstawić dowolne zdania. Stawiając przed ostatnią formułą spójnik negacji

$$(p \vee \neg p), \quad \neg (p \& \neg p)$$

otrzymaliśmy dwa schematy zdań zawsze prawdziwych, niezależnie od treści zdania p. Nie są to jedyne schematy takich zdań

i można ich podać znacznie więcej. Schematy zdań zawsze prawdziwych będziemy nazywali **twierdzeniami, tezami** albo **tautologiami** rachunku zdań. Zgodnie z poprzednio podaną umową, twierdzenia teorii będziemy poprzedzać symbolem $\vdash$. A więc dla rachunku zdań

$$\vdash (p \vee \neg p), \quad \vdash \neg (p \,\&\, \neg p)$$

Rachunek zdań jest teorią badającą schematy zdań zawsze prawdziwych. Aby teorię tę określić, musimy podać jej aksjomaty, tj. wyjściowe schematy zdań zawsze prawdziwych oraz reguły pozwalające z aksjomatów otrzymywać twierdzenia, tj. schematy nowych zdań zawsze prawdziwych. W ten sposób możemy produkować dowolne schematy zdań zawsze prawdziwych.

Jak to uzasadnialiśmy w poprzedniej części, dla celów maszynowych konieczne jest przyjęcie takich reguł wnioskowania, które pozwalają na szukanie przesłanek dla zadanej hipotezy; a więc reguły takie pozwalają dla dowolnej formuły rachunku zdań znaleźć jej przesłanki, dla tych przesłanek — dalsze przesłanki itd., aż do otrzymania atomu.

Podaliśmy tutaj jedno sformułowanie rachunku zdań. Dla interesujących nas celów określenie to jest jednak nie wystarczające. W dalszym ciągu podamy inne sformułowanie rachunku zdań, na podstawie którego będziemy mogli badać w czysto formalny sposób czy zadana formuła rachunku zdań jest twierdzeniem tego rachunku czy też nie. Sformułowanie to będzie jednakże odbiegać od powszechnie spotykanego określenia rachunku zdań. Na ogół rachunek zdań jest formułowany w postaci teorii dedukcyjnej, do naszych celów zaś jest potrzebne sformułowanie w postaci teorii redukcyjnej, tj. w postaci takiej teorii, w której możliwe jest, za pomocą odpowiednich reguł, szukanie przesłanek dla zadanego wyrażenia. Oba sformułowania są oczywiście równoważne. Dla naszych celów jednakże pierwsze sformułowanie byłoby nieprzydatne, gdyż reguły teorii dedukcyjnej pozwalają na wyprowadzanie wniosków z przesłanek, a dla nas są niezbędne reguły pozwalające na postępowanie odwrotne.

Rozdział IX

SYSTEM GENTZENA (W SFORMUŁOWANIU HAO-WANGA)

§ 1. Pojęcie sekwentu

Przypominamy, że symbolami pierwotnymi rachunku zdań są:

$\neg$ — negacje,
$\vee$ — alternatywy,
$\&$ — koniunkcje,
$\supset$ — implikacje,
$\equiv$ — równoważności.

oraz litery $p, q, r, s, t, \ldots$. Małe litery będziemy nazywali **formułami elementarnymi.**

Jeżeli Φ i Ψ są formułami, to również następujące wyrażenia są formułami:

1. $\neg (\Phi)$,
2. $(\Phi \vee \Psi)$,
3. $(\Phi \,\&\, \Psi)$,
4. $(\Phi \supset \Psi)$,
5. $(\Phi \equiv \Psi)$.

Litery Π, Γ będą oznaczały ciągi formuł rachunku zdań. Ciągi formuł w szczególnym przypadku mogą się składać z jednej formuły, lub też formuły pustej, tj. formuły nie zawierającej żadnego symbolu.

Wyrażenie $\Pi \vdash \Gamma$ będziemy nazywali **sekwentem.** Ciąg Π nazwiemy **poprzednikiem,** ciąg Γ — **następnikiem** sekwentu. Np. sekwentami są następujące wyrażenia:

$$p \vee q \vdash q, p$$
$$p, q \vdash q$$
$$\vdash p \vee \neg q$$

Po obu stronach symbolu $\vdash$ w sekwencie mogą stać ciągi formuł, pojedyncze formuły zmienne, czy też formuły puste, jak np. w sekwencie $\vdash p \vee \neg q$. W systemie Gentzena operujemy nie formułami, a sekwentami. Aksjomaty jak i reguły wnioskowania odnosić się więc będą do sekwentów.

§ 2. Aksjomaty rachunku zdań

W rozpatrywanym systemie jest tylko jeden aksjomat następującej postaci:

$$A_1\colon\ \Delta \vdash \Theta$$

gdzie Δ i Θ są ciągami formuł elementarnych (tj. liter $p, q, r, s, t, \ldots$), zawierającymi choć jedną wspólną formułę elementarną. Np. sekwent

$$p, q \vdash p$$

jest aksjomatem, gdyż w poprzedniku i następniku są tylko formuły elementarne i jedna z nich, p, występuje zarówno w poprzedniku, jak i w następniku. Na tej samej zasadzie następujące sekwenty są aksjomatami.

$$p, q, r \vdash r, r$$
$$p \vdash p, q, r$$
$$r \vdash p, r$$

Natomiast sekwenty poniższe aksjomatami nie są:

$$p \vdash r$$
$$p, q, r \vdash s, t$$
$$q, p \vdash r, s$$

Jeżeli po obu stronach symbolu $\vdash$ występują tylko formuły elementarne, to sekwent taki nazwiemy **atomem.**

§ 3. Reguły wnioskowania

W systemie Gentzena stosowanych jest 10 reguł wnioskowania, które kolejno omówimy. Reguły te dotyczą przekształcania sekwentów. Ponieważ w twierdzeniach rachunku zdań występują spójniki logiczne, a w aksjomacie spójników tych nie ma, więc jeżeli chcemy z aksjomatu wyprodukować jakieś twierdzenie, to reguły twierdzenia muszą dotyczyć zasad wprowadzania spójników do sekwentu. Odwrotnie: jeżeli

chcemy stwierdzić, czy jakaś formuła rachunku zdań jest twierdzeniem, powinniśmy umieć rugować spójniki tak, aby w rezultacie dojść do aksjomatu. A więc w rozpatrywanym systemie rachunku zdań, reguły wnioskowania dotyczą wprowadzania i redukowania spójników. Jak wiemy, z maszynowego punktu widzenia interesujący jest przypadek redukowania formuły do aksjomatów. A więc będziemy się interesowali regułami eliminowania spójników. Zresztą w omawianym systemie reguły wnioskowania są „symetryczne" i można je stosować w postępowaniu „od aksjomatów do twierdzenia", jak i odwrotnie, tj. „od twierdzenia do aksjomatów".

Dla każdego spójnika podamy dwie reguły jego eliminowania. Jedną regułę będziemy stosować, gdy rugowany spójnik znajduje się w poprzedniku, drugą zaś, gdy rugowany spójnik jest w następniku. Ponieważ przyjęliśmy 5 spójników logicznych, więc potrzeba dziesięciu reguł wnioskowania.

Reguły są stosowane zawsze do spójnika głównego pierwszej z lewej strony formuły nieelementarnej w redukowanym sekwencie. Spójnik ten będziemy nazywali **głównym spójnikiem** sekwentu. Proces eliminowania powtarzany jest tak długo, aż otrzymamy w wyniku same atomy, tj. sekwenty nie zawierające żadnych spójników logicznych. Jeżeli wszystkie atomy otrzymane w wyniku redukcji są aksjomatami, to badana formuła jest twierdzeniem rachunku zdań. Wyjaśnimy to dokładniej na przykładach, podając przed tym wszystkie reguły wnioskowania.

We wszystkich podanych dalej regułach litery Δ i Θ oznaczają ciągi formuł elementarnych, np. p, p, q, r itp. Reguły dotyczące spójnika w następniku będziemy oznaczali literą **R** z odpowiednim wskaźnikiem u dołu, natomiast reguły dotyczące spójnika w poprzedniku, oznaczymy literą **R** ze wskaźnikiem u dołu oraz gwiazdką u góry. Wygodnie jest pisać reguły wnioskowania nie w postaci takiej, jak to robiliśmy dotychczas, lecz pisząc je jak gdyby w postaci ułamka, którego licznik zawsze będzie zawierał badany sekwent, a mianownik — jego przesłanki.

1. Negacja

$$\mathbf{R}_1\ \frac{\Delta \vdash \Theta, \neg \Phi, \Psi}{\Phi, \Delta \vdash \Theta, \Psi} \qquad \overset{*}{\mathbf{R}}_1\ \frac{\Delta, \neg \Phi, \Gamma \vdash \Pi}{\Theta, \Gamma \vdash \Gamma, \Phi}$$

Reguły te mimo groźnego wyglądu są bardzo proste. Pierwsza z nich mówi:

Jeżeli głównym spójnikiem sekwentu jest negacja w następniku, to formułę będącą argumentem tej negacji możemy przenieść do poprzednika.

Np. stosując do sekwentu

$$p \vdash \neg (r \,\&\, s), q$$

regułę $\mathbf{R}_1$, otrzymamy

$$(r \,\&\, s), p \vdash q$$

W tym przykładzie Δ jest ciągiem składającym się z jednej litery p, Θ — jest ciągiem pustym, Φ ma postać $(r \,\&\, s)$, oraz Ψ jest również formułą elementarną q. Pisząc inaczej, otrzymamy

$$\frac{p \vdash \neg (r \,\&\, s), q}{(r \,\&\, s), p \vdash q}$$

Druga reguła R_1^* jest podobna, z tą różnicą, że odnosi się ona do negacji w poprzedniku. Reguła ta brzmi:

Jeżeli głównym spójnikiem sekwentu jest negacja w poprzedniku, to formuła będąca argumentem tej negacji może być przeniesiona do następnika.

Np. jeżeli sekwent ma postać

$$p, \neg (p \vee q) \vdash (r \,\&\, s)$$

to w wyniku reguły $\overset{*}{\mathbf{R}}_1$ otrzymamy

$$p \vdash (r \,\&\, s), (p \vee q)$$

lub, pisząc krócej

$$\frac{p, \neg (p \vee q) \vdash (r \,\&\, s)}{p \vdash (r \,\&\, s), (p \vee q)}$$

Obie te reguły przypominają nieco przenoszenie liczby ze zmianą znaku na drugą stronę równości. Równości odpowiada tutaj znak $\vdash$.

Zwracamy jeszcze raz uwagę, że do zadanego sekwentu można zastosować tylko jedną z reguł, gdyż wyznaczają jednoznacznie główny spójnik w sekwencie. Taki spójnik jest oczywiście jeden, a więc postępowanie jest jednoznaczne.

2. Koniunkcja

$$\mathbf{R}_2 \quad \frac{\Delta \vdash \Theta, \Phi \,\&\, \Psi, \Gamma}{\Delta \vdash \Theta, \Phi, \Gamma; \Delta \vdash \Theta, \Psi, \Gamma}$$

$$\overset{*}{\mathbf{R}}_2 \quad \frac{\Theta, \Phi \,\&\, \Psi, \Gamma \vdash \Pi}{\Theta, \Phi, \Psi, \Gamma \vdash \Pi}$$

Reguły te są również bardzo proste. Regułę $\mathbf{R}_2$ należy rozumieć tak:

Jeżeli głównym spójnikiem sekwentu jest koniunkcja w następniku, to sekwent ten możemy zastąpić dwoma sekwentami, z których pierwszy będzie zawierał lewy argument koniunkcji, a drugi — prawy argument koniunkcji.

Np. sekwent

$$p, q \vdash r, ((p \vee q) \,\&\, r), r$$

możemy zastąpić sekwentem

$$p, q \vdash r, (p \,\&\, q), r$$

oraz

$$p, q \vdash r, r, r$$

co zapiszemy

$$\frac{p, q \vdash r, ((p \vee q) \,\&\, r), r}{p, q \vdash r, (p \,\&\, q), r; p, q \vdash r, r, r}$$

Sekwenty stojące w „mianowniku" są więc przesłankami dla sekwentu napisanego w „liczniku".

Reguła $\overset{*}{\mathbf{R}}_2$ jest jeszcze prostsza. Brzmi ona:

Jeżeli głównym spójnikiem sekwentu jest koniunkcja w poprzedniku, to symbol koniunkcji możemy zastąpić przecinkiem.

Np. Na podstawie reguły $\overset{*}{\mathbf{R}}_2$ z sekwentu

$$p, (q \,\&\, r), (p \vee s) \vdash \neg (p \,\&\, s)$$

otrzymamy

$$p, q, r, (p \vee s) \vdash \neg (p \,\&\, s)$$

lub, pisząc krócej:

$$\frac{p, (q \,\&\, r), (p \vee s) \vdash \neg (p \,\&\, s)}{p, q, r, (p \vee s) \vdash \neg (p \,\&\, s)}$$

3. Alternatywa

$$\mathbf{R}_3\ \frac{\Delta \vdash \Theta, \Phi \vee \Psi, \Gamma}{\Delta \vdash \Theta, \Phi, \Psi, \Gamma}$$

$$\overset{*}{\mathbf{R}}_3\ \frac{\Theta, \Phi \vee \Psi, \Gamma \vdash \Pi}{\Theta, \Phi, \Gamma \vdash \Pi;\ \Theta, \Psi, \Gamma \vdash \Pi}$$

Obie te reguły $\mathbf{R}_3$ i $\overset{*}{\mathbf{R}}_3$ bardzo przypominają reguły eliminowania koniunkcji. Są względem nich tylko w pewnym sensie odwrotne. Reguła $\mathbf{R}_3$ jest następująca:

Jeżeli głównym spójnikiem sekwentu jest alternatywa w następniku, to możemy ją zastąpić przecinkiem.

Np. stosując do sekwentu

$$p \vdash r, (p \vee q), s$$

regułę $\mathbf{R}_3$, otrzymamy

$$p \vdash r, p, q, s$$

Pisząc inaczej, otrzymamy

$$\frac{p \vdash r, (p \vee q), s}{p \vdash r, p, q, s}$$

Reguła $\overset{*}{\mathbf{R}}_3$ jest podobna do reguły $\mathbf{R}_2$:

Jeżeli głównym spójnikiem sekwentu jest alternatywa w poprzedniku, to sekwent ten możemy zastąpić dwoma sekwentami, z których pierwszy będzie zawierał lewy argument alternatywy, a drugi — prawy argument alternatywy.

Np. z sekwentu

$$p, (p \vee q), (p \,\&\, s) \vdash r$$

stosując regułę $\mathbf{R}_3$, otrzymamy sekwent

$$p, p, (p \,\&\, s) \vdash r$$

oraz

$$p, q, (p \,\&\, s) \vdash r$$

czyli

$$\frac{p, (p \vee q), (p \,\&\, s) \vdash r}{p, p, (p \,\&\, s) \vdash r;\ p, q, (p \,\&\, s) \vdash r}$$

4. Implikacja

$$\mathbf{R}_4\ \frac{\Delta \vdash \Theta, \Phi \supset \Psi, \Gamma}{\Delta\ \Phi \vdash \Theta, \Psi, \Gamma}$$

$$\overset{*}{\mathbf{R}}_4\ \frac{\Theta, \Phi \supset \Psi, \Gamma \vdash \Pi}{\Theta, \Psi, \Gamma \vdash \Pi;\ \Theta, \Gamma \vdash \Pi, \Phi}$$

Regułę $\mathbf{R}_4$ należy rozumieć następująco:

Jeżeli głównym spójnikiem sekwentu jest implikacja w następniku, to pierwszy człon implikacji możemy przenieść do poprzednika.

Np. w sekwencie

$$p \vdash r, p \supset q, r$$

głównym spójnikiem jest implikacja, a więc stosując regułę $\mathbf{R}_4$, otrzymamy

$$p, p \vdash r, q, r$$

albo, pisząc inaczej:

$$\frac{p \vdash r, (p \supset q), r}{p, p \vdash r, q, r}$$

Reguła $\overset{*}{\mathbf{R}}_4$ brzmi:

Jeżeli głównym spójnikiem sekwentu jest implikacja w poprzedniku, to sekwent ten możemy zastąpić dwoma sekwentami, z których pierwszy zawiera drugi człon implikacji, w drugim natomiast zawarty jest pierwszy człon implikacji.

Np. z sekwentu

$$p, p \supset q \vdash r$$

otrzymamy sekwent

$$p, q \vdash r$$

oraz sekwent

$$p \vdash r, p$$

czyli

$$\frac{p, p \supset q \vdash r}{p, q \vdash r;\ p \vdash r, p}$$

Pozostały nam do omówienia jeszcze dwie ostatnie reguły dotyczące równoważności.

$$\mathbf{R}_5 \quad \frac{\Delta \vdash \Theta, \Phi \equiv \Psi, \Gamma}{\Phi, \Delta \vdash \Theta, \Psi, \Gamma; \Psi, \Delta \vdash \Theta, \Phi, \Gamma}$$

$$\mathbf{R}_5^* \quad \frac{\Theta, \Phi \equiv \Psi, \Gamma \vdash \Pi}{\Phi, \Psi, \Theta, \Gamma \vdash \Pi; \Theta, \Gamma \vdash \Pi, \Phi, \Psi}$$

Reguła $\mathbf{R}_5$ brzmi:

Jeżeli głównym spójnikiem sekwentu jest równoważność w następniku, to sekwent ten możemy zastąpić dwoma sekwentami, z których pierwszy zawiera lewy argument równoważności, a drugi — prawy argument równoważności.

Np. stosując regułę $\mathbf{R}_5$ do sekwentu

$$p \vdash p \equiv q, r$$

otrzymamy

$$p \vdash p, r$$

oraz

$$p \vdash q, r$$

lub krócej

$$\frac{p \vdash p \equiv q, r}{p \vdash p, r; p \vdash q, r}$$

Wreszcie ostatnia reguła mówi co następuje:

Jeżeli głównym spójnikiem sekwentu jest równoważność w poprzedniku, to sekwent ten możemy zastąpić dwoma sekwentami, z których pierwszy zawiera oba argumenty równoważności w poprzedniku, a drugi — oba argumenty równoważności w następniku.

Np. sekwent

$$p, p \equiv q \vdash r$$

po zastosowaniu reguły R_5^* da w wyniku

$$p, p, q \vdash r$$

oraz

$$p \vdash r, p, q$$

A więc

$$\frac{p, p \equiv q \vdash r}{p, p, q \vdash r; p \vdash r, p, q}$$

Reguły te są w istocie bardzo proste i posługiwanie się nimi nie sprawia trudności. Pozwalają one w sposób systematyczny na rugowanie wszystkich spójników z formuły. Niestety z braku miejsca nie podaliśmy uzasadnienia tych reguł, jednakże z interesującego nas punktu widzenia nie jest to może zbyt ważne. Interesują nas bowiem ciągi symboli, sposoby ich przekształcania oraz maszynowa realizacja takich procesów [1]).

Zastosowanie podanych reguł do znajdywania dowodów twierdzeń w rachunku zdań podamy w następnym paragrafie.

§ 4. Przykłady dowodów

Przerobienie kilku dowodów za pomocą zadanych reguł pozwoli Czytelnikowi na powiązanie metody Gentzena z rozważaniami na początku książki. Zacznijmy od najprostszego przykładu twierdzenia, które już cytowaliśmy w ostatnim paragrafie poprzedniego rozdziału, a mianowicie

$$\vdash p \vee \neg p$$

Twierdzenie to jest szczególnym przypadkiem sekwentu. Poprzednikiem tego sekwentu jest formuła pusta, to jest formuła nie zawierająca żadnego symbolu. Następnikiem sekwentu jest zaś jedna formuła $p \vee \neg p$. Głównym spójnikiem sekwentu jest spójnik $\vee$. Ponieważ występuje on w następniku, możemy do tego sekwentu zastosować regułę $\mathbf{R}_3$ i otrzymamy

$$\frac{\vdash p \vee \neg p}{\vdash p, \neg p}$$

W otrzymanym sekwencie $\vdash p, \neg p$ jedynym spójnikiem jest ne-

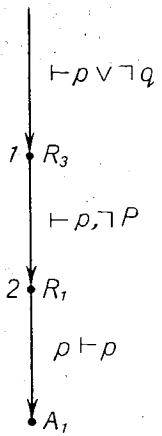
[1]) Niestety w literaturze polskiej system Gentzena w potrzebnym nam sformułowaniu nie był dotąd omówiony, nie możemy więc Czytelnika odesłać do odpowiednich podręczników.

gacja w następniku, a więc na podstawie reguły $\mathbf{R}_1$ otrzymamy

$$\frac{\vdash p, \neg p}{p \vdash p}$$

Ponieważ w sekwencie $p \vdash p$ nie występują już żadne spójniki logiczne, więc jest on atomem. W atomie tym litera p powtarza się po obu stronach znaku $\vdash$, więc atom ten jest aksjomatem. Formuła $p \vee \neg p$ jest więc twierdzeniem rachunku zdań, a dowód tego twierdzenia przebiega następująco:

1.	$\vdash p \vee \neg p$	$\mathbf{R}_3$, 2
2.	$\vdash p, \neg p$	$\mathbf{R}_1$, 3
3.	$p \vdash p$	A_1



Rys. 15

W dowodzie podkreślono wiersz będący aksjomatem. Drzewo tego dowodu jest linią prostą, pokazaną na rys. 15, gdzie numery obok punktów oznaczają kolejność wykonywania operacji.

Przypominamy, że z prawej strony tabelki podano zastosowaną regułę oraz numer wiersza, w którym znajduje się otrzymana przesłanka.

Rozpatrzmy drugie, podawane już twierdzenie rachunku zdań:

$$\vdash \neg (p \,\&\, \neg p)$$

Poprzednikiem, jak i w poprzednim twierdzeniu, jest formuła pusta. Następnik tego sekwentu jest również jedną formułą $\neg (p \,\&\, \neg p)$. Głównym spójnikiem tego sekwentu jest negacja przed nawiasem w następniku, a więc do jej wyeliminowania zastosujemy regułę $\mathbf{R}_1$ i otrzymamy

$$\frac{\vdash \neg (p \,\&\, \neg p)}{(p \,\&\, \neg p) \vdash}$$

W otrzymanym sekwencie $(p \,\&\, \neg p)$ głównym spójnikiem jest koniunkcja w poprzedniku; na podstawie reguły R_2^* otrzymamy

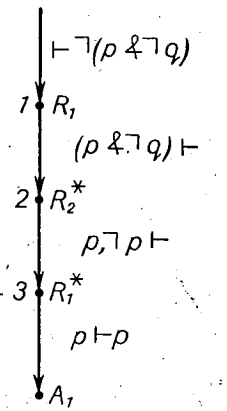
$$\frac{(p \,\&\, \neg p) \vdash}{p, \neg p \vdash}$$

Stosując do otrzymanego sekwentu regułę $\mathbf{R}_1^*$, mamy

$$\frac{p, \neg p \vdash}{p \vdash p}$$

Formuła $\neg (p \,\&\, \neg p)$ jest więc również twierdzeniem rachunku zdań. Dowód jej ma postać:

1.	$\vdash \neg (p \,\&\, \neg p)$	$\mathbf{R}_1$, 2
2.	$(p \,\&\, \neg p) \vdash$	R_2^*, 3
3.	$p, \neg p \vdash$	R_1^*, 4
4.	$p \vdash p$	A_1



Rys. 16

Drzewo tego dowodu przedstawia rys. 16.

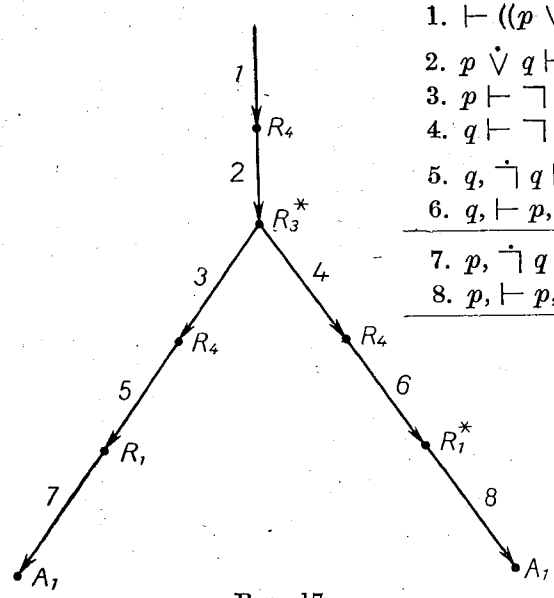
Rozpatrzmy teraz nieco trudniejszy dowód twierdzenia

$$\vdash ((p \vee q) \supset (\neg q \supset p))$$

Nie będziemy już szczegółowo omawiali kolejności operacji, a podamy od razu dowód tego twierdzenia. Dla ułatwienia, w każdym kroku będziemy zaznaczali kropką u góry spójnik główny sekwentu.

1.	$\vdash ((p \vee q) \dot{\supset} (\neg q \supset p))$	$\mathbf{R}_4$, 2
2.	$p \dot{\vee} q \vdash \neg q \supset p$	$\mathbf{R}_3^*$, 3, 4
3.	$p \vdash \neg q \dot{\supset} p$	$\mathbf{R}_4$, 5
4.	$q \vdash \neg q \supset p$	$\mathbf{R}_4$, 6
5.	$p, \dot{\neg} q \vdash p$	$\mathbf{R}^*$, 7
6.	$q, \dot{\neg} q \vdash p$	$\mathbf{R}_1$, 8
7.	$p, \vdash p, q$	A_1
8.	$q, \vdash p, q$	A_1

Badana formuła jest więc twierdzeniem. Drzewo twierdzenia podano na rys. 17, gdzie dla prostoty nie pisano odpowiadających im formuł. Zamiast nich zostały podane numery wierszy formuł. Wystąpił tu już problem porządku wykonywania operacji. Widzimy, że operacje są tu wykonane w porządku $\overline{P}$. Numery wierszy można uważać też za numery operacji. Dla porządku $\overline{W}$ dowód ten ma postać:



Rys. 17

1. $\vdash ((p \vee q) \dot{\supset} (\neg q \supset p))$	$\mathbf{R}_4$, 2
2. $p \dot{\vee} q \vdash \neg q \supset p$	R_3^*, 3, 4
3. $p \vdash \neg q \dot{\supset} p$	$\mathbf{R}_4$, 7
4. $q \vdash \neg q \dot{\supset} p$	$\mathbf{R}_4$, 5
5. $q, \dot{\neg} q \vdash p$	R_1^*, 6
6. $q, \vdash p, q$	A_1
7. $p, \dot{\neg} q \vdash p$	R_1^*, 8
8. $p, \vdash p, q$	A_1

Drzewo tego dowodu jest identyczne jak poprzednio, inna jest tylko numeracja jego gałęzi.

Wszędzie tam, gdzie otrzymamy dwie przesłanki, występuje problem porządku stosowania reguł wnioskowania $\overline{P}$ lub $\overline{W}$. Będziemy stosować w dalszych przykładach tylko porządek $\overline{P}$, proponując Czytelnikowi przerobienie tych dowodów dla porządku $\overline{W}$.

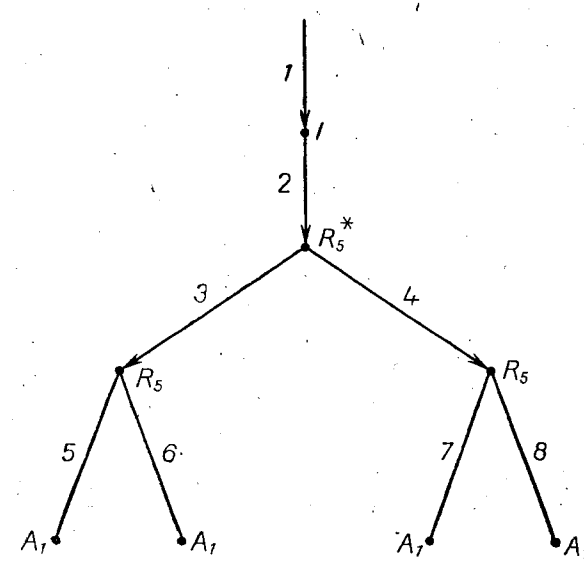
Jeszcze kilka prostych przykładów dowodów pozwoli nam na zapoznanie się z wszystkimi regułami wnioskowania.

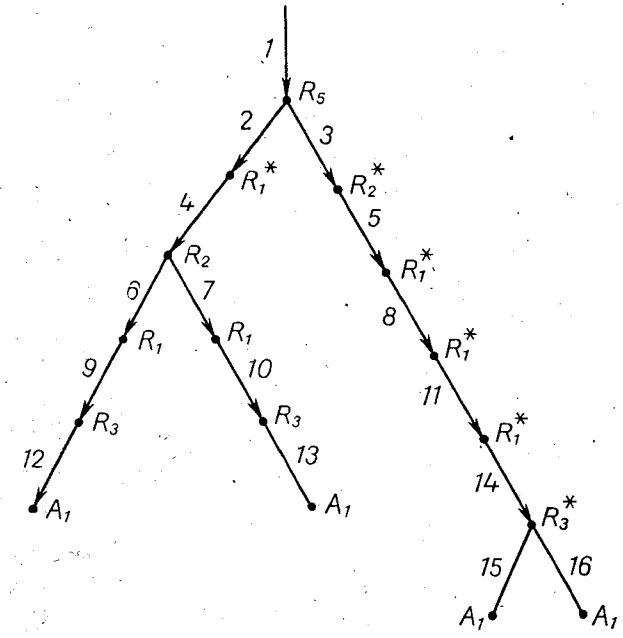
Dowód twierdzenia $\vdash ((p \equiv q) \supset (q \equiv p))$

1. $\vdash ((p \equiv q) \dot{\supset} (q \equiv)) \; p$	$\mathbf{R}_4$, 2
2. $(p \dot{\equiv} q) \vdash (q \equiv p)$	R_5^*, 3, 4
3. $p, q \vdash (q \dot{\equiv} p)$	$\mathbf{R}_5$, 5, 6
4. $\vdash (q \dot{\equiv} p), p, q$	$\mathbf{R}_5$, 7, 8,
5. $q, p, q \vdash p$	A_1
6. $p, p, q \vdash q$	A_1
7. $q \vdash p, p, q$	A_1
8. $p \vdash q, p, q$	A_1

Rys. 18 przedstawia drzewo tego dowodu.

Rys. 18





Rys. 19

I wreszcie ostatni dowód twierdzenia $\vdash \neg (p \vee q) \equiv (\neg p \& \neg q)$

1. $\vdash \neg(p \vee q) \dot{\equiv} (\neg p \& \neg q)$	$\mathbf{R}_5$, 2, 3
2. $\dot{\neg} (p \vee q) \vdash \neg p \& \neg q$	$\mathbf{R}_1$, 4
3. $\neg p \dot{\&} \neg q \vdash \neg (p \vee q)$	$\mathbf{R}_2$, 5
4. $\vdash \neg p \dot{\&} \neg q, p \vee q$	$\mathbf{R}_2$, 6, 7
5. $\dot{\neg} p, \neg q \vdash \neg (p \vee q)$	$\mathbf{R}_1$, 8
6. $\vdash \dot{\neg} p, p \vee q$	$\mathbf{R}_1$, 9
7. $\vdash \dot{\neg} q, p \vee q$	$\mathbf{R}_1$, 10
8. $\dot{\neg} q \vdash \neg (p \vee q), p$	$\mathbf{R}_1$, 11
9. $p \vdash p \dot{\vee} q$	$\mathbf{R}_3$, 12
10. $q \vdash p \dot{\vee} q$	$\mathbf{R}_3$, 13
11. $\vdash \dot{\neg} (p \vee q), p, q$	$\mathbf{R}_1$, 14
12. $p \vdash p, q$	A_1
13. $q \vdash p, q$	A_1
14. $p \dot{\vee} q \vdash p, q$	$\mathbf{R}_3$, 15, 16
15. $p \vdash p, q$	A_1
16. $q \vdash p, q$	A_1

Drzewo tego dowodu pokazano na rys. 19. Na tym przykłady dowodów zakończymy. Na zakończenie tego paragrafu zbadajmy jeszcze jedną formułę rachunku zdań, która, jak się okaże, twierdzeniem nie jest. Rozpatrzmy mianowicie formułę

$$((p \& q \vee q) \supset (p \& q))$$

Dowodu będziemy szukać tak, jak to czyniliśmy dotychczas.

1. $\vdash (((p \& q) \vee q) \dot{\supset} (p \& q))$	$\mathbf{R}_4$, 2
2. $((p \& q) \dot{\vee} q) \vdash (p \& q)$	R_3^*, 3, 4
3. $(p \dot{\&} q) \vdash (p \& q)$	$\mathbf{R}_2$, 5
4. $q \vdash (p \dot{\&} q)$	$\mathbf{R}_2$, 6, 7
5. $p, q \vdash (p \dot{\&} q)$	
6. $q \vdash p$	fałszywe
7. $q \vdash q$	

Sekwent 6. $q \vdash p$ jest atomem, ale nie jest aksjomatem. Badana formuła nie jest więc twierdzeniem rachunku zdań.

Pokazana metoda pozwala dla każdej formuły rachunku zdań orzec, czy formuła ta jest twierdzeniem, czy też nie. Metoda ta jest jednoznaczna. Na podstawie analizy formuły wiemy, którą z reguł wnioskowania mamy stosować w każdym kroku dowodzenia. Szukanie dowodu nie jest tutaj uzależnione od przypadku, czy szczęścia i przypomina nieco wykonywanie rachunku.

Mając taką metodę, można zbudować maszynę, która by tę metodę realizowała. Schemat ogólny takiej maszyny już dyskutowaliśmy. Przypominamy, że schemat ten nie zależy od konkretnej teorii. A więc schemat maszyny pokazany na rys. 14 ważny jest również dla rachunku zdań. Jedynie operator O musiałby być tak zbudowany, aby mógł realizować wszystkie dziesięć reguł systemu Gentzena. Gdybyśmy chcieli zastosować do dowodzenia twierdzeń w rachunku zdań nie maszynę bezadresową a maszynę adresową, również ogólna struktura programu dowodzącego nie uległaby zmianie. Natomiast inaczej, niż to opisaliśmy dla teorii $\mathbf{T}_1'$, wyglądałyby programy realizujące poszczególne reguły wnioskowania. W teorii $\mathbf{T}_1'$ sytuacja była bardzo prosta: mieliśmy jedną regułę wnioskowania i reguła ta była całkiem prymitywna. W rachunku zdań reguł wnioskowania jest dziesięć, a więc program na podstawie analizy badanego sekwentu musi wybrać odpowiednią regułę. Realizacja poszczególnych reguł jest również, z punktu widzenia programu, dość skomplikowana. Nie będziemy dlatego tej sprawy omawiać szczegółowo. Tym niemniej w następnym rozdziale spróbujemy pokazać niektóre ciekawe fragmenty programu realizującego reguły wnioskowania w systemie Gentzena, pozwoli to bowiem Czytelnikowi na lepsze zorientowanie się w wymaganiach odnośnie struktury reguł wnioskowania w teoriach matematycznych, które mają być badane maszynowo. Okaże się, jak to już wielokrotnie podkreślano, że reguły wygodne do manipulowania przez człowieka, nie zawsze są najlepsze do celów maszynowych i odwrotnie.

Z drugiej strony, zapoznanie się z niektórymi fragmentami programu, realizującego reguły wnioskowania, pozwoli na lepsze poznanie własności maszyn cyfrowych.

Rozdział X

PROGRAMY OBLICZANIA PRZESŁANEK

§ 1. Ogólna struktura programu

W rozdziale tym będziemy zajmowali się zagadnieniem, w jaki sposób należy ułożyć program, aby dla zadanego sekwentu wyszukał on spójnik główny, sprawdził, czy spójnik ten znajduje się przed, czy po symbolu $\vdash$ i zależnie od rezultatu sprawdzenia, zastosował właściwą regułę wnioskowania — a ściślej program tę regułę realizujący — i obliczył przesłanki tego sekwentu.

W przypadku, jeżeli sekwent żadnych logicznych spójników nie zawiera, program powinien sprawdzić, czy sekwent ten jest aksjomatem.

Zadanie to jest więc bardzo proste, ale — jak zobaczymy — program jego rozwiązania nie jest już tak bardzo prosty, jak programy rozważane do tej pory. Przy studiowaniu tego programu warto zwrócić uwagę na fakt, jak dalece istniejące maszyny cyfrowe są w gruncie rzeczy nie przystosowane do wykonywania najprostszych nawet manipulacji na symbolach i ile trzeba wykonać zabiegów, aby znaleźć wreszcie przesłanki badanego sekwentu. Człowiek posługujący się tymi samymi regułami wnioskowania nie traktuje każdego symbolu sekwentu oddzielnie, a rozpatruje je w całości, dlatego sposób postępowania człowieka i maszyny jest tutaj całkiem inny. Uwagi te są adresowane przede wszystkim do tych, którzy w maszynach matematycznych widzą mniej czy więcej udaną kopię mózgu ludzkiego. Rzecz jasna, porównanie takie dzisiaj nie może być traktowane poważnie.

Wróćmy do tematu głównego. Przyjmiemy, że symbole badanego sekwentu znajdują się w kolejnych miejscach magazynu. W podobny sposób mają być umieszczone w magazynie przesłanki.

Pierwszym zadaniem programu jest odszukanie głównego spójnika sekwentu.

Szukanie głównego spójnika następuje w ten sposób, że sekwent jest czytany symbol po symbolu tak długo, aż zostanie odczytany pierwszy nawias lewostronny lub symbol negacji. Pierwsza odczytana negacja przed nawiasem jest oczywiście spójnikiem głównym. Jeżeli natomiast najpierw został odczytany nawias lewostronny, to maszyna przechodzi do innego programu, który czyta dalsze symbole sekwentu, obliczając dla nich jednocześnie wartość funkcji $F(\sigma_i)$. Czynność ta jest powtarzana tak długo, aż $F(\sigma_i) = 0$. Wtedy σ_i jest spójnikiem głównym sekwentu.

Znalezienie spójnika głównego jest czynnością dość ciekawą i charakterystyczną dla tego typu zastosowań maszyn, dlatego omówimy je dość szczegółowo w następnym paragrafie.

§ 2. Program szukania głównego spójnika

Zanim przystąpimy do omówienia programu, wprowadzimy najpierw potrzebne oznaczenia. Przede wszystkim, symbole języka rachunku zdań są ponumerowane i w magazynie są zapisywane nie symbole, a ich numery. Dla naszych celów wygodne będzie ponumerowanie wszystkich symboli języka kolejnymi liczbami nieparzystymi, według poniższej tabelki:

Symbol	Nazwa	Nr symbolu N
$\neg$	Negacja	1
$\&$	Koniunkcja	3
$\vee$	Alternatywa	5
$\supset$	Implikacja	7
$\equiv$	Równoważność	9
p	Zmienna	11
q	Zmienna	13
r	Zmienna	15
)	Nawias prawostronny	17

Symbol	Nazwa	Nr symbolu N
(	Nawias lewostronny	19
$\vdash$	Symbol asercji	21
,	Przecinek	23
;	Średnik	25

Dla prostoty, w tabelce podaliśmy tylko trzy zmienne p, q, r. Nic nie przeszkadza, aby ilość zmiennych była większa.

Ponadto przyjmiemy, że:

h_0 — jest adresem pierwszego symbolu sekwentu,
h_1 — jest adresem pierwszego symbolu pierwszej przesłanki,
h_2 — jest adresem pierwszego symbolu drugiej przesłanki (o ile taka istnieje).

Dalej założymy, że pierwsza instrukcja programu realizującego regułę $\mathbf{R}_i$ znajduje się w miejscu o adresie r_i i podobnie dla reguły $\mathbf{R}_i^*$, tzn. programy realizujące odpowiednie reguły wnioskowania zaczynają się w miejscach:

Adres pierwszej instrukcji programu	Reguła wnioskowania	
r_1	$\mathbf{R}_1$	Negacja
r_1^*	$\mathbf{R}_1^*$	
r_2	$\mathbf{R}_2$	Koniunkcja
r_2^*	R_2^*	
r_3	$\mathbf{R}_3$	Alternatywa
r_3^*	$\mathbf{R}_3^*$	
r_4	$\mathbf{R}_4$	Implikacja
r_4^*	$\mathbf{R}_4^*$	
r_5	$\mathbf{R}_5$	Równoważność
r_5^*	$\mathbf{R}_5^*$	

Poza tym będzie nam potrzebne ustalenie adresów pierwszych instrukcji programu obliczania funkcji $F(\sigma_i)$ oraz programu sprawdzającego, czy atom jest aksjomatem:

b_2 — adres pierwszej instrukcji programu obliczania $F(\sigma_i)$,
c_2 — adres pierwszej instrukcji programu sprawdzającego, czy sekwent jest aksjomatem.

Teraz już możemy przystąpić do omówienia programu szukania głównego spójnika. Program podzielimy dla przejrzystości na trzy bloki A_1, A_2, A_3. Adresy w blokach będziemy oznaczali zgodnie z poprzednio przyjętymi zasadami, tj. pierwszym adresem każdego bloku A_i jest a_i. Dla bloku A_1 pierwszym adresem jest więc a_1, dla bloku A_2 — a_2 i dla bloku A_3 — a_3. Dalsze kolejne adresy uzyskamy zwiększając adres poprzedni o 1. Rolę i znaczenie każdego bloku omówimy oddzielnie.

Blok A_1

W bloku A_1 znajdują się wszystkie stałe i wielkości pomocnicze, konieczne do zrealizowania programu szukania głównego spójnika.

a_1	h_0
$a_1 + 1$	i
$a_1 + 2$	k
$a_1 + 3$	a_3

Pod adresem a_1 znajduje się adres pierwszego symbolu badanego sekwentu.

W miejscu o adresie $a_1 + 1$ znajduje się adres aktualnie czytanego symbolu sekwentu.

Następne miejsce $a_1 + 2$ zawiera liczbę k, która może być tylko zerem lub jedynką. $k = 0$ wtedy i tylko wtedy, gdy aktualnie czytany symbol sekwentu znajduje się przed znakiem asercji, oraz $k = 1$, gdy czytany symbol jest po symbolu asercji. A więc na początku przyjmiemy $k = 0$. Jeżeli w trakcie czytania sekwentu pojawi się symbol $\vdash$, to do miejsca $a_1 + 2$ dodawana jest jedynka.

W następnym miejscu $a_1 + 3$ bloku A_1 znajduje się adres początkowy bloku A_3.

Blok A_2

W bloku A_2 znajduje się właściwy program czytania symboli sekwentu. Program ten jest następujący:

a_2	$\leftarrow a_1$
$a_2 + 1$	$\rightarrow a_1 + 1$
$a_2 + 2$	$\leftarrow (a_2 + 1)$
$a_2 + 3$	$+ a_1 + 2$
$a_2 + 4$	$+ a_1 + 3$
$a_2 + 5$	$N a_1 + 1$
$a_2 + 6$	! (0)
$a_2 + 7$	$N a_1 + 2$
$a_2 + 8$	$! a_2 + 2$

Pierwsze dwie instrukcje $\leftarrow a_1$, $\rightarrow a_2 + 1$ przesyłają h_0 do miejsca $a_1 + 1$. Z miejsca tego będzie zawsze brany adres symbolu, który ma być odczytany.

Następna instrukcja $\leftarrow (a_2 + 1)$ powoduje przesłanie pierwszego symbolu sekwentu do akumulatora. Jak pamiętamy, przyjęliśmy założenie, że w magazynie są zapisane nie symbole, ale ich numery według podanej w poprzednim paragrafie tabelki. Tak więc, w wyniku instrukcji $\leftarrow (a_2 + 1)$ znajdzie się w akumulatorze nie kolejny symbol sekwentu, a jego numer. Np. jeżeli pierwszym czytanym symbolem była litera p, to w akumulatorze znajdzie się liczba 11.

Następna instrukcja $+ a_1 + 2$ dodaje do akumulatora liczbę k, zapisaną pod adresem $a_1 + 2$. Ponieważ w naszym przypadku $k = 0$, więc w akumulatorze nadal znajduje się liczba 11.

Dalsza instrukcja $+ a_1 + 3$ dodaje do akumulatora adres początkowy a_3 bloku A_3, tak że dla rozpatrywanego przykładu znajdzie się w akumulatorze liczba $a_3 + 11$.

Instrukcja $N a_1 + 1$ przygotowuje następny symbol sekwentu do odczytu.

Wreszcie instrukcja ! (0) powoduje przejście maszyny do bloku A_3, pod adres uformowany w akumulatorze, czyli w rozpatrywanym przykładzie pod adres $a_3 + 11$.

A więc blok A_2 formuje adres w bloku A_3, do którego ma przejść maszyna zależnie od odczytanego symbolu sekwentu. Jeżeli odczytywany symbol znajdowałby się po znaku asercji, to w akumulatorze zostałby uformowany adres o jeden większy, np. zamiast $a_3 + 11$ — adres $a_3 + 12$.

Odczytanie znaku asercji za pomocą instrukcji $N a_1 + 2$ powoduje umieszczenie w miejscu $a_1 + 2$ jedynki.

Najważniejsza część programu znajduje się w bloku A_3.

Blok A_3

W bloku A_3, każdemu symbolowi języka rachunku zdań odpowiadają dwa miejsca o numerach $a_3 + N$ oraz $a_3 + N + 1$, gdzie N jest numerem symbolu. Np. symbolowi &, dla którego $N = 3$, odpowiadają miejsca $a_3 + 3$ oraz $a_3 + 4$. Symbolowi występującemu przed znakiem asercji odpowiada miejsce $a_3 + N$, symbolowi występującemu po znaku asercji — $a_3 + N + 1$. W każdym miejscu bloku A_3 znajduje się instrukcja skokowa, mówiąca, co należy wykonać po odczytanym symbolu. Np. jeżeli w akumulatorze została uformowana instrukcja $a_3 + 4$, znaczy to, że został odczytany symbol koniunkcji po znaku asercji. Maszyna powinna więc przejść do realizowania programu eliminującego koniunkcję w następniku. Ponieważ program ten zaczyna się od miejsca r_2^*, więc w miejscu $a_3 + 4$ powinna znajdować się instrukcja $!r_2^*$. Postępując podobnie dla wszystkich możliwych symboli języka rachunku zdań, otrzymamy blok A_3.

A_3	a_3		
	$a_3 + 1$	$!r_1$	Negacja
	$a_3 + 2$	$!r_1^*$	

$a_3 + 3$	$!r_2$	Koniunkcja
$a_3 + 4$	$!r_2^*$	
$a_3 + 5$	$!r_3$	Alternatywa
$a_3 + 6$	$!r_3^*$	
$a_3 + 7$	$!r_4$	Implikacja
$a_3 + 8$	$!r_4^*$	
$a_3 + 9$	$!r_5$	Równoważność
$a_3 + 10$	$!r_5^*$	
$a_3 + 11$	$!a_2 + 2$	p
$a_3 + 12$	$!a_2 + 2$	
$a_3 + 13$	$!a_2 + 2$	q
$a_3 + 14$	$!a_2 + 2$	
$a_3 + 15$	$!a_2 + 2$	r
$a_3 + 16$	$!a_2 + 2$	
$a_3 + 17$	—	Nawias prawostronny)
$a_3 + 18$	—	
$a_3 + 19$	$!b_2$	Nawias lewostronny (
$a_3 + 20$	$!b_2$	
$a_3 + 21$	—	Znak asercji $\vdash$
$a_3 + 22$	$!a_2 + 7$	
$a_3 + 23$	$!a_2 + 2$	Przecinek ,
$a_3 + 24$	$!a_2 + 2$	
$a_3 + 25$	niepoprawne	Średnik ;
$a_3 + 26$	$!c_2$	

A_3

Miejsca od $a_3 + 1$ do $a_3 + 10$ zawierają instrukcje powodujące przejście do odpowiednich programów eliminacji głównego spójnika sekwentu.

Miejsca od $a_3 + 11$ do $a_3 + 16$ zawierają instrukcję $! a_2 + 2$, która powoduje przejście do bloku A_2 i odczytanie następnego symbolu sekwentu, bowiem po odczytaniu zmiennej należy odczytać następny symbol sekwentu.

Przy szukaniu głównego spójnika nie może być odczytany nawias prawostronny, dlatego miejsca $a_3 + 17$ i $a_3 + 18$ są puste.

W przypadku odczytania nawiasu lewostronnego, zarówno w następniku jak i w poprzedniku, maszyna przechodzi do obliczania funkcji $F(\sigma_i)$, która wskaże, gdzie znajduje się w tym wyrażeniu spójnik główny. Dlatego w miejscach $a_3 + 19$ i $a_3 + 20$ są umieszczone instrukcje $! p_0$ powodujące przejście do programu obliczania funkcji $F(\sigma_i)$.

Odczytanie znaku asercji $\vdash$ powoduje przejście do instrukcji $a_2 + 7$ w bloku A_2, która wpisuje do miejsca $a_1 + 2$ jedynkę i za pomocą następnej instrukcji $! a_2 + 2$ powoduje czytanie następnego symbolu sekwentu.

Po przecinku, podobnie jak po zmiennych, następuje czytanie kolejnego symbolu sekwentu i dlatego w miejscach $a_3 + 23$ i $a_3 + 24$ znajdują się instrukcje $! a_2 + 2$.

Jeżeli maszyna odczytała średnik sygnalizujący koniec sekwentu, a przedtem nie był odczytany znak asercji, to badane wyrażenie jest niepoprawne. W miejscu $a_3 + 25$ może się więc znajdować specjalna instrukcja, sygnalizująca na wyjściu maszyny, że wyrażenie jest niepoprawne. Jeżeli natomiast został odczytany średnik, a przed nim był znak asercji, znaczy to, że w badanym sekwencie nie ma żadnych spójników logicznych, a więc sekwent jest atomem. Należy więc zbadać, czy jest on aksjomatem. Dlatego w miejscu $a_3 + 26$ znajduje się instrukcja $! p_1$, powodująca przejście do programu badającego, czy atom jest aksjomatem.

Program ten pozwala więc na wyszukanie głównego spójnika logicznego w sekwencie i przejścia do odpowiedniego programu eliminującego ten spójnik. Jeżeli taki spójnik nie istnieje, maszyna przechodzi do badania, czy sekwent jest aksjomatem. W następnych paragrafach pokażemy, jak jest zbudowany program realizujący funkcje $F(\sigma_i)$ oraz program badający, czy sekwent jest aksjomatem.

§ 3. Program realizujący funkcję $F(\sigma_i)$

Celem tego programu jest odszukanie głównego spójnika dwuargumentowego w formule wchodzącej w skład sekwentu.

Program ten podzielimy również na trzy bloki B_1, B_2 i B_3, których oznaczenie jest podobne do bloków A_1, A_2, A_3 w poprzednim programie. Dlatego od razu przystąpimy do omówienia kolejnych bloków.

Blok B_1

B_1	b_1	b_3
	$b_1 + 1$	$F(\sigma_i)$
	.....	.....

W miejscu b_1 podany jest adres b_3 pierwszego miejsca bloku. W miejscu $b_1 + 1$ znajduje się aktualna wartość funkcji $F(\sigma_i)$ dla badanego symbolu. Na początku w miejscu $b_1 + 1$ jest zapisane zero.

Blok B_2

B_2	b_2	$\sim N\, a_1 + 1$
	$b_2 + 1$	$\leftarrow (a_1 + 1)$
	$b_2 + 2$	$+ b_1$
	$b_2 + 3$	$! (0)$
	$b_2 + 4$	$N\, b_1 + 1$
	$b_2 + 5$	$! b_3 + 7$
	$b_2 + 6$	$P\, b_1 + 1$
	$b_2 + 7$	$= b_1 + 1$
	$b_2 + 8$	$? b_2$
	$b_2 + 9$	$- b_1$
	$b_2 + 10$	$! a_2 + 3$

Ponieważ wartość początkowa miejsca $b_1 + 1$ jest zerem, więc w przypadku gdy program A_2 wprowadzi do akumulatora nawias lewostronny (a właściwie jego numer) nie potrzebujemy dla tego nawiasu obliczać wartości funkcji $F(\sigma_i)$. Dlatego program B_2 zaczyna się od instrukcji $N\, a_1 + 1$, która wraz z instrukcją $\leftarrow (a_1 + 1)$ powoduje wprowadzenie do akumulatora następnego symbolu sekwentu. Dzięki następnej instrukcji $+ b_1$, do numeru odczytywanego symbolu dodawany jest adres b_3 tak, że w rezultacie w akumulatorze znajduje się obecnie liczba $b_3 + N$. Następna instrukcja powoduje odczytanie odpowiedniej instrukcji z bloku B_3. Budowę bloku B_3 omówimy w następnym paragrafie, a tymczasem przyjmiemy, że zależnie od odczytanego symbolu z bloku B_3 maszyna wraca do jednej z instrukcji: $N\, b_1 + 1$, $P\, b_1 + 1$ lub $N\, a_1 + 1$ (pierwsza instrukcja programu). Znaczy to, że do miejsca $b_1 + 1$ jest dodawana jedynka, odejmowana jedynka, lub też zawartość tego miejsca pozostaje bez zmiany, zgodnie z określeniem funkcji $F(\sigma_i)$. Następne instrukcje $= b_1 + 1$ oraz $? b_2$ powodują cytowanie następnego symbolu; jeżeli $F(\sigma_i) \neq 0$, bądź też jeżeli $F(\sigma_i) = 0$, od liczby w akumulatorze odejmowany jest adres b_3 tak, że w rezultacie w akumulatorze znajdzie się numer symbolu N i następna instrukcja $! a_2 + 3$ powoduje przejście do bloku A_2 w programie podanym w poprzednim paragrafie. Dalsze postępowanie znamy z paragrafu poprzedniego. Pozostała nam jeszcze do omówienia struktura bloku B_3.

Blok B_3

W bloku B_3 miejsca o adresach $b_3 + 2i$, gdzie $i = 1, 2, \ldots, 13$, są niewykorzystane i dlatego nie podano ich w programie. W pozostałych miejscach znajdują się rozkazy skokowe, powodujące — zależnie od odczytanego symbolu — przejście do czytania następnego symbolu sekwentu.

Po odczytaniu negacji maszyna przechodzi do odczytania następnego symbolu sekwentu. Jeżeli odczytany został symbol spójnika dwuargumentowego, lub nawias prawostronny, to w miejscu $b_1 + 1$ dodawana jest jedynka. Jeżeli zaś odczytany jest nawias lewostronny lub jakakolwiek zmienna, to od zawartości miejsca $b_1 + 1$ odejmowana jest jedynka. Symbol asercji, przecinek lub średnik nie mogą wystąpić przed spójnikiem głównym, stąd w miejscach $b_3 + 21$, $b_3 + 23$,

	$b_3 + 1$	$! b_2$	Negacja
	$b_3 + 3$	$! b_2 + 6$	Koniunkcja
	$b_3 + 5$	$! b_2 + 6$	Alternatywa
	$b_3 + 7$	$! b_2 + 6$	Implikacja
	$b_3 + 9$	$! b_2 + 6$	Równoważność
	$b_3 + 11$	$! b_2 + 4$	p
B_3	$b_3 + 13$	$! b_2 + 4$	q
	$b_3 + 15$	$! b_2 + 4$	r
	$b_3 + 17$	$! b_2 + 4$	(
	$b_3 + 19$	$! b_2 + 6$	)
	$b_3 + 21$	$! -$	$\vdash$
	$b_3 + 23$	$-$	,
	$b_3 + 25$	$-$	;

$b_3 + 25$ nie ma żadnych instrukcji. Tak zbudowany program realizuje więc funkcję $F(\sigma_i)$.

§ 4. Program badania aksjomatów

Jak pamiętamy, atom jest aksjomatem tylko wtedy, jeżeli w poprzedniku i następniku występuje choć raz taka sama formuła elementarna. Musimy więc ułożyć taki program, który będzie badał kolejne symbole w poprzedniku i porównywał je z symbolami w następniku. Ponieważ porównywanie symboli jest niewygodne, zrobimy to w inny sposób, który opiszemy w dalszym ciągu. Program ten podzielimy również na trzy bloki C_1, C_2, C_3, zgodnie z poprzednio przyjętymi zasadami.

Blok C_1

W bloku C_1 przeznaczymy trzy miejsca na zmienne występujące w poprzedniku oraz trzy miejsca na zmienne występujące w następniku. Każdej zmiennej w poprzedniku odpowiada więc jedno miejsce oraz każdej zmiennej w następniku odpowiada również jedno miejsce w bloku C_1. Pozostałe miejsca bloku C_1 mają charakter pomocniczy.

	c_1	p
	$c_1 + 1$	p^*
	$c_1 + 2$	q
C_1	$c_1 + 3$	q^*
	$c_1 + 4$	r
	$c_1 + 5$	r^*
	$c_1 + 6$	c_3
	$c_1 + 7$	k

Rolę miejsc c_1 do $c_1 + 5$ poznamy przy omawianiu następnego bloku C_2.

W miejscu $c_1 + 7$ jest liczba k, która może mieć wartość 0 lub 1 zależnie od tego, czy badany symbol sekwentu jest w poprzedniku bądź w następniku.

Blok C_2

Program w bloku C_2 powoduje czytanie sekwentu symbol po symbolu. Jeżeli odczytana jest jakaś zmienna w poprzedniku, to w bloku C_1, w miejscu odpowiadającym tej zmiennej, dodawana jest jedynka. Jeżeli więc zmienna występuje n razy w poprzedniku, to w odpowiadającym jej miejscu znajdzie się liczba n. Jeżeli zmienna występuje zero razy, to w miejscu jej odpowiadającym jest liczba zero. Podobnie jest dla zmiennych w następniku z tą różnicą, że liczba wystąpień danej zmiennej wpisywana jest w miejscu zaznaczonym gwiazdką. W stanie początkowym, w miejscach $c_1 \ldots c_1 + 5$ są zapisane zera.

Po przeanalizowaniu całego sekwentu, jeżeli sekwent ten jest aksjomatem, w bloku C_1 znajdzie się choć jedna taka para miejsc, odpowiadająca jednej zmiennej w następniku i poprzedniku, że zawartość obu tych miejsc jest różna od zera, tzn.

$Cc_1 \neq 0$ i $Cc_1 + 1 \neq 0$ lub
$Cc_1 + 2 \neq 0$ i $Cc_1 + 3 \neq 0$ lub
$Cc_1 + 4 \neq 0$ i $Cc_1 + 5 \neq 0$ [1]).

Program realizujący powyższą myśl jest następujący:

C_2

c_2	$\leftarrow a_1$
$c_2 + 1$	$\rightarrow a_1 + 1$
$c_2 + 2$	$\leftarrow (a_1 + 1)$
$c_2 + 3$	$+ c_1 + 7$
$c_2 + 4$	$+ c_1 + 6$
$c_2 + 5$	$N a_1 + 1$
$c_2 + 6$	! (0)
$c_2 + 7$	$N c_1$
$c_2 + 8$	$! c_2 + 2$
$c_2 + 9$	$N c_1 + 1$
$c_2 + 10$	$! c_2 + 2$
$c_2 + 11$	$N c_1 + 2$
$c_2 + 12$	$! c_2 + 2$
$c_2 + 13$	$N c_2 + 3$
$c_2 + 14$	$! c_2 + 2$
$c_2 + 15$	$N c_2 + 4$
$c_2 + 16$	$! c_2 + 2$
$c_2 + 17$	$N c_2 + 5$
$c_2 + 18$	$! c_2 + 2$
$c_2 + 19$	$N c_1 + 7$

[1]) Przypominamy, że litera C przed adresem oznacza zawartość tego adresu.

C_2

$c_2 + 20$	$! c_2 + 2$
$c_2 + 21$	$\leftarrow c_1$
$c_2 + 22$	$. c_1 + 1$
$c_2 + 23$	$. = 0$
$c_2 + 24$	$? c_2 + 34$
$c_2 + 25$	$\leftarrow c_1 + 2$
$c_2 + 26$	$. c_1 + 3$
$c_2 + 27$	$= 0$
$c_2 + 28$	$? c_2 + 34$
$c_2 + 29$	$\leftarrow c_1 + 4$
$c_2 + 30$	$. c_1 + 5$
$c_2 + 31$	$= 0$
$c_2 + 32$	$? c_2 + 34$
$c_2 + 33$	Nie jest aksjomatem
$c_2 + 34$	Jest aksjomatem

Instrukcje znajdujące się w miejscach c_2, ..., $c_2 + 6$ powodują pobranie kolejnego symbolu sekwentu do akumulatora i utworzenie odpowiedniego adresu mówiącego czy symbol ten jest w poprzedniku, czy też w następniku. Poprzednio proces ten był szczegółowo wyjaśniony, nie będziemy więc raz jeszcze go powtarzać.

Instrukcje, znajdujące się w miejscach $c_2 + 7$ do $c_2 + 18$ powodują dodanie jedynki do odpowiedniego miejsca w bloku C_1, zależnie od odczytanej zmiennej.

Instrukcja $Nc_1 + 7$ zapisana w miejscu $c_2 + 19$ ma na celu dodanie jedynki do miejsca $c_1 + 7$, o ile odczytany symbol jest znakiem asercji.

Pozostałe instrukcje umieszczone od adresu $c_2 + 21$ do $c_2 + 34$ służą do sprawdzenia, czy istnieje taka para miejsc w bloku C_1, odpowiadająca tej samej zmiennej w poprzedniku i następniku, których

iloczyn zawartości jest różny od zera. Jeżeli iloczyn ten jest równy zeru, to znaczy, że badana zmienna nie występuje po obu stronach znaku asercji. Gdy iloczyn ten jest różny od zera chociaż dla jednej zmiennej, badany sekwent jest aksjomatem.

Dla uproszczenia w miejscach $c_2 + 33$ i $c_2 + 34$ nie podano instrukcji, lecz objaśnienie wyniku działania programu. W rzeczywistości powinny tu znajdować się instrukcje, mówiące co należy zrobić w jednym i drugim przypadku. Sprawa ta jednak była już omawiana przy okazji programu dowodzącego dla teorii T_1' i dlatego nie będziemy do niej wracać.

Pozostał nam jeszcze do omówienia blok C_3.

Blok C_3

C_3	$c_3 + 11$	$! c_2 + 7$	p
	$c_3 + 12$	$! c_2 + 9$	p^*
	$c_3 + 13$	$! c_2 + 11$	q
	$c_3 + 14$	$! c_2 + 13$	q^*
	$c_3 + 15$	$! c_2 + 15$	r
	$c_3 + 16$	$! c_2 + 17$	r^*
	$c_3 + 17$	—	
	$c_3 + 18$	—	
	$c_3 + 19$	—	
	$c_3 + 20$	—	
	$c_3 + 21$	$! c_2 + 19$	$\vdash$
	$c_3 + 22$	—	
	$c_3 + 23$	$! c_2 + 2$	,
	$c_3 + 24$	$! c_2 + 2$	.
	$c_3 + 25$	niepoprawne	;
	$c_3 + 26$	$! c_2 + 21$	:

Ponieważ w atomie nie występują spójniki logiczne, więc blok C_3 rozpoczyna się od miejsca $c_3 + 11$. Grupa instrukcji zawarta pod adresami od $c_3 + 11$ do $c_3 + 16$ powoduje, łącznie z odpowiednimi instrukcjami w bloku C_2, dodanie jedynki do miejsca odpowiadającego każdej zmiennej w bloku C_1.

Miejsca $c_3 + 17$ do $c_3 + 20$ są nie wykorzystane, gdyż odpowiadają one nawiasom, a nawiasy — podobnie jak spójniki logiczne — w atomie nie występują.

Po odczytaniu znaku asercji instrukcja $! c_2 + 19$, wraz z instrukcją $Nc_1 + 7$ w bloku C_2 (adres $c_2 + 19$), powoduje wpisanie jedynki do miejsca $c_1 + 7$ w bloku C_1.

Ponieważ po odczytaniu przecinka należy czytać następny symbol sekwentu, instrukcja $! c_2 + 2$ kieruje maszynę do programu czytania sekwentu.

Jeżeli został odczytany średnik i poprzednio nie było znaku asercji, to wyrażenie nie jest sekwentem i dlatego w miejscu $c_3 + 25$ umieszczono instrukcję „niepoprawne".

Przeanalizowanie wszystkich symboli sekwentu powoduje — za pośrednictwem instrukcji $! c_2 + 21$ — przejście do tego fragmentu programu w bloku C_2, który sprawdza, czy jakaś litera wystąpiła po obu stronach znaku asercji.

Dla całkowitego zrozumienia działania programu dowodzącego powinniśmy opisać programy realizujące wszystkie 10 reguł wnioskowania. Nie będziemy jednak nużyć Czytelnika i wszystkich tych programów nie podamy. Dla ilustracji pokażemy tylko program eliminowania negacji z poprzedników. Wszystkie pozostałe można ułożyć w podobny sposób. Czytelnik, dysponujący dostatecznie dużą dozą cierpliwości, zrobi to z łatwością samodzielnie.

§ 5. Program eliminowania negacji z poprzednika

Wyeliminowanie negacji z poprzednika polega na przepisaniu symboli sekwentu najpierw bez negacji oraz jej argumentu i dopisaniu na końcu otrzymanego ciągu argumentu opuszczonej negacji. Mówiąc dokładniej, najpierw należy przepisywać kolejno wszystkie symbole sekwentu do ustalonych miejsc w magazynie, aż do odczytania negacji. Następnie dalszych symboli, poczynając od negacji aż do naj-

bliższego przecinka, należy nie wpisywać. Dalsze zaś symbole — od przecinka do średnika — należy znów przepisać, średnika jednak nie przepisując. W ten sposób otrzymamy sekwent pozbawiony negacji oraz jej argumentu. Zgodnie z regułą $\mathbf{R}_1$ musimy teraz argument negacji dopisać na końcu otrzymanego sekwentu. A więc należy czytać pierwotny sekwent od początku i postępować odwrotnie niż za pierwszym czytaniem, tzn. symboli od początku do najbliższej negacji nie przepisywać; począwszy od negacji do najbliższego przecinka — przepisać do dalszych miejsc zarezerwowanych na przesłankę i na końcu dopisać średnik.

Program tych czynności przedstawimy w dwu blokach: R'_1 i R_1.

Blok R'_1

R'_1

r'_1	h_0
$r'_1 + 1$	h_1
$r'_1 + 2$	i
$r'_1 + 3$	j
$r'_1 + 4$	1
$r'_1 + 5$	23
$r'_1 + 6$	25
$r'_1 + 7$	

W miejscach r'_1 i $r'_1 + 1$ podano adresy pierwszego symbolu sekwentu oraz przesłanki.

W miejscach $r'_1 + 2$ i $r'_1 + 3$ są obliczane aktualne adresy i symbole aktualnie czytanego i w sekwencie oraz adres j symbolu aktualnie zapisywanego w przesłance.

W miejscach $r'_1 + 4$, $r'_1 + 5$, $r'_1 + 6$ są zapisane kolejne numery negacji, przecinka i średnika. Właściwy program znajduje się w bloku $\mathbf{R}_1$.

Miejsce $r'_1 + 7$ służy jako miejsce pomocnicze.

Blok R_1

R_1

r_1	$\leftarrow r'_1$
$r_1 + 1$	$\rightarrow r'_1 + 2$
$r_1 + 2$	$\leftarrow r'_1 + 1$
$r_1 + 3$	$\rightarrow r'_1 + 3$
$r_1 + 4$	$\leftarrow (r'_1 + 2)$
$r_1 + 5$	$\rightarrow r'_1 + 7$
$r_1 + 6$	$- r'_1 + 4$
$r_1 + 7$	$= 0$
$r_1 + 8$	$? r_1 + 35$
$r_1 + 9$	$N r'_1 + 2$
$r_1 + 10$	$\leftarrow (r'_1 + 2)$
$r_1 + 11$	$- (r'_1 + 5)$
$r_1 + 12$	$= 0$
$r_1 + 13$	$? r_1 + 40$
$r_1 + 14$	$N r'_1 + 2$
$r_1 + 15$	$\leftarrow (r'_1 + 2)$
$r_1 + 16$	$\rightarrow r'_1 + 7$
$r_1 + 17$	$- r'_1 + 6$
$r_1 + 18$	$= 0$
$r_1 + 19$	$? r_1 + 41$
$r_1 + 20$	$\leftarrow r'_1$
$r_1 + 21$	$\rightarrow r'_1 + 2$
$r_1 + 22$	$\leftarrow (r'_1 + 2)$
$r_1 + 23$	$- r'_1 + 4$
$r_1 + 24$	$= 0$

R_1

$r_1 + 25$	$?\, r_1 + 45$
$r_1 + 26$	$N\, r_1' + 2$
$r_1 + 27$	$\leftarrow (r_1' + 2)$
$r_1 + 28$	$\rightarrow r_1' + 7$
$r_1 + 29$	$- r_1' + 5$
$r_1 + 30$	$= 0$
$r_1 + 31$	$?\, r_1 + 47$
$r_1 + 32$	$\leftarrow r_1' + 6$
$r_1 + 33$	$\rightarrow (r_1' + 3)$
$r_1 + 34$	stop
$r_1 + 35$	$\leftarrow r_1' + 7$
$r_1 + 36$	$\rightarrow (r_1' + 3)$
$r_1 + 37$	$N\, r_1' + 2$
$r_1 + 38$	$N\, r_1' + 3$
$r_1 + 39$	$!\, r_1 + 4$
$r_1 + 40$	$!\, r_1 + 9$
$r_1 + 41$	$\leftarrow r_1' + 7$
$r_1 + 42$	$\rightarrow (r_1' + 3)$
$r_1 + 43$	$N\, r_1' + 3$
$r_1 + 44$	$!\, r_1 + 44$
$r_1 + 45$	$N\, r_1' + 2$
$r_1 + 46$	$!\, r_1 + 22$
$r_1 + 47$	$\leftarrow r_1' + 7$
$r_1 + 48$	$\rightarrow (r_1' + 3)$
$r_1 + 49$	$N\, r_1' + 3$
$r_1 + 50$	$!\, r_1 + 26$

W bloku R_1 instrukcje od adresu r_1 do $r_1 + 8$, powodują wraz z instrukcjami o adresach $r_1 + 35$ do $r_1 + 39$ przepisywanie symboli sekwentu tak długo, aż pojawi się symbol negacji. Wtedy instrukcje $r_2 + 9, \ldots, r_2 + 13$ oraz instrukcja pod adresem $r_1 + 40$ powodują szukanie najbliższego przecinka [1]), ale odczytywane symbole nie są przez program wpisywane.

Instrukcje $r_1 + 14, \ldots, r_1 + 19$ oraz instrukcje $r_1 + 41$, $r_1 + 44$ powodują przepisywanie pozostałych symboli sekwentu.

Dalsze instrukcje $r_1 + 20, \ldots, r_1 + 25$ oraz $r_1 + 45, r_1 + 46$ powodują ponowne czytanie sekwentu od początku, z tym, że symbole odczytywane nie są przepisywane tak długo, aż pojawi się symbol negacji. Wtedy instrukcje $r_1 + 26$, $r_1 + 31$ oraz $r_1 + 47, \ldots, r_1 + 50$ czytają dalsze symbole, przepisując je jako dalsze symbole przesłanki tak długo, aż zostanie odczytany przecinek. Przecinek nie jest wpisywany jako symbol końca sekwentu, lecz zamiast niego, za pomocą instrukcji $r_1 + 32$, $r_1 + 33$, zostaje zapisany średnik powodujący, że obliczanie przesłanki zostało zakończone.

Czytelnik, który przebrnął, chociażby powierzchownie, poprzez gąszcz symboli pokrywających gęsto strony drugiej części książki czuje się być może zawiedziony automatycznym dowodzeniem twierdzeń jak i maszynami matematycznymi w ogóle. Cóż to za automatyka, w której najprostsze wnioskowanie trzeba najpierw opisać za pomocą setek symboli? Niestety, taka jest rzeczywistość. Komu ona nie odpowiada, niech sobie poczyta książkę Stanisława Lema, *Summa technologiae*, w której maszyny są znacznie doskonalsze niż te, które są budowane obecnie.

---

[1]) Dla uproszczenia programu przyjmujemy, że przecinek stawiamy również przed symbolem $\vdash$, np. $p, q, \vdash r$.

ZAKOŃCZENIE

W związku z próbami zastosowania maszyn matematycznych do dowodzenia twierdzeń powstało szereg pytań. Najciekawsze niewątpliwie jest pytanie postawione we wstępie: jaką rolę mogą odegrać maszyny matematyczne w twórczej pracy matematyków? Nadzieja na zastąpienie żywego matematyka maszyną tworzącą teorie matematyczne jest nieuzasadniona. Maszyny mogą odegrać tu rolę jedynie użytecznego narzędzia, formalizując dowody twierdzeń już znanych, bądź też mogą być użyteczne w badaniu teorii matematycznych, pozwalając na szybkie znalezienie dowodu twierdzenia, jeżeli badana formuła jest twierdzeniem. Niestety w większości teorii matematycznych nie mamy metody rozstrzygania, czy formuła należąca do języka teorii jest twierdzeniem tej teorii, czy też nie: *Jeżeli badana formuła jest twierdzeniem, to po skończonej liczbie kroków maszyna się zatrzyma i znajdzie dowód. Tak jest dla większości teorii matematycznych. Jeżeli zaś formuła twierdzeniem nie jest, to maszyna nie zatrzyma się i będzie ciągle znajdowała według reguł wnioskowania coraz to nowe przesłanki, nie dochodząc nigdy do aksjomatów.* Teorie takie nazywamy teoriami **nierozstrzygalnymi.** Przykłady teorii rozpatrywanych w tej książce były rozstrzygalne. Dlatego, jeżeli badaliśmy wyrażenie nie będące twierdzeniem, to maszyna dochodziła do atomu, którego dalej już nie mogła rozkładać i sygnalizowała, że badana formuła nie jest twierdzeniem. W teoriach nierozstrzygalnych takie postępowanie jest niemożliwe. W związku z powyższym, przystępując do badania formuły w teorii nierozstrzygalnej, nie wiemy, czy maszyna się zatrzyma i znajdzie dowód, czy też nie. A więc jeżeli maszyna działa dość długo i dowodu jeszcze nie znalazła, to nie wiemy, czy czekać na wynik dłużej czy też zatrzymać maszynę, gdyż badana formuła nie jest twierdzeniem. Tym niemniej przy dostatecznie szybkich maszynach i przy dostatecznie długim czasie działania maszyny można

przyjąć z dużym prawdopodobieństwem, że większa liczba twierdzeń badanej teorii zostanie znaleziona. W związku z powyższym powstaje ważne zagadnienie długości dowodu oraz częstości występowania twierdzeń w języku teorii. Przez długość dowodu rozumiemy liczbę kroków, potrzebnych do jego znalezienia. Liczba ta zależy oczywiście od reguł wnioskowania stosowanych w teorii. Dla celów maszynowych byłoby więc interesujące, aby większa część twierdzeń teorii miała krótkie dowody. Szanse znalezienia ich wtedy za pomocą maszyny byłyby duże. Z drugiej strony, dla celów maszynowych byłoby pożądane, aby teorie matematyczne miały jak największą „sprawność", tj. aby częstość występowania twierdzeń pośród formuł języka była jak największa. Wtedy również można oczekiwać większej przydatności maszyn do badania teorii matematycznych.

Można również zastosować maszyny w inny sposób niż ten, który podaliśmy w tej książce. Maszyna może produkować w systematyczny sposób wszystkie formuły jakiejś teorii i sprawdzać, które z nich są twierdzeniami tej teorii. W ten sposób można by produkować kolejne twierdzenia badanej teorii. Jak już wspominaliśmy, nie wszystkie jednak twierdzenia są jednakowo interesujące. Dlatego produkowanie wszystkich możliwych twierdzeń jest z punktu widzenia matematyka nie celowe. W jaki sposób ograniczyć więc produkcje maszyny tak, aby dawała ona tylko interesujące rezultaty? Wang próbował odpowiedzieć na to pytanie, określając co to znaczy, że twierdzenie jest interesujące. Nie wchodząc bliżej w tę sprawę, zasygnalizujemy, że uzyskane tą drogą twierdzenia nie były wszystkie rzeczywiście ciekawe. Należałoby więc zdaniem Wanga zastosować mocniejsze kryterium „ciekawości" twierdzenia. Załóżmy, że kryterium takie udało się znaleźć i że maszyna produkuje rzeczywiście ciekawe twierdzenia. Przy dzisiejszej szybkości liczenia maszyna matematyczna może w krótkim czasie wyprodukować kilkaset tysięcy twierdzeń teorii. Pojawia się więc pytanie, kto będzie mógł te twierdzenia czytać, rozumieć i wykorzystywać? Właściwie należałoby zapytać, czy w jakiejkolwiek teorii może być rzeczywiście sto tysięcy interesujących twierdzeń? Sprawa ta nie jest całkiem jasna. Aby na pytanie to można było odpowiedzieć, konieczne jest głębsze zbadanie teorii matematycznych. Obecny stan wiedzy o strukturze teorii matematycznych nie pozwala na te pytania odpowiedzieć.

Zastosowanie maszyn do szukania dowodów konkretnych formuł

teorii wydaje się nie nasuwać tylu zastrzeżeń, jakkolwiek i tutaj nie wszystko jest całkiem jasne. Aby stosować maszyny do dowodzenia konkretnych twierdzeń, konieczna jest jeszcze olbrzymia ilość pracy badawczej, polegającej na takim przeformułowaniu teorii matematycznych, aby stosowane w nich reguły wnioskowania nadawały się do wygodnej realizacji w maszynie. Można zaryzykować opinię, że wszystkie niemal istniejące obecnie teorie matematyczne w dotychczasowej postaci nie nadają się do maszynowego traktowania. Przeformułowanie ich nie jest sprawą prostą. Nasuwa się więc nieodparcie pytanie, czy i w jakim stopniu stosowanie maszyn może być tutaj celowe, opłacalne. Jeżeli bowiem konieczne jest włożenie dużego wysiłku w ponowne opracowanie teorii matematycznych po to, aby w końcowym wyniku otrzymać znane już rezultaty, cała sprawa może być niepotrzebną zabawą. Znalezienie zaś nowych ważnych twierdzeń dzięki maszynom matematycznym wydaje się mało prawdopodobne.

Myślę jednak, że mimo tych pesymistycznych prognoz badania nad automatycznym dowodzeniem twierdzeń będą się rozwijały. Konieczność bowiem spojrzenia na znane fakty, ale z innego punktu widzenia, może pogłębić **rozumienie** struktury teorii matematycznych oraz zdolności twórczych człowieka. Tak więc maszyny matematyczne niezależnie od ich praktycznego znaczenia, mogą odegrać pewną rolę pośrednią w badaniach struktury teorii matematycznych.

LITERATURA

Davis M. Putnam, H. „A computing procedure for quantification theory". *Journal of the Association for the Computing Machinery*, 7(1960), 201—215,

Gelernter H. „Realization of a geometry theorem proving machine". *Proceeding of the International Conference on Information Processing*, UNESCO, Paris, 1959.

Glimor P. C. „A proof method for quantification theory". IBM *Journal of Research and Development*, 4(1960), 28—35.

Grzegorczyk A. *Logika popularna*. Warszawa, 1960.

Pogorzelski W., Słupecki J., *O dowodzie matematycznym*. Warszawa, 1960.

Pawlak Z., *Maszyna i język*. Warszawa, 1964.

Prawitz D., Prawitz H., Voghera N. „A mathematical proof procedure and its realization in an electronic computer". *Journal of the Association for Computing Machinery*, 7(1960), 102—139.

Wang H. „Toward Mechanical Mathematics", *IBM Journal of Research and Development*, 4(1960), 2 —22.

Wang H. „Proving theorem by pattern recognition", *I, Communication of ACM*, 3(1960), 220—234.

Wang H. „Proving theorem by pattern recognition", *II, Bell System Technical Journal*, 40(1961), 1—41.

SPIS TREŚCI

CZĘŚĆ II