ZDZISŁAW PAWLAK

GRAMATYKA I MATEMATYKA

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

Redaktor
Gustawa Maślankiewicz

Redaktor techniczny
Stefania Rzęcka

BIBLIOTEKA
Instytutu Matematycznego U.W.
Nr inw. 11362

Copyright by
Państwowe Zakłady Wydawnictw Szkolnych — Warszawa 1965

Wydanie pierwsze Nakład 5 000+240 egz. Ark. druk. 7; wyd. 5,84
Oddano do składania 22. X. 1964 r. Podpisano do druku 14. V. 1965 r.
Druk ukończono w czerwcu 1965 r.
Papier druk. sat. 86×122 cm, 70 g, kl. V z fabryki w Skolwinie
Zam. 5818/924 Cena książki zł 16,50 G-6

Zakłady Graficzne PZWS w Bydgoszczy, ulica Jagiellońska 1.

K-51/65

Motto

„Językoznawstwo, podobnie jak każda inna nauka empiryczna, jest złożonym konglomeratem teorii i obserwacji. Dokładny charakter tego konglomeratu nie jest jeszcze dostatecznie dobrze rozumiany".

Yehoshua Bar-Hillel

PRZEDMOWA

W ostatnich kilkunastu latach powstało wiele nowych gałęzi nauki i techniki. Przyczyniły się do tego również w pewnym stopniu maszyny matematyczne.

Powszechnie znana jest rola maszyn matematycznych, jako wysoce sprawnego narzędzia obliczeniowego. Oddają one nieocenione usługi nauce i technice, pozwalając na szybkie wykonywanie skomplikowanych obliczeń, koniecznych do rozwiązania jakiegoś problemu teoretycznego lub praktycznego. Rola maszyn matematycznych w nauce nie sprowadza się jednakże tylko do użytecznego narzędzia.

Odegrały one inną jeszcze, mniej znaną, ale równie ważną rolę w nauce. Polegała ona na zbliżeniu do siebie wielu nawet odległych dziedzin. Okazało się, że problemy występujące w związku z konstrukcją i zastosowaniem maszyn matematycznych są interesujące nie tylko dla inżynierów, ale i dla ludzi różnych zawodów: psychologów, biologów, językoznawców, matematyków i wielu innych. Każdy z nich znalazł w maszynach matematycznych odbicie cząstki problematyki z własnej dziedziny. Na przykład neurologowie zauważyli, że elementy maszyn matematycznych w pewnym stopniu przypominają podstawowe komórki systemu nerwowego organizmów żywych. Sposób opisu zjawisk zachodzących w maszynach matematycznych okazał się w pewnym stopniu przydatny do opisu przebiegu reakcji biochemicznych.

Zastosowanie maszyn do tłumaczenia zmusiło do sprecyzowania podstawowych pojęć gramatycznych i do głębszych badań nad strukturą języków naturalnych. Pod wpływem zastosowań maszyn do skomplikowanych obliczeń powstały nowe metody rachunkowe. Przykładów takich można podać wiele.

Na marginesie warto odnotować jeszcze jedną rolę maszyn matematycznych, mniej ważną od poprzednich, ale równie ciekawą — wpływ na fantazję. Wpływ ten był dwojakiego rodzaju.

Ci, którzy zetknęli się z problematyką maszyn matematycznych tylko powierzchownie, zostali zafascynowani możliwościami maszyn, nie widząc zupełnie ich braków i ograniczeń. Było to przyczyną wybuchu niehamowanej fantazji. Na temat możliwości maszyn matematycznych zaczęto pisać zupełnie nieodpowiedzialne rzeczy; głoszono, że zbudowanie maszyny, mogącej wykonywać dowolne czynności mózgu ludzkiego, jest tylko kwestią czasu. Wystarczy zbudować dostatecznie dużą maszynę o odpowiednim stopniu skomplikowania i będzie ona w stanie wyręczać człowieka w pracy umysłowej. Gdyby poglądy o wszechmożliwościach maszyn były głoszone jedynie przez dziennikarzy lub pisarzy powieści fantastycznych, skłonnych z racji swojego zawodu do przesady, sprawa nie byłaby może tak groźna. Jednakże do rozpowszechniania fantastycznych wiadomości o maszynach przyłączyli się również ludzie z tytułami naukowymi, o poważnym nierzadko dorobku w swojej dyscyplinie, którzy — nie zapoznawszy się dostatecznie z problematyką maszyn matematycznych — próbowali ją przenieść do innych dziedzin, powodując w ten sposób sporo zamieszania i nieporozumień.

Odwrotny zupełnie wpływ miały maszyny na ludzi, którzy zetknęli się z nimi bliżej: na ich konstruktorów i użytkowników. Tutaj spowodowały one ograniczenie fantazji. Maszyny matematyczne okazały się bowiem bezwzględnym filtrem dla wszelkich błędnych pomysłów. Jeżeli ktoś podał niepoprawną metodę rozwiązania jakiegoś zagadnienia, po zastosowaniu jej do maszyny zostało to od razu ujawnione. Otrzymano wyniki pozbawione zupełnie sensu. Łatwo pisać czy mówić o „myślących maszynach", jednakże do tej pory nikt nie pokusił się o zbudowanie chociażby najprostszej maszyny myślącej. Znacznie wygodniej jest o tym rozprawiać na papierze. Maszyny matematyczne — jak to powiedział matematyk Hao Wang — zawęziły znaczenie słowa „można". Co innego jest napisać, że coś „można" zrobić, a co innego to zrobić.

Inaczej używa słowa „można" ktoś, komu nie grozi praktyczne zweryfikowanie głoszonych przez niego hipotez, a zupełnie inny sens posiada to słowo w ustach człowieka zmuszanego do rychłego sprawdzania swoich idei.

Jedną z dziedzin, na której rozwój w ostatnich latach wywarły dość duży wpływ maszyny matematyczne, jest językoznawstwo. Struktura języka, jego stosunek do rzeczywistości od dawna był przedmiotem zainteresowań filozofów i logików, nie mówiąc oczywiście o językoznawcach. Stworzenie dostatecznie ścisłej teorii języka wymagało odpowiednio precyzyjnych metod. Mogła ich dostarczyć jedynie współczesna matematyka. Pierwszą próbę matematyzacji pojęć gramatycznych podjął filozof i logik polski Kazimierz Ajdukiewicz około 1929 roku. Idee jego nie znalazły jednak wówczas kontynuatorów. Dopiero po drugiej wojnie światowej, próby zastosowania maszyn matematycznych do tłumaczenia spowodowały zwrócenie uwagi na koncepcje Ajdukiewicza. Rozwinął je i zastosował praktycznie do pierwszego maszynowego tłumaczenia Yehoshua Bar-Hillel, profesor uniwersytetu w Jerozolimie.

Studia nad mechanicznym tłumaczeniem spowodowały intensywny rozwój matematycznej analizy pojęć gramatycznych. W rezultacie powstał szereg nowych koncepcji w tej dziedzinie, różnych od koncepcji Ajdukiewicza i Bar-Hillela. Do najciekawszych należą prace prof. Noama Chomsky'ego z Massachusetts Institute of Technology (USA) oraz koncepcja pochodząca od Michaela Rabina i Dana Scotta. Pierwszy z nich jest profesorem logiki w uniwersytecie w Jerozolimie (Izrael), drugi zaś profesorem logiki w uniwersytecie w Stanfordzie (USA).

Dziedzina ta powoli przekształciła się w samoistną dyscyplinę naukową, zwaną lingwistyką matematyczną, badającą język metodami matematycznymi, nie tylko na użytek zastosowań do maszynowego tłumaczenia.

Jaki jest cel takich badań? Niewątpliwie pozwalają one na głębsze zrozumienie struktury języka, aniżeli byłoby to możliwe bez użycia metod matematycznych.

Jednocześnie pojęcie języka zostało znacznie rozszerzone. Przez język rozumie się nie tylko język ludzki, ale i wszelki system sygnalizacji zarówno w maszynie, jak i dowolnym organizmie biologicznym. Oczywiście inne własności posiada określony język ludzki, a inne język, w którym zachodzi przesyłanie sygnałów między różnymi elementami organizmu. Tym niemniej, mimo olbrzymich różnic, łączą je pewne cechy wspólne. Stworzenie ogólnych podstaw matematycznych dla szeroko rozumianego języka jest jednym z najbliższych celów lingwistyki matematycznej.

Lingwistyka matematyczna rozwija się najintensywniej w USA, Izraelu oraz ZSRR. W Polsce problematyką tą zajmuje się dr Irena Bellert, dr Andrzej Ehrenfeucht, prof. Roman Suszko oraz doc. Olgierd Wojtasiewicz.

W książce tej podano elementarne wiadomości z lingwistyki matematycznej. Rozdział I zawiera wstępne pojęcia, takie jak określenie języka, gramatyki, zdania, zdania zbudowanego poprawnie itp. W rozdziale II, III i IV podano omówienie najprostszych gramatyk, gramatyki struktur frazowych (zwanej tu gramatyką prostą), gramatyki kategorialnej oraz gramatyki skończenie stanowej. W następnym V rozdziale omówiono język z punktu widzenia przydatności do opisywania pewnej klasy czynności. Rozdział VI zawiera opis języka, przydatnego do opisywania najprostszych przejawów życia — tworzenia się białka. Wreszcie ostatni VII rozdział informuje o językach maszyn matematycznych. W zakończeniu podano kilka uwag o mechanicznym tłumaczeniu oraz o innych językach, nie omawianych szerzej w tej książce.

Książkę można więc podzielić na dwie części. Rozdziały I i IV dotyczą różnych metod badania struktury języka i nie wchodzą w sprawy związane ze znaczeniem. Zajmujemy się w nich wyłącznie zagadnieniami gramatycznymi. W pozostałych rozdziałach V—VII podane są różne przykłady języków oraz związku z opisywanym przez nie „światem". Szczególnie interesujące wydają się tu języki opisujące syntezy białka oraz języki maszyn matematycznych. Warto jeszcze zwrócić uwagę na rolę, jaką odgrywają maszyny matematyczne w badaniach języków. Maszyna służy tu jako pojęcie równoważne z pojęciem gramatyki (rozdział V). Maszyna gra więc rolę pojęcia abstrakcyjnego, służącego do określania zdań poprawnych języka.

Do opisu rozwiązywania zadań matematycznych za pomocą maszyn używany jest specjalny język, w którym zapisywany jest przebieg rozwiązania. Język ten jest omówiony w rozdziale VII.

Ponadto maszyny matematyczne są stosowane do tłumaczeń języków naturalnych. W każdym z tych trzech przypadków chodzi oczywiście o inne języki.

Książka jest przeznaczona w zasadzie dla uczniów szkół średnich i nauczycieli interesujących się matematyką i jej nowoczesnymi zastosowaniami. Do jej czytania wystarczą wiadomości z matematyki z zakresu szkoły średniej. Dokładniejsze zrozumienie poruszanych problemów wymaga jednak pewnej samodzielności myślenia, polegającej na ciągłej konfrontacji podawanych faktów i wiadomości z innych dziedzin, np. gramatyki czy biologii.

Wydaje się, że książka może być z pożytkiem przeczytana również przez studentów interesujących się maszynami matematycznymi.

Na końcu każdego rozdziału (z wyjątkiem ostatniego) podano wykaz literatury, w której omawiane są wyczerpująco poruszane zagadnienia. Książkę można więc traktować jako wstęp do lingwistyki matematycznej przeznaczony dla tych Czytelników, którzy by chcieli zająć się poważniej tą dziedziną.

Na zakończenie chciałbym wyrazić podziękowanie drowi A. Ehrenfeuchtowi i drowi A. W. Mostowskiemu za szereg uwag dotyczących strony matematycznej książki oraz dr Irenie Bellert i mgr Krystynie Łyczkowskiej za uwagi natury lingwistycznej.

Rozdział I

POJĘCIA WSTĘPNE

§ 1. Języki naturalne

Język jest narzędziem porozumiewania się ludzi między sobą. Historia powstania języka ludzkiego nie jest do tej pory zbadana dostatecznie szczegółowo i zawiera wiele luk i niejasności. Szczególnie niejasne są początki powstawania języka i być może pozostaną one na zawsze tajemnicą.

Język od chwili swych narodzin przechodził różne fazy rozwoju i zależnie od wielu, często trudnych do uchwycenia czynników lokalnych przybierał najrozmaitsze postacie. Niekiedy rozwój ten był szybszy, niekiedy wolniejszy. Czasem język zatrzymywał się na pewnym etapie rozwoju i trwał prawie niezmieniony przez wieki.

Zatrzymywanie się rozwoju języka ma miejsce wyłącznie w językach martwych, jak np. łacina. Czasem znów język, po osiągnięciu określonego stopnia rozwoju, ginął. W innych okolicznościach z mieszaniny słownictwa różnych języków powstawał nowy język. W rezultacie w dniu dzisiejszym istnieje duża liczba języków, którymi posługują się różne grupy ludzkie.

Każdy żywy język rozwija się. Rozwój ten dotyczy zarówno struktury gramatycznej, jak i słownictwa. Zmiany w strukturze gramatycznej zachodzą bardzo wolno: potrzeba czasu przynajmniej jednego pokolenia, by je zauważyć.

Duże różnice istniejące między językami oraz ich ciągłe zmienianie się utrudniają znacznie studia nad językiem.

Językoznawstwo — nauka o języku — może zajmować się badaniem jednego języka w określonym stadium jego rozwoju, może też interesować się pewnymi cechami wspólnymi różnych języków. Szczególnie interesujące wydaje się pytanie, czy w strukturze różnych języków istnieje jakiś wspólny, zasadniczy szkielet, na którego podstawie

oparta jest cała konstrukcja języka, czy też struktury różnych języków są całkowicie niezależne, nie mające żadnych elementów wspólnych.

Do niedawna trudno byłoby powiedzieć, że istnieje jakiś precyzyjny pogląd na strukturę języka w ogóle, niezależnie od tego, czy mamy na myśli język polski, angielski czy jakikolwiek inny. Badania nad strukturą języka datują się począwszy od lat dwudziestych bieżącego stulecia. Rzecz jasna były one zupełnie niezależne od zagadnień związanych z mechanicznym tłumaczeniem. W ostatnim dziesiątku lat, w związku z tłumaczeniem za pomocą maszyn matematycznych, zrobiono w tym kierunku pewien postęp. Rozpoczęto badania, które chociaż robione pod kątem zastosowania do konkretnych języków, mogą stanowić zaczątki ogólnych podstaw językoznawstwa.

Kierunek tych badań niektórzy nazywają językoznawstwem teoretycznym, inni — językoznawstwem matematycznym, jeszcze inni — językoznawstwem kombinatorycznym. Pozostaniemy tutaj przy terminie — **językoznawstwo matematyczne** (bądź lingwistyka matematyczna).

Za twórcę lingwistyki matematycznej powszechnie uchodzi zmarły w 1963 r. polski logik i filozof Kazimierz Ajdukiewicz.

Około 30 lat temu przedstawił on pewne koncepcje teorii języka, które stanowią podstawy dzisiejszej lingwistyki matematycznej. Jednakże koncepcje te nie natrafiły na podatny grunt i przez wiele lat nie były dalej rozwijane. Były one, jak to często w historii nauki się zdarza, przedwczesne i nie znajdowały społecznego zapotrzebowania.

Idee Ajdukiewicza zostały podjęte i dalej rozwinięte dopiero po drugiej wojnie światowej, w związku z zastosowaniem maszyn matematycznych do tłumaczenia. Uczynił to profesor uniwersytetu w Jerozolimie (Izrael) — Yehoshua Bar-Hillel. Koncepcje Ajdukiewicza i Bar-Hillela przedstawimy w rozdziale III.

Nieco inne podejście do lingwistyki matematycznej niż Ajdukiewicza i Bar-Hillela zaproponował prof. Noam Chomsky z Massachusetts Institut of Technology (USA). Z ideami Chomsky'ego zapoznamy się w rozdziale II, z ideami natomiast Rabina i Scotta zapoznamy się w rozdziale IV.

Badania, o których mówiliśmy do tej pory, dotyczyły struktury języka. W badaniach strukturalnych nie interesuje nas strona znaczeniowa języka, jego związek z rzeczywistością. Takie podejście do języka, w którym nie interesuje nas strona znaczeniowa, lecz zagadnienia fonetyczne, morfologiczne i syntaktyczne (fleksyjne i słowotwórcze), nazywamy **syntaktycznym.**

Możemy również badać stronę znaczeniową języka. Mówimy wtedy o **semantyce.** Pewne uwagi z tej dziedziny znajdzie Czytelnik w dalszych rozdziałach.

§ 2. Język mówiony i język pisany

O ile powstanie mowy wiąże się z pojawieniem się człowieka, to pismo jest zjawiskiem — w skali rozwoju człowieka — całkowicie współczesnym. Powstało ono zaledwie kilka tysięcy lat temu. Początki języka pisanego wiążą się z **pismem obrazkowym,** polegającym na porozumiewaniu się za pomocą rysunków. Rysunek nie musiał być oczywiście wiernym odbiciem przedstawianego przedmiotu, lecz jego uproszczonym schematem. Pismo rysunkowe mogło być rozumiane przez ludzi używających różnych języków mówionych, których ustne

Tekst, którym posługiwali się Indianie Cuna (Panama) przy śpiewie rytualnym. Wiersze czytano na przemian z prawa na lewo i z lewa na prawo. Przedruk z „L'art de l'écriture". Le Courrier. Unesco. Mars 1964.

porozumienie się między sobą byłoby niemożliwe. A więc w przypadku pisma obrazkowego nie było bezpośrednich związków między językiem mówionym i pisanym. Przykład pisma obrazkowego podany jest w tablicy 1.

Dalszym etapem w ewolucji sposobów zapisywania języka było **pismo ideograficzne.** W piśmie tym każdemu **pojęciu** odpowiada odrębny **znak.** Język taki zawiera, więc bardzo dużą liczbę znaków, które należy zapamiętać, aby swobodnie się nim posługiwać.

Pismo ideograficzne nie zawiera również wskazówek o sposobie jego czytania. A więc język mówiony i pisany są w tym przypadku również słabo powiązane. Przykładem pisma ideograficznego jest pismo chińskie. Np. symbol 山 oznacza górę, a symbol 水 wodę. W zasadzie każdemu pojęciu odpowiada inny symbol. Jednakże często zestawienie symboli oznacza nowe pojęcie, np. 水山 — „góra" — „woda" — oznacza „krajobraz".

I wreszcie ostatni, dobrze już nam znany etap rozwoju języka pisanego, to **pismo alfabetyczne,** w którym elementarne symbole — litery alfabetu — odpowiadają nie pojęciom, lecz klasom dźwięków. Języki o piśmie alfabetycznym stanowią niewątpliwie kulminacyjny punkt w rozwoju języka pisanego. Pismo alfabetyczne, dzięki zapisowi wymowy, okazało się wyjątkowo odpowiednie do utrwalenia języka mówionego. Nie podaje ono wprawdzie wszystkich elementów wymowy, jak np. akcentu intonacji, tym niemniej stanowi zasadniczy krok w ewolucji sposobów zapisywania języka.

Teoria, którą mamy zamiar przedstawić w niniejszej książce, dotyczy w zasadzie zapisywania języków za pomocą pisma alfabetycznego, przy czym nie będzie nas interesował związek badanych języków pisanych i odpowiadających im języków mówionych. Przyjmiemy, że język to zbiór odpowiednich napisów, składających się z liter ustalonego alfabetu, i będziemy badali niektóre własności tak rozumianego języka, nie interesując się zupełnie sprawami jego wymowy i związaną z tym problematyką. [1])

W dalszym ciągu przez „język" będziemy więc rozumieli wyłącznie „pisany język alfabetyczny".

---

[1]) Taka definicja języka jest charakterystyczna dla lingwistyki matematycznej i w zasadzie różni się od rozumienia języka i definicji w językoznawstwie ogólnym.

§ 3. Języki formalne

Oprócz języków naturalnych, pisanych czy mówionych, istnieje jeszcze jeden język, którym posługuje się pewna grupa ludzi, niezależnie od swojego języka ojczystego. Jest to **język matematyki.** Język ten jest również pisanym językiem alfabetycznym, do którego alfabetu należą cyfry, zmienne, symbole działań i wiele innych.

Nie ma jednolitej wymowy tego języka, jednakże w pewnym kręgu ludzi język ten jest zrozumiały. A więc mamy tu sytuację nieco zbliżoną do zapisu obrazkowego lub ideograficznego. Symbole alfabetu języka matematycznego oznaczają raczej pewne pojęcia aniżeli dźwięki. Jeżeli jednak pominiemy stronę semantyczną tego języka, to zarówno na pisany język naturalny, jak i na symboliczny język matematyczny możemy patrzeć w jednakowy sposób: jako na kombinacje pewnych symboli, zestawianych w myśl zadanych reguł.

Oczywiście w praktyce język matematyczny jest mieszaniną języka naturalnego i języka formalnego. Jednakże dla naszych celów możemy przyjąć, że język matematyczny to język posiadający alfabet, który się składa wyłącznie z symboli matematycznych, bez jakichkolwiek elementów języka naturalnego.

Powstają więc dodatkowe pytania: Czy zachodzi jakieś podobieństwo między językiem matematycznym a językiem naturalnym? Na czym polegają różnice między tymi językami? Czy do ich badania można stosować jednakowe metody?

Lingwistyka matematyczna w zakres swych rozważań włącza również języki matematyczne, jako szczególny przypadek języków naturalnych. Tak więc nasze dalsze rozważania będą dotyczyły zarówno języków naturalnych, jak i matematycznych.

§ 4. Inne języki

W pierwszym paragrafie tego rozdziału napisaliśmy, że język to narzędzie porozumiewania się między ludźmi. Ostatnio jednak, głównie na skutek badań nad maszynami matematycznymi, pojęcie języka zostało znacznie rozszerzone. Przez język rozumie się wszelki system przesyłania informacji, niezależnie od tego, czy przekazywanie informacji odbywa się między osobnikami ludzkimi, czy między fragmentami organizmu biologicznego, czy też między różnymi częściami maszyny.

Jeżeli na przykład jakiś fragment maszyny znajduje się w określonym stanie, a informacja o tym stanie jest przesyłana za pomocą sygnałów elektrycznych do innego fragmentu tejże maszyny i wywołuje tu określoną reakcję fizyczną, możemy również mówić o funkcji języka: o przenoszeniu informacji między różnymi elementami technicznymi. Nic w istocie nie stoi na przeszkodzie, aby pojęcie języka rozumieć w ten sposób, że zamiast ludzi elementami nadającymi i odbierającymi są maszyny. W języku maszyny alfabet będzie składał się nie z symboli pisanych na przykład atramentem na papierze, lecz z impulsów elektrycznych w odpowiednich urządzeniach elektronowych. Możemy jednak pominąć stronę techniczną zagadnienia i dla uproszczenia przyjąć, że alfabet języka maszyny to zwykły alfabet pisany atramentem na papierze. Nie będzie to miało istotnego wpływu na bieg naszych rozumowań, podobnie jak nieistotne jest, czy założymy, że w języku pisanym piszemy atramentem lub ołówkiem na papierze, czy też rylcem na tabliczce glinianej.

Podobnie rzecz się przedstawia w biologii. Wzajemna wymiana informacji między poszczególnymi częściami organizmu odbywa się również za pomocą języka. Tutaj symbolami mogą być zarówno impulsy elektryczne, jak i związki chemiczne. Z punktu widzenia struktury języka postać fizyczna symboli jest oczywiście nieistotna. Powstają więc pytania: Czy struktura języka naturalnego, języka maszyny, czy też języka biologicznego ma zbliżoną budowę? Czy język może być badany podobnymi środkami, niezależnie od tego, czy jest on językiem społeczeństwa, maszyny, czy też organizmu biologicznego?

Lingwistyka matematyczna próbuje stworzyć ogólne podstawy języka, ważne zarówno dla języka naturalnego, jak i wszelkich innych języków, a więc języka maszyny czy języka biologicznego. Oczywiście każdy z języków posiada swoiste własności, nie występujące w innych językach, tym niemniej we wszystkich językach można znaleźć pewien wspólny trzon, który w różnych szczególnych przypadkach może być rozmaicie modyfikowany.

W niniejszej książeczce nie będziemy pojęcia „język" ograniczali wyłącznie do języka naturalnego; przez „język" będziemy rozumieli zarówno język naturalny, jak język maszyny, jak i język organizmu. Ściślej biorąc, przez „język" chcielibyśmy rozumieć pewien twór abstrakcyjny, który zależnie od przyjętych założeń może być interpretowany jako język w zupełnie dowolnym rozumieniu.

A więc język określimy jako nieskończony zbiór napisów, z których każdy jest pewną kombinacją skończonej liczby symboli ustalonego alfabetu.

Jeszcze raz przypominamy, że symbolami alfabetu niekoniecznie muszą być znaki na papierze.

§ 5. Formalne określenie języka

Obecnie określimy język nieco ściślej, aniżeli uczyniliśmy to w poprzednim paragrafie.

Słownikiem będziemy nazywali skończony zbiór **symboli.** Słownik oznaczymy literą S, natomiast symbole słownika będziemy oznaczali małymi literami a, b, ..., x, y. Skończone ciągi symboli słownika S będziemy oznaczali literami greckimi Φ, Ψ itp., ewentualnie ze wskaźnikami u dołu Φ_1, Φ_2, Ψ_1, Ψ_2 itp. Zbiór wszystkich skończonych ciągów symboli słownika S będziemy nazywali **językiem** określonym na zbiorze S. Języki będziemy oznaczali J_1, J_2, ... itp. [1])

Rozpatrzmy przykłady prostych języków określonych według podanej definicji.

Przykład 1. Język J_1 posiada słownik S_1 składający się z symboli a, b, c. Zdaniami w języku J_1 będą na przykład następujące ciągi symboli:

$a\ b\ b\ b \qquad a\ c\ a\ a \qquad a\ b \qquad a\ a\ a\ a\ b\ c\ c.$

Przykład 2. Język J_2 posiada słownik S_2 składający się z symboli a, p, x. Zdaniami w tym języku będą np. następujące ciągi:

$a\ a\ p\ x \qquad x\ x\ p \qquad a\ p\ p\ a$

itp.

[1] Podana terminologia jest nieco niezgodna z terminologią znaną nam z nauki o językach naturalnych. Punktem bowiem wyjściowym jest tam alfabet. Ciągi symboli alfabetu są nazywane słowami, a ciągi słów — zdaniami. W lingwistyce matematycznej przyjęto jednak pewne uproszczenia zakładając, że wszystkie słowa słownika są jednoliterowe, tak jak to podano w powyższej definicji języka. W literaturze fachowej można też spotkać inną terminologię. Zbiór symboli zwany jest **alfabetem.** Skończone natomiast ciągi symboli nazywane są **słowami.**

Natomiast ciągi

$a\ b\ x\ p$
$p\ x\ c\ a$
$a\ a\ b\ x$

nie należą do języka J_2, gdyż wchodzą do nich symbole nie należące do słownika języka J_2.

Przykład 3. Język J_3 posiada słownik S_3 składający się z symboli (,), $+$, $-$, $\cdot$,), x, y, z. Zdaniami w tym języku będą więc np. następujące ciągi:

$((+ - x\ x\ ( )\cdot$
$+ - - x\ y\ z$
$(x + y)$
$((x + -$

Jeszcze jeden przykład pozwoli nam na lepsze zorientowanie się w znaczeniu podanej definicji języka.

Przykład 4. Język J_4 posiada słownik S_4:

dom *niebo* *góra* *cegła* *jest* [1])

Zdaniami w tym języku będą więc ciągi:

dom niebo góra *cegła jest góra* *niebo cegła*

Oczywiście zdania języka J_4 nie są zdaniami w języku polskim, chociaż występujące w nim słowa należą do słownika języka polskiego. Pozbawione są one również sensu. Jednakże — jak to już wspominaliśmy — zagadnieniem znaczenia chwilowo interesować się nie będziemy. Język dla nas to zbiór odpowiednio pozestawianych symboli.

Zanim przejdziemy do następnego paragrafu, uczynimy kilka uwag na temat podanej definicji języka.

Do dalszych rozważań wygodnie jest przyjąć, że pojedynczy symbol słownika jest także zdaniem. Czasem wprowadzone jest również pojęcie zdania pustego, które składa się z zerowej ilości symboli słownika. Zdania puste oznaczymy symbolem Λ.

Warto zauważyć, że chociaż słownik zawiera skończoną liczbę symboli i chociaż każde zdanie składa się ze skończonej liczby symboli, to liczba zdań w języku jest zbiorem nieskończonym.

[1]) W słowniku tym symbole są wieloliterowe Nic nie stoi jednak na przeszkodzie, abyśmy zapomnieli o alfabecie i napisy takie, jak *dom* czy *niebo*, traktowali jako pojedyncze symbole. Ułatwi to nam znacznie dalsze rozważania.

I na zakończenie jedna uwaga. Przyjęliśmy tutaj, że słownik języka jest raz na zawsze ustalony, co jest oczywiście uproszczeniem, szczególnie w odniesieniu do języków naturalnych. Założenie to jest wygodne, gdyż jego brak komplikowałby nadmiernie badanie języka, z drugiej zaś strony zmiany słownika w czasie — z punktu widzenia struktury języka — wydają się nieistotne. Badamy bowiem wtedy język w jego określonym stanie rozwoju i staramy się znaleźć takie jego własności, które by również były prawdziwe przy innym, zmienionym słowniku.

§ 6. Zdania zbudowane poprawnie

Spośród wszystkich zdań języka wyróżniliśmy podklasę **zdań zbudowanych poprawnie.** Inaczej mówiąc, nie wszystkie zdania, które można utworzyć z symboli słownika, będziemy dopuszczali w języku, a tylko pewne ich zestawienia spełniające specjalne warunki. Na przykład dla języka polskiego nie wszystkie dowolne kombinacje słów ze słownika stanowią zdania zbudowane poprawnie. Z trzech podanych niżej ciągów słów tylko ostatni uznamy za zdanie zbudowane poprawnie. W myśl naszej definicji języka dwa pierwsze ciągi będą również zdaniami, jednakże zdaniami niepoprawnymi.

ławica	*ławicę*	*na*
przed	*przed*	*krótko*
wschodni	*wschodniej*	*przed*
krótko	*krótko*	*zachodem*
chmura	*chmur*	*wiatr*
po	*po*	*rozpędził*
niebo	*nieba*	*ławicę*
zachód	*zachodem*	*chmur*
rozpędzić	*rozpędził*	*ciemniejącą*
ciemnieć	*ciemniejącą*	*po*
wiatr	*wiatr*	*wschodniej*
na	*na*	*stronie*
strona	*stronie*	*nieba* [1]).

[1]) Przykład zaczerpnięty ze skryptu Janiny Żlabowej: *Gramatyka opisowa języka polskiego*. PWN, 1960.

W języku J_3, podanym w poprzednim paragrafie, zdania to po prostu formuły matematyczne. Wśród wszystkich możliwych formuł języka J_3 wyróżniamy także pewną podklasę formuł, które nazywamy klasą formuł zbudowanych poprawnie. Np. ciągi symboli

$$((+ - x\ x()$$
$$((x + -$$

są formułami zbudowanymi niepoprawnie, w języku natomiast J_3 formuły

$$(x + y)$$
$$((x + y) \cdot z)$$

są formułami poprawnymi.

Powstaje więc pytanie ogólne, jak w językach odróżnić zdania poprawne od niepoprawnych. Odpowiedź na to pytanie daje **gramatyka języka.**

§ 7. Gramatyka

Przez gramatykę języka będziemy rozumieli skończony zbiór reguł, pozwalający odróżniać zdania poprawne od zdań niepoprawnych w tym języku.

W szczególnym przypadku w niektórych językach wszystkie zdania, które w tym języku można zbudować (tzn. wszystkie ciągi słów), mogą być zdaniami poprawnymi. Nie ma więc potrzeby odróżniania zdań poprawnych od niepoprawnych. Można również rozważać takie języki, w których zdania składają się tylko z pojedynczych znaków słownika i każdy pojedynczy symbol jest zdaniem poprawnym. Gramatyki obu rodzajów języków nazwalibyśmy trywialnymi. Języki takie są ogólnie spisem ustalonych znaków, przewidzianych na wszystkie możliwe interesujące nas okoliczności, o których chcemy kogoś informować.

Na marginesie uwaga na temat odbiegający nieco od głównego nurtu naszych rozważań. Chodzi mianowicie o to, czy zwierzęta posługują się językiem. W świetle naszej definicji niewątpliwie tak. Istnieje jednak pytanie, czy język ten posiada gramatykę nietrywialną, czy też jest on zbiorem prostych sygnałów, przewidzianych na sygnalizowanie różnych sytuacji występujących w życiu zwierzęcia.

Wróćmy jednak do tematu zasadniczego. Języki, posiadające gramatyki trywialne, są bardzo ubogie; nie będziemy się tutaj nimi zajmować. Przez języki będziemy zawsze rozumieli język posiadający jakąś gramatykę nietrywialną.

Dla języków naturalnych gramatyka, podobnie jak i słownik, nie jest w czasie niezmienna. Inna jest gramatyka języka polskiego z wieku XVI, inna jest gramatyka języka współczesnego i inna jeszcze będzie za lat na przykład sto. Dla uproszczenia przyjmiemy, że zarówno słownik, jak i gramatyka badanych języków są ustalone.

W związku z gramatyką istnieje szereg interesujących problemów teoretycznych. Szczególnie interesujący, przynajmniej w stosunku do języków naturalnych, wydaje się problem następujący: dana jest skończona liczba zdań danego języka; czy można na podstawie takiej „próbki" języka podać gramatykę tegoż języka? Mówiąc ściślej, czy istnieje jakaś metoda — algorytm, jak mówią matematycy — pozwalająca na podstawie skończonej liczby zdań języka znaleźć jego gramatykę? Sytuacja taka spotykana jest w praktyce bardzo często. Na przykład w wyniku prac archeologicznych odkryto jakieś tabliczki z napisami w nieznanym, nie istniejącym już języku. Czy na podstawie takich próbek języka można określić jego gramatykę? Zresztą z podobną sytuacją spotykamy się nie tylko w przypadku odkrycia dawnych wymarłych już języków. Językoznawca badający współczesne języki ma w istocie do rozwiązania identyczne niemal zagadnienie. Świadomość reguł gramatycznych jest bowiem w stosunku do języka czymś późniejszym. Język naturalny nie powstał przecież według założonych z góry reguł gramatycznych. Językoznawca jest tutaj w sytuacji zbliżonej do fizyka, który usiłuje znaleźć prawidłowości w masie pozornie nie powiązanych ze sobą zjawisk. Poszukiwanie gramatyki odbywa się na podstawie analizy dużej ilości materiału językowego.

Podobnych pytań w odniesieniu do teorii gramatyki można by postawić bardzo wiele. Poruszane zagadnienie, jakkolwiek bardzo ciekawe, nie mieści się w ramach niniejszej książki, nie będziemy się nim więc zajmowali. Interesować się natomiast będziemy różnymi rodzajami gramatyk, przydatnych zarówno dla języków naturalnych, jak i innych. Rozważane dalej gramatyki podzielimy na dwie zasadnicze klasy: na **gramatyki syntetyczne** i **gramatyki analityczne**.

Przez gramatykę syntetyczną będziemy rozumieli zespół reguł, pozwalających tworzyć, czyli generować wszystkie zdania poprawne w badanym języku. Natomiast gramatyką analityczną będziemy nazywali — inaczej niż w językoznawstwie — zespół reguł, pozwalających odróżniać zdania poprawne od zdań niepoprawnych. Różnica między obiema gramatykami polega więc na tym, że w pierwszym przypadku reguły gramatyki pozwalają na **konstruowanie, syntezę** zdań poprawnych, natomiast w przypadku gramatyki drugiego rodzaju mamy do czynienia z **analizą** zdania już istniejącego, skonstruowanego; reguły gramatyki analitycznej pozwalają na ustalenie, czy badane zdanie jest zbudowane poprawnie. Można by tu mówić zamiast o gramatyce syntetycznej i analitycznej — o gramatyce **mówiącego** i gramatyce **słuchającego**. Człowiek mówiący konstruuje bowiem zdanie według pewnych reguł, niekoniecznie sobie te reguły uświadamiając, natomiast słuchający analizuje odebrane zdanie, również całkowicie automatycznie, nieświadomie. Konstruowanie i analizowanie zdań poprawnych przebiegają w różny sposób. Dla lepszego uchwycenia różnicy między obu rodzajami gramatyk wyobraźmy sobie, że chcemy zbudować dwie maszyny. Jedna z nich ma tworzyć dowolne poprawne zdania w języku polskim, druga zaś ma badać poprawność zdań wyprodukowanych przez maszynę pierwszą. Pierwsza z maszyn powinna więc produkować zdania według reguł gramatyki syntetycznej, druga natomiast maszyna, analizująca zdania, powinna postępować według reguł gramatyki analitycznej.

Podajemy proste przykłady obu rodzajów gramatyk, które wyjaśnią ewentualne niejasności.

Przykład 1. Rozpatrzmy język J_5 posiadający słownik, składający się z symboli a, b, c, oraz gramatykę syntetyczną w postaci następujących dwu reguł:

$\mathbf{R}_1$: Symbol a jest zdaniem poprawnym w języku J_5.

$\mathbf{R}_2$: Jeżeli α jest zdaniem poprawnym w języku J_5, to zdanie $b\ \alpha\ c$ jest również zdaniem poprawnym w języku J_5.

Reguła $\mathbf{R}_2$ mówi, że dopisując do dowolnego zdania, poprawnego w języku J_5, na początku literę b i jednocześnie na końcu literę c, otrzymamy w języku J_5 również zdanie poprawne. Np. ponieważ a na podstawie reguły $\mathbf{R}_1$ jest zdaniem poprawnym w języku J_5, a więc na podstawie reguły $\mathbf{R}_2$ również poprawnym w J_5 jest zdanie $b\ a\ c$.

Stosując do zdania $b\ a\ c$ ponownie regułę $\mathbf{R}_2$, otrzymamy nowe zdanie poprawne $b\ b\ a\ c\ c$, itd. A więc gramatyka ta pozwala na konstruowanie zdań poprawnych w języku J_5. Za pomocą reguł $\mathbf{R}_1$, $\mathbf{R}_2$ nie wyprodukujemy np. zdania $a\ a\ b\ c$. Jest ono więc w tym języku niepoprawne.

Przykład 2. Niech J_6 będzie językiem posiadającym słownik a, b, c oraz następującą gramatykę analityczną:

$\mathbf{Q}_1$: Jeżeli zdanie ma postać $b\ \alpha\ c$, to zdanie $b\ \alpha\ c$ nazwiemy **redukowalnym** w J_6, a zdanie α — redukcją zdania $b\ \alpha\ c$.

$\mathbf{Q}_2$: Jeżeli pierwszy symbol zdania β jest różny od b lub ostatni symbol zdania β jest różny od c, to zdanie β jest nieredukowalne w J_6.

$\mathbf{Q}_3$: Symbol a jest zdaniem poprawnym w J_6.

$\mathbf{Q}_4$: Jeżeli zdanie γ można zredukować do zdania poprawnego w J_6, to γ jest zdaniem poprawnym w J_6. Jeżeli zaś w wyniku redukcji zdania γ otrzymamy zdanie nieredukowalne różne od a, to γ nie jest zdaniem poprawnym w języku J_6.

Ciąg $b\ b\ a\ c\ c$ będziemy redukowali kolejno, jak następuje:

1. $b\ b\ a\ c\ c$
2. $b\ a\ c$
3. a

Zdanie $b\ b\ a\ c\ c$ przez zastosowanie reguły $\mathbf{Q}_1$ zostało zredukowane do zdania $b\ a\ c$. Stosując do otrzymanego zdania ponownie regułę $\mathbf{Q}_1$, otrzymamy zdanie a. Zgodnie z regułą $\mathbf{Q}_3$ jest to zdanie poprawne w języku J_6. A więc ponieważ zdanie wyjściowe $b\ b\ a\ c\ c$ za pomocą reguł gramatyki języka J_6 zredukowaliśmy do zdania poprawnego a, zdanie wyjściowe $b\ b\ a\ c\ c$ jest zdaniem poprawnym w języku J_6.

Sprawdzimy jeszcze, czy zdanie $b\ b\ b\ b\ a\ c\ c\ c$ jest poprawne w języku J_6. Redukując kolejno zdanie wyjściowe, otrzymamy:

1. $b\ b\ b\ b\ a\ c\ c\ c$
2. $b\ b\ b\ a\ c\ c$
3. $b\ b\ a\ c$
4. $b\ a$

Ostatnie zdanie $b\ a$ jest, zgodnie z regułą $\mathbf{Q}_2$, zdaniem nieredukowalnym, a więc zdania wyjściowego nie udało się nam sprowadzić do zdania

poprawnego. Zdanie wyjściowe jest więc, zgodnie z regułą $\mathbf{Q}_4$, zdaniem niepoprawnym w języku J_6.

Przytoczone przykłady języków były bardzo proste i właściwie poprawność zdania można było w nich ocenić na pierwszy rzut oka. Sprawdzanie więc poprawności zdania według podanych reguł mogłoby się wydawać zbędnym skomplikowaniem prostej sprawy. Jednakże w językach bardziej skomplikowanych określenie poprawności zdania nie może być na ogół dokonane za pomocą prostej obserwacji. Z drugiej strony, mając na uwadze maszynowe badanie poprawności zdań, musimy podać szczegółowy sposób postępowania nawet w najprostszych dla człowieka przypadkach, maszyna bowiem niczego się nie domyśli i może postępować tylko ściśle według podanych reguł.

Reguły gramatyki syntetycznej pozwalają więc na **produkowanie** pewnych zespołów symboli z zadanych symboli wyjściowych. W istocie produkcja taka nie różni się zbytnio od jakiejkolwiek innej produkcji przedmiotów materialnych. Gdybyśmy przyjęli na przykład, że zamiast symboli alfabetu mamy części jakiejś maszyny, a zamiast reguł gramatycznych — reguły składania tych części, zdaniom odpowiadałyby wtedy gotowe urządzenia złożone z wyjściowych elementów. Produkowanie zdań ze słów słownika przypomina więc produkowanie dowolnych przedmiotów z elementów podstawowych, według odpowiednich reguł produkcyjnych. Dlatego też niektórzy nazywają gramatyki syntetyczne **gramatykami produkcyjnymi.** [1])

Gramatyce analitycznej odpowiadałoby rozkładanie przedmiotu na jego części składowe. Gdyby chcieć np. sprawdzić, czy samochód został zbudowany poprawnie, należałoby go w określony sposób rozłożyć na elementy składowe. Gdyby wszystkie te elementy dalej nierozkładalne okazały się właściwymi, to moglibyśmy powiedzieć, że samochód został zmontowany poprawnie.

Podanych analogii nie należy rozumieć zbyt dosłownie, ostatecznie bowiem budowa samochodu i gramatyka to sprawy różne, chociaż występują w obu przypadkach pewne elementy wspólne.

Reasumując, język będziemy określali przez podanie jego słownika S, składającego się z symboli $s_1, s_2, \ldots, s_n$, oraz gramatyki $\mathbf{G}$, zawie-

[1]) Oczywiście można by tu — zamiast przenosić terminologię produkcyjną do gramatyki — uczynić odwrotnie i mówić o **gramatyce produkcji.**

rającej reguły $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_m$. Pisząc krócej, język J będziemy oznaczali: $J = (S, G)$.

W odróżnieniu od innych definicji języka mających zastosowanie w językoznawstwie, przyjmiemy tu, że język to zbiór odpowiednich napisów.

Na marginesie warto dodać, że w związku z pojęciem gramatyki istnieje ważny problem, czy znając na przykład gramatykę syntetyczną jakiegoś języka, możemy znaleźć jego gramatykę analityczną — i odwrotnie. Badania nad tym zagadnieniem są w toku i nie jest ono jeszcze całkowicie rozwiązane, dlatego tutaj nim się zajmować nie będziemy.

W dalszym ciągu zapoznamy się z najprostszymi gramatykami rozważanymi w lingwistyce matematycznej.

LITERATURA

M. Cohen: *Pismo, zarys dziejów*. PWN, Warszawa 1953.
M. Cohen: *Język, jego budowa i rozwój*. PWN, Warszawa 1956.
W. Doroszewski: *Podstawy gramatyki polskiej*. PWN, Warszawa 1952.
J. Fridrich: *Zapomniane pisma i języki*. PWN, Warszawa 1958.
Z. Klemensiewicz: *Podstawowe wiadomości z gramatyki i języka polskiego*. Wyd. IV. PWN, Warszawa 1962.
A. Martinet: *Eléments de linguistique générale*. Wyd. II. A. Colin, Paryż 1962.
T. Milewski: *Wstęp do językoznawstwa*. Wyd. IV. PWN, Łódź 1962.
F. de Saussure: *Kurs językoznawstwa ogólnego*. PWN, Warszawa 1961.
J. Vendryès: *Język*. PWN. Warszawa 1956.
L'art de l'écriture. Le Courrier. UNESCO. Mars 1964.

Rozdział II

GRAMATYKI PROSTE

§ 8. Definicja gramatyki prostej

Gramatykę (syntetyczną) określiliśmy jako metodę produkowania zdań poprawnych. Mówiąc inaczej, gramatyka — to metoda produkowania ciągów symboli o określonej strukturze. Można podać bardzo wiele różnych gramatyk, pozwalających produkować z symboli słownika najrozmaitsze ich zestawienia. Aby tak rozumiane gramatyki nie były tylko formalną zabawą, muszą być one tworzone pod kątem określonych zastosowań.

W rozdziale tym przedstawimy bardzo prosty zespół reguł, który będziemy nazywali gramatyką prostą. Gramatyka ta jest jedną z gramatyk stworzonych i studiowanych przez wspomnianego już w pierwszym rozdziale językoznawcę — Chomsky'ego. Miała ona stanowić uproszczony model dla wielu języków, a przede wszystkim dla języka angielskiego. Okazało się jednak, że gramatyka prosta do opisu nawet języka angielskiego nie jest wystarczająca i Chomsky stworzył dalsze gramatyki, znacznie lepiej opisujące strukturę języka angielskiego. Tym niemniej gramatyki proste wydają się bardzo interesujące, pozwalają bowiem zrozumieć podstawową strukturę języków naturalnych. Z drugiej strony różne modyfikacje gramatyk prostych doskonale nadają się do opisu języków maszyn matematycznych i w tym zakresie są one zupełnie wystarczające.

W rozdziale tym zapoznamy się z zasadniczą ideą tej klasy gramatyk oraz podamy najprostsze ich zastosowania.

Rozważmy język, którego słownik S składa się z dwóch rodzajów symboli: mianowicie, symboli **końcowych** $k_1, k_2, \ldots, k_n$ oraz symboli **pomocniczych** $p_1, p_2, \ldots, p_m$. Zbiór symboli końcowych słownika S będziemy oznaczali przez K_s lub krótko K, natomiast zbiór symboli pomocniczych tegoż słownika przez P_s lub krócej P. Przyjmijmy ponadto, że wśród symboli pomocniczych znajduje się dokładnie jeden wyróżniony symbol z, który nazwiemy symbolem początkowym słownika S.

Gramatykę języka J posiadającego słownik S, który składa się z symboli końcowych, pomocniczych oraz symbolu początkowego, nazwiemy prostą, jeżeli wszystkie reguły tej gramatyki posiadają postać

(1) $\alpha \neq X$ $\qquad \alpha \to X, \qquad$ $K_S = K$, $P_S = P$, $K \cup P = S$

gdzie α oznacza jeden dowolny symbol pomocniczy słownika S, X oznacza dowolny skończony ciąg, składający się z symboli słownika S, oraz spełniony jest warunek, że $\alpha \neq X$.

Regułę (1) należy rozumieć jako regułę przepisywania. Znaczy ona, że w dowolnym wyrażeniu języka J symbol α można zastąpić ciągiem symboli X. Reguły gramatyczne typu (1) pozwalają więc na kolejne przekształcanie ciągów symboli.

Idea gramatyki prostej polega na tym, aby za pomocą reguł gramatycznych postaci (1), wychodząc z symbolu początkowego języka, można było otrzymać wszystkie możliwe poprawne zdania języka J i tylko takie zdania.

Zanim przejdziemy do dalszych rozważań, podamy kilka przykładów reguł gramatyki prostej.

Np. niech język J_7 posiada następujący słownik:

$$\left.\begin{array}{ll} z & \text{symbol początkowy} \\ x & \\ y & \end{array}\right\} \text{symbole pomocnicze}$$

$$\left.\begin{array}{l} a \\ b \end{array}\right\} \text{symbole końcowe}$$

oraz gramatykę w postaci reguł

$$\begin{array}{l} \mathbf{R}_1\colon z \to a\ a \\ \mathbf{R}_2\colon z \to x \\ \mathbf{R}_3\colon x \to a \\ \mathbf{R}_4\colon x \to x\ b \\ \mathbf{R}_5\colon y \to a\ b \\ \mathbf{R}_6\colon z \to y\ x\ x \end{array}$$

Łatwo sprawdzić, że zgodnie z podaną definicją gramatyka języka J_7 jest prosta, gdyż wszystkie reguły $\mathbf{R}_1, \ldots, \mathbf{R}_6$ tej gramatyki mają postać (1). Po lewej stronie strzałki występują symbole pomocnicze, po prawej zaś — dowolne ciągi symboli słownika języka J_7.

§ 9. Produkowanie zdań poprawnych

Gramatyka prosta jest gramatyką syntetyczną, służy więc do produkowania zdań poprawnych. Zanim wyjaśnimy zasadę produkowania zdań poprawnych za pomocą reguł gramatyki prostej, wprowadzimy jeszcze kilka dodatkowych pojęć.

Powiemy, że ciąg symboli Y **wynika bezpośrednio** z ciągu symboli X w gramatyce $\mathfrak{G}$ (symbolicznie $X \Rightarrow Y$), jeżeli istnieją takie ciągi U i V, że $Y = U\,A\,V$, $X = U\,a\,V$ oraz $a \to A$ jest regułą gramatyki $\mathfrak{G}$.

Mówiąc inaczej: jeżeli ciąg symboli Y został otrzymany z ciągu symboli X, przez zastąpienie w ciągu symboli X jednego symbolu pomocniczego odpowiednim ciągiem symboli, zgodnie z regułami gramatyki, to mówimy, że ciąg Y wynika bezpośrednio z ciągu X.

Na przykład w podanym w poprzednim paragrafie języku J_7 ciąg $x\ x\ b\ y$ wynika z ciągu $x\ x\ y$, gdyż można go otrzymać przez zastosowanie do drugiego symbolu x w ciągu $x\ x\ y$ reguły $\mathbf{R}_4$. Podobnie ciąg $a\ b\ x\ x\ z$ wynika z ciągu $a\ b\ x\ z\ z$ przez zastosowanie do przedostatniego symbolu z reguły $\mathbf{R}_2$.

Ciąg Y **wynika** z ciągu X (symbolicznie $X \to Y$), jeżeli istnieją takie ciągi $X_0, X_1, \ldots, X_n$, że $X = X_0$, $Y = X_n$ oraz dla każdego i $(0 \leqslant i \leqslant n-1)$ $X_i \Rightarrow X_{i+1}$. $\bigvee_{X_i\, i=1,n}$ $X = X_0$, $Y = X_n$ $\bigwedge_{0 \le i \le n-1}$ $X_i \Rightarrow X_{i+1}$

Wyrażenie

$$X_0,\ X_1,\ \ldots,\ X_n$$

będziemy nazywali **wywodem** Y z X w oparciu o gramatykę $\mathfrak{G}$.

Mówiąc krócej, ciąg Y wynika z ciągu X, jeżeli da się przekształcić za pomocą reguł gramatycznych w ciąg Y.

Pojęcie wynikania jest tutaj zbliżone do pojęcia konsekwencji przy dowodzeniu. Gramatyka gra wtedy rolę podobną do reguł wnioskowania w systemie formalnym[1]).

[1]) Patrz np. Z. Pawlak: *Automatyczne dowodzenie twierdzeń*. PZWS.

Np. wywód w języku J_7 może mieć postać:

1. $z\ a\ b\ x\ y\ \overset{*}{z}$ $\quad \mathbf{R}_1$
2. $z\ a\ b\ \overset{*}{x}\ y\ a\ a$ $\quad \mathbf{R}_3$
3. $z\ a\ b\ a\ \overset{*}{y}\ a\ a$ $\quad \mathbf{R}_5$
4. $\overset{*}{z}\ a\ b\ a\ a\ b\ a\ a$ $\quad \mathbf{R}_1$
5. $a\ a\ a\ b\ a\ a\ b\ a\ a$

Z prawej strony kolumny podano zastosowane reguły gramatyczne, gwiazdką natomiast u góry zaznaczono symbol, do którego podana reguła się odnosi.

Ostatniego wyrażenia dalej już za pomocą reguł gramatycznych przekształcać nie można, nie zawiera ono bowiem żadnego symbolu pomocniczego.

Inny przykład wywodu w języku J_7:

1. $\overset{*}{z}\ z\ x\ y\ y\ a$ $\quad \mathbf{R}_1$
2. $a\ a\ \overset{*}{z}\ x\ y\ y\ a$ $\quad \mathbf{R}_1$
3. $a\ a\ a\ a\ \overset{*}{x}\ y\ y\ a$ $\quad \mathbf{R}_4$
4. $a\ a\ a\ a\ \overset{*}{x}\ b\ y\ y\ a$ $\quad \mathbf{R}_4$
5. $a\ a\ a\ a\ \overset{*}{x}\ b\ b\ y\ y\ a$ $\quad \mathbf{R}_3$
6. $a\ a\ a\ a\ a\ b\ b\ \overset{*}{y}\ y\ a$ $\quad \mathbf{R}_5$
7. $a\ a\ a\ a\ a\ b\ b\ a\ b\ \overset{*}{y}\ a$ $\quad \mathbf{R}_5$
8. $a\ a\ a\ a\ a\ b\ b\ a\ b\ a\ b\ a$

Ciąg zawierający tylko symbole końcowe nazwiemy **ciągiem końcowym.** W obu przykładach wywodu ostatnie wiersze wywodów były ciągami końcowymi.

Ciąg końcowy X nazwiemy **zdaniem poprawnym** w języku J, jeżeli istnieje wywód $z \to X$ w oparciu o reguły gramatyki języka J.

Wyjaśnimy to dokładniej na przykładzie wywodu zdań poprawnych w języku J_7. Najprostszym zdaniem poprawnym w języku J_7 jest wyrażenie $a\ a$, gdyż na podstawie reguły $\mathbf{R}_1$ wynika ono bezpośrednio z z. Wyrażenie $a\ a\ a\ b$ jest zdaniem poprawnym w J_7, gdyż istnieje wywód:

1. $\overset{*}{z}$ $\quad \mathbf{R}_2$
2. $\overset{*}{x}$ $\quad \mathbf{R}_4$
3. $\overset{*}{x}\ b$ $\quad \mathbf{R}_4$
4. $\overset{*}{x}\ x\ b$ $\quad \mathbf{R}_4$

5. $\overset{*}{x}\ x\ x\ b$ $\quad \mathbf{R}_3$
6. $a\ \overset{*}{x}\ x\ b$ $\quad \mathbf{R}_3$
7. $a\ a\ \overset{*}{x}\ b$ $\quad \mathbf{R}_3$
8. $a\ a\ a\ b$

W wierszu 8 znajduje się już wyrażenie końcowe. Ponieważ wywiedliśmy je z symbolu początkowego z, jest ono w języku J_7 zdaniem poprawnym.

Przykład innego wywodu zdania poprawnego w języku J_7:

1. $\overset{*}{z}$ $\quad \mathbf{R}_6$
2. $\overset{*}{y}\ x\ x$ $\quad \mathbf{R}_5$
3. $a\ b\ \overset{*}{x}\ x$ $\quad \mathbf{R}_4$
4. $a\ b\ \overset{*}{x}\ x\ b$ $\quad \mathbf{R}_3$
5. $a\ b\ a\ \overset{*}{x}\ b$ $\quad \mathbf{R}_3$
6. $a\ b\ a\ a\ b$

Założyliśmy tutaj milcząco, że gramatyki proste nie są **cykliczne,** tzn. reguły nie mają postaci takiej, jak np. w poniższej gramatyce:

$$\mathbf{Q}_1\colon\ z \to x$$
$$\mathbf{Q}_2\colon\ x \to y\ a$$
$$\mathbf{Q}_3\colon\ y \to x\ b$$

gdzie z, x, y są symbolami pomocniczymi, natomiast a i b — symbolami końcowymi. Wtedy bowiem wywody byłyby nieskończone i otrzymanie zdania poprawnego byłoby niemożliwe, jak to widać z powyższego wywodu:

1. $\overset{*}{z}$ $\quad \mathbf{Q}_1$
2. $\overset{*}{x}$ $\quad \mathbf{Q}_2$
3. $\overset{*}{y}\ a$ $\quad \mathbf{Q}_3$
4. $\overset{*}{x}\ b\ a$ $\quad \mathbf{Q}_2$
5. $\overset{*}{y}\ a\ b\ a$ $\quad \mathbf{Q}_3$

. . . .

Mówiąc ściślej, żądamy, aby gramatyki miały taką własność, że dla każdego symbolu pomocniczego x istnieje ciąg końcowy Φ, tzn. ciąg składający się z symboli końcowych, taki że $x \to \Phi$.

§ 10. Gramatyka formuł arytmetycznych

Rozpatrzmy język posiadający następujący słownik:

$$\left.\begin{array}{c} z \\ o \end{array}\right\} \text{symbole pomocnicze} \qquad \left.\begin{array}{c} a \\ b \\ c \\ + \\ - \\ \cdot \\ / \\ (\\ ) \end{array}\right\} \text{symbole końcowe}$$

oraz gramatykę składającą się z reguł:

$$\begin{array}{l} \mathbf{A}_1\colon z \to (z\ o\ z) \\ \mathbf{A}_2\colon z \to a \\ \mathbf{A}_3\colon z \to b \\ \mathbf{A}_4\colon z \to c \\ \mathbf{A}_5\colon o \to + \\ \mathbf{A}_6\colon o \to - \\ \mathbf{A}_7\colon o \to \cdot \\ \mathbf{A}_8\colon o \to / \end{array}$$

Jest to oczywiście gramatyka prosta. Wywód zdania poprawnego w tym języku może mieć np. postać:

1.	$\overset{*}{z}$	$\mathbf{A}_1$
2.	$(z\ o\ \overset{*}{z})$	$\mathbf{A}_1$
3.	$(z\ o\ (\overset{*}{z}\ o\ z))$	$\mathbf{A}_1$
4.	$(\overset{*}{z}\ o\ ((z\ o\ z)\ o\ z))$	$\mathbf{A}_2$
5.	$(a\ \overset{*}{o}\ ((z\ o\ z)\ o\ z))$	$\mathbf{A}_5$
6.	$(a + ((\overset{*}{z}\ o\ z)\ o\ z))$	$\mathbf{A}_3$
7.	$(a + ((b\ \overset{*}{o}\ z)\ o\ z))$	$\mathbf{A}_7$
8.	$(a + ((b \cdot \overset{*}{z})\ o\ z))$	$\mathbf{A}_2$
9.	$(a + ((b \cdot a)\ \overset{*}{o}\ z))$	$\mathbf{A}_6$
10.	$(a + ((b \cdot a) - \overset{*}{z}))$	$\mathbf{A}_4$
11.	$(a + ((b \cdot a) - c))$	

W ostatnim wierszu otrzymaliśmy poprawne zdanie, które w naszym przypadku jest poprawną formułą arytmetyczną. Dla uproszczenia przyjęliśmy tylko trzy zmienne a, b, c oraz cztery działania. Oczywiście liczba zmiennych i symboli operacji może być dowolna.

W formułach tych występują wszystkie nawiasy. Normalnie w zapisie formuł arytmetycznych nie wszystkie nawiasy są konieczne. Np. zamiast

$$(a + ((b \cdot c) - c))$$

możemy zapisać

$$a + b \cdot c - c.$$

Polecamy Czytelnikowi zastanowienie się, jaka powinna być gramatyka takiego uproszczonego języka arytmetycznego.

Zgodnie z regułami gramatyki pojedyncza zmienna jest również formułą poprawną. Natomiast sam symbol działania formułą poprawną nie jest.

§ 11. Gramatyka języka naturalnego

Czy za pomocą odpowiedniej prostej gramatyki można produkować poprawne zdania np. języka polskiego? W pewnym zakresie tak. Weźmy pod uwagę bardzo prosty słownik następującej postaci:

$$\left.\begin{array}{l} z \\ F_R \\ F_c \\ C \end{array}\right\} \text{symbole pomocnicze}$$

$$\left.\begin{array}{l} \textit{Anna} \\ \textit{lubi} \\ \textit{książki} \end{array}\right\} \text{symbole końcowe}^{1)}$$

Przyjmijmy następnie gramatykę posiadającą reguły:

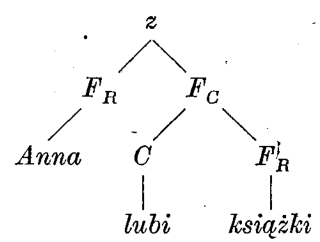
$$\begin{array}{l} \mathbf{R}_1\colon z \to F_R F_C \\ \mathbf{R}_2\colon F_C \to C F_R \\ \mathbf{R}_3\colon F_R \to \textit{Anna} \\ \mathbf{R}_4\colon F_R \to \textit{książki} \\ \mathbf{R}_5\colon C \to \textit{lubi} \end{array}$$

[1]) Jak już wspominaliśmy, w takim przypadku całe słowo, np. *Anna*, uważamy za jeden symbol.

Gramatyka ta jest oczywiście prosta. Postępując według reguł przyjętej gramatyki, możemy otrzymać następujący wywód:

1.	z	$\mathbf{R}_1$
2.	$F_R F_C$	$\mathbf{R}_3$
3.	*Anna* F_C	$\mathbf{R}_2$
4.	*Anna* $C F_R$	$\mathbf{R}_5$
5.	*Anna lubi* F_R	$\mathbf{R}_4$
6.	*Anna lubi książki*	

A więc otrzymaliśmy poprawne zdanie w języku polskim. Z podanego przykładu widać łatwo znaczenie symboli pomocniczych. z oznacza po prostu zdanie, F_R — rzeczownik, C — czasownik oraz F_C — tzw. frazę czasownikową. Przez frazę rozumiemy tu zespół słów, stanowiący pewną spójną całość, grającą określoną rolę w zdaniu. Strukturę wyprowadzonego zdania łatwiej sobie uświadomimy, przedstawiając wywód graficznie w postaci drzewa:



Rozpatrywany przykład jest bardzo prosty i nie pretenduje do wyjaśnienia struktury gramatycznej języka polskiego. Chodzi tu tylko o zilustrowanie myśli przewodniej, matematycznej formalizacji gramatyki.

Dla otrzymywania bardziej skomplikowanych zdań, szczególnie w języku polskim, konieczny jest zespół znacznie bogatszych reguł i poza tym prosta gramatyka tu nie wystarcza[1]).

Inna trudność w stosowaniu tego typu gramatyk do języków naturalnych polega na tym, że są one niejednoznacznie wyznaczane.

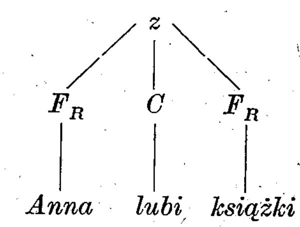
[1]) Model sformalizowanej gramatyki relacyjnej oraz jego zastosowanie do podstawowych zdań języka polskiego był tematem pracy doktorskiej Ireny Bellert.

Np. zdanie *Anna lubi książki* można wywieść według reguł następującej gramatyki:

$\mathbf{P}_1$: $z \to F_R\, C\, F_R$
$\mathbf{P}_2$: $F_R \to$ *Anna*
$\mathbf{P}_3$: $F_R \to$ *książki*
$\mathbf{P}_4$: $C \to$ *lubi*

1.	z	$\mathbf{P}_1$
2.	$F_R C F_R$	$\mathbf{P}_2$
3.	*Anna* $C F_R$	$\mathbf{P}_4$
4.	*Anna lubi* R_R	$\mathbf{P}_3$
5.	*Anna lubi książki*	

Drzewo tego wywodu pokazano niżej:



Czy pierwszy, czy też drugi rodzaj gramatyki jest właściwszy do opisu języków naturalnych? W związku z matematyzacją gramatyki powstaje szereg trudnych do rozstrzygnięcia problemów. Na rozwiązanie ich musimy jeszcze dość długo poczekać. Matematyzacja gramatyki języków naturalnych, mimo że nie obejmuje całości struktury żadnego języka, a w istocie jej stosunkowo małe fragmenty, niewątpliwie znacznie przyczynia się do lepszego rozumienia struktury syntaktycznej języka. Bez takiej formalizacji niemożliwe byłoby prowadzenie badań nad zastosowaniem maszyn matematycznych do tłumaczenia. Z drugiej strony byłoby naiwnością sądzić, że cała skomplikowana, ciągle się zmieniająca struktura języka naturalnego da się całkowicie ująć w karby matematycznej formalizacji. Poprawność zdań języka żywego nie polega na zgodności z obowiązującymi zasadami gramatyki; zdania są uznawane za poprawne na zasadzie zgodności z przyjętymi normami społecznymi, które są różne w zależności od dialektu i historycznie zmienne.

Matematyzacja gramatyki, przynajmniej w odniesieniu do języków naturalnych, ma więc dość ograniczone perspektywy zastosowań, tym niemniej prowadzenie tego rodzaju badań wydaje się jak najbardziej celowe.

Badanie natomiast składni języków sztucznych, takich np. jak języki matematyczne czy języki maszynowe, wydaje się tym ograniczeniom nie podlegać. I tutaj już dzisiaj, przynajmniej w odniesieniu do maszyn matematycznych, formalizacja gramatyki oddaje duże praktyczne usługi.

LITERATURA[1])

Y. Bar-Hillel, M. Perles and E. Shamir: *On formal properties of simple phrase structure grammars*. Applied Logic Branch, The Hebrew University of Jerusalem, Technical Report nr 4, Jerozolima 1960.

Y. Bar-Hillel, C. Gaifman and E. Shamir: *On categorial and phrasestructural grammars*. The Bulletin of the Research Council of Israel, June 1960, 9F, nr 1.

N. Chomsky: *On certain formal properties of grammars*. Information and Control, 2, 137–167, 1959.

N. Chomsky: *A note on phrase structure grammars*. Information and Control, 2, 393–395, 1959.

N. Chomsky: *Three models for the description of language*. IRE Trans. on Information Theory, 1956, IT-2, 113–124.

N. Chomsky, G. A. Miller: *Introduction to the formal analysis of natural languages*. Handbook of mathematical psychology, New York–Londyn 1963, 269–322.

N. Chomsky: *Formal properties of grammars*. Ibid. 323–418.

N. Chomsky, G. A. Miller: *Finitary models of language users*, Ibid. 419–491.

H. Curry: *Foundations of mathematical logic*. Mc Graw-Hill Book Comp. New York 1963.

W. Dingwall: *Transformational grammars: form and theory, a contribution to the history of linguistics*. Lingua, 12 (1963) No. 3, str. 233–275.

R. Péter: *Über die Rekursivität der Begriffe der mathematischen Grammatiken*. A Magyar Tudományos Akadémia, Matematikai Kutató Intézetének Közleményei, 1963 VIII, Ser A. Fasc. 1–2, str. 213–228.

[1]) Większość podanych tu pozycji dotyczy również dwu następnych rozdziałów.

E. M. Uhlenbeck: *An appraisal of transformation theory*. Lingua, 12 (1963) nr 1, str. 1–18.

Computational Linguistics I, Computing Centre of the Hungarian Academy of Sciences. Budapeszt 1963.

Structure of language and its mathematical aspects, Proc. 12th Sympos. in App. Math., Providence, R. J. American Mathematical Society, 1961.

Rozdział III

GRAMATYKI KATEGORIALNE

W poprzednim rozdziale zapoznaliśmy się z pewnym przykładem gramatyki syntetycznej, obecnie zaś poznamy prosty przykład gramatyki analitycznej, tzn. gramatyki, której reguły pozwalają rozstrzygnąć, czy zadany ciąg symboli stanowi zdanie w określonym języku, czy też nie. Przedstawiona tu koncepcja pochodzi od Ajdukiewicza i była rozwijana przez Bar-Hillela i innych. Stanowiła ona teoretyczne podstawy pierwszego tłumaczenia za pomocą maszyny matematycznej z języka angielskiego na rosyjski, dokonanego w 1956 roku w USA. Punktem wyjściowym jest tutaj pojęcie **kategorii syntaktycznej**.

§ 12. Pojęcie kategorii syntaktycznej

Jeżeli w zdaniu *Anna lubi książki* słowo *lubi* zastąpimy słowem *czyta*, otrzymamy nadal zdanie poprawne: *Anna czyta książki*. Słowo *lubi* możemy też zastąpić innym słowem, np. *niesie*, i otrzymamy wtedy zdanie: *Anna niesie książki*.

Podobnie słowo *Anna* możemy zastąpić słowem *Hania* czy też jakimkolwiek innym imieniem i otrzymamy nadal zdanie poprawne. Również słowo *książki* możemy zastąpić szeregiem innych słów, nie naruszając poprawnej struktury zdania. Nie możemy natomiast słowa *lubi* zastąpić słowem np. *ładnie* czy *mama*, otrzymamy bowiem wtedy zdania: *Anna ładnie książki* oraz *Anna mama książki*, które nie są zdaniami poprawnymi w języku polskim.

Z gramatycznego punktu widzenia wszystkie słowa, które możemy wstawić na miejsce jakichś słów w poprawnie zbudowanym zdaniu,

tak aby zdanie pozostało nadal poprawne, spełniają jednakową funkcję gramatyczną. Inaczej mówiąc, posiadają one jednakową **kategorię syntaktyczną**. Jeżeli więc w zdaniu poprawnym jakieś słowo zastąpimy innym słowem takiej samej kategorii syntaktycznej, to poprawność zdania nie zostanie naruszona.

Podobnie możemy mówić o kategorii symboli w formule matematycznej, np. w $(a + (b \cdot c))$ symbol a możemy zastąpić symbolem x i otrzymamy nadal formułę poprawną $(x + (b \cdot c))$. Podobnie symbol $+$ możemy zastąpić symbolem $-$ i otrzymane w ten sposób wyrażenie $(x - (b \cdot c))$ jest nadal poprawną formułą matematyczną. Nie możemy natomiast symbolu x zastąpić znakiem $+$, otrzymamy bowiem wtedy wyrażenie $(+ + (b \cdot c))$, które nie jest poprawną formułą matematyczną. A więc i w przypadku formuł matematycznych możemy operować pojęciem kategorii syntaktycznej.

Oczywiście przy każdym takim podstawieniu zmieniamy sens zdania lub formuły, lecz jak już to kilkakrotnie podkreślaliśmy, sensem zdań chwilowo się nie interesujemy. Struktura zdania natomiast przy takich podstawieniach pozostaje bez zmiany.

Fakty te stanowią podstawę gramatyki kategorialnej.

§ 13. Kategorie syntaktyczne ważniejszych części mowy

W omawianej przez nas koncepcji dwie kategorie przyjęte są jako podstawowe, a wszystkie inne są tworzone z tych dwu kategorii wyjściowych. Podstawowymi kategoriami są: **kategoria zdania** oznaczona symbolem z oraz **kategoria nazwy** oznaczona symbolem n. Kategorie gramatyczne pozostałych części mowy ustala się według następującej reguły:

Jeżeli część mowy C_0 tworzy zdanie z częściami mowy $C_1, C_2, \ldots, C_n$, to C_0 posiada kategorię syntaktyczną $z/k_1, k_2, \ldots, k_n$, gdzie $k_1, k_2, \ldots, k_n$, są kategoriami części mowy $C_1, C_2, \ldots, C_n$.

Np. w zdaniu *Anna czyta książkę* czasownik będzie miał kategorię z/nn, gdyż z dwoma rzeczownikami tworzy zdanie. W zdaniu *deszcz pada* czasownik *pada* posiada kategorię syntaktyczną z/n, gdyż z jednym rzeczownikiem tworzy zdanie. Podwójna kategoria syntaktyczna czasownika wiąże się z faktem, że czasownik może w zdaniu spełniać dwie funkcje: może oznaczać relacje między dwoma przedmiotami, a może też określać stan jednego przedmiotu. W obu tych przypadkach rola czasownika jest całkiem inna i rzecz oczywista oba rodzaje czasowników nie są wymienne.

Przysłówek posiada kategorię syntaktyczną

$$\frac{z}{n\frac{z}{n}}\ ^{1)}$$

gdyż z nazwy i czasownika typu z/n tworzy zdanie. Np. *deszcz bardzo pada*, słowo *bardzo* będzie miało kategorię

$$\frac{z}{n\frac{z}{n}}$$

Spójniki międzyzdaniowe, takie jak *i*, *lub*, *albo* itp., posiadają kategorię syntaktyczną z/zz, gdyż z dwu zdań tworzą nowe zdanie.

W podobny sposób można przypisać kategorie syntaktyczne dowolnym częściom mowy. Jako pouczające ćwiczenie polecamy Czytelnikowi określenie kategorii syntaktycznej przymiotnika.

§ 14. Schematy zdań

Zgodnie z regułami podanymi w poprzednim paragrafie każdemu wyrazowi w zdaniu możemy przyporządkować kategorię syntaktyczną. Zamiast więc badać poprawność zdania możemy na miejsce każdego wyrazu w zdaniu wpisać odpowiadającą mu kategorię syntaktyczną i badać poprawność tak otrzymanego wyrażenia. Wyrażenie otrzymane przez zastąpienie wyrazów odpowiadającymi im kategoriami nazwiemy **schematem zdania.**

Np. zdanie

Anna czyta książkę

posiada schemat

$$n \qquad z/nn \qquad n$$

Schemat zdania

Deszcz bardzo pada

---

[1] Zamiast z/n używamy oznaczenia $\frac{z}{n}$ i podobnie dla bardziej skomplikowanych kategorii.

jest następujący:

$$n \quad \frac{z}{n\frac{z}{n}} \quad \frac{z}{n}$$

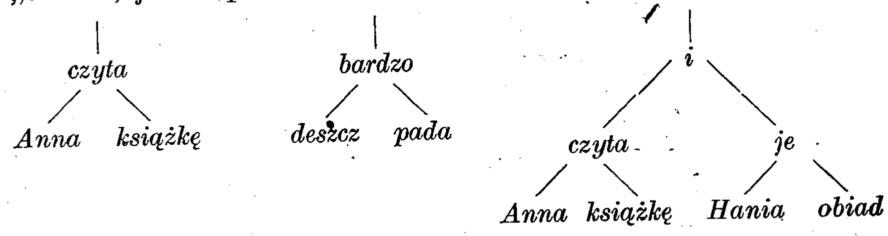
Zdanie

Anna czyta książkę i Hania je obiad

posiada schemat

$$n \quad \frac{z}{nn} \quad n \quad \frac{z}{zz} \quad n \quad \frac{z}{nn} \quad n$$

Do niektórych celów wygodnie jest przedstawić zdania w postaci „drzew", jak to pokazano niżej:



Wyraźnie wtedy widać związki między poszczególnymi wyrazami zdania. Zamiast rysunku można by też pisać nawiasy grupujące wyrazy w zdaniu, podobnie jak to czynimy w formułach matematycznych.

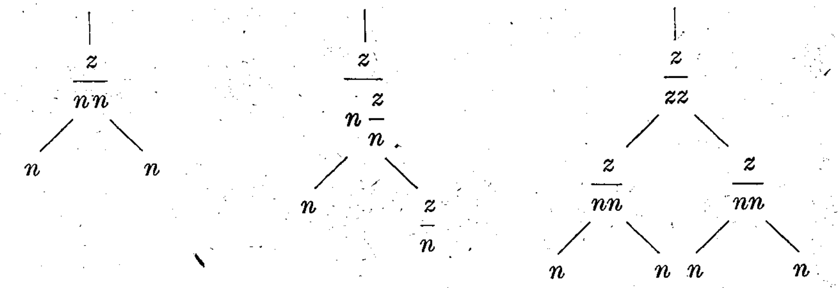
Używając nawiasów, zapiszemy dyskutowane zdania w postaci:

((*Anna*) *czyta* (*książkę*))
((*Deszcz*) *bardzo* (*pada*))
(((*Anna*) *czyta* (*książkę*)) *i* ((*Hania*) *je* (*obiad*)))

Schematy zdań będą miały wtedy postać:

1. $((n)\frac{z}{nn}(n))$
2. $\left((n)\frac{z}{n\frac{z}{n}}\left(\frac{z}{n}\right)\right)$
3. $\left(\left((n)\frac{z}{nn}(n)\right)\frac{z}{zz}\left((n)\frac{z}{nn}(n)\right)\right)$

Schematy zdań wraz z grupującymi kategorie nawiasami mają już postać zwykłych formuł matematycznych i do sprawdzenia ich poprawności można podać wygodne reguły. Schematy zdań wygodnie jest czasem przedstawiać również w postaci drzew, jak to pokazano niżej:



§ 15. Redukowanie schematów zdań

Pojęcie redukcji wyjaśniliśmy już w rozdziale pierwszym. Ma ona na celu, poprzez „upraszczanie" redukowanego wyrażenia, otrzymanie zapisu, który się już dalej redukować nie da. Zależnie od otrzymanego wyniku redukcji wyrażenie początkowe jest poprawne bądź też nie.

Dla schematów zdań przyjmiemy następującą regułę redukcji

$$\mathbf{R}_1:\quad \left((\alpha)\,\frac{\gamma}{\alpha\beta}\,(\beta)\right) \to (\gamma),$$

gdzie α, β i γ są dowolnymi kategoriami syntaktycznymi.

Reguła ta mówi, że jeżeli w jakimś nawiasie, po obu stronach kategorii syntaktycznej znajdującej się w środku nawiasu, kategorie są identyczne jak w „mianowniku" kategorii środkowej, to obie skrajne kategorie oraz „mianownik" środkowej kategorii możemy „skrócić". Reguła ta przypomina rzeczywiście skracanie ułamków.

Dlatego regułę $\mathbf{R}_1$ będziemy nazywali regułą **skracania kategorii syntaktycznych.** Zgodnie z tą regułą możemy wykonać np. skrócenie:

$$1.\quad \left((n)\,\frac{z}{nn}\,(n)\right) \to (z)$$

$$2.\quad \left((n)\,\frac{z}{n\,\frac{z}{n}}\,\left(\frac{z}{n}\right)\right) \to (z)$$

Schemat zdania nazwiemy zredukowanym jednokrotnie, jeżeli dowolną parę jego nawiasów zredukujemy zgodnie z regułą $\mathbf{R}_1$.

Schemat zdania niezredukowany nazwiemy zredukowanym zero razy. Zdanie zredukowane jednokrotnie możemy zredukować ponownie, aż otrzymamy schemat, który dalej się już redukować nie da. Schemat zdania, którego dalej już redukować nie można, nazwiemy **schematem końcowym.** Schemat zredukowany i-krotnie oznaczymy Φ_i. W szczególności Φ_0 oznacza schemat zdania niezredukowany.

Gramatyka kategorialna składa się z dwu reguł. Pierwsza reguła tej gramatyki, to podana reguła skracania kategorii syntaktycznych $\mathbf{R}_1$, druga reguła tej gramatyki jest natomiast następująca:

$\mathbf{R}_2$: Jeżeli schemat końcowy zdania ma postać (z), to zdanie jest zbudowane poprawnie, jeżeli zaś schemat końcowy zdania jest różny od (z), to zdanie nie jest zbudowane poprawnie.

Przykład 1

(((*Anna*) *czyta* (*książkę*)) *i* ((*Hania*) *je* (*obiad*)))

$$\Phi_0\quad \left(\left((n)\quad \frac{z}{nn}\quad (n)\right)\frac{z}{zz}\left((n)\quad \frac{z}{nn}\quad (n)\right)\right)$$

$$\Phi_1\quad \left((z)\quad \frac{z}{zz}\quad \left((n)\quad \frac{z}{nn}\quad (n)\right)\right)$$

$$\Phi_2\quad \left((z)\quad \frac{z}{zz}\quad (z)\right)$$

$$\Phi_3\quad (z)$$

A więc zdanie to jest zbudowane poprawnie. Natomiast zdanie

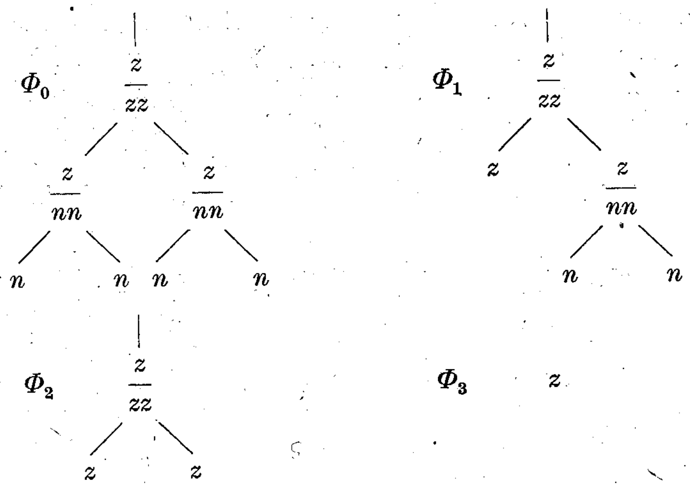
Anna czyta bardzo

nie jest poprawne, gdyż redukcja schematu tego zdania

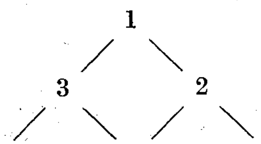
$$\left((n)\,\frac{z}{nn}\left(\frac{z}{n\,\frac{z}{n}}\right)\right)$$

nie prowadzi do kategorii z.

Przebieg redukcji schematu zdania wygodnie jest prześledzić na „drzewie".



Redukowanie schematu przebiegało w rozpatrywanym przykładzie w kolejności według numerów 3, 2, 1, jak to pokazano niżej



Oczywiście redukcję można przeprowadzić również w innej kolejności, np. 2, 3, 1, gdyż kolejność redukcji nie odgrywa tu żadnej roli.

Redukcja, jeżeli prześledzi się ją na drzewie, przypomina obcinanie gałęzi drzewa, aż do otrzymania samego pnia.

Przedstawiona metoda obejmuje bardzo wąski fragment gramatyki. Nie uwzględnione są tutaj odmiany rzeczownika, czasownika i szereg innych, bardziej skomplikowanych form gramatycznych. Dla uzyskania szerszego zakresu formalizacji gramatyki konieczne jest zastosowanie znacznie bogatszych środków matematycznych niż te, których użyliśmy w naszych prostych przykładach. Dla różnych języków formalne ujęcie gramatyki jest różne. Inaczej na przykład wygląda formalizacja języka rosyjskiego, inaczej angielskiego. Idea jednakże w obu przypadkach jest jednakowa.

Przedstawione w tym rozdziale pojęcia są przystępnie wyjaśnione w książce H. Stonerta pt. *Język i nauka.*

Metoda Ajdukiewicza została rozwinięta przez Bar-Hillel'a i innych (w Polsce przez prof. Romana Suszkę) i jak to wspominaliśmy, stanowiła podstawę pierwszego mechanicznego tłumaczenia.

LITERATURA

K. Ajdukiewicz: *Język i poznanie.* PWN, Warszawa 1960.

Y. Bar-Hillel: *A quasi-arithmetical notation for syntactic description.* Language, 1953, 29, 47–58.

R. Ł. Dobruszyn: *Matematyczne metody w lingwistyce.* Wiadomości Matematyczne, Seria II, VI. 2, 1963 str. 217–242.

O. S. Kułagina: *Ob odnom sposobie opriedielenija gramaticzeskich poniatij na bazie mnożestwa.* Problemy Kibiernietiki. Moskwa 1959, str. 203–234.

J. Lambek: *The mathematics of sentence structure.* Amer. Mathematical Monthly, 1958, 65, 154–170.

H. Stonert: *Język i nauka.* Wiedza Powszechna, Warszawa 1964.

Rozdział IV

GRAMATYKI SKOŃCZENIE STANOWE

§ 16. Maszyna Turinga

Gramatykę określiliśmy jako zbiór reguł, pozwalający **konstruować** zdania poprawne z symboli słownika, bądź też jako zbiór reguł, pozwalający **sprawdzać** poprawność zdań składających się z symboli ustalonych słownika. Konstruowanie bądź sprawdzanie zdań jest więc czynnością mechaniczną, wykonywaną ściśle według reguł gramatyki języka. Obie te czynności mogą być więc wykonywane za pomocą odpowiedniej maszyny, realizującej odpowiednie reguły gramatyczne. Inaczej mówiąc gramatykę można również określić przez podanie konstrukcji maszyny do sprawdzania bądź też produkowania zdań. Dlatego czasem w takim przypadku mówi się skrótowo, że gramatyka to maszyna realizująca odpowiednie reguły.

Może być jednak również sytuacja odwrotna. Mamy zadaną jakąś maszynę, która może produkować ciągi symboli. Pytamy, jaka jest gramatyka języka takiej maszyny, tzn. jakie są reguły tworzenia zdań przez maszynę. Odpowiedzią na to pytanie jest właściwie sama konstrukcja maszyny. Znając konstrukcję maszyny, zasadę jej działania, będziemy wiedzieli, jakie zdania mogą być przez maszynę produkowane.

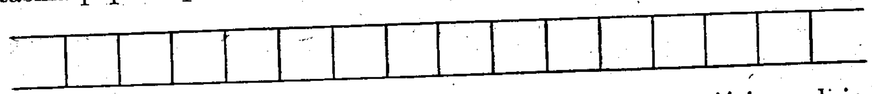
Najprostszą maszyną, która może produkować lub sprawdzać ciągi symboli, jest tzw. **maszyna Turinga.** Matematyk angielski A. Turing podał w 1937 roku koncepcję bardzo prostej maszyny do operowania na symbolach. Od jego nazwiska nazwano ją maszyną Turinga. Maszyna Turinga nie była pomyślana jako projekt nadający się do technicznej realizacji, lecz raczej jako uproszczony maksymalnie model tego, co można rozumieć przez maszynę matematyczną.

Pojęcie maszyny Turinga odgrywa dzisiaj dużą rolę w teorii maszyn

matematycznych. Nic więc dziwnego, że matematycy zajmujący się lingwistyką próbowali zbadać, jakiego rodzaju analizę gramatyczną mogą przeprówadzać maszyny Turinga. Ściślej biorąc, w rozważaniach tych chodzi raczej o pewną uproszczoną wersję maszyn Turinga, tzw. automaty skończone. Możliwości automatów skończonych są znacznie uboższe aniżeli maszyn Turinga, jednakże tutaj nie będziemy tych spraw bliżej dyskutowali. Gramatyki automatów skończonych są nazywane gramatykami skończenie stanowymi.

W dalszym ciągu wyjaśnimy bliżej te pojęcia.

Podstawowym elementem maszyny Turinga jest **nieskończona** taśma papieru podzielona na **kratki.**



Założenie nieskończoności taśmy nie może być oczywiście spełnione w rzeczywistości. Jak już wspominaliśmy, maszyna Turinga nie jest jednak projektem technicznego urządzenia, a raczej pewnym pojęciem abstrakcyjnym i dlatego założenie nieskończoności taśmy może być przyjęte bez zastrzeżeń.

W każdej kratce taśmy może być zapisany jeden z symboli s_0, s_1, ..., s_n słownika S maszyny. Słownik maszyny jest skończony. s_0 będziemy nazywali symbolem pustym i oznaczymy go przez Λ.

Przyjmiemy ponadto, że maszyna może znajdować się w jednym z m możliwych **stanów** q_1, q_2, ..., q_m. Co to jest stan maszyny — nie będziemy wyjaśniać bliżej. Po zapoznaniu się z działaniem maszyny Czytelnik zrozumie z łatwością pojęcie stanu.

Maszyna w każdej chwili może „obserwować" jedną kratkę taśmy. Jeżeli maszyna aktualnie znajduje się w stanie q_j i w kratce „obserwowanej" wtedy przez maszynę jest zapisany symbol s_i z alfabetu maszyny, to mówimy, że maszyna jest w sytuacji (q_j, s_i).

Wszystkie stany maszyny dzielą się na dwie klasy, na **stany czynne** i **stany bierne.** Jeżeli maszyna znajduje się w stanie czynnym, to wykonuje ona **ruch.** Każdy ruch maszyny składa się z trzech elementów:

1. Zmiany stanu wewnętrznego z q_j na q_j'.
2. Zmiany obserwowanego symbolu. Zmiana ta polega na wydrukowaniu na miejscu symbolu s_i nowego symbolu słownika s_i'. W szczególności może to być symbol pusty Λ, co oznacza po prostu wymazanie poprzedniego symbolu s_i.

3. Zmiany obserwowanej kratki taśmy. Zmiana kratki może być trojakiego rodzaju:

P — obserwacja kratki sąsiedniej z prawej strony,

L — obserwacja kratki sąsiedniej z lewej strony,

N — obserwacja tej samej kratki.

W ostatnim przypadku obserwowana kratka po prostu nie ulega zmianie.

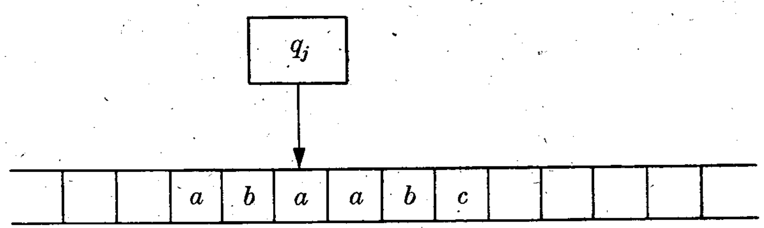
W wyniku ruchu maszyna przechodzi od jednej sytuacji (q_j, s_i) do nowej sytuacji (q_j', s_i'), co symbolicznie zapiszemy

$$(q_j, s_i) \rightarrow (q_j', s_i')$$

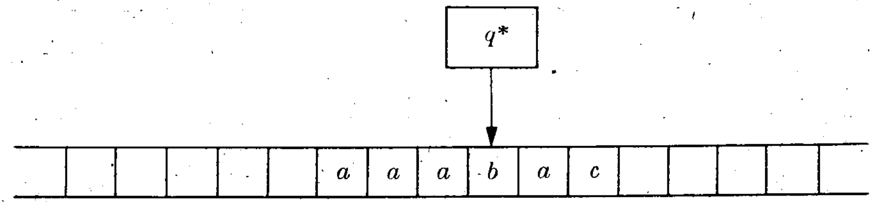
Jeżeli maszyna znajduje się w stanie biernym, nie wykonuje ona ruchu i przerywa działania. Dlatego stany bierne maszyny nazwiemy też **stanami końcowymi.**

Działanie maszyny Turinga jest więc następujące: na taśmie zapisane są przed rozpoczęciem działania maszyny jakieś symbole oraz maszyna znajduje się w stanie q_j. Ponadto maszyna obserwuje jeden z kwadratów taśmy.

Sytuacja ta jest przedstawiona poniżej:



W prostokącie podano stan maszyny. Strzałka wychodząca z tego prostokąta wskazuje obserwowany symbol na taśmie. Jeżeli q_j jest stanem czynnym, maszyna wpisuje na miejscu obserwowanego symbolu inny symbol słownika, zmienia swój stan oraz przechodzi do obserwowania nowego kwadratu, przechodząc w ten sposób do nowej sytuacji. W tej nowej sytuacji wszystko powtarza się jak w przypadku sytuacji poprzedniej. W rezultacie maszyna wykonuje ruchy tak długo, aż znajdzie się w stanie końcowym q^*. Wtedy maszyna zatrzymuje się i na taśmie znajduje się jakiś ciąg symboli na ogół inny, aniżeli był przed rozpoczęciem działania maszyny.



Aby działanie maszyny było określone, każdej sytuacji musi być jednoznacznie przyporządkowany ruch maszyny. Tzn. dla każdego stanu i dla każdego symbolu alfabetu muszą być określone:

a) następny stan maszyny,

b) jaki symbol ma być wydrukowany w obserwowanej kratce,

c) następna kratka, która ma być przez maszynę obserwowana.

Pisząc symbolicznie, każdej parze (q_j, s_i) musimy przyporządkować trójkę (x, q_j', s_i'), gdzie x oznacza jedną z liter L, P, N. Wyrażenie (x, q_j, s_i) nazwiemy **instrukcją** maszyny. Jeżeli takie przyporządkowanie istnieje, ruch maszyny dla każdej sytuacji jest jednoznacznie określony. Przyporządkowanie takie jest pewną funkcją. Ponieważ liczba stanów maszyny jest skończona i słownik jest skończony, więc liczba sytuacji jest również skończona i wynosi $(n + 1) \cdot m$, gdzie n jest liczbą symboli słownika, a m — liczbą stanów maszyny. Funkcję tę można więc przedstawić w postaci tabelki

	q_1	q_2	⋮	q_m
s_0	$r_{0,1}$	$r_{0,2}$	⋮	$r_{0,m}$
s_1	$r_{1,1}$	$r_{1,2}$	⋮	$r_{1,m}$
....	....	....	....	....
s_n	$r_{n,1}$	$r_{n,2}$	⋮	$r_{n,m}$

gdzie $r_{i,j}$ oznacza instrukcję, którą wykona maszyna, jeżeli znajdzie się w sytuacji (s_i, q_j). Tablica taka określa więc jednoznacznie zachowanie się maszyny we wszystkich możliwych sytuacjach. Tablicę tę będziemy więc nazywać **tablicą charakterystyczną** maszyny.

W następnym paragrafie poznamy przykład maszyny Turinga.

§ 17. Przykład maszyny Turinga

Podany tu przykład maszyny Turinga jest niezmiernie prosty. Ma on głównie na celu zilustrowanie poprzednio wprowadzonych pojęć: stanu, alfabetu, sytuacji itp.

Rozpatrzmy działanie maszyny Turinga, którą oznaczymy przez $\mathfrak{M}_1$. Słownik S_1 maszyny $\mathfrak{M}_1$ składa się z trzech symboli: Λ, a, b (Λ — jest symbolem pustym). Zbiór stanów Q_1 maszyny $\mathfrak{M}_1$ jest dwuelementowy, tj. maszyna może znajdować się w jednym ze stanów q_0, q_1^*. Stan q_0 jest stanem czynnym, stan q_1^* — stanem biernym (końcowym).[1]

Tablica charakterystyczna maszyny $\mathfrak{M}_1$ ma postać

	q_0	q_1^*
Λ	$Pq_1^*\Lambda$	$Nq_1^*\Lambda$
a	Pq_0b	$Nq_1^*\Lambda$
b	Pq_0a	$Nq_1^*\Lambda$

Działanie maszyny jest więc następujące:

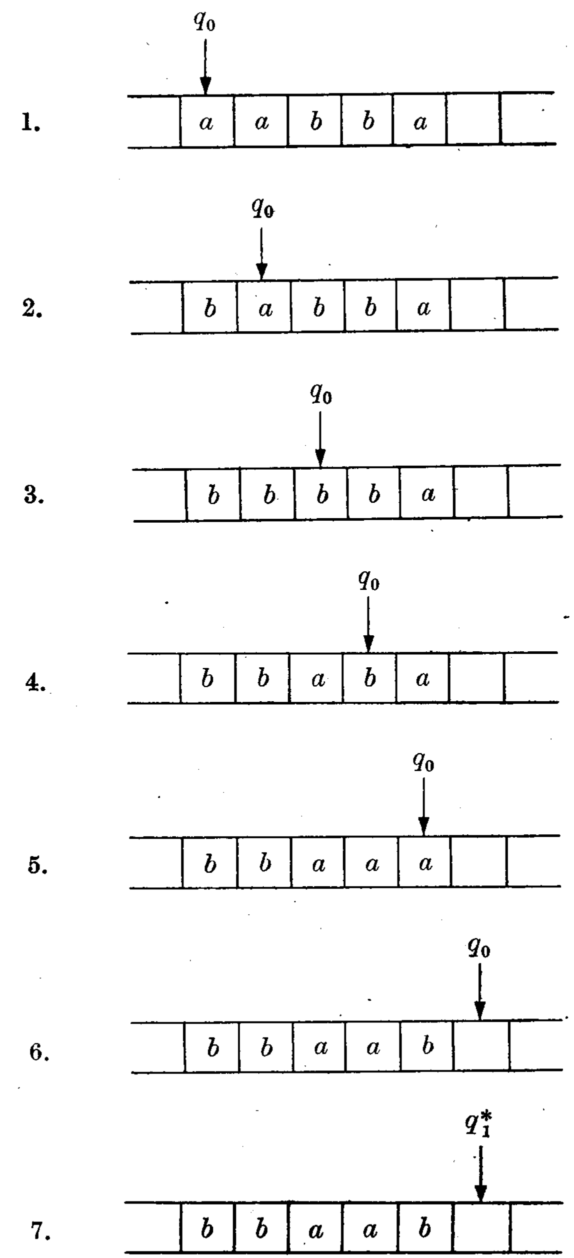
Jeżeli maszyna $\mathfrak{M}_1$ znajduje się w stanie q_0 i obserwuje symbol pusty, przechodzi do następnej kratki po prawej stronie i zatrzymuje się. Jeżeli zaś w stanie q_0 maszyna obserwuje symbol a, to na jego miejsce wpisuje symbol b i przechodzi do obserwowania sąsiedniej komórki z prawej strony; jeżeli natomiast w stanie q_0 maszyna obserwowała symbol b, to zastępuje go symbolem a i również przechodzi do obserwowania sąsiedniej komórki z prawej strony.[2]

Mówiąc krócej, jeżeli na taśmie przed rozpoczęciem liczenia zapisany jest ciąg symboli składający się z dowolnej kombinacji liter a i b, to maszyna $\mathfrak{M}_1$ zamienia litery a na b i odwrotnie; zatrzymuje się na pierwszym pustym miejscu.

Kolejne ruchy maszyny dla ciągu początkowego na taśmie $a\,a\,b\,b\,a$ są następujące:

[1] W dalszym ciągu stany końcowe będziemy oznaczali gwiazdką.

[2] Pustą kratkę maszyna czyta jak symbol Λ i na odwrót.



W kolejnych ruchach maszyny podano symbole znajdujące się na taśmie. Strzałką zaznaczono odczytywaną komórkę w każdym ruchu, a nad strzałką podano odpowiedni stan maszyny. Jakkolwiek przykład jest bardzo prosty, podamy jeszcze ciąg sytuacji oraz kolejne instrukcje wykonywane przez maszynę w każdym ruchu.

Ruch	Sytuacja	Instrukcja
1.	(q_0, a)	$(\mathrm{P}\ q_0\ b)$
2.	(q_0, a)	$(\mathrm{P}\ q_0\ b)$
3.	(q_0, b)	$(\mathrm{P}\ q_0\ a)$
4.	(q_0, b)	$(\mathrm{P}\ q_0\ a)$
5.	(q_0, a)	$(\mathrm{P}\ q_0\ b)$
6.	(q_0, Λ)	$(\mathrm{N}\ q_1^*\ \Lambda)$

Instrukcję dla każdej sytuacji wypisano oczywiście z tablicy charakterystycznej maszyny $\mathfrak{M}_1$.

W podobny sposób można określić dowolne manipulacje na symbolach.

Istnieje hipoteza, że wszystkie algorytmy [1]) mogą być realizowane za pomocą odpowiedniej maszyny Turinga.

Ponieważ maszyna Turinga wykonuje bardzo proste operacje elementarne, więc tablica charakterystyczna maszyny dla nieco bardziej skomplikowanych czynności jest już bardzo skomplikowana i praktycznie nieużyteczna. Ale jak to już wspominaliśmy, nie chodzi tutaj o rzeczywiste rozwiązywanie jakichś zagadnień matematycznych za pomocą maszyn Turinga, lecz o badanie ogólnych własności tego rodzaju maszyn. Do tego celu podana przez Turinga postać maszyny matematycznej jest całkowicie odpowiednia. Zresztą wielu matematyków uważa, że maszyny Turinga można uważać za wyidealizowany model współczesnych elektronicznych maszyn matematycz-

[1]) Algorytm to przepis wykonywania pewnych czynności według podanych reguł.

nych. Tak więc teoria maszyn Turinga obejmuje w zasadzie wszelkie istniejące maszyny.

Maszyna Turinga, posiadająca skończoną taśmę o z góry zadanej liczbie kratek, jest nazywana **automatem skończonym.** W dalszym ciągu dla uproszczenia nie będziemy rozróżniać pojęć maszyny Turinga oraz automatu skończonego.

Automaty skończone mogą być używane do produkowania bądź analizowania ciągów symboli. Ciągi tych symboli zależą od struktury automatu. A więc możemy tu mówić o gramatyce tych ciągów, nie są one bowiem dowolne, a budowa ich posiada pewne prawidłowości. Nie więc dziwnego, że matematycy zajmujący się lingwistyką zaczęli bliżej badać języki, które mogą być produkowane przez automaty skończone, ich gramatyki oraz związki między gramatykami.

Gramatyki języków automatów skończonych nazywane są **gramatykami skończenie stanowymi.**

W następnych paragrafach zapoznamy się z tą klasą gramatyk, syntetycznych, jak i analitycznych.

§ 18. Analiza zdań za pomocą automatów skończonych

Dla prostoty nie będziemy w tym paragrafie rozpatrywali języków naturalnych, lecz bardzo proste języki sztuczne. Uprościmy również nieco pojęcie automatu skończonego. Przyjmiemy mianowicie, że instrukcje w automatach rozpatrywanych mają postać P q_i, to znaczy, że w każdym ruchu maszyna przechodzi do badania sąsiedniej kratki z prawej strony (a więc przejście do sąsiedniej kratki z lewej strony, bądź pozostanie w miejscu jest niedopuszczalne). Ponadto maszyna nie zmienia napisu na taśmie, tzn. nic nie zapisuje na miejscu obserwowanych symboli, a tylko przechodzi do odpowiedniego stanu.

Praca więc takiego uproszczonego automatu skończonego polega na czytaniu kolejnych symboli zapisanych na taśmie i przechodzeniu do odpowiednich stanów.

Ciąg symboli jest zdaniem zbudowanym poprawnie, jeżeli po odczytaniu ostatniego symbolu tego ciągu maszyna znajdzie się w stanie końcowym. Automat taki będziemy nazywali **analizującym.**

Mówiąc nieco dokładniej, automat analizujący posiada skończony słownik S symboli $s_0, s_1, \ldots, s_n$, skończony zbiór stanów wewnętrznych $Q = q_0, q_1, \ldots, q_i^*, \ldots, q_m^*$. Stan q_0 będziemy nazywali stanem począstkowym. Stany $q_i^*, \ldots, q_m^*$ są stanami końcowymi, stany $q_0, q_1, \ldots, q_{i-1}$ są stanami czynnymi.

Ciąg symboli
$$X = a_0, a_1, \ldots, a_k$$
alfabetu S jest **akceptowany** przez automat skończony $\mathfrak{A}$ (lub X jest zdaniem poprawnym według automatu $\mathfrak{A}$), jeżeli ciąg stanów $p_0, p_1, \ldots, p_{k+1}$ jest określony następująco:

1. $p_0 = q_0$
2. $p_{i+1} = T(p_i, a_i)$,

gdzie $0 \leqslant i \leqslant k$, a T jest tablicą charakterystyczną automatu oraz p_{k+1} jest stanem końcowym.

Znaczy to, że automat, zaczynając analizę ciągu symboli, jest zawsze w stanie początkowym q_0. Potem odczytuje następny z prawej strony symbol ciągu i zgodnie z tablicą charakterystyczną automatu przechodzi do nowego stanu, zależnego od poprzedniego symbolu i poprzedniego stanu. Jeżeli po odczytaniu ostatniego symbolu badanego ciągu automat znajdzie się w stanie końcowym, to badany ciąg jest przez automat akceptowany lub — mówiąc inaczej — badany ciąg jest według automatu $\mathfrak{A}$ zdaniem poprawnym.

Rozpatrzmy prosty przykład automatu analizującego ciągi wyrażeń. Automat ten oznaczymy przez $\mathfrak{M}_2$. Słownik automatu $\mathfrak{M}_2$ składa się z trzech symboli: a, b, c. Automat $\mathfrak{M}_2$ posiada pięć stanów: $q_0, q_1, q_2, q_3, q_4^*$. Stan q_4^* jest stanem końcowym, pozostałe stany są czynne. Stan q_0 jest stanem początkowym automatu $\mathfrak{M}_2$.

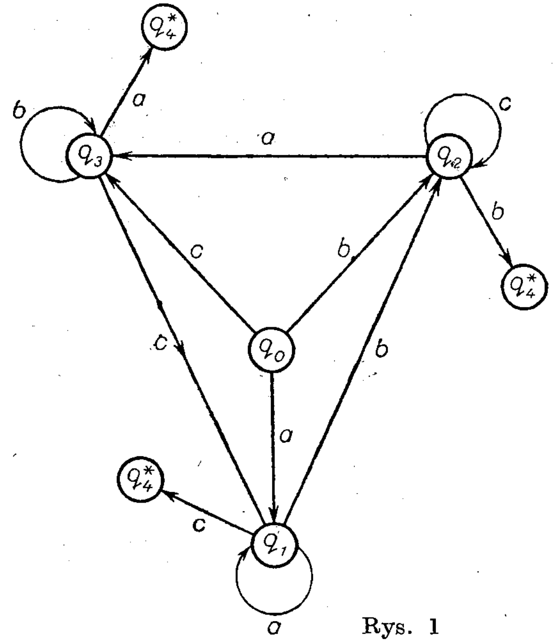
Automat $\mathfrak{M}_2$ posiada następującą tablicę charakterystyczną:

	q_0	q_1	q_2	q_3	q_4^*
a	P q_1	P q_1	P q_3	N q_4^*	N q_4^*
b	P q_2	P q_2	N q_4^*	P q_3	N q_4^*
c	P q_3	N q_4^*	P q_2	P q_1	N q_4^*

Automat ten działa w ten sposób, że wychodząc ze stanu q_0, zależnie od odczytanej litery przechodzi do jednego ze stanów q_1, q_2 bądź q_3. W każdym z tych stanów przy odczytaniu jednej z liter,

pozostaje w tym samym stanie, przechodzi do stanu końcowego q_4^* lub też do jednego ze stanów q_1, q_2, q_3.

Tablica taka jest nieco nieprzejrzysta i dlatego działanie automatu wygodniej jest przedstawiać w postaci rysunku, jak to pokazano na



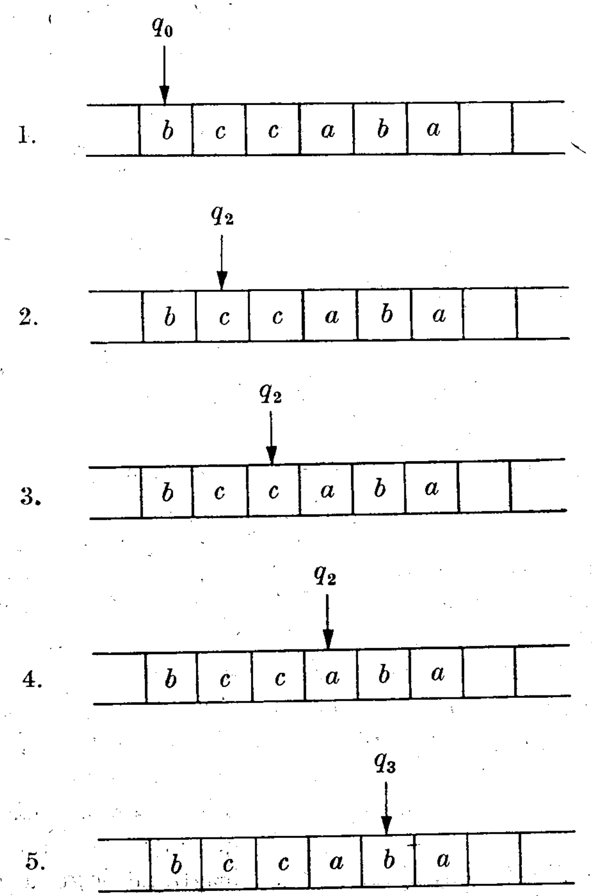
Rys. 1

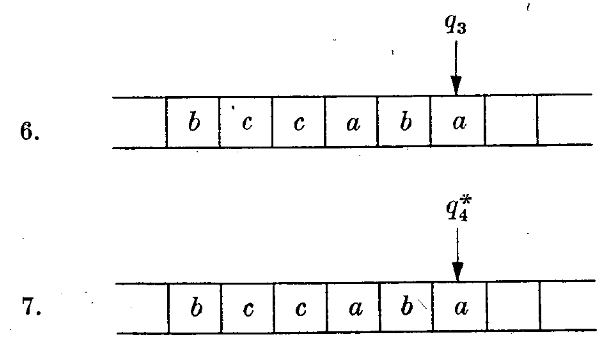
rysunku 1. Dla prostoty na rysunku nie wpisano symboli P i N występujących w tablicy.

Stany automatu na tym rysunku przedstawione są w postaci kółek z napisanym w środku symbolem odpowiedniego stanu. Kółka są połączone za pomocą strzałek. Jeżeli automat jest np. w stanie q_0 i po odczytaniu symbolu b ma przejść do stanu q_2, to kółka reprezentujące stany q_0 i q_2 są połączone strzałką skierowaną od q_0 do q_2 i strzałkę tę oznaczymy literą b. W szczególnym przypadku, jeżeli po odczytaniu jakiegoś symbolu automat ma nie zmienić stanu, to strzałka odpowiadająca temu symbolowi zaczyna się i kończy na tym samym kółku. Ponieważ w każdym stanie automat może odczytać jeden z trzech symboli słownika, z każdego kółka stanu muszą wychodzić trzy strzałki. Z drugiej strony po odczytaniu dowolnego symbolu automat może przejść tylko do jednego stanu, więc z każdego kółka może wychodzić tylko jedna strzałka oznaczona danym symbolem słownika.

Tak zbudowany rysunek nazywamy **grafem** automatu. Graf automatu, podobnie jak tablica charakterystyczna, jednoznacznie określa postępowanie automatu. Z grafu wyraźnie widać, jak będzie działał automat przy czytaniu dowolnego ciągu symboli.

Prześledzimy np. działanie automatu $\mathfrak{M}_2$ przy sprawdzaniu, czy ciąg $b\ c\ c\ a\ b\ a\ a$ jest zdaniem poprawnym. Kolejne ruchy automatu będą miały postać następującą:





Po odczytaniu ostatniego symbolu ciągu automat przeszedł do stanu końcowego q_4^*, a więc badany ciąg jest zdaniem poprawnym według automatu $\mathfrak{M}_2$.

Kolejne sytuacje oraz odpowiadające im instrukcje w kolejnych ruchach analizy wyrażenia $b\ c\ c\ a\ b\ a$ mają postać jak niżej:

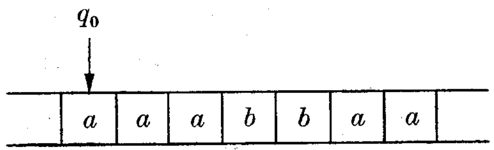
Ruch	Sytuacja	Instrukcja
1.	(q_0, b)	$(\mathrm{P}\ q_2)$
2.	(q_2, c)	$(\mathrm{P}\ q_2)$
3.	(q_2, c)	$(\mathrm{P}\ q_2)$
4.	(q_2, a)	$(\mathrm{P}\ q_3)$
5.	(q_3, b)	$(\mathrm{P}\ q_3)$
6.	(q_3, a)	$(\mathrm{N}\ q_4^*)$

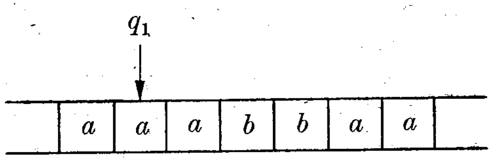
Gdyby ciąg ten zawierał jeszcze jedną dowolną literę na końcu, byłby on już niepoprawny, automat bowiem zatrzymał się na przedostatnim symbolu i ostatniego symbolu już by nie analizował.

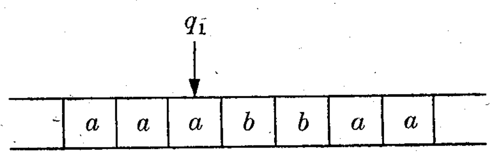
A więc, jeżeli automat przejdzie do zdania końcowego przy czytaniu symbolu różnego od symbolu końcowego, to ciąg taki jest niepoprawny.

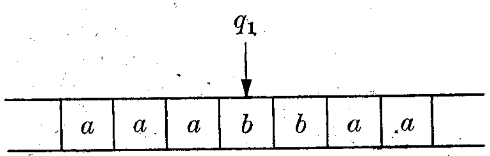
Zwróćmy jednak uwagę, że przy takim określeniu poprawności zdania, decyzję o poprawności podejmuje nie analizujący automat, a człowiek. Jeżeli bowiem automat przeszedł po odczytaniu jakiegoś symbolu do stanu końcowego, to automat „nie wie", że jest to symbol ostatni badanego ciągu; fakt ten widzi dopiero obserwator zewnętrzny stwierdzający, że maszyna się zatrzymała i że nie wszystkie symbole zostały przeanalizowane.

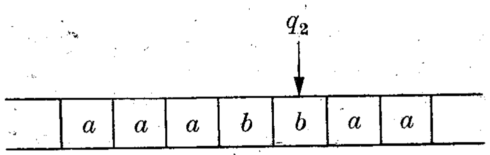
Wyrażenie *a a a b b a a* jest natomiast zdaniem niepoprawnym. Kolejne ruchy maszyny przy analizowaniu tego ciągu są następujące:

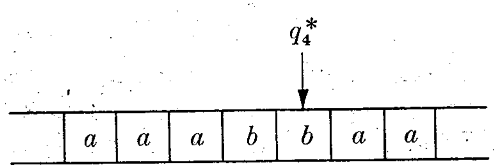
1.


2.


3.


4.


5.


6.


Maszyna zatrzymała się na symbolu trzecim od końca, a więc wyrażenie jest niepoprawne.

Sytuacje i instrukcje dla tej analizy są następujące:

Ruch	Sytuacja	Instrukcja
1.	(q_0, a)	(P q_1)
2.	(q_1, a)	(P q_1)
3.	(q_1, a)	(P q_1)
4.	(q_1, b)	(P q_2)
5.	(q_2, b)	(N q_4^*)

W ten sposób wszystkie ciągi składające się z liter *a*, *b*, *c*, *d* mają być analizowane przy pomocy maszyny $\mathfrak{M}_2$ i zależnie od wyniku tej analizy uznane za poprawne bądź nie.

Przedstawioną koncepcję gramatyki można również interpretować nieco inaczej. Mianowicie, możemy przyjąć, że nie mamy żadnego automatu analizującego poprawność zdań, a gramatyka języka jest przedstawiona w postaci wykresu, jak np. na rysunku 1. Inaczej mówiąc, gramatyka nie jest tu zbiorem reguł zapisanych w jakiejś symbolice, a rysunkiem. Badany ciąg symboli jest zdaniem poprawnym, jeżeli wychodząc z punktu początkowego q_0 można dojść do punktu końcowego q_4^* w ten sposób, że każdemu kolejnemu symbolowi badanego ciągu odpowiada przejście odcinka grafu oznaczonego tym samym symbolem (wielokrotne przechodzenie tego samego odcinka jest dozwolone). Np. zdanie

$$b\ c\ c\ c\ a\ b\ b\ b\ a$$

jest poprawne, gdyż w grafie na rysunku 1 istnieje następująca droga z q_0 do q_4^*:

$$q_0\ b\ q_2\ c\ q_2\ c\ q_2\ c\ q_2\ a\ q_3\ b\ q_3\ b\ q_3\ b\ q_3\ a\ q_4^*$$

Ciąg ten opisuje sposób przejścia od punktu q_0 do punktu q_4^*. Przez odcinek *c* przechodzimy tu trzykrotnie raz za razem, podobnie jak i przez odcinek *b*.

Napiszemy jeszcze kilka przykładów sprawdzania poprawności zdań według tej metody.

$$\begin{cases} c\ c\ c \\ q_0\ c\ q_3\ c\ q_1\ c\ q_4^*. \end{cases}$$

$$\begin{cases} a\ a\ b\ c\ a\ b\ a \\ q_0\ a\ q_1\ a\ q_1\ b\ q_2\ c\ q_2\ a\ q_3\ b\ q_3\ a\ q_4^*. \end{cases}$$

$$\begin{cases} b\ a\ c\ b\ b \\ q_0\ b\ q_2\ a\ q_3\ c\ q_1\ b\ q_2\ b\ q_4^*. \end{cases}$$

$$\begin{cases} c\ c\ a\ b\ c\ a\ a \\ q_0\ c\ q_3\ c\ q_1\ a\ q_1\ b\ q_2\ c\ q_2\ a\ q_3\ a\ q_4^*. \end{cases}$$

Podane przykłady dokładnie wyjaśniają istotę opisanej metody. Badanie poprawności zdań można przeprowadzić jeszcze nieco inaczej, wychodząc z gramatyki przedstawionej na rysunku 1. Z grafu tego wynika, że zdania poprawne nie mogą zaczynać się następująco:

$$a\ b\ b\ \ldots$$
$$b\ a\ a\ \ldots$$
$$c\ c\ c\ \ldots$$
$$\underbrace{a\ \ldots\ a}_{n \text{ razy}}\ c\ \ldots$$
$$b\ \underbrace{c\ \ldots\ c}_{n \text{ razy}}\ b\ \ldots$$
$$c\ \underbrace{b\ \ldots\ b}_{n \text{ razy}}\ a\ \ldots$$

a więc niepoprawne są zdania:

$$\underline{a\ b\ b}\ a\ c\ c$$
$$\underline{b\ a\ a}\ a\ b\ a\ c$$
$$\underline{a\ a\ c}\ a\ b$$
$$\underline{b\ c\ c\ c\ b}\ a\ b$$
$$\underline{c\ b\ b\ b\ a}\ c\ b$$

W zdaniach podkreślono niedozwolone kombinacje początkowe. Oczywiście podane kombinacje nie wyczerpują wszystkich niedozwo-

lonych zestawień symboli. Dla niektórych gramatyk można podać skończony zbiór takich niedozwolonych zestawień symboli i wtedy badanie poprawności zdania może polegać na szukaniu w zdaniu niedozwolonych kombinacji symboli. Jeżeli jedna taka kombinacja w zdaniu występuje — zdanie jest niepoprawne, jeżeli natomiast w zdaniu nie ma żadnej niedozwolonej kombinacji, zdanie jest poprawne.

§ 19. Synteza zdań za pomocą automatów skończonych

Automat produkujący zdania poprawne działa podobnie do automatu analizującego zdania, z tą różnicą, że zamiast odczytywania symboli z taśmy — drukuje on symbole na taśmie według pewnego schematu. Ruch takiego automatu składa się więc z następujących elementów:

1) wydrukowanie symbolu w obserwowanej komórce,
2) przejście do nowego stanu,
3) zmiana obserwowanej komórki na komórkę sąsiednią z prawej strony.

Tablicę charakterystyczną takiego automatu, podaną poniżej

	q_0	q_1	$\cdots$	q_n
s_0	$q_{0,0}$	$q_{0,1}$		$q_{0,n}$
s_1	$q_{1,0}$	$q_{1,1}$		$q_{1,n}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
s_m	$q_{m,0}$	$q_{m,1}$		$q_{m,n}$

należy więc teraz interpretować w ten sposób, że jeżeli automat znajduje się w stanie q_i, to przechodzi on do jednego ze stanów $q_{0,i}$, $q_{1,i}$, …, $q_{m,i}$ — np. $q_{j,i}$ — i drukuje w obserwowanej komórce symbol s_j.

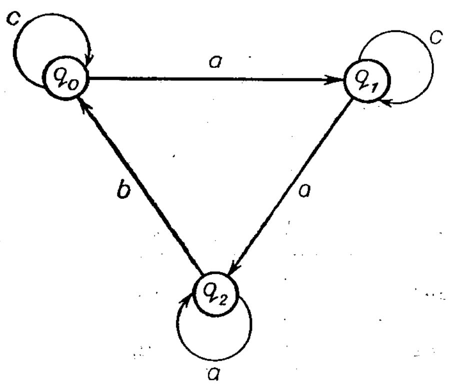
W tablicy dla prostoty nie podano symbolu P oznaczającego przejście do obserwowania nowej komórki.

Automat drukujący zdania może więc mieć np. następującą tablicę charakterystyczną

	q_0	q_1	q_2
a	q_1	q_2	q_2
b	—	—	q_0
c	q_1	q_1	—

q_0 jest stanem początkowym.

Kreski w tablicy oznaczają, że w niektórych stanach pewne symbole nie mogą być drukowane. Na przykład jeżeli automat jest w stanie q_1, to może on wydrukować symbol a przechodząc do stanu q_2, bądź też może on wydrukować symbol c przechodząc do stanu q_1, natomiast symbol b nie może być wydrukowany przy stanie q_1. Działanie takiego automatu nie jest więc ściśle zdeterminowane.



Rys. 2

Graf tego automatu przedstawiony jest na rysunku 2. Znaczenie tego grafu jest podobne do omawianego w poprzednim paragrafie. Różnica polega jedynie na tym, że symbole przyporządkowane strzałkom są teraz symbolami drukowanymi przez automat, a nie symbolami odczytywanymi, jak to miało miejsce przy analizie zdań.

Wszystkie ciągi symboli produkowane przez automat skończony $\mathfrak{A}$ nazwiemy zdaniami poprawnymi według automatu $\mathfrak{A}$.

Oznaczmy przez s' stan początkowy automatu $\mathfrak{A}$.

Ciąg

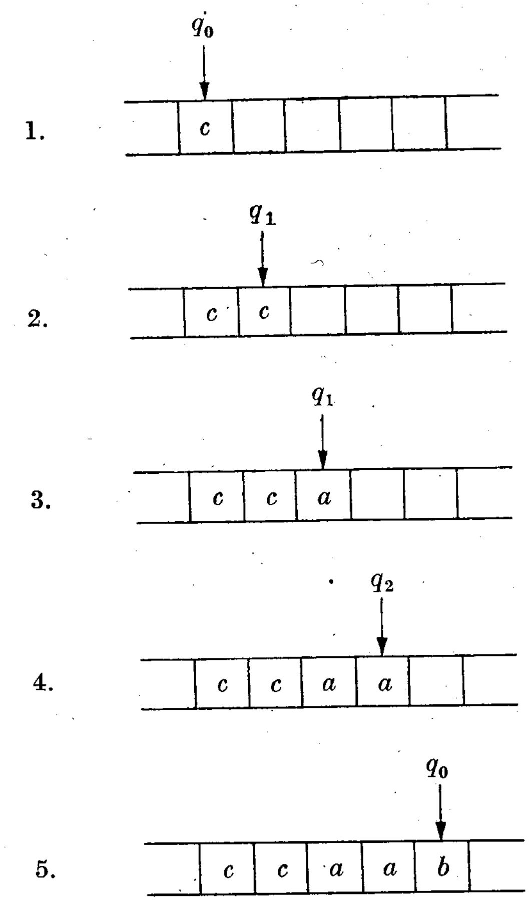
$$X = x_0,\ x_1,\ \ldots,\ x_n$$

jest wyprodukowany przez automat $\mathfrak{A}$, jeżeli istnieje taki ciąg stanów $s_0, s_1, \ldots, s_n$ automatu $\mathfrak{A}$, że

1. $s_0 = s_n = s'$
2. $s_i \neq s'$, jeżeli $0 < i < n$
3. $s_{i+1} = T\,(x_i s_i)$

Podane określenie mówi po prostu, że jeżeli automat, startując ze stanu początkowego q_0, będzie przechodził do kolejnych stanów według tablicy charakterystycznej (lub grafu automatu), drukując jednocześnie na taśmie odpowiednie symbole i powróci do stanu początkowego q_0, to wydrukowany w ten sposób ciąg symboli jest zdaniem poprawnym, według automatu $\mathfrak{A}$.

Dla podanego przykładu automatu drukowanie zdania może więc przebiegać następująco:



W każdym ruchu automat przechodzi do następnego stanu, drukuje według tablicy charakterystycznej odpowiedni symbol i przechodzi do obserwowania następnej komórki z prawej strony. Ponieważ w każdym stanie możliwych jest kilka przejść do następnego stanu, stan następny może być wybierany dowolnie, na przykład przypadkowo. W rozpatrywanym przykładzie automat może przejść ze stanu q_1 do stanu q_2 lub wrócić ponownie do q_1. Pierwszym razem przyjęliśmy dowolnie, że przeszedł on ze stanu q_1 do stanu q_1, za drugim zaś razem przyjęliśmy przejście ze stanu q_1 do stanu q_2. Podobnie postąpiliśmy w pozostałych krokach.

Ponieważ stan pierwszy i ostatni są stanami początkowymi q_0, więc ciąg $c\ c\ a\ a\ b$ jest zdaniem poprawnym, wyprodukowanym przez automat $\mathfrak{A}$.

Również w przypadku produkcji zdań możemy pominąć automat i powiedzieć, że graf przejścia od stanu do stanu przedstawia gramatykę języka. Wychodząc z punktu początkowego (stanu początkowego) i idąc dowolnie wzdłuż gałęzi grafu tak, aby po pewnej ilości kroków powrócić znów do punktu wyjściowego — produkujemy zdanie poprawne. Zdanie to jest ciągiem symboli przyporządkowanych kolejno przechodzonym gałęziom.

Np. w myśl tej zasady możemy wyprodukować następujące zdanie:

$$a\ c\ c\ a\ a\ a\ b\ a$$
$$q_0\ a\ q_1\ c\ q_1\ c\ q_1\ a\ q_2\ a\ q_2\ a\ q_2\ b\ q_0\ a$$

Również w tym przypadku możemy próbować szukać pewnych prawidłowości w strukturze zdań zbudowanych poprawnie i prawidłowości te stosować jako kryteria poprawności zdań.

*

Podaliśmy przykład trzech rodzajów gramatyk. Jest ich znacznie więcej. Lingwistyka matematyczna zajmuje się badaniem własności gramatyk oraz ich wzajemnych związków między sobą. W zasadzie problem, w jakim stopniu te matematyczne modele są odpowiednie do opisu rzeczywistych języków, do lingwistyki matematycznej nie należy. Sprawa przedstawia się tutaj podobnie jak i w innych dziedzinach zastosowań matematyki, np. fizyce, gdzie przyjmuje się pewien uproszczony „idealny" obraz świata i bada się metodami matematycznymi jego właściwości. Uproszczenia takie są niezbędne, badanie bowiem

jakiegoś zjawiska w całej jego złożoności może być bardzo trudne, jeżeli nie niemożliwe. Jednakże mimo przyjętych uproszczeń otrzymane rezultaty mają wartości praktyczne, dają bowiem możliwość uchwycenia zasadniczych własności interesującego nas zjawiska. Podobnie rzecz się przedstawia w lingwistyce matematycznej. Badanie języków naturalnych bez żadnych uproszczeń, ze wszystkimi szczegółami, byłoby niemożliwe i być może niepotrzebne. Dlatego wprowadzane jest pojęcie języka, które ma się tak do języka naturalnego, jak np. w fizyce pojęcie gazu „idealnego" do gazu rzeczywistego — i bada się tak rozumiany język. Oczywiście im model bliższy jest rzeczywistości, tym wyniki jego badania mogą znaleźć szersze praktyczne zastosowanie.

Rozpatrywane przykłady języków były bardzo proste. W lingwistyce matematycznej badane języki, jakkolwiek są uproszczone w stosunku do języków rzeczywistych, to jednak są znacznie bardziej skomplikowane od tych, które przedstawiono tutaj. Chodziło nam bowiem o pokazanie zasadniczych idei. A do tego celu podane przykłady są wystarczające.

Badanie własności języka tylko z punktu widzenia struktury zdań poprawnych jest niewystarczające. Celem bowiem języka jest przekazywanie wiadomości, bardzo więc ważne jest zagadnienie, w jakim stopniu struktura języka zależy od tego, co on opisuje, a zatem zależność między strukturą a semantyką. Jednym z elementów struktury języka jest składnia. Badania nad składnią są jednak obecnie znacznie bardziej zaawansowane niż badania dotyczące semantyki.

LITERATURA

S. C. Kleene: *Introduction to metamathematics*. Amsterdam 1952.

A. Trachtenbrot: *Algorytmy i automatyczne rozwiązywanie zadań*. Warszawa 1961.

N. Chomsky, G. A. Miller: *Finite state languages*. Information and Control 1958, 1, 91—112.

Y. Bar-Hillel, E. Shamir: *Finite state languages: Formal representations and adequacy problems*. The Bulletin of the Research Council of Israel 8F, nr 3 Feb. 1960.

M. Θ. Rabin, D. Scott: *Finite automata and their decision problems*, IBM Journal of Research and Development, 3, 1959, str. 115—125.

Rozdział V

OPISYWANIE CZYNNOŚCI

§ 20. Proces prosty

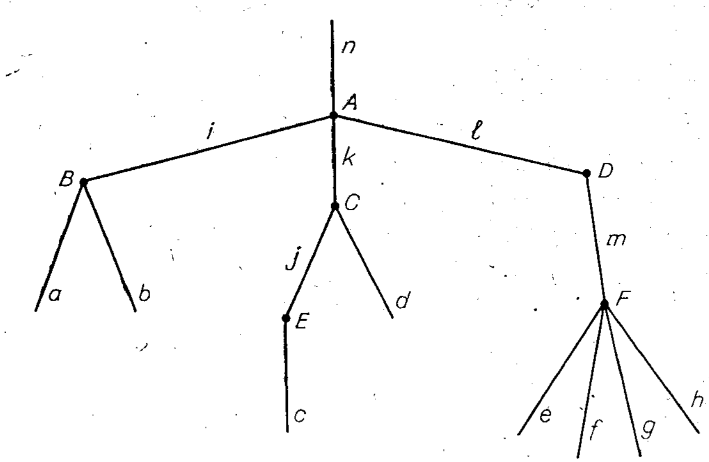
Jednym z ciekawszych „zastosowań" języka jest opisywanie czynności. Łatwo sprawdzić, próbując szczegółowo opisać przebieg wykonania jakiejkolwiek czynności, że sprawa nie jest zbyt prosta. Języki naturalne są bowiem dość niewygodne do opisywania czynności. Dla uniknięcia nieporozumień wyjaśnijmy od razu, że przez czynność rozumiemy tu taki rodzaj działania, w którego wyniku z jednych przedmiotów powstają jakieś nowe przedmioty.

A więc mamy tu na myśli opis jakiegoś procesu wytwarzania czy produkcji. Wydaje się wysoce prawdopodobne, że człowiek w historycznym rozwoju potrzebował najpierw języka nadającego się do wyrażania stosunków, a konieczność opisywania przebiegu czynności pojawiła się znacznie później i nie wpłynęła ona już istotnie na kształtowanie się struktury języka.

W rozdziale tym spróbujemy się zastanowić, w jaki sposób można określić język przydatny do opisywania niektórych czynności. Punktem wyjściowym naszych rozważań będzie pojęcie **procesu.**

Z pojęciem procesu spotykamy się niemal na każdym kroku. Mimo bardzo częstego używania tego pojęcia nie jest ono zbyt jasno sprecyzowane. Ogólnie biorąc, z procesem są związane jakieś zmiany w czasie. Tutaj przez proces będziemy rozumieli wytwarzanie z zadanych elementów, za pomocą ustalonych operacji, określonego przedmiotu. Nie będziemy podawali ścisłej definicji tego pojęcia, a wyjaśnimy je podając jego interpretację geometryczną. Jeżeli mianowicie przedmioty, na których są wykonywane operacje i które są wynikiem operacji, będziemy przedstawiać w postaci odcinków, a operacje w postaci

punktów, to każdy proces wytwarzania jakiegoś przedmiotu i jego części składowych można przedstawić w postaci „drzewa", jak to pokazano na rysunku 3. Operacje wykonywane w procesie oznaczono dużymi literami, przedmioty natomiast — literami małymi. Przedmioty, które nie są wynikiem zadanej operacji w procesie, nazwiemy **elementami.** Na rysunku 3 elementami są przedmioty a, b, c, d, e, f,



Rys. 3

g, h. Przedmioty, które są wynikiem zastosowania jednej z operacji procesu do przedmiotów już utworzonych w procesie (również elementów), nazwiemy **podzespołami.** W przykładzie na rysunku 3 podzespołami są przedmioty oznaczone literami i, j, k, l, m, n. Taki podzespół, który nie jest już poddawany żadnej operacji, nazwiemy **wytworem końcowym** procesu. W przykładzie wytworem końcowym jest przedmiot oznaczony literą n.

Przedmiot n powstał w wyniku zastosowania operacji A do przedmiotów i, k, l. Przedmiot i jest z kolei wynikiem zastosowania operacji B do przedmiotów a i b (elementów) itd.

Tak rozumiany proces będziemy nazywali **procesem prostym.**

Proces prosty charakteryzuje się tym, że w jego wyniku powstaje **jeden** wytwór końcowy oraz że w wyniku każdej operacji powstaje **jeden** podzespół.

Przyjmiemy ponadto, że w każdej chwili wykonywana jest tylko **jedna** operacja w procesie. Proces taki nazwiemy **sekwencyjnym.**[1]

Jeszcze jedno założenie odnośnie procesu: ciąg operacji w procesie prostym jest wykonywany jednokrotnie. To znaczy po otrzymaniu jednego wytworu końcowego proces zostaje zakończony.[2]

§ 21. Opis procesu prostego

Opis procesu prostego będziemy nazywali **programem** tego procesu. Załóżmy, że chcemy opisać przebieg procesu w ten sposób, aby inna osoba, postępując według tego opisu, mogła opisany proces zrealizować. Jakie informacje o procesie musimy zawrzeć w programie? Oczywiście konieczne jest podanie:

1) jakie należy wykonać operacje,
2) na jakich przedmiotach powinna być wykonana każda operacja,
3) w jakim porządku należy wykonać operacje.

Te właśnie informacje wystarczają, aby proces był opisany jednoznacznie.

Gdybyśmy chcieli zastosować do opisu język naturalny, przebieg procesu wyrazilibyśmy w postaci ciągu zdań typu:

1) wykonaj operację A na przedmiotach $a, b, \ldots, c$;
2) wykonaj operację B na przedmiotach $a', b', \ldots, c'$,

. . . .

itd., aż do wyczerpania w opisie wszystkich operacji potrzebnych do wykonania wytworu końcowego. Na przykład proces przedstawiony na rysunku 3 opisalibyśmy następująco:

1) wykonaj operację F na przedmiotach e, f, g, h;
2) następnie wykonaj operację E na przedmiocie c;

. . . .

5) wykonaj operację A na przedmiotach i, k, l.

---

[1] Można sobie również wyobrazić, że jednocześnie są wykonywane dwie lub trzy operacje w procesie. Takim przypadkiem procesu nie będziemy się jednak interesować.

[2] Można sobie wyobrazić proces w ten sposób, że po uzyskaniu jednego wytworu końcowego wszystkie operacje są ponownie wykonywane od początku; otrzymujemy w ten sposób nowy, identyczny z poprzednim wytwór końcowy itd. Byłby to więc rodzaj produkcji seryjnej. My zaś jesteśmy zainteresowani jednokrotną realizacją wszystkich operacji i otrzymaniem jednego wytworu końcowego.

Wynik ostatniej operacji jest wytworem końcowym.

W opisie tym dla prostoty nie podawaliśmy, co otrzymujemy w wyniku każdej operacji. Jednakże informacja ta powinna być w zasadzie zawarta w naszym opisie.

Podany sposób opisu jest rozwlekły i nieprzejrzysty i zupełnie z niego nie widać przebiegu powstawania wytworu końcowego. Można by go nieco uprościć, stosując zamiast zapisu słownego programu — zapis symboliczny, np. podając kolejno wykonywane operacje wraz z nazwami przedmiotów, na których mają one być wykonywane — oraz nazwę przedmiotu otrzymanego w wyniku tej operacji. Wtedy proces przedstawiony na rysunku 3 można by przedstawić w postaci jak niżej:

1. $e\ f\ g\ h\ F\ m$
2. $c\ E\ j$
3. $j\ d\ C\ k$
4. $m\ D\ l$
5. $a\ b\ B\ i$
6. $i\ k\ l\ A\ n$

Po lewej stronie każdego symbolu operacji podano nazwy przedmiotów, na których operacja jest wykonywana, a po stronie prawej symbolu operacji — nazwę przedmiotu otrzymanego w wyniku tej operacji. Np. operacja C jest wykonana na przedmiotach j, d i w wyniku tej operacji otrzymujemy przedmiot k.

Ten opis jest już nieco wygodniejszy i mógłby być on traktowany jako zdanie w pewnym języku, jednakże zgodnie z naszymi intencjami odnośnie do języków chcielibyśmy, aby struktura programu opisującego proces prosty była w pewien sposób związana ze strukturą opisywanego procesu. W podanym opisie trudno byłoby doszukać się takich zależności.

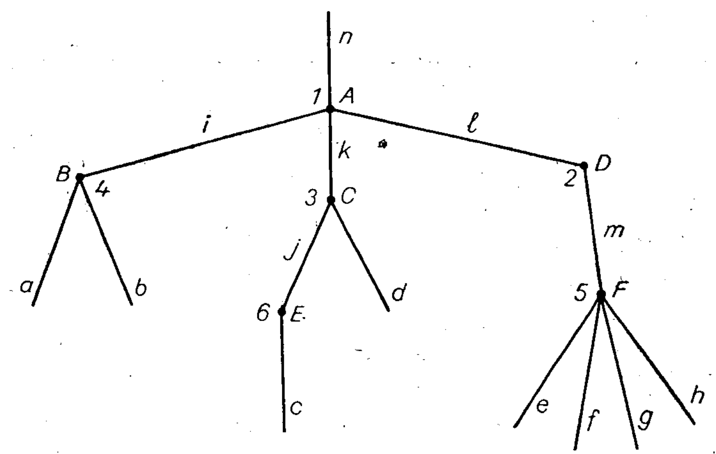
Okazało się, że dla określenia języka procesów prostych uzasadnioną rolę gra kolejność wykonywania operacji w procesie.

§ 22. Kolejność operacji

W procesie prostym operacje możemy wykonać w różnej kolejności. Na przykład dla procesu przedstawionego na rysunku 3 w poprzednim paragrafie podaliśmy jedną kolejność wykonywania operacji, ale również operacje te mogą być wykonywane np. w kolejności jak niżej:

1. $a \; b \; B \; i$
2. $c \; E \; j$
3. $e \; f \; g \; h \; F \; m$
4. $m \; D \; l$
5. $j \; d \; C \; k$
6. $i \; k \; l \; A \; n$

Możemy oczywiście przyjąć jeszcze inne kolejności niż te, które tu przedstawiono. Mogłoby się wydawać, że z punktu widzenia opisu, kolejność tu nie odgrywa istotnej roli, jednakże tak nie jest. Okazało się, że cztery z możliwych wszystkich kolejności wykonywania operacji mają tutaj szczególne wyróżnione znaczenia. Dwie z nich w dalszym ciągu przedstawimy; dwie dalsze są podobne, nie będziemy ich więc omawiać. Kolejności te nazwiemy: kolejnością **poprzeczną** oraz kolejnością **wzdłużną** i będziemy je oznaczali literami P oraz W. Ustalenie kolejności wykonywania działań sprowadza się do odpowiedniego ponumerowania rozgałęzień drzewa przedstawiającego proces prosty. Zasadę numerowania działań dla obu kolejności wyjaśnimy na przykładzie dyskutowanego procesu.



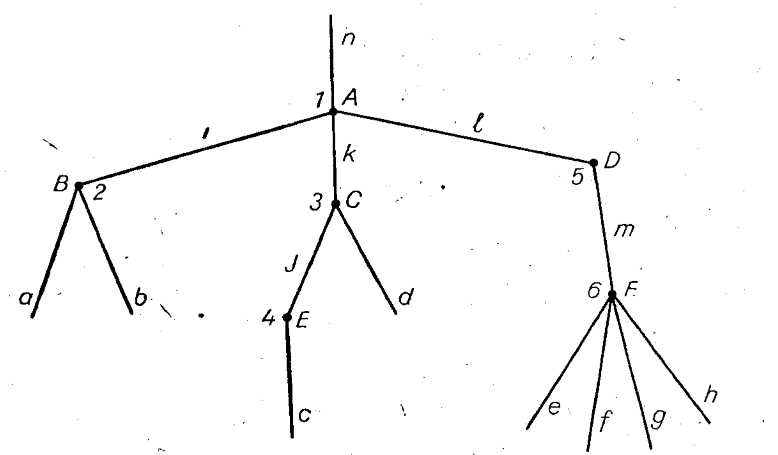
Rys. 4

Kolejność poprzeczna (P). Zasadę numeracji poprzecznej wyjaśnia rysunek nr 4, operacja końcowa ma numer 1. Pozostałe operacje numerujemy według następującej zasady: drzewo dzielimy na „piętra", na każdym piętrze operacje numerujemy kolejnymi liczbami od strony prawej ku lewej. Na każdym piętrze najmniejszy numer operacji jest o jeden większy od największego numeru operacji na piętrze wyższym. A więc operacje numerujemy od góry w dół i z prawa na lewo.

Jeżeli więc w procesie przedstawionym na rysunku 3 przyjmiemy poprzeczną kolejność wykonywania operacji, otrzymamy program:

6. $c \; E \; j$
5. $e \; f \; g \; h \; F \; m$
4. $a \; b \; B \; i$
3. $j \; d \; C \; k$
2. $m \; D \; l$
1. $i \; k \; l \; A \; n$

Najpierw jest wykonywana operacja o największym numerze, dalej zaś operacje o numerach malejących 6, 5 itd. aż do 1. A więc operacje są wykonywane na drzewie z dołu do góry i z lewa na prawo.



Rys. 5

Kolejność wzdłużna (W). Przykład wzdłużnej numeracji operacji pokazano na rysunku nr 5.

Nietrudno rozszyfrować zasadę tej numeracji. Jeżeli przyjmiemy porządek W, proces przedstawiony na rysunku 3 będzie przebiegał następująco:

6. *e f g h F m*
5. *m D l*
4. *c E j*
3. *j d C k*
2. *a b B i*
1. *i k l A n*

Zakładamy, że w procesach prostych, które będziemy rozważali w dalszym ciągu, operacje mogą być wykonywane tylko w kolejności P lub W.

§ 23. Język procesów prostych

W paragrafie tym podamy pewien język bardzo wygodny do opisywania przebiegu procesu prostego. Jeżeli mianowicie ograniczymy się do wspomnianych poprzednio dwu kolejności wykonywania operacji, to podanie w programie nazwy przedmiotu będącego wynikiem operacji jest, jak się przekonamy, zbędne. Programy procesu dla dwu kolejności będą więc miały teraz postać:

Kolejność P	Kolejność W
c E	*e f g h F*
e f g h F	*m D*
a b B	*c E*
j d C	*j d C*
m D	*a b B*
i k l A n	*i k l A n*

W programach tych, po każdym symbolu operacji z wyjątkiem operacji ostatniej, nazwy wyniku operacji już nie pisaliśmy. Programy te wygodniej jest pisać w jednej linii w sposób następujący:

Kolejność P

c E e f g h F a b B j d C m D i k l A n

Kolejność W

e f g h F m D c E j d C a b B i k l A n

Analizując program od strony lewej do prawej, wiemy, jakie należy wykonać operacje i na jakich obiektach, brak nam natomiast informacji o tym, co otrzymujemy w wyniku każdej operacji.

Jeżeli jednak potrafimy odróżnić nazwy elementów od nazw podzespołów, można podać proste reguły przyporządkujące każdemu symbolowi operacji nazwę przedmiotu otrzymanego w jej wyniku. Przyjmijmy więc, że podzespoły oznaczamy przez napisanie kreski nad odpowiednią literą, jak to pokazano niżej:

Kolejność P

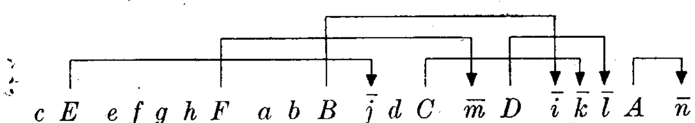
$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\ \bar{n}$

Kolejność W

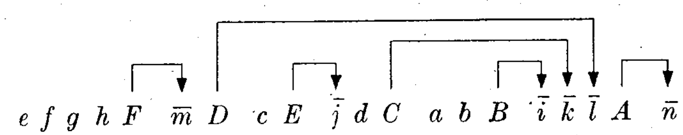
$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\ \bar{n}$

Każdemu symbolowi operacji w programie odpowiada więc nazwa podzespołu. Mówiąc inaczej, każdej dużej literze w programie należy przyporządkować małą literę z kreską u góry w ten sposób, aby litera ta była nazwą otrzymanego — w wyniku przyporządkowanej operacji — przedmiotu. Przyporządkowanie to możemy otrzymać na podstawie analizy drzewa, zaznaczając wynik każdego działania strzałką w sposób następujący:

Kolejność P



Kolejność W



A więc np. wynikiem operacji B jest $\bar{i}$, wynikiem operacji D jest $\bar{l}$ itp.

Analizując to rozmieszczenie strzałek w programie, łatwo zauważyć, że nie są one rozmieszczone przypadkowo, a w pewien prawidłowy, łatwy do uchwycenia sposób.

Dla kolejności P, każdy symbol operacji łączymy strzałką z kolejnym symbolem podzespołu, poczynając od strony lewej programu, jak to pokazano niżej:

Kolejność P

$$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$c\ E\quad e\ f\ g\ h\ F\quad a\ b\ B\quad \bar{j}\ d\ C\quad \bar{m}\ D\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

Dla kolejności W reguła stawiania strzałek jest zbliżona: każdy symbol operacji łączymy z najbliższym z prawej strony wolnym symbolem podzespołu, zaczynając stawianie strzałek od prawej strony programu, jak następuje:

Kolejność W

$$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

$$e\ f\ g\ h\ F\quad \bar{m}\ D\quad c\ E\quad \bar{j}\ d\ C\quad a\ b\ B\quad \bar{i}\ \bar{k}\ \bar{l}\ A\quad \bar{n}$$

Reguły te obowiązują nie tylko dla podanego przykładu, lecz i dla programów dowolnych procesów prostych.

Jeżeli mamy zadany program procesu i nie wiemy, czy jest on napisany w kolejności P czy W, program taki jest oczywiście niejednoznaczny. Nie wiemy bowiem z tego programu, jaki przedmiot otrzymamy w wyniku każdej operacji. Np. program

$$a\ b\ A\quad c\ B\quad \bar{d}\ \bar{e}\ F\quad \bar{g}\ h\ i\ C\quad \bar{f}$$

możemy odczytać jako

$$a\ b\ A\quad c\ B\quad \bar{d}\ \bar{e}\ F\quad \bar{g}\ h\ i\ C\quad \bar{f}$$

jeżeli przyjmiemy, że operacje są wykonywane w kolejności P, bądź jako

$$a\ b\ A\quad c\ B\quad \bar{d}\ \bar{e}\ F\quad \bar{g}\ h\ i\ C\quad \bar{f}$$

zakładając, że przyjęto kolejność W wykonywania operacji.

Programy procesów prostych tworzą język. Zdaniami poprawnymi w tym języku są właśnie programy opisywanych procesów. Można badać, jakie reguły rządzą poprawnością zdań, tzn. poprawnością programów w tym języku, inaczej mówiąc — jaką ten język posiada gramatykę.

Rozważany język jest doskonałą ilustracją faktu, że badanie tylko struktury zdań języka, jakkolwiek bardzo ważne, nie jest wystarczające. Z reguł gramatycznych języka nie wynika bowiem **znaczenie**, a to dla nas jest ostatecznie najważniejsze.

Poprawność programów możemy badać tutaj zarówno na podstawie ich **struktury**, jak również na podstawie **znaczenia**. Możemy więc powiedzieć, że program jest zbudowany poprawnie, jeżeli jest on zgodny z regułami gramatycznymi lub jeżeli jest on opisem procesu prostego.

W pierwszym przypadku mamy do czynienia z poprawnością **syntaktyczną** (składniową), w drugim zaś — z poprawnością **semantyczną** (znaczeniową).

Dla innych bardziej skomplikowanych języków badania dotyczące znaczenia języka nie są tak proste, jak w przytoczonym przykładzie. Jest to przyczyna, dla której stosowanie metod matematycznych do semantyki napotyka znaczne trudności i badania w tym kierunku są znacznie mniej zaawansowane od badań strukturalnych [1]).

---

[1]) Z innym opisem procesów prostych Czytelnik może się zapoznać z książek: Z. Pawlak: *Maszyny i język*. Omega, PWN 1964.
Z. Pawlak: *Organizacja maszyn bezadresowych*. PWN (w druku).

Rozdział VI

GRAMATYKI GENETYCZNE

§ 24. Alfabet życia

W dotychczasowych rozdziałach zajmowaliśmy się językami naturalnymi lub językami sztucznie stworzonymi do określonych celów, na przykład językiem do opisywania procesów. W ostatnich latach w zakres zainteresowań językoznawców wkraczają: biologia, genetyka i biochemia. Sprawa ta wydaje się bardzo ciekawa i ważna. Chodzi tu mianowicie o badanie podstawowych procesów dotyczących rozwoju żywych organizmów.

Wiadomo, że podstawową najważniejszą cegiełką, z których zbudowany jest każdy żywy organizm na ziemi: roślina, bakteria, czy człowiek — jest białko. Białka są olbrzymimi cząsteczkami chemicznymi, składającymi się z setek tysięcy atomów. Liczba rozmaitych białek jest również olbrzymia i bliżej jeszcze nie znana.

Białko jest istotnym składnikiem żywej materii. Dlatego zbadanie jego struktury i procesu powstawania ma zasadnicze znaczenie dla rozważań zagadki życia.

Wiadomo, że każde białko składa się z 20 podstawowych związków zwanych **aminokwasami**. Aminokwasy te łączą się w gigantycznej długości łańcuchy. W zależności od rodzaju występujących w białku aminokwasów, ich rozmieszczenia, kolejności, sposobu powtarzania się itp. otrzymujemy białko o najrozmaitszych własnościach. Najdrobniejsza nawet zmiana struktury białka, np. zastąpienie w nich tylko jednego aminokwasu innym, powoduje całkowitą zmianę jego własności.

Każdy gatunek wytwarza prawdopodobnie swoiste dla siebie rodzaje białek i mechanizm ten jest przekazywany bez żadnych zmian z pokolenia na pokolenie.

Łączenie aminokwasów w białko nie odbywa się jednak samoistnie, jak np. łączenie wodoru i tlenu, a wymaga obecności innych specyficznych związków biochemicznych. Związkami tymi są chromosomy. Stanowią one przepis — program, według którego przebiega proces syntezy białek.

Chromosomy są związkami o strukturze długich nitek, w skład których wchodzą cztery podstawowe związki chemiczne zwane **bazami**. Związki chemiczne wchodzące w skład chromosomów można uważać za litery alfabetu, a ciągi tych liter — za zdania w pewnym języku. A więc chromosomy to ciągi pewnych zdań, tekst opisujący produkcję białka w organizmie. Niestety język, którym ten proces produkcyjny natury jest opisany, nie został do tej pory zbadany dostatecznie. Odczytanie jego jest więc niestety niemożliwe. Wiemy tylko, że alfabet tego języka składa się z czterech liter i wiemy także, że jest on opisem składania 20 aminokwasów w białku. Sytuacja jest podobna do tej, w jakiej znajdują się czasem archeologowie, jednakże rozwiązanie tej zagadki wydaje się o wiele ważniejsze.

W wielu laboratoriach świata prowadzone są intensywne badania, mające na celu ustalenie związków między strukturą białka a opisującą ją strukturą chromosomów. W badaniach tych biorą udział uczeni najrozmaitszych specjalności: biochemicy, genetycy, matematycy i inni. Mimo to do tej pory nie osiągnięto zadowalających rezultatów. Sprawa wydaje się bardzo trudna i chyba nie należy liczyć na bardzo szybkie jej rozwiązanie. Tym niemniej istnieje szereg koncepcji takiego języka. Jedną z nich chcielibyśmy tutaj przedstawić.

§ 25. Mozaiki i białka

Nie wiemy dokładnie, jak przebiega i na czym polega proces syntezy białka z aminokwasów. Jest to zjawisko bardzo skomplikowane i zbadanie go od razu w całej złożoności jest bardzo trudne. Dlatego też zajmiemy się zagadnieniem nieco prostszym, które mamy nadzieję, że jest przybliżonym obrazem interesującego nas procesu.

Można przypuszczać, że przebieg procesu syntezy białka z aminokwasów przypomina nieco budowanie domu z cegieł, układanie mozaiki z kafelków, czy też zabawę w domino, w której z pewnej liczby elementów, według określonych reguł składane są bardziej skomplikowane twory. Wiemy, jakie mamy kafelki do dyspozycji, nie znamy natomiast zasady ich układania. Nie każde bowiem zestawienie ami-

nokwasów w białku jest dozwolone. Wiemy ponadto, że opis tego procesu składania jest wyrażany za pomocą czteroliterowego alfabetu.

Mamy więc do rozwiązania zadanie zbliżone do zadania poruszanego w poprzednim rozdziale, w którym chodziło nam o znalezienie języka, opisującego pewne procesy.

Spróbujemy najpierw zastanowić się, jakiego rodzaju proces może być tu brany pod uwagę. W myśl uwag poprzedniego rozdziału każdy proces składania, w którym z pewnych elementów tworzone są nowe obiekty, będzie procesem prostym. Czym będą jednak obiekty tego procesu? Białko mimo liniowego charakteru jest niewątpliwie konstrukcją przestrzenną. Aminokwasy można sobie wyobrazić jako pewne przestrzenne figury geometryczne, a białko — jako konfigurację tych podstawowych brył, złożoną według pewnych prawideł. Różnym aminokwasom mogą oczywiście odpowiadać różne figury, co jeszcze bardziej komplikuje sprawę.

Dla uproszczenia przyjmijmy, że każdy aminokwas można sobie wyobrazić w postaci płaskiej figury-trójkąta równobocznego. Białku natomiast będzie odpowiadać mozaika otrzymana ze sklejania trójkątów krawędziami. Zamiast więc badać proces syntezy białka z aminokwasów, będziemy rozpatrywać składanie mozaik z trójkątów oraz spróbujemy znaleźć język pozwalający na opisywanie procesu konstruowania mozaik, zakładając, że w zdaniach tego języka mogą występować tylko cztery rodzaje liter.

§ 26. Mozaiki zbudowane poprawnie

Mówiąc, że wyrażenie jest zbudowane poprawnie, mieliśmy na myśli, że zostało ono otrzymane z symboli według określonych reguł. Podobne pojęcie możemy wprowadzić dla mozaik. Jeżeli trójkąty będą składane w myśl określonych reguł, powiemy, że tworzą one mozaikę zbudowaną poprawnie [1]).

Przyjmiemy mianowicie następującą indukcyjną definicję mozaiki zbudowanej poprawnie:

1. Jeden trójkąt równoboczny jest mozaiką zbudowaną poprawnie.

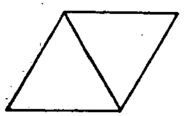
[1]) Reguły składania mozaik można by nazwać gramatyką mozaik, a odpowiadające im reguły składania aminokwasów — gramatyką białkową.

2. Jeżeli dwie mozaiki zbudowane poprawnie skleimy jedną krawędzią zewnętrzną, to otrzymamy mozaikę zbudowaną poprawnie.

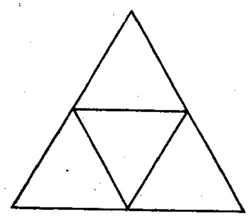
Nasza gramatyka pozwala na budowanie mozaik przez sklejanie ich krawędziami zewnętrznymi. Można podać wiele interesujących własności mozaik zbudowanych poprawnie, jednakże tutaj z braku miejsca nie będziemy ich rozpatrywali, lecz ograniczymy się do najprostszych przykładów ilustrujących podstawowe własności tego rodzaju tworów.

Na rysunku 6 pokazano przykłady mozaik zbudowanych poprawnie. Jeżeli dowolne z tych mozaik skleimy jedną krawędzią zewnętrzną, jak to pokazano na rysunku 7, otrzymamy również mozaiki zbudowane poprawnie. Gruba krawędź oznacza miejsce sklejenia.

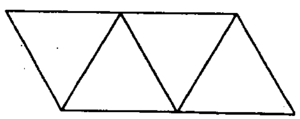
Natomiast na rysunku 8 pokazano mozaiki, które nie są zbudowane poprawnie.



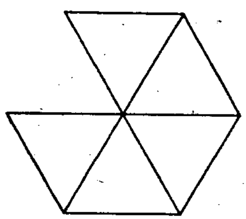
a)



b)

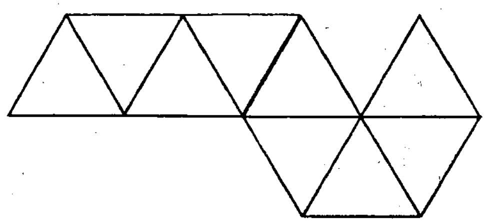


c)

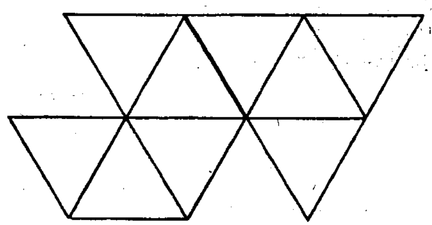


d)

Rys. 6



a)

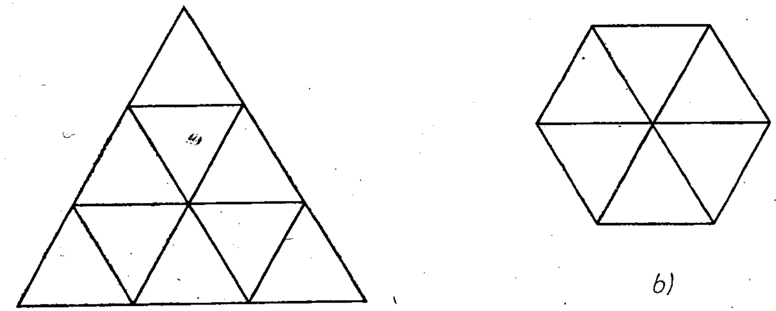


b)

Rys. 7

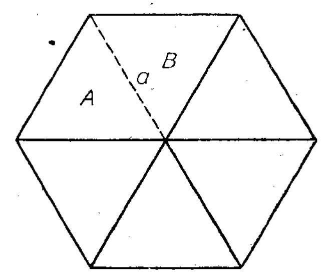
Można sprawdzić, że żadnej z tych mozaik nie da się otrzymać przez sklejanie trójkątów tylko jedną krawędzią.

Zwróćmy uwagę na różnicę między operacją **sklejania** krawędzi a operacją **stykania** krawędzi. Dwa trójkąty mogą się stykać krawędziami, ale nie być sklejane wzdłuż tej krawędzi. Na przykład jeżeli w mozaice przedstawionej na rysunku 9 trójkąty są sklejane wzdłuż krawędzi oznaczonych linią ciągłą, a stykają się wzdłuż krawędzi oznaczonej linią przerywaną, mozaika ta jest zbudowana poprawnie, możemy ją bowiem uzyskać, sklejając trójkąty tylko jedną krawędzią. Do krawędzi a tej mozaiki możemy dokleić nowy trójkąt itd., otrzymując w rezultacie spiralną taśmę. Rozważane więc przez nas mozaiki, chociaż składają się z elementów płaskich, nie muszą być tworami dającymi się przedstawić na płaszczyźnie. Drugą ważną cechą mozaik zbudowanych poprawnie jest to, że przecinając je wzdłuż dowolnej jednej krawędzi wewnętrznej otrzymamy dwie mozaiki poprawne. Mozaiki pokazane na rysunku 8 nie posiadają tej własności. Przecięcie jednej ich krawędzi nie zawsze rozdziela mozaikę na dwie części.



a) b)

Rys. 8

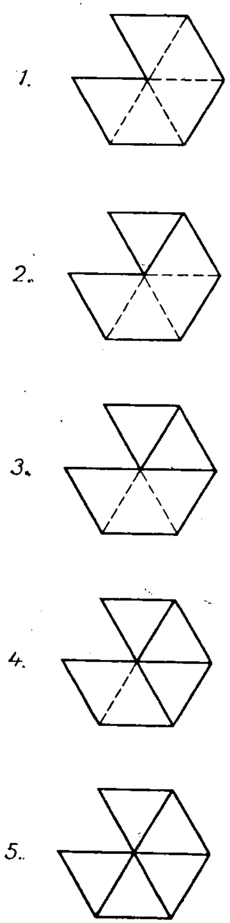


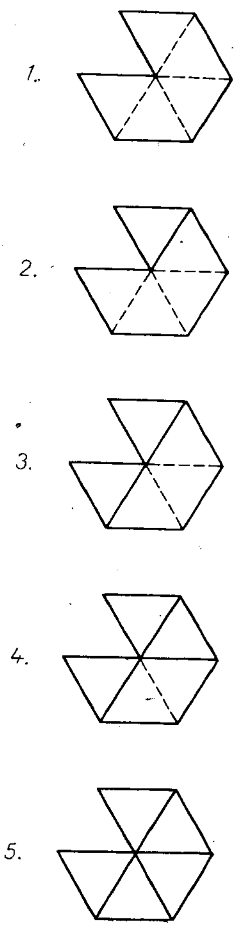
Rys. 9

A więc mozaiki poprawne są tworami „włóknistymi", tzn. nie tworzą one dużych powierzchni, a struktura ich przypomina dowolnie poskręcaną w przestrzeni taśmę jednakowej szerokości, od której w różnych miejscach odchodzą boczne rozgałęzienia. Mozaika taka jest o tyle interesująca, że struktura białka jest również włóknista, a więc przyjęte przez nas założenie nie jest sprzeczne z faktami biochemicznymi.

§ 27. Proces składania mozaiki poprawnej

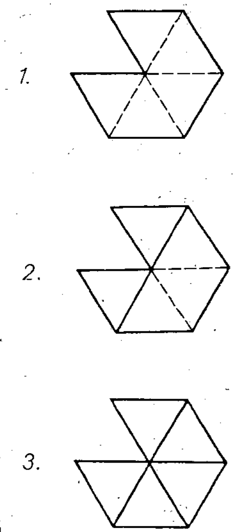
Wiemy, co to są mozaiki poprawne. Zastanówmy się teraz, jak może przebiegać proces składania takiej mozaiki z trójkątów. Proces ten może mieć rozmaity charakter. Dla konstrukcji rozpatrzmy różne możliwości składania mozaiki przedstawionej na rysunku 6d. Jedna z możliwości pokazana jest na rysunku 10. Najpierw narysowano wszystkie trójkąty oddzielnie, a następnie w każdym kroku doklejany jest **jeden** trójkąt.


Rys. 10


Rys. 11

Inna możliwość sklejania pokazana jest na rysunku 11. Różnica między poprzednim składaniem a obecnym polega na tym, że teraz składanie nie odbywa się po jednym trójkącie, a w czwartym kroku sklejane są dwie mozaiki, z których jedna zawiera trzy trójkąty, druga zaś — dwa. Ogólnie biorąc, drugi rodzaj składania polega na tym, że najpierw z trójkątów składane są „małe" mozaiki, z nich zaś składane są większe itd.

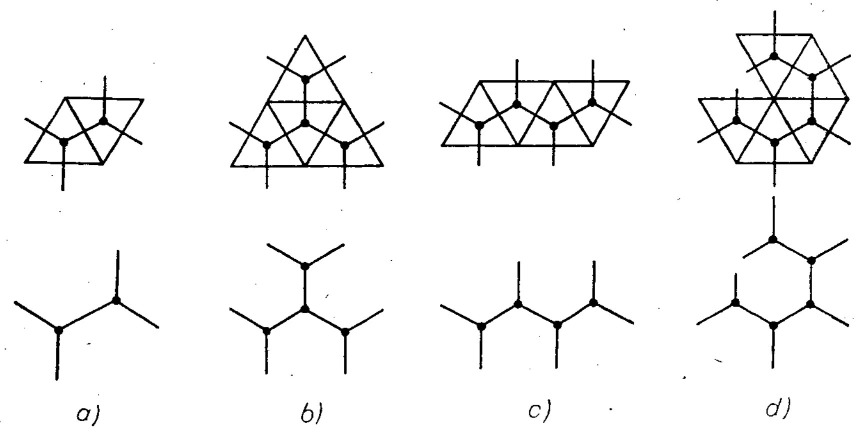
Inny jeszcze rodzaj składania pokazano na rysunku 12. W tym procesie w każdym kroku sklejane są cztery trójkąty, a nie jak poprzednio dwa. Ogólnie, jednocześnie może się odbywać sklejanie mozaiki nie w jednym, lecz w wielu miejscach. Który z tych procesów przyjąć jako najbardziej odpowiedni dla zilustrowania syntezy białka? Wydaje się, że najprawdopodobniejszy jest schemat drugi. Tzn. z aminokwasów są tworzone większe jednostki, z tych zaś jeszcze większe itd. aż do określonej wielkości, i dopiero z tych ostatnich jednostek budowane jest białko. Proces ten jest jednak zbyt skomplikowany, aby od niego rozpoczynać nasze rozważania, i dlatego przyjmiemy, że białko powstaje według schematu pierwszego, tj. w każdej chwili do zbudowanej już cząsteczki białka przyłącza się tylko **jeden** aminokwas. Założenie to nie ogranicza w niczym naszych rozważań. Gdyby się bowiem okazało, że należy przyjąć raczej schemat drugi, to i tak należałoby najpierw rozwiązać zagadnienie, w jaki sposób z aminokwasów — jednostek „budowlanych" powstaje synteza następnego rzędu, a temu właśnie odpowiada schemat pierwszy.


Rys. 12

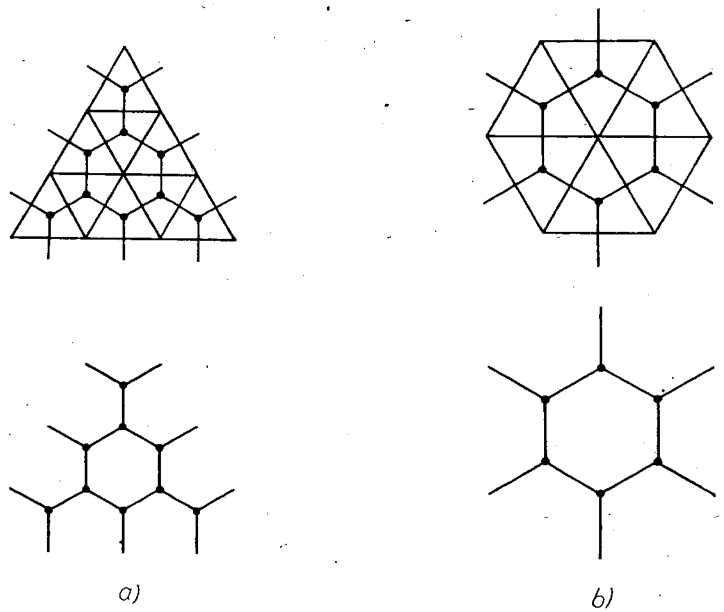
Tak więc zbadanie pierwszego schematu syntezy jest nieodzowne w każdym przypadku.

§ 28. Mozaiki i drzewa

Przedstawianie mozaik w postaci rysunków, jak to czyniliśmy dotychczas, dla bardziej skomplikowanych przypadków jest niewygodne i nieprzejrzyste. Dlatego też wprowadzimy inny sposób graficznego przedstawiania mozaik. Trójkąty będziemy rysowali w postaci punktów, zaś ich krawędzie — w postaci wychodzących z punktów odcinków. Jeżeli dowolne dwa trójkąty mozaiki posiadają krawędź wspólną, to punkty odpowiadające tym trójkątom połączymy jednym odcinkiem. Mozaiki przedstawione na rysunku będziemy więc rysowali tak, jak pokazano na rysunku 13. Natomiast mozaiki pokazane na rysunku 8 przedstawimy tak, jak to pokazano na rysunku 14. Przedstawienie to nazwiemy **grafem** mozaiki. Jeżeli mozaikę zbudowano poprawnie, to jej graf nie zawiera zamkniętych dróg i stanowi tzw. drzewo, tzn. z każdego punktu grafu do innego punktu



Rys. 13

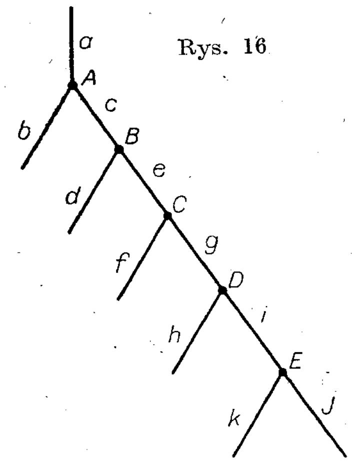


Rys. 14

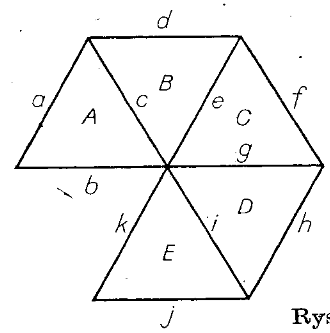
można dojść tylko jedną drogą. Grafy mozaik niepoprawnych nie spełniają tego warunku. Widać to na rysunku 14. Każdej mozaice zbudowanej poprawnie możemy więc przyporządkować drzewo.

W każdym punkcie tego drzewa schodzą się trzy gałęzie. I odwrotnie: każdemu drzewu, którego rozgałęzienia łączą trzy gałęzie, można przyporządkować mozaikę zbudowaną poprawnie. Sposób narysowania drzewa jest oczywiście nieistotny, aby tylko rozgałęzienia były ze sobą połączone gałęzią. Mówi to bowiem o tym, które trójki i którymi gałęziami są ze sobą połączone.

Proces wzrostu mozaiki możemy teraz traktować jako proces wzrostu drzewa. Jeżeli zgodnie z przyjętymi założeniami mozaika wzrosła przez doklejanie do niej po jednym trójkącie, to odpowiadające drzewo będzie „rosło“ w ten sposób, że po doklejeniu trójkąta do zewnętrznej krawędzi mozaiki, w drzewie z odpowiedniej gałęzi wyrastają dwie nowe gałęzie, ilustrujące dwie nowe krawędzie zewnętrzne mozaiki. Oznaczmy trójkąty mozaiki dużymi literami,



Rys. 16



Rys. 15

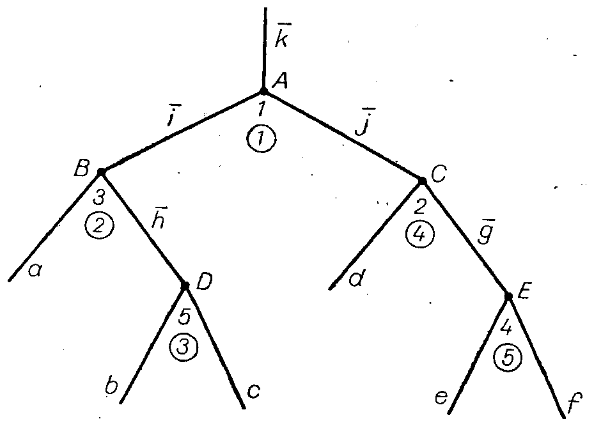
a krawędzie małymi i rozpatrzmy proces tworzenia mozaiki przedstawionej na rysunku 15. Drzewo tej mozaiki pokazano na rysunku 16. Z drzewa tego można odczytać, z jakich trójkątów jest złożona mozaika; które trójkąty i którymi krawędziami są sklejone oraz które krawędzie są zewnętrzne. Np. trójkąty C i D są sklejone krawędzią g, natomiast krawędź f trójkąta C jest zewnętrzna. Drzewo daje więc dokładny opis struktury mozaiki.

Mozaika ta mogła zostać skonstruowana w ten sposób, że do trójkąta A jest doklejony krawędzią c trójkąt B; do trójkąta B jest doklejony krawędzią e trójkąt C, itd. Przy takim rozumieniu drzewa jest więc ono nie tylko opisem struktury mozaiki, ale i opisem procesu jej powstawania.

W dalszym więc ciągu, zamiast posługiwać się rysunkiem mozaiki, będziemy stosować jej drzewo.

§ 29. Języki liniowe mozaik poprawnych

Ponieważ mozaiki poprawne dla nas to tyle, co drzewa, więc liniowe przedstawienie struktury drzew nie przedstawia już trudności. Możemy zastosować tu po małej modyfikacji rozważania z poprzedniego rozdziału. Obecnie istnieje również problem kolejności sklejania trójkątów. Na przykład drzewo mozaiki, przedstawione na rysunku 17,

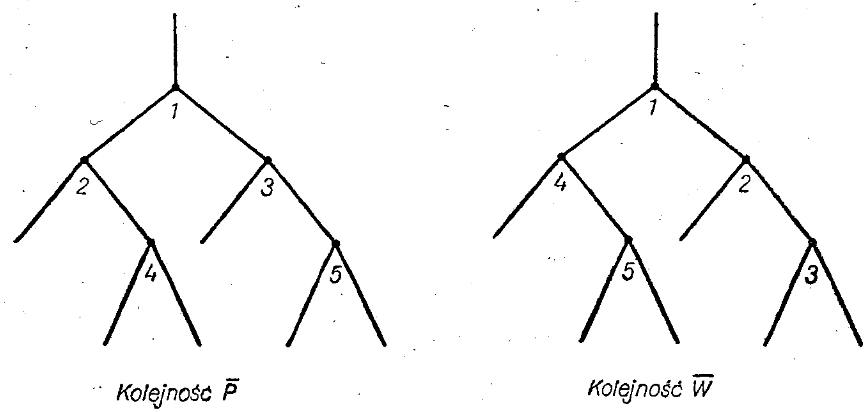


Rys. 17

możemy ponumerować w kolejności P lub W. Liczby bez kółek oznaczają numerację P, liczby z kółkami — numerację W. Litery z kreskami u góry oznaczają krawędzie wewnętrzne mozaiki, litery zaś bez kresek — krawędzie zewnętrzne.

Jednakże teraz trójkąty muszą być sklejone w kolejności numerów rosnących. Gdybyśmy bowiem zaczęli sklejanie od trójkąta nr 5 (przy numeracji P), to okazałoby się, że nie jest spełnione nasze założenie początkowe. Mozaika nie byłaby bowiem budowana przez doklejanie zawsze tylko jednego trójkąta. Przebieg sklejania byłby wtedy następujący: do trójkąta D byłby doklejony trójkąt B krawędzią $\bar{h}$,

tworząc mozaikę składającą się z dwu trójkątów. Następnie do trójkąta E zostałby doklejony trójkąt C krawędzią $\bar{g}$, tworząc drugą mozaikę. Dalej zaś krawędź $\bar{\imath}$ trójkąta B zostałaby doklejona do trójkąta A i w ten sposób powstałaby nowa mozaika, posiadająca trzy trójkąty, i ta mozaika krawędzią $\bar{\jmath}$ zostałaby sklejona z trójkątem C.



Rys. 18

A więc nie jest tu spełniony warunek, że mozaika powstaje przez doklejanie zawsze tylko jednego trójkąta. Dlatego musimy przyjąć, że trójkąty są sklejane poczynając od trójkąta nr 1 (A) aż do trójkąta o największym numerze, przy zadanej numeracji (P lub W). Z tego powodu wygodnie jest przyjąć inną nieco numerację rozgałęzień drzewa, którą oznaczymy przez $\overline{\mathrm{P}}$ i $\overline{\mathrm{W}}$, a która jest jak gdyby odbiciem lustrzanym poprzednich numeracji P i W. Wyjaśnienie tej numeracji pokazano na rysunku 18. W związku ze zmianą numeracji będziemy również formułę drzewa pisali w kierunku przeciwnym, aniżeli to czyniliśmy w poprzednim rozdziale. Program składania mozaiki przedstawionej na rysunku 17 będzie miał więc postać:

Kolejność $\overline{\mathrm{P}}$

$$k\ A\ \bar{\imath}\ \bar{\jmath}\quad B\ a\ \bar{h}\quad C\ d\ \bar{g}\quad D\ b\ c\quad E\ e\ f$$

Kolejność $\overline{\mathrm{W}}$

$$k\ A\ \bar{\imath}\ \bar{\jmath}\quad C\ d\ \bar{g}\quad E\ e\ f\quad B\ a\ \bar{h}\quad D\ b\ c$$

Aby program ten był zrozumiały, musimy zdołać odczytać każdą trzecią krawędź każdego trójkąta, gdyż z wyjątkiem trójkąta A na pierwszy rzut oka) nie są one z programu widoczne. Do tego celu możemy oczywiście zastosować tę samą zasadę stawiania strzałek, co w poprzednim rozdziale.

Kolejność $\overline{\mathrm{P}}$

$k \quad A\,\bar{i}\,\bar{j} \quad B\,a\,\bar{h} \quad C\,d\,\bar{g} \quad D\,b\,c \quad E\,e\,f$

Kolejność $\overline{\mathrm{W}}$

$k \quad A\,\bar{i}\,\bar{j} \quad C\,d\,\bar{g} \quad E\,e\,f \quad B\,a\,\bar{h} \quad D\,b\,c$

Dla każdego trójkąta strzałki wskazują jego trzecią krawędź, którą jest on sklejony z innym trójkątem. Np. z programu możemy odczytać, że trójkąt C ma być sklejony z trójkątem A krawędzią $\bar{j}$[1].

Program ten zaś należy czytać od strony lewej do prawej, symbol po symbolu i sklejać kolejno trójkąty krawędziami wskazanymi przez strzałki. Np. dla kolejności $\overline{\mathrm{P}}$, należy do trójkąta A dokleić trójkąt B krawędzią $\bar{i}$; następnie do trójkąta A dokleić trójkąt D krawędzią $\bar{p}$, itd. Ponieważ każdy trójkąt jest wyznaczany przez jego krawędzie, w programach możemy opuścić duże litery i zostaną one nadal jednoznaczne.

Kolejność $\overline{\mathrm{P}}$

$k \quad \bar{i}\,\bar{j} \quad a\,\bar{h} \quad d\,\bar{g} \quad b\,c \quad e\,f$

Kolejność $\overline{\mathrm{W}}$

$k \quad \bar{i}\,\bar{j} \quad d\,\bar{g} \quad e\,f \quad a\,\bar{h} \quad b\,c$

Programy te nie są zbyt czytelne, ale przy odrobinie wysiłku można według nich postępując skleić mozaikę z trójkątów.

W rozpatrywanym języku użyliśmy alfabetu o dowolnej ilości liter. Jak teraz uwzględnić fakt, że alfabet naszego języka może być tylko czteroliterowy?

[1]) Tzn. krawędź $\bar{j}$ jest wspólną krawędzią trójkątów A i C.

§ 30. Alfabet czteroliterowy

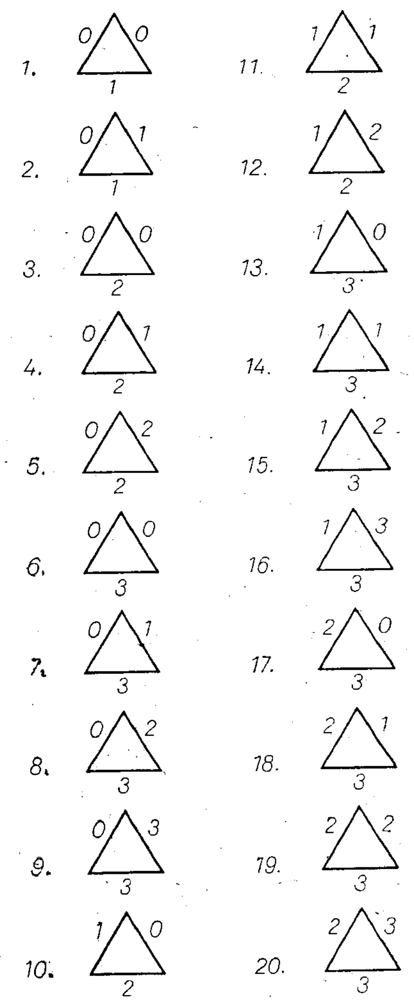
Jako symbole czteroliterowego alfabetu możemy przyjąć zupełnie dowolne symbole. Przyjmijmy więc cyfry 0, 1, 2, 3. Alfabet ten będziemy nazywali **genetycznym.** Ponieważ symbolami naszego języka oznaczaliśmy krawędzie trójkątów, obecnie więc krawędzie trójkątów będziemy oznaczali cyframi 0, 1, 2, 3. Jednakże nie w sposób dowolny, lecz według specjalnych reguł. Będziemy mianowicie w każdym trójkącie wyróżniać podstawę, lewy bok i prawy bok. Krawędzie trójkąta będziemy numerować według zasady następującej:

$$(1) \qquad N(k_0) < N(k_1) \geqslant N(k_2)$$

gdzie k_0 oznacza lewy bok trójkąta, k_1 — podstawę, k_2 — prawy jego bok, zaś $N(k_0)$ numer lewego boku, a $N(k_1)$ i $N(k_2)$ odpowiednie numery podstawy i prawego boku.

Krawędzie mozaiki składającej się z jednego trójkąta możemy wtedy ponumerować na 20 sposobów, jak to pokazano na rysunku 19. Dzięki tak wprowadzonej numeracji mamy dwadzieścia różnych trójkątów, z których teraz możemy budować rozmaite mozaiki. Trójkąty te są więc odpowiednikami dwudziestu podstawowych aminokwasów. Zamiast rysować trójkąty, czasem wygodnie jest pisać liczby przyporządkowane ich krawędziom w ustalonym porządku:

$$< N(k_0), \quad N(k_1), \quad N(k_2) >$$



Rys. 19

a więc nasza mozaika może się składać z 20 następujących trójkątów:

1.	$<0\ 1\ 0>$	11.	$<1\ 2\ 1>$
2.	$<0\ 1\ 1>$	12.	$<1\ 2\ 2>$
3.	$<0\ 2\ 0>$	13.	$<1\ 3\ 0>$
4.	$<0\ 2\ 1>$	14.	$<1\ 3\ 1>$
5.	$<0\ 2\ 2>$	15.	$<1\ 3\ 2>$
6.	$<0\ 3\ 0>$	16.	$<1\ 3\ 3>$
7.	$<0\ 3\ 1>$	17.	$<2\ 3\ 0>$
8.	$<0\ 3\ 2>$	18.	$<2\ 3\ 1>$
9.	$<0\ 3\ 3>$	19.	$<2\ 3\ 2>$
10.	$<1\ 2\ 0>$	20.	$<2\ 3\ 3>$

Przy przyjętej zasadzie numeracji krawędzi, inne trójkąty nie istnieją. Jest więc ich dokładnie tyle, ile jest aminokwasów. Trójki liczb w nawiasach można by zamienić więc na nazwy każdego aminokwasu.

Jak wprowadzenie alfabetu genetycznego wpływa na zasady łączenia trójkątów w mozaiki? Oczywiście reguł łączenia tak ponumerowanych trójkątów można podać bardzo wiele. Podamy jedną z takich zasad, która jest bardzo prosta i wydaje się prawdopodobne, że może być ona przybliżonym obrazem łączenia aminokwasów. Dla odróżnienia od trójkątów nieponumerowanych, trójkąty z ponumerowanymi krawędziami nazwiemy γ-trójkątami.

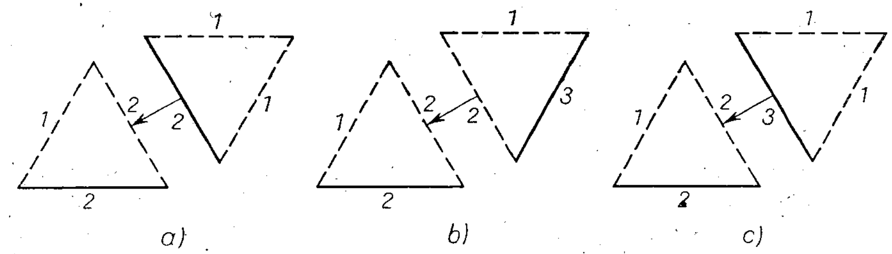
§ 31. Łączenie γ-trójkątów

Mozaiki składające się z γ-trójkątów nazwiemy γ-mozaikami. Zanim podamy dokładne określenie γ-mozaiki, pokażemy najpierw, w jaki sposób można ze sobą sklejać γ-trójkąty.

Niech b będzie dowolnym bokiem trójkąta T_1 i niech p będzie podstawą γ-trójkąta T_2. Krawędź p trójkąta T_2 można przykleić do krawędzi b trójkąta T_1 wtedy i tylko wtedy, jeżeli $N(p) = N(b) \neq 0$. Operacja ta jest więc niesymetryczna, gdyż trójkąt T_2 przyklejamy do trójkąta T_1, a nie odwrotnie. Nie możemy więc przykleić do podstawy jakiegoś trójkąta krawędzi innego trójkąta, lecz tylko do dowolnego boku trójkąta możemy przykleić inny trójkąt podstawą,

jeżeli ponadto obie sklejane krawędzie mają jednakowe numery różne od zera.

Na rysunku 20 pokazano dozwolone i niedozwolone sklejanie γ-trójkątów. Sklejane są przeciwległe krawędzie. Dla jasności — podstawy trójkątów są oznaczone linią ciągłą, boki natomiast — przerywaną. Rysunek 20 a przedstawia złożenie poprawne, gdyż jedna krawędź jest bokiem, druga podstawą i obie krawędzie mają jednakowy



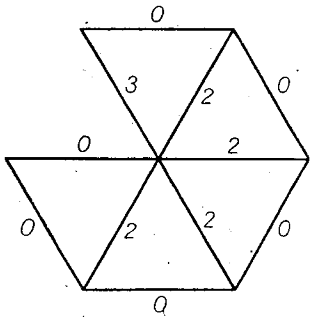
Rys. 20

numer 2. Sklejenie na rysunku 20 b jest niepoprawne, gdyż obie krawędzie są bokami, zaś na rysunku 20 c obie przeciwległe krawędzie posiadają różne numery, a więc również ich sklejanie jest niedozwolone. Inne przypadki niedozwolonego sklejania trójkątów są oczywiste.

Określimy obecnie indukcyjnie pojęcie γ-mozaiki.

1. Każdy γ-trójkąt jest γ-mozaiką.
2. Jeżeli γ-trójkąt T_1 jest elementem mozaiki M i krawędź k trójkąta T_1 jest krawędzią zewnętrzną mozaiki M, to mozaika otrzymana przez doklejenie do krawędzi k γ-trójkąta T_2 jest również γ-mozaiką (zakładamy przy tym, że trójkąty T_1 i T_2 są sklejone zgodnie z regułą sklejania γ-trójkątów).

Warto zauważyć, że składania γ-mozaiki nie możemy zacząć od dowolnego γ-trójkąta. Nie możemy zacząć składania γ-mozaiki od trójkątów $<010>$, $<020>$, $<030>$, gdyż do żadnej z krawędzi tych trójkątów nie możemy dokleić żadnego innego γ-trójkąta, nie pozwala na to bowiem reguła sklejania γ-trójkątów. Trójkąty te mogą być natomiast doklejane krawędzią różną od zera do innych trójkątów. Widać tu wyraźnie asymetrię reguły sklejania γ-trójkątów.



Rys. 21

W wyniku sklejania γ-trójkątów możemy otrzymać mozaiki, do których nic już dokleić nie można. Nie pozwalają na to reguły sklejania. Mozaikę taką nazwiemy **kompletną**. Przykład mozaiki kompletnej pokazano na rysunku 21. Jedna krawędź zewnętrzna tej mozaiki jest oznaczona numerem 3, a wszystkie pozostałe krawędzie zewnętrzne są oznaczone numerem 0. Zgodnie z regułami tworzenia γ-mozaik, do krawędzi zewnętrznych oznaczonych numerem 0 nie można dokleić żadnego γ-trójkąta. Pozostała jeszcze krawędź 3. Krawędź ta jest podstawą trójkąta <032>, jednakże do krawędzi, która jest podstawą, niezależnie od jej numeru, również nie możemy dokleić żadnego γ-trójkąta, składanie tej mozaiki jest więc całkowicie zakończone i dlatego nazywamy ją mozaiką kompletną. Mozaiki kompletne będziemy uważali za odpowiedniki białek.

Uformowana już ostatecznie mozaika nie może się dalej rozbudowywać i stanowi zupełną całość. Prawdopodobnie podobna sytuacja jest obowiązująca dla białek. Do białka ostatecznie już ukształtowanego nie można dalej doczepiać aminokwasów. Łańcuch białkowy ma wtedy zamkniętą strukturę i dalszemu rozwojowi nie podlega.

§ 32. Gramatyka języka genetycznego

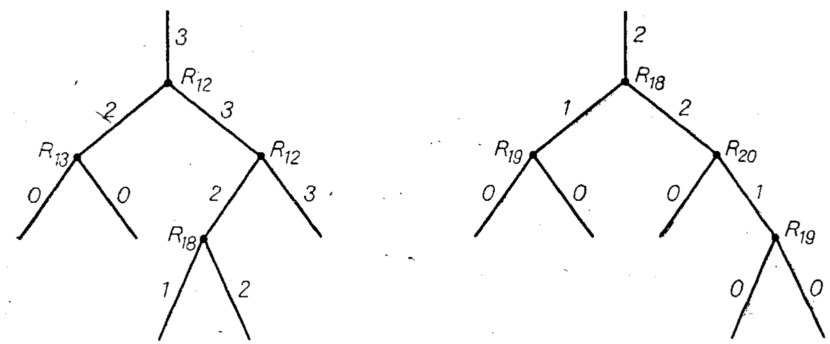
W paragrafie tym zajmiemy się gramatyką języka opisującego proces rozwoju γ-mozaik kompletnych, które — raz jeszcze przypominamy — uważamy za przybliżony obraz białka. A więc białkiem nie może być każda struktura utworzona z aminokwasów według jakichś reguł formowania, a tylko taka struktura, która dalej już rozwijać się nie może, której budowa została zakończona. W dalszym ciągu często zamiast „γ-mozaiki kompletne“ będziemy mówili „białko“.

Celem prześledzenia procesu formowania się białka, przejdźmy do opisu tego procesu za pomocą drzew. Tak jak poprzednio, zamiast mówić o tworzeniu się białka, możemy mówić o wzroście odpowiedniego drzewa. Podamy więc reguły tworzenia drzew przedstawiających

białko. Reguły te sformułujemy na wzór gramatyki struktur frazowych. Gramatyka ta składa się z dwudziestu reguł:

$\mathbf{R}_1: 3 \to 00$	$\mathbf{R}_{11}: 3 \to 22$
$\mathbf{R}_2: 3 \to 01$	$\mathbf{R}_{12}: 3 \to 23$
$\mathbf{R}_3: 3 \to 02$	$\mathbf{R}_{13}: 2 \to 00$
$\mathbf{R}_4: 3 \to 03$	$\mathbf{R}_{14}: 2 \to 01$
$\mathbf{R}_5: 3 \to 10$	$\mathbf{R}_{15}: 2 \to 02$
$\mathbf{R}_6: 3 \to 11$	$\mathbf{R}_{16}: 2 \to 10$
$\mathbf{R}_7: 3 \to 12$	$\mathbf{R}_{17}: 2 \to 11$
$\mathbf{R}_8: 3 \to 13$	$\mathbf{R}_{18}: 2 \to 12$
$\mathbf{R}_9: 3 \to 20$	$\mathbf{R}_{19}: 1 \to 00$
$\mathbf{R}_{10}: 3 \to 21$	$\mathbf{R}_{20}: 1 \to 01$

Reguły te należy rozumieć następująco: jeżeli gałąź drzewa ma numer napisany po lewej stronie strzałki, to „wyrastające“ z niej dwie gałęzie mogą mieć tylko numery napisane po prawej stronie strzałki, przy czym lewa gałąź — numer lewy, a prawa — prawy. Np. z gałęzi nr 3 według reguły $\mathbf{R}_9$ może wyrastać lewa gałąź o numerze 2 i prawa gałąź o numerze 0.

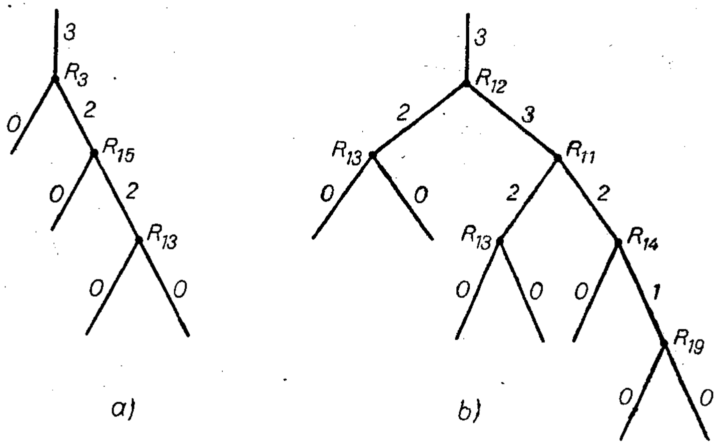


Rys. 22

Przykłady drzew zbudowanych według tej gramatyki przedstawiono na rysunku 22. Drzewo, do którego nie można już stosować reguł $\mathbf{R}_1$ — $\mathbf{R}_{20}$, nazwiemy **kompletnym**. Można przypuszczać, że proces powstawania białka da się opisać drzewem takiego typu.[1])

1) Drzewo kompletne jest jednocześnie obrazem struktury białka.

Przykłady drzew kompletnych pokazano na rysunku 23. Z drzew tych łatwo odczytać przebieg formowania białka. Np. według rysunku 23a trójkąt <032> jest sklejany z trójkątem <022> krawędzią 2; dalej krawędź 2 trójkąta <022> jest sklejana z krawędzią 2 trójkąta <020>.



Rys. 23

Program ten możemy również przedstawić w języku procesów następująco:

3 02 02 00

Ciąg ten jest więc zdaniem w języku genetycznym. Odczytanie jego nie sprawia trudności. Musimy tylko najpierw postawić strzałki. Ponieważ kolejności $\overline{\text{P}}$ i $\overline{\text{W}}$ są w tym przypadku identyczne, więc strzałki ustawimy następująco:

3 02 02 00

Każdej parze cyfr przyporządkowana jest trzecia cyfra 1 lub 2, według reguł podanych przy omawianiu języków procesów.

Proces zaś przedstawiony na rysunku 23b będzie miał program:

Kolejność $\overline{\text{P}}$

3 23 00 22 00 01 00

Kolejność $\overline{\text{W}}$

3 23 22 01 00 00 00

Przyjmując więc model białka w postaci γ-mozaiki kompletnej, proces jego tworzenia możemy opisać w określonym języku. Zdaniami tego języka są ciągi symboli 0, 1, 2, 3. Ciągi te mają ściśle określoną strukturę i prawidłową budowę. Można by je nazwać zdaniami genetycznymi. Symbolami w tych zdaniach są związki biochemiczne. Ciąg zdań stanowi tekst, według którego odbywa się rozwój każdego organizmu.

Czy rzeczywisty język, według którego jest zaprogramowany rozwój organizmu, ma charakter zbliżony do przedstawionego tutaj — nie wiadomo. Odkrycie gramatyki tego języka jest bardzo trudne i wymaga nie tylko badań struktury chromosomów i białek, ale również badań matematycznych. Można podać wiele innych jeszcze języków, które z logicznego punktu widzenia mogą stanowić punkt wyjścia do badań języków genetycznych. Tutaj nie będziemy ich omawiać, ale na jeden z nich warto zwrócić uwagę. Jeżeli mianowicie wypiszemy wszystkie trójkąty w kolejności $\overline{\text{P}}$ lub $\overline{\text{W}}$, to otrzymane tak wyrażenie jest również jednoznacznym programem syntezy białka.

Dla drzewa pokazanego na rysunku 23b programy te będą miały postać:

Kolejność $\overline{\text{P}}$

233 020 232 020 021 010

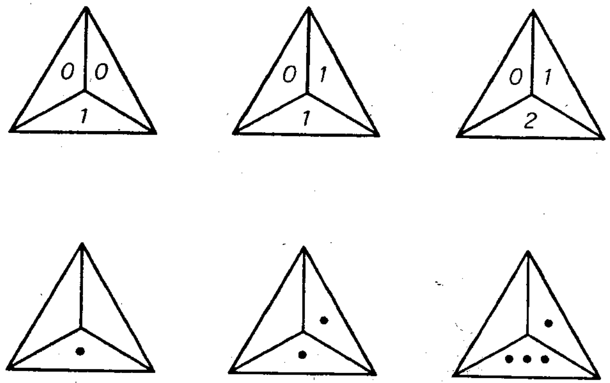
Kolejność $\overline{\text{W}}$

233 232 021 010 020 020.

Odczytanie tych programów pozostawiamy jako zadanie Czytelnikowi. Odstępy między trójkami symboli są zbędne, daliśmy je tylko dla ułatwienia czytania programów.

Dla zdobycia lepszego wyobrażenia o otrzymywanych tu mozaikach polecamy Czytelnikowi wykonanie z papieru trójkątów o krawędziach ponumerowanych według podanych zasad, np. tak jak to pokazano na rysunku 24, oraz sklejenie mozaiki według programu:

011 011 011 011 011 010



Rys. 24

Sklejanie mozaiki przypomina zabawę w domino, z tą różnicą, że klocki są w naszym przykładzie nie prostokątne, a trójkątne.

LITERATURA

G. Gamow: *Possible mathematical relation between deoxyribonucleic acid and proteins.* Biol. Medd. Dan. Vid. Selsk. 22(1955)8.

F. G. C. Crick, J. Griffith, L. Orgel: *Codes without commas.* Proc. Nat. Acad. Sci., US, 43(1957)416.

S. Golomb, B. Gordon, L. Welch: *Comma-free codes.* Can. J. Math., 10(1958), 202—209.

S. Golomb: *Proceedings of the Symposium on Math. problems in the biological science,* Am. Math. Soc. 1961.

C. Woese: *The Genetic code* — 1963. ICSU Review of World Science 5 (1963) str. 210—252.

Z. Pawlak: *Genetics languages* (w przygotowaniu do druku).

Rozdział VII

JĘZYKI MASZYN MATEMATYCZNYCH

§ 33. Maszyny cyfrowe

W rozdziale tym zapoznamy Czytelnika z ostatnim językiem dyskutowanym w tej książce. Język ten będzie zupełnie inny niż języki przedstawione w poprzednich rozdziałach.

Omawiane do tej pory języki określaliśmy przez podanie ich gramatyki bądź też określaliśmy je semantycznie.

Języki określane tą drugą metodą miały również pewną gramatykę, którą można było sformułować na podstawie analizy zdań poprawnych języka. Istnieją ciekawe języki, dla których do tej pory nie udało się określić reguł gramatyki. Poprawność zdań w tych językach można badać tylko poprzez szczegółową analizę treści znaczeniowych. Są to języki współczesnych maszyn matematycznych, tzw. maszyn cyfrowych. Aby rozwiązać jakiekolwiek zadanie za pomocą takiej maszyny, należy ułożyć szczegółowy program rozwiązania zadania w języku maszyny, która ma być użyta do rozwiązywania problemu. Następnie program jest wprowadzony do maszyny i maszyna postępuje według niego, krok po kroku rozwiązując w ten sposób postawione zadanie.

Programy takie zawierają olbrzymią ilość symboli, więc układanie programów prowadzi do częstych pomyłek. Aby sprawdzić, czy program jest zbudowany poprawnie, byłoby wygodne posiadać odpowiednią gramatykę takiego języka i postępując czysto formalnie według reguł gramatycznych sprawdzać, czy ułożony program jest poprawny. Niestety do tej pory nie udało się wykryć reguł gramatycznych tego języka. Każdy ułożony program należy więc sprawdzać dokładnie od strony znaczeniowej, analizując, czy rzeczywiście opisuje on to, o co nam chodzi. Jest to jeden z czynników znacznie utrudniających naprawdę szerokie zastosowanie maszyn matematycznych. Znaleziono wprawdzie częściowe wyjście z tej sytuacji w postaci tzw. automatycznego programowania, polegającego na tym, że problemy są zapisywane nie w języku maszyny, lecz w tzw. języku zewnętrznym, zbliżonym do potocznie używanego języka w matematyce. Dało to wprawdzie znaczne uproszczenie obsługi maszyny, kosztem jednakże jej szybkości działania i ogólnej sprawności. Maszyna bowiem musi wtedy najpierw przetłumaczyć program z języka zewnętrznego na język wewnętrzny, a czynność ta jest wysoce czasochłonna.

W dalszym ciągu zapoznamy się z językiem wewnętrznym maszyn cyfrowych. Uprzednio jednak musimy się zapoznać z zasadą działania maszyn cyfrowych.

Nie będziemy szczegółowo wyjaśniali zasady działania maszyny cyfrowej, lecz ograniczymy się tylko do elementów wiążących się bezpośrednio z poruszanym przez nas tematem. [1])

Jako alfabet maszyny przyjmiemy następujące symbole: +, —, ·, /, →, ←, !, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Są to symbole czterech działań arytmetycznych, cyfry od 0 do 9. Trzy pozostałe symbole →, ←, ! wyjaśnimy później.

Podobnie jak w maszynie Turinga, podstawowym elementem maszyny cyfrowej jest taśma (skończona) podzielona na kratki. [2])

Wszystkie kratki pamięci są kolejno ponumerowane liczbami naturalnymi, poczynając od zera. [3])

0	1	2	3	4			n

Numery kratek nazywane są **adresami.** Kratkę o adresie zero będziemy

[1]) Czytelnik interesujący się bliżej działaniem maszyn cyfrowych znajdzie bardziej wyczerpujące informacje na temat ich działania np. w książce B. A. Trachtenbrota, *Algorytmy i automatyczne rozwiązywanie zadań.* PWN, Warszawa 1961.

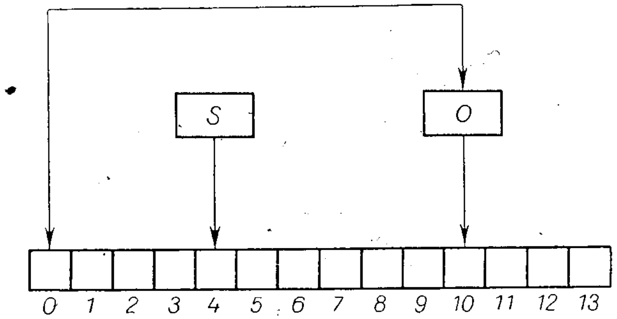
[2]) W rzeczywistej maszynie cyfrowej rolę taśmy gra skomplikowane urządzenie elektronowe zwane pamięcią maszyny; w pamięci maszyny można bardzo szybko zapisywać i odczytywać symbole. Dla naszych celów wystarczy, jeżeli pamięć będziemy uważali za pokratkowaną taśmę papieru.

[3]) Taśma ta różni się więc od taśmy w maszynie Turinga tym, że poprzednio kratki nie były numerowane.

nazywali **akumulatorem.** W każdej kratce może być zapisany jeden symbol alfabetu maszyny.

Maszyna posiada ponadto **sterowanie i operatora.** W każdej chwili sterowanie maszyny obserwuje jedną kratkę taśmy, a operator obserwuje dwie kratki, z których jedną jest zawsze kratka 0 (akumulator), jak to pokazano na rysunku 25. (Na rysunku tym sterowanie obserwuje kratkę o adresie 4, a operator — kratkę o adresie 10 i akumulator).

Rys. 25



Sterowanie maszyny może się znajdować w jednym z $k+1$ możliwych stanów wewnętrznych $q_0, q_1, \ldots, q_k$. Stan q_0 nazwiemy stanem **początkowym,** a q_k stanem **końcowym.** Stan końcowy będziemy oznaczali q^*. Stany $q_0, q_1, \ldots, q_{k-1}$ będziemy też nazywali stanami **czynnymi,** a stan q^* — stanem **biernym.**

Jeżeli sterowanie, znajdując się w stanie q_i, obserwuje kratkę o adresie α_s, w której jest zapisany symbol σ_s, to powiemy, że sterowanie jest w sytuacji $(q_i, \alpha_s, \sigma_s)$. Podobnie **sytuację operatora** określimy jako $(\alpha_0, \sigma_0, \sigma_A)$, gdzie α_0 oznacza adres komórki obserwowanej przez operatora, σ_0 — symbol zapisany w komórce obserwowanej przez operatora, σ_A — symbol zapisany w akumulatorze.

Jeżeli sterowanie jest w stanie czynnym, maszyna wykonuje **ruch.** Ruch maszyny zmienia sytuację sterowania z $(q_i, \alpha_s, \sigma_s)$ na $(q_i', \alpha_s', \sigma_s')$ oraz sytuację operatora z $(\alpha_0, \sigma_0, \sigma_A)$ na $(\alpha_0', \sigma_0', \sigma_A')$. Pisząc symbolicznie,

$$(q_i, \alpha_s, \sigma_s) \Rightarrow (q_i', \alpha_s', \sigma_s')$$
$$(\alpha_0, \sigma_0, \sigma_A) \Rightarrow (\sigma_0', \sigma_0', \sigma_A')$$

Znaczy to, że w wyniku ruchu sterowanie przechodzi do innego stanu, obserwuje inne miejsce na taśmie i odczytuje inny symbol;

również operator zmienia obserwowany kwadrat, a w nim nowy symbol oraz zmienia symbol w akumulatorze.

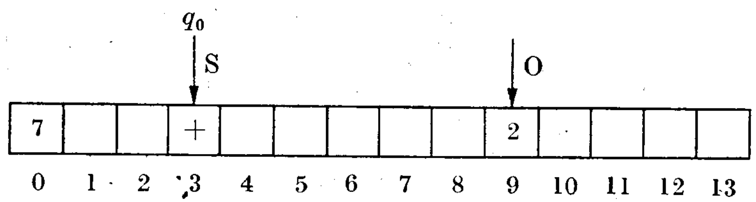
Jeżeli sterowanie jest w stanie końcowym, maszyna zatrzymuje się i przerywa działanie. Ponieważ alfabet oraz liczba stanów maszyny są skończone, zachowanie maszyny możemy opisać w postaci tablicy charakterystycznej, podobnie jak to czyniliśmy dla maszyny Turinga. W tym przypadku jednak opis taki byłby nieprzejrzysty i dlatego posłużymy się opisem słownym.

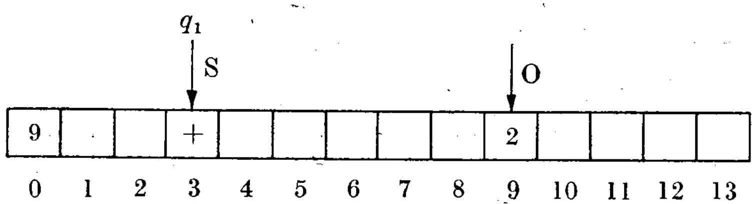
Przyjmiemy, że sterowanie maszyny cyfrowej może znajdować się w jednym z trzech stanów q_0, q_1, q^*.

Jeżeli sterowanie znajduje się w sytuacji (q_0, α_s, ω), gdzie ω jest jednym z symboli $+$, $-$, $.$, $/$, a operator jest w sytuacji α_0, σ_0, σ_A, to maszyna wykonuje ruch składający się z dwu elementów:

$$(\alpha_0, \sigma_0, \sigma_A) \Rightarrow (\alpha_0, \sigma_0, \sigma_A\omega\sigma_0)$$
$$(q_0, \alpha_s, \omega) \Rightarrow (q_1, \alpha_s, \omega)$$

Znaczy to, że na symbolach σ_0 i σ_A zostanie wykonane działanie ω i wynik tego działania zostaje zapisany w akumulatorze, sterowanie zaś przechodzi do stanu q_1, nie zmieniając α_s ani ω, jak to pokazano niżej:





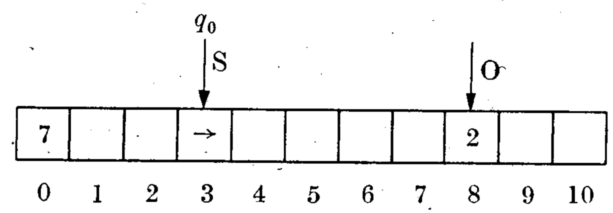
Strzałki →, ← oznaczają odpowiednie przepisanie symbolu z akumulatora do miejsca α_0 i odwrotnie — z miejsca α_0 do akumulatora. Jeżeli w stanie q_0 odczytany zostanie symbol ←, to pierwszy etap ruchu maszyny będzie miał postać

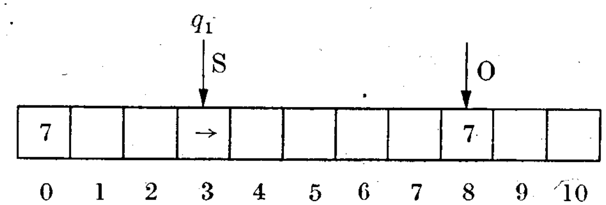
$$(\alpha_0, \sigma_0, \sigma_A) \Rightarrow (\alpha_0, \sigma_A, \sigma_A)$$

podobnie dla symbolu →

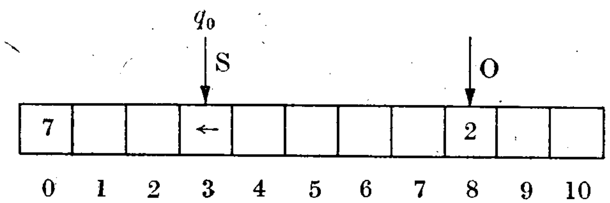
$$(\alpha_0, \sigma_0, \sigma_A) \Rightarrow (\alpha_0, \sigma_0, \sigma_0)$$

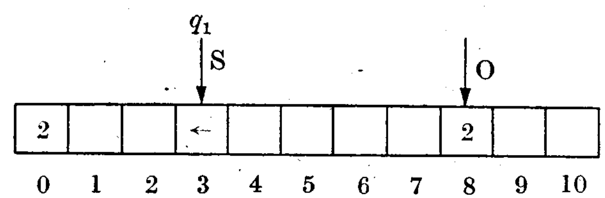
Druga część ruchu pozostanie bez zmiany. Ruch maszyny wygląda więc dla symbolu → następująco:





i dla symbolu ← jest podobny





W przypadku odczytania symbolu ! maszyna przechodzi do stanu końcowego q^*.

Jeżeli sterowanie jest w sytuacji $(q_1,\ \alpha_s,\ \sigma_s)$, to ruch sterowania ma postać:

$$(q_1,\ \alpha_s,\ \sigma_s) \Rightarrow (q_0,\ \alpha_{s+1},\ \sigma_s')$$

a operator nie wykonuje żadnego ruchu.

Znaczy to, że sterowanie przechodzi ponownie do stanu q_0 i obserwuje następną kratkę na taśmie. Np.:

				q_1 S ↓				O ↓		
7				→	9			5		
0	1	2	3	4	5	6	7	8	9	10

					q_0 S ↓			O ↓		
7				→	9			5		
0	1	2	3	4	5	6	7	8	9	10

Jeżeli sterowanie znajduje się w sytuacji $(q_0,\ \alpha_s,\ \lambda)$, gdzie λ jest liczbą, to powoduje to ruch operatora

$$(\alpha_0,\ \sigma_0,\ \sigma_A) \Rightarrow (\lambda,\ \sigma_0',\ \sigma_A)$$

oraz ruch sterowania

$$(q_0,\ \alpha_s,\ \lambda) \Rightarrow (q_1,\ \alpha_s,\ \lambda)$$

Następujący przykład ilustruje ten ruch:

			q_0 S ↓							O ↓
2			7				5			
0	1	2	3	4	5	6	7	8	9	10

			q_1 S ↓				O ↓			
2			7				5			
0	1	2	3	4	5	6	7	8	9	10

A więc odczytanie przez sterowanie liczby 7 powoduje przejście operatora do obserwowania komórki o adresie 7.

Jeżeli sterowanie jest w stanie q_1, to niezależnie od symbolu σ_s wykonywany jest zawsze ruch.

$$(q_1,\ \alpha_s,\ \sigma_s) \Rightarrow (q_0,\ \alpha_{s+1},\ \sigma_s')$$
$$(\alpha_0,\ \sigma_0,\ \sigma_A) \Rightarrow (\alpha_0,\ \sigma_0,\ \sigma_A)$$

Działanie maszyny cyfrowej polega więc na tym, że jeżeli sterowanie odczyta symbol operacji, to operator wykonuje odczytaną przez sterowanie operację i sterowanie przechodzi do odczytania następnego symbolu na taśmie. Jeżeli sterowanie odczytuje liczbę λ, to operator jest nastawiany na odczytanie komórki o adresie λ i sterowanie przechodzi do odczytania następnego symbolu na taśmie.

Zastanówmy się teraz, jak możemy opisać w języku maszyny cyfrowej wykonanie najprostszych operacji arytmetycznych.

§ 34. Programy maszyn cyfrowych

Rozpatrzmy, w jaki sposób maszyna cyfrowa wykona proste obliczenie

$$(2 + (3 \cdot 2)) / 4$$

Przed rozpoczęciem liczenia sytuacja maszyny jest następująca:

	q_0 S ↓												O ↓
	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Sterowanie jest nastawione na odczytywanie symbolu, znajdującego się w komórce 1. Operator odczytuje dowolną komórkę taśmy.

Wykonywanie obliczenia przez maszynę przebiega następująco:

1.

	q_1 S ↓									O ↓			
	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

2.

		q_0 S ↓								O ↓			
	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

3.

		q_1 S ↓								O ↓			
2	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

4.

			q_0 S ↓							O ↓			
2	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

5.

			q_1 S ↓								O ↓		
2	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

6.

				q_0 S ↓							O ↓		
2	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

7.

				q_1 S ↓							O ↓		
6	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

8.

					q_0 S ↓						O ↓		
6	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

9.

					q_1 S ↓					O ↓			
6	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

10.

						q_0 S ↓				O ↓			
6	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

11.

						q_1 S ↓				O ↓			
8	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

12.

							q_0 S ↓			O ↓			
8	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

13.

							q_1 S ↓					O ↓	
8	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

14.

								q_0 S ↓				O ↓	
8	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

15.

								q_1 S ↓				O ↓	
2	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

16.

									q^* S ↓			O ↓	
2	10	←	11	·	10	+	12	/	!	2	3	4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Gdy maszyna przeszła do stanu końcowego, w akumulatorze znalazł się wynik obliczenia.

Programem opisującym przebieg obliczenia jest tutaj ciąg symboli

$$10 \leftarrow 11 \cdot 10 + 12 \; / \; !$$

zapisanych na taśmie maszyn od miejsca o adresie 1 do 9.

Nietrudno podać programy innych obliczeń lub też dowolnych manipulacji na symbolach.

Zastanówmy się teraz nad własnościami języka maszyn cyfrowych.

§ 35. Języki maszyn cyfrowych

Języki maszyn cyfrowych posiadają dość dziwne nie spotykane w dotychczas omawianych językach własności. Nie będziemy rozpatrywali wszystkich własności takich języków, lecz tylko te, które się wiążą z poruszanym przez nas tematem: gramatyką oraz znaczeniem. Dla krótkości języki takie nazwiemy językami **cyfrowymi.**

Co to jest wyrażenie poprawne w języku cyfrowym? Z semantycznego punktu widzenia jest to program opisujący wykonanie jakichś przekształceń symboli zapisanych na taśmie maszyny. A więc jeżeli program opisuje zamierzone przekształcanie symboli wyjściowych i po wykonaniu tego przekształcania maszyna się zatrzyma — program jest zbudowany poprawnie. Jeżeli program nie realizuje zamierzonego przekształcenia, nie jest on poprawny. A więc sprawdzanie poprawności programu wymaga szczegółowego prześledzenia działania maszyny dla każdego symbolu badanego programu. Ponieważ programy takie mogą zawierać tysiące symboli, sprawdzanie poprawności programów nie jest czynnością zbyt prostą.

Podobną sytuację mieliśmy, rozpatrując języki procesów prostych. Wychodząc z pojęcia utworzyliśmy język, który ten proces opisywał. Okazało się potem, że język ten posiadał regularną budowę, tak że można było dla niego ustalić gramatykę, pozwalającą w sposób formalny tworzyć lub sprawdzać programy procesów prostych. Czy możliwe jest stworzenie gramatyki języka cyfrowego? Sprawa ta nie była do tej pory dokładnie badana, wydaje się jednak, że nie. Co możemy na przykład powiedzieć na podstawie jedynie struktury o poprawności następującego programu:

$$5 + 8 - 9 - 4 \cdot 5 \; / \; !$$

Oczywiście nawet w tym najprostszym przykładzie występują pewne prawidłowości strukturalne. Na przykład na końcu programu musi się znajdować wykrzyknik; pozostałe symbole są na przemian symbolami operacji i adresami. Jednakże poprawność programu istotnie zależy od adresów w nim występujących. Jakie powinny być zależności w programie poprawnym między adresami? Dla działań arytmetycznych zależności takie można podać. Jednakże maszyna cyfrowa może wykonywać zupełnie dowolne operacje na symbolach i w takim ogólnym przypadku podanie jakichś prawidłowości strukturalnych wydaje się niemożliwe.

Z drugiej strony, jeżeli mamy na przykład program zbudowany poprawnie dla określonego rozmieszczenia danych początkowych na taśmie i jeżeli program pozostawimy bez zmiany, a wszystkie dane umieścimy w innym miejscu na taśmie, czy program pozostanie poprawny? Oczywiście nie. Nie będzie on bowiem realizował założonego zadania. Alfabet rzeczywistych maszyn cyfrowych jest znacznie bogatszy niż ten, który podaliśmy tutaj, tak że występuje szereg dalszych komplikacji przy próbie formalnego określenia poprawności programu. Np. poprawność programu może zależeć od położenia nie tylko danych na taśmie, ale również od położenia samego programu na taśmie. Ten sam program, poprawny przy jednym położeniu na taśmie, po przeniesieniu go bez żadnych zmian w inne miejsce taśmy przestaje być poprawny.

Dla języka tego, przynajmniej dotychczas, gramatyki nie znaleziono. Dlatego posługiwanie się nim wymaga dokładnej znajomości maszyny cyfrowej, realizującej ten język.

Warto zwrócić uwagę na różnice między językami automatów skończonych a językami maszyn cyfrowych. Wyrażenie zapisane na taśmie automatu skończonego było poprawne, jeżeli automat po jego przeanalizowaniu zatrzymał się lub jeżeli to wyrażenie mogło zostać wyprodukowane przez odpowiedni automat. W maszynach cyfrowych sytuacja jest całkiem inna. Z poprawnością programu w maszynie cyfrowej wiąże się wykonanie określonego zadania, polegającego na przekształcaniu symboli.

Języki automatów skończonych posiadają gramatykę określoną strukturą automatu, ale są pozbawione znaczenia. Inaczej przedstawia się sprawa z językami maszyn cyfrowych: nie posiadają one ściśle określonej struktury gramatycznej, natomiast ich znaczenie jest dokładnie określone.

Języki maszyn cyfrowych są najmniej zbadane spośród języków sztucznych. Wydaje się, że dokładne ich poznanie może mieć znaczenie nie tylko dla maszyn matematycznych, ale i dla wielu innych dziedzin.

ZAKOŃCZENIE

Z poruszanym przez nas problemem formalizacji gramatyki wiąże się zastosowanie maszyn matematycznych do tłumaczeń z języka na język. Kilka lat temu z mechanicznym tłumaczeniem wiązano duże nadzieje. Dziś sądy dotyczące zastosowania maszyn matematycznych w tej dziedzinie są daleko ostrożniejsze, jeżeli nie pesymistyczne. W świetle uwag, które uczyniliśmy na temat formalizacji gramatyk, nie jest to niespodzianką. Mechaniczne tłumaczenie wymaga bowiem dokładnego sprecyzowania gramatyki dwu języków i dopiero wtedy można mówić o użyciu maszyny zamiast tłumacza. Matematyczne ujęcie całej gramatyki jakiegoś języka naturalnego dziś nie jest jeszcze możliwe. Zresztą nawet najdoskonalsze zmatematyzowanie gramatyki nie jest wystarczające do wykonania poprawnego tłumaczenia. Tłumaczenie dzieła literackiego wymaga znajomości wielu innych dodatkowych czynników, jak na przykład znajomości epoki, tematu, stylu itp. W takim przypadku zastosowanie maszyny, niezależnie od stopnia sprecyzowania gramatyki, niewiele pomoże.

Istnieje jednak szereg dziedzin, w których używany język jest stosunkowo ubogi, prymitywny. Np. w podręcznikach technicznych, matematycznych czy w innych dziedzinach nauki stosowane jest dość znormalizowane słownictwo; zdania mają niezbyt skomplikowaną budowę, a często nawet w różnych podręcznikach, w tych samych okolicznościach, używane są identyczne zwroty. W takich przypadkach poprawne tłumaczenie może być wykonane przy stosunkowo słabej znajomości języka.

Nasunęło to myśl opracowania słownika i gramatyki w ten sposób, aby osoba nie znająca zupełnie obcego języka mogła w prostszych przypadkach dokonywać poprawnych tłumaczeń w sposób całkowicie mechaniczny. Wiemy bowiem, że zastąpienie wszystkich słów w tłumaczonym zdaniu odpowiadającymi im słowami w innym języku nie daje jeszcze poprawnego tłumaczenia. Konieczne są ponadto odpowiednie zabiegi „kosmetyczne", które nadają poszczególnym słowom właściwą formę gramatyczną, a całemu zdaniu — poprawną składnię. Powstał więc problem przedstawienia gramatyki, przynajmniej w pewnym zakresie, w postaci reguł pozwalających w sposób całkowicie automatyczny nadać zdaniu przetłumaczonemu, tylko przy użyciu słownika, poprawną postać, zgodną z regułami gramatyki danego języka. Zadanie to zostało częściowo rozwiązane w latach trzydziestych. Metoda ta nie znalazła jednakże praktycznego zastosowania, okazało się bowiem, że aby uzyskać tu jakąś użyteczność, należałoby zastosować zbyt dużą liczbę reguł tłumaczenia, przy czym reguły te musiałyby mieć bardzo skomplikowaną postać.

Do spraw tych wrócono ponownie w latach pięćdziesiątych w związku z maszynami matematycznymi. Okazało się, że maszyny pozwalają zastosować dostatecznie dużo skomplikowanych reguł tłumaczenia, tak że cała powyższa koncepcja niespodziewanie nabrała znaczenia praktycznego.

W pamięci maszyny jest zapisany w odpowiedni sposób słownik oraz reguły tłumaczenia. Do maszyny wprowadzane jest tłumaczone zdanie. Maszyna najpierw wszystkie słowa zastępuje według słownika słowami w innym języku, a następnie w tak otrzymanym zdaniu, które oczywiście nie jest poprawne, analizuje słowo po słowie i postępując według zapisanych w niej reguł, nadaje słowom odpowiednią postać gramatyczną, a także odpowiednią kolejność.

Im większy słownik posiada maszyna i im większy zakres gramatyki został w maszynie uwzględniony w postaci reguł tłumaczenia, tym bardziej skomplikowane teksty mogą być za pomocą maszyny tłumaczone.

W wielu ośrodkach badawczych pracuje się więc nad tym, aby jak największą część gramatyki ująć w precyzyjne reguły; pozwoliłoby to bowiem na zwiększenie praktycznej przydatności maszyn w tej dziedzinie. Sprawa ta jednak nie jest tak prosta, jakby to się w pierwszej chwili wydawało. Okazuje się bowiem, że nawet nowoczesne maszyny matematyczne są do tych celów za małe. Uwzględnienie w maszynowym tłumaczeniu dostatecznie dużego słownika i dużej porcji gramatyki wymaga bardzo dużych maszyn, jest zatem kosztowne i nieekonomiczne. Pojawiają się więc propozycje, aby maszyny stosować tylko do wstępnego tłumaczenia, a ostateczną adiustację tekstu pozostawić człowiekowi.

Jakie jest więc znaczenie i perspektywy mechanicznego tłumaczenia? Niewątpliwie zakres stosowania maszyn w tej dziedzinie jest ograniczony, chociaż można przypuszczać, że w niektórych dziedzinach mechaniczne tłumaczenie będzie miało w przyszłości praktyczne znaczenie. Myślę jednak, że maszyny mogą tu pośrednio odegrać doniosłą rolę. Po pierwsze, zastosowanie maszyn do tłumaczenia w znacznym stopniu zintensyfikowało studia nad formalizacją gramatyki, po drugie — być może okazałoby się celowe znormalizowanie w niektórych dziedzinach języka tak, aby można było w nich stosować maszyny do tłumaczenia.

*

Ciekawym przykładem języka jest muzyka. Co to jest zdanie muzyczne? Jaka jest jego struktura? Jak wygląda gramatyka tego języka? Jaki jeszcze inny niż obecny zapis muzyczny można stosować do zapisywania sekwencji dźwięków? Takie i wiele innych pytań nasuwa się w związku z tym tematem. Od strony gramatycznej język muzyki, o ile mi wiadomo, nie był do tej pory badany. Stosowano tu jedynie badania statystyczne, polegające na analizie częstości występowania określonych dźwięków i na tej drodze próbowano syntetyzować utwory muzyczne za pomocą maszyn matematycznych. Z reprezentowanego tu punktu widzenia próby tego rodzaju nie dają jednak obrazu struktury gramatycznej utworów muzycznych.

Warto zwrócić uwagę, że gramatyka kompozycji muzycznych może mieć charakter zbliżony do gramatyki automatów skończonych. Znaczy to, że jest ona określana nie przez podanie reguł przekształcania symboli, lecz przez strukturę pewnego automatu, produkującego bądź analizującego dźwięki. Co jest tym automatem dla człowieka? Oczywiście ucho ludzkie. Przez „ucho" rozumiem tu ucho z odpowiednim fragmentem mózgu, analizującym przychodzące dźwięki. A więc gramatyka muzyki może być po prostu określana anatomią organów słuchu. Wyjaśnienie struktury muzyki może pomóc w zrozumieniu mechanizmu słyszenia i odwrotnie. Innym zagadnieniem jest rozumienie muzyki. Czy z utworem muzycznym może się kojarzyć jakiś sens? Czy też chodzi tu tylko o formalną strukturę następujących po sobie dźwięków?

Wszystkie rozważane przez nas do tej pory języki były liniowe. Czy liniowość jest nieodzowną własnością języka? Oczywiście nie. Można sobie z łatwością wyobrazić języki „płaskie“. Przykładem języka na płaszczyźnie jest mapa. Czy można podać formalne kryteria poprawności mapy? Czy mapy posiadają określoną strukturę? Jaka jest gramatyka map? Można podać pewne reguły poprawności formalnej map. Rysunek, który tych reguł nie spełnia, nie jest mapą. Mapa więc to zdanie poprawne w odpowiednim języku.

Podobne pytania można zadać odnośnie do dowolnego obrazu. W każdym obrazie, niezależnie od konwencji artystycznej, w jakiej jest namalowany, istnieją pewne prawidłowiści formalne, tworzące gramatykę reprezentowanego kierunku artystycznego. Mogą one być trudne do sformułowania w języku naturalnym. U podstaw ich mogą leżeć również fakty dotyczące mechanizmu widzenia. Sytuacja ta może być podobna do sytuacji w automatach skończonych. Obraz jest zbudowany poprawnie z formalnego punktu widzenia, jeżeli określona „maszyna“ po jego percepcji przechodzi do przewidzianego stanu. Prawidłowość formalna obrazu może więc zależeć od psychologii widzenia, od struktury mechanizmu postrzegania obrazów przez człowieka.

Malarstwo jest więc również pewnym językiem, w którym zdaniami są obrazy. Formalna analizą tego języka wydaje się bardzo trudna, jeżeli nie niemożliwa.

Język jest więc pojęciem bardzo szerokim i dokładniejsze zbadanie jego struktury może mieć duże znaczenie dla wielu dziedzin. Obecny stan badań w tym zakresie niewątpliwie należy uznać za początkowy. Być może w najbliższych latach wiele wspomnianych tu problemów zostanie już rozwiązanych.

SPIS RZECZY

INSTYTUT MATEMATYCZNY • W.M. • U.W.

Biblioteka WMIM UW
1094024697