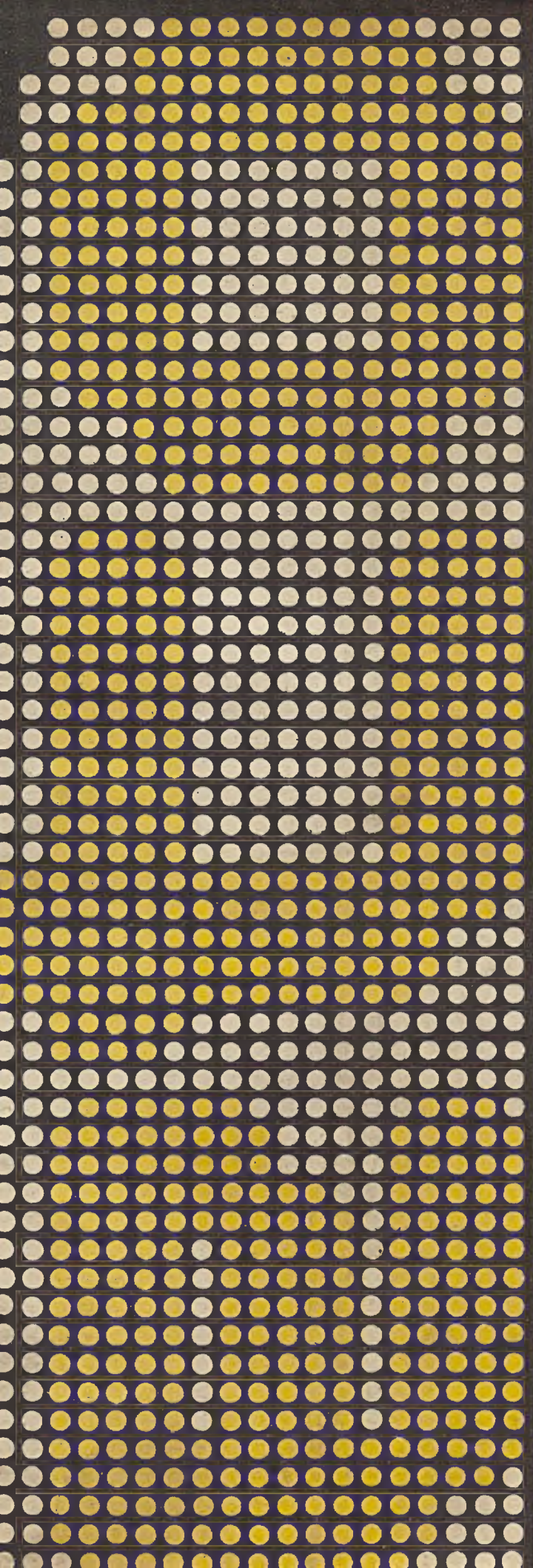


elektroniczna technika obliczeniowa

NOWOŚCI NR 1 1979

ZJEDNOCZENIE
PRZEMYSŁU
AUTOMATYKI
I APARATURY
POMIAROWEJ „MERA"

INSTYTUT MASZYN
MATEMATYCZNYCH
BRANŻOWY
OŚRODEK INTE

ELEKTRONICZNA TECHNIKA OBLICZENIOWA

N O W O Ś C I

DWUMIESIĘCZNIK

Rok XIII Nr 1 1974

S p i s t r e ś c i

Od Redakcji 3

Mgr inż. Jerzy DAŃDA. Niektóre problemy następnych generacji komputerów. Część III 5

Zagadnienia rozwoju maszyn matematycznych. Dyskusja w Instytucie Maszyn Matematycznych 29

Perspektywy rozwoju architektury komputerów. Oprac. mgr inż. A. Kojemski 57

Mgr inż. Tomasz RAWIŃSKI: Rozszerzenie koncepcji Jednolitego Systemu Elektronicznych Maszyn Cyfrowych. Wprowadzenie EMC ODRA do JS EMC 77

JAK PRACUJE TRANSLATOR

Mgr Tomasz KRAWCZYK: Metody analizy składni oparte na relacjach pierwszeństwa 83

Wydaje

INSTYTUT MASZYN MATEMATYCZNYCH

B r a n ż o w y O ś r o d e k I n f o r m a c j i

N a u k o w e j T e c h n i c z n e j i E k o n o m i c z n e j

Adres redakcji: 02-078 Warszawa, ul. Krzywickiego 34
tel. 28-37-29 lub 21-84-41 w. 431

Druk IMM z. 116/74 n. 1200 egz. pap. offset. kl. III g. 80 . W-121. INDEKS-35804.

O d R e d a k c j i

Kraj nasz rozpoczął produkcję maszyn III generacji, produkowane są również nowoczesne urządzenia peryferyjne. Wkrótce biura konstrukcyjne zaplecza badawczo-rozwojowego przemysłu zaczną pracować nad komputerami i urządzeniami, które będziemy zaliczać do czwartej generacji. Zespoły naukowców i teoretyków podjęły już prace podstawowe, których wyniki będą mogły być wykorzystane przy następnych generacjach komputerów. Aby kształt przyszłych maszyn, ich właściwości użytkowe, wygoda posługiwania się nimi była jak najbliższa oczekiwaniom - konstruktorzy sprzętu i twórcy oprogramowania muszą nawiązać jak najściślejszą współpracę. Opinie takie powtarzają się jak refren we wszystkich wypowiedziach i artykułach, krajowych i zagranicznych, które dotyczą ogólnych zagadnień ukierunkowywania rozwoju (ewolucji technologicznej) następnych generacji komputerów.

Jesteśmy przekonani o konieczności ścisłej współpracy wszystkich grup osób, związanych z postępem informatyki: jej użytkowników, twórców oprogramowania, konstruktorów sprzętu, technologów informatyki i twórców nowych koncepcji zajmujących się pracami podstawowymi tej dziedziny. Dlatego staramy się przez dobór artykułów i pozyskiwanie coraz to nowych autorów, ktorzy interesują się współtworzeniem przyszłych koncepcji i produktów informatycznych - przyczyniać się w miarę swych możliwości do nawiązywania i umacniania takiej współpracy. Świadczyć o tym może profil naszego pisma, szczególnie aktywnie w tym kierunku kształtowany w ciągu ostatnich dwóch lat.

Zaczęliśmy zamieszczać coraz więcej artykułów z dziedziny ogólnych zagadnień programowania komputerów; z braku własnych prac w tym zakresie - posługujemy się także opracowaniami najlepszych w dziedzinie zagadnień ogólnych rozwoju informatyki artykułów zagranicznych. Próbujemy również organizować dyskusje na aktualne tematy, zapraszając na nie najlepszych fachowców, chcących z nami w ten sposób współpracować. Przebieg jednej z takich dyskusji publikujemy w tym zeszycie wraz z artykułem, którego dyskusja dotyczyła. Zamierzeniem redakcji jest, aby najbliższe lata naszej działalności oznaczały jeszcze dalsze udoskonalenie funkcji, które chcemy pełnić. A pragniemy stworzyć na łamach naszego czasopisma możliwość wypowiadania ciekawszych pomysłów i koncepcji merytorycznych, które mogą wspomóc rozwój informatyki w naszym kraju.

Aby sprawnie pełnić taką funkcję - coraz pilniej potrzebujemy sprzężenia zwrotnego. Musimy wiedzieć, czy pismo spełnia oczekiwania, a przynajmniej - czy od tych oczekiwań zbyt daleko nie odbiega. Prosimy - przekażcie nam swoje opinie o naszej działalności.

Od tego roku zaczynamy wydawać zeszyty co dwa miesiące, przy niezmienionej objętości zeszytu. Artykuły, które poruszają interesujące treści, bliskie naszemu profilowi, staramy się zamieszczać niezależnie od ich objętości, zdając sobie sprawę, że w pewnych dziedzinach nadmierne restrykcje dotyczące zwięzłości wypowiedzi powodują, że mogą w niej być zawarte tylko niektóre myśli, nie zawsze tworzące spójną całość.

Od wszystkich Czytelników oczekujemy więc opinii, a mamy nadzieję, że wielu Czytelników stanie się naszymi Autorami.

ETO NOWOŚCI nr 1/1974

Mgr inż. Jerzy DAŃDA
Instytut Maszyn Matematycznych

681.322.001.13

NIEKTÓRE PROBLEMY NASTĘPNYCH GENERACJI KOMPUTERÓW
Część III

1. Wprowadzenie

Niniejsza część opracowania zawiera, zgodnie z zapowiedzią zawartą w II części, pewną propozycję architektury systemów wielodostępnych. Jest ona oparta na wnioskach, wynikających z krytycznego spojrzenia na dotychczasowy ewolucyjny rozwój architektury systemów liczących. Analiza taka została przeprowadzona w II części tego opracowania, gdzie podano również zarys architektury systemu wielodostępnego, który nie byłby obciążony (przynajmniej niektórymi) wadami dotychczasowych rozwiązań. Krytykowane uprzednio niedostatki istniejących systemów, to zachowywanie przez cały czas rozwoju komputerów zasady, że właściwe wykorzystanie sprzętu systemu liczącego uzyskuje się na drodze przełączania jednej, potężnej (w przypadku systemów wielodostępnych) jednostki centralnej na obsługę wielu użytkowników. Konsekwentne trzymanie się przez konstruktorów tej zasady powodowało nadmierne obciążenie oprogramowania takich systemów funkcjami związanymi z zarządzaniem pracą systemu, a częste przełączanie z jednego użytkownika na drugiego obciążało system w znacznej mierze funkcjami pamiętania stanu procesu w momencie jego przerwania oraz przekazywaniem informacji między pamięciami poszczególnych poziomów.

W literaturze dotyczącej architektury systemów liczących brak jest (jak dotychczas) pozycji zawierających ilościową analizę ogólnie pojętych kosztów, jakie się ponosi rozwiązując większość złożonych problemów sterowania pracą dużego systemu liczącego np. wielodostępnego przede wszystkim na drodze softwareowej. Jednakże niewątpliwe sukcesy, jakie odnoszą firmy Control Data Corporation oraz Burroughs (obie stosujące w swoich maszynach-gigantach na szeroką skalę pracę wielopro-

cesorową oraz sprzętową realizację nawet złożonych funkcji) nakazują zwrócić na ten kierunek szczególną uwagę. Można przypuszczać, że bardzo szybko postępująca obniżka kosztów układów scalonych średniej i wielkiej integracji oraz obniżające się koszty pamięci półprzewodnikowych spowodują, że rozwiązania, które można było stosować ekonomicznie w latach sześćdziesiątych i siedemdziesiątych tylko w odniesieniu do maszyn-gigantów - w latach osiemdziesiątych będą mogły być stosowane w systemach znacznie bardziej powszechnego użytku.

Przedstawiając propozycję architektury systemu liczącego, który mógłby być realizowany w przyszłości, warto wymienić explicite założenia dotyczące stanu technologii, który warunkuje ekonomiczną realizację takiego systemu. Aczkolwiek dopiero dokładna analiza techniczno-ekonomiczna oraz znacznie bardziej szczegółowe opracowania projektowe mogłyby dać odpowiedź, czy prezentowana propozycja w ogóle jest warta realizacji - dla wygody czytelnika podamy najważniejsze założenia; pominiemy te, które wynikają z krytyki dotychczasowej ewolucji architektury i te omówione już w poprzedniej części opracowania.

Założymy, po pierwsze, że stan technologii układów scalonych osiągnie w okresie, którego dotyczy omawiana propozycja, stan tak wysoki, że koszty realizacji procesora, odpowiadającego swoimi własnościami funkcjonalnymi współczesnym minikomputerom, będą wynosiły 1 - 10 dolarów. Wg opinii C. Fostera [1] [1] taki stan rozwoju technologii półprzewodnikowych zostanie osiągnięty w USA w końcu lat osiemdziesiątych.

Drugim założeniem, wynikającym częściowo z pierwszego, będzie przyjęcie znacznej redundancji, na poziomie procesorów. Celem wprowadzenia tej redundancji będzie głównie zmniejszenie nadmiernego obciążenia oprogramowania takiego systemu; obciążenie to związane jest z przebiegiem wielu procesów roboczych i systemowych sterujących w jednym, czy najwyżej w niewielkiej liczbie procesorów. Wydaje się, że rewelacyjnie niski koszt sprzętu w porównaniu z obecnym jego poziomem pozwoli na globalnie ekonomiczną realizację systemów liczących nawet w warunkach, jeśli wykorzystanie sprzętu będzie wynosiło 10 - 20%, co oznacza, że przy normalnej eksploatacji pracuje tylko co dziesiąty czy co piąty procesor roboczy. W jakim stopniu redundancja będzie wykorzystywana równocześnie do zapewnienia dużej niezawodności działania systemu jest oddzielną kwestią, która nie będzie szczegółowo rozpatrywana w tej części opracowania. Nie ulega jednakże wątpliwości, że założenie o

---

[1] Zob. również str. 57

znacznej nadmiarowości sprzętu pozostawi konstruktorom takich systemów znaczną swobodę w zapewnieniu dużej niezawodności pracy systemów.

Trzecim założeniem, niezwykle istotnym - bo umożliwiającym ekonomiczną realizację systemów liczących o architekturze omawianej w tym odcinku, jest przyjęcie możliwości opracowania przez technologów układów umożliwiających szybkie przesyłanie informacji między dowolnymi podsystemami. Inaczej mówiąc - aby realizować takie systemy - konstruktorzy muszą dysponować elementami, które można łączyć sposobem "każdy z każdym". Obecnie najpowszechniej stosowanym sposobem jest łączenie elementów za pomocą "szyn" lub "magistrali" (buses). Ten ostatni sposób - niewątpliwie ekonomiczny - przy założeniu, że elementami łączącymi są podzespoły elektroniczne - nie będzie mógł być stosowany w omawianych systemach.

Wydaje się, że najłatwiej będzie znaleźć potrzebny typ elementów w grupie technik i technologii optoelektronicznych. Jedno z możliwych rozwiązań, jeszcze może niezbyt doskonałych, lecz wskazujących jeden z możliwych kierunków poszukiwań, zawarte jest w polskim patencie [2].

Czwarte założenie, że pamięć systemu jest złożona z wielu modułów o jednakowej pojemności - ok. 2-8 kilobajtów, wyposażonych w oddzielne układy adresowania oraz wprowadzania i odczytywania informacji - przyjmiemy głównie dla jasności i zwięzłości opisu; ma ono również swe uzasadnienie w obserwowanej od lat obniżce kosztów pamięci. Kwestie, czy będzie to jedyna pamięć systemu, czy też będzie on wyposażony dodatkowo w pamięci pojemniejsze, lecz o dłuższym czasie dostępu - nie będą w tym odcinku opracowania szczegółowiej rozpatrywane. Wydaje się bowiem, że konieczność zorganizowania odpowiednich przesłań między ewentualną wielopoziomową pamięcią takiego systemu nie powinna nastręczać bardzo dużych trudności i będzie mogła być rozwiązana w sposób zgodny z ogólną koncepcją organizacji. Czytelnik, który uzna, że przyjęcie takiego założenia wydaje się nieuzasadnione może znaleźć pewne argumenty za takim rozwiązaniem w odcinku 6 i następnych. Można też traktować te odcinki jako kolejną "warstwę" krytyki dotychczasowych rozwiązań, w tym przypadku dotyczącą przede wszystkim niedoskonałości wymiany informacji między konstruktorami podzespołów komputerów a projektantami ich architektury logicznej.

2. Opis struktury systemu

Ogólny schemat struktury systemu wielodostępnego z dynamiczną strukturalizacją przedstawiony jest na rys. 1. Lokalne procesy robocze[1] realizowane sę przez minikomputery oznaczone $MINI_1$,... $MINI_k$ i wyposażone w różne urządzenia peryferyjne oraz pamięci pomocnicze w zależności od klasy procesów roboczych, które w nich są najczęściej wykonywane. Każdy z minikomputerów lokalnych połączonych jest linią transmisji danych (indywidualną lub przełączaną) z mini- lub mikrokomputerem, umieszczonym w kompleksie centralnym. Komputery te będziemy nazywali "dublerami". Ogólnie biorąc funkcje dublera polegają na "reprezentowaniu" użytkownika wewnątrz kompleksu centralnego; opiszemy je bardziej szczegółowo przy okazji opisu funkcji systemu.

Kompleks centralny złożony jest z "zasobów elementów" trzech rodzajów oraz systemu sterującego. Pierwszy rodzaj zasobów, to zbiór procesorów roboczych, na rys. 1 oznaczonych P_1,... P_9. Każdy z nich jest wyspecjalizowany funkcjonalnie przez zastosowanie odpowiedniej listy rozkazów w kierunku wykonywania danej klasy procesów roboczych. Charakterystyczną cechą tych procesorów jest przechowywanie informacji o procesie, realizowanym przez procesor - w pamięciach z nim połączonych oraz w dublerze. Umożliwia to praktycznie natychmiastowe odłączanie procesora od zespołu przydzielonych mu modułów pamięciowych, bez długotrwałego rejestrowania stanu procesu roboczego, który pozostaje w pamięci oraz w dublerze. Jest to rozwinięcie i przeniesienie na sprzęt zasady wykorzystywanej w oprogramowaniu i znanej jako koncepcja "pure procedure". Analogicznie procesory tego typu można byłoby nazwać "czystymi procesorami".

Drugi rodzaj zasobów to zbiór modułów pamięciowych o natychmiastowym dostępie. Każdy moduł wyposażony jest w indywidualne układy adresowe, umożliwiające wybranie dowolnej informacji oraz układy interface'u umożliwiające podłączenie modułu za pomocą elementów łączących do dowolnego procesora. Moduły pamięciowe oznaczone są na rys. 1 symbolami M_1,... M_{16}.

---

[1] Zob. "Niektóre problemy następnych generacji komputerów" cz. II, ETO Nowości 1/1973, gdzie określono niektóre pojęcia wykorzystywane w niniejszym odcinku pracy.

Trzeci rodzaj zasobów stanowi zbiór elementów łączących, oznaczonych $Ł_1 \ldots Ł_{14}$. Elementy te służą do łączenia ze sobą procesorów i modułów pamięciowych. Każdy z elementów łączących jest sterowany przez minikomputer strukturalizacji określający, które układy kompleksu centralnego mają być połączone danym elementem. Połączenie następuje praktycznie natychmiast po skierowaniu do elementu łączącego sygnału sterującego i trwa do chwili podania sygnału kasującego połączenie. Przyjmiemy, zgodnie z tym co powiedziano we wprowadzeniu, że elementy łączące są realizowane technikami optoelektronicznymi.

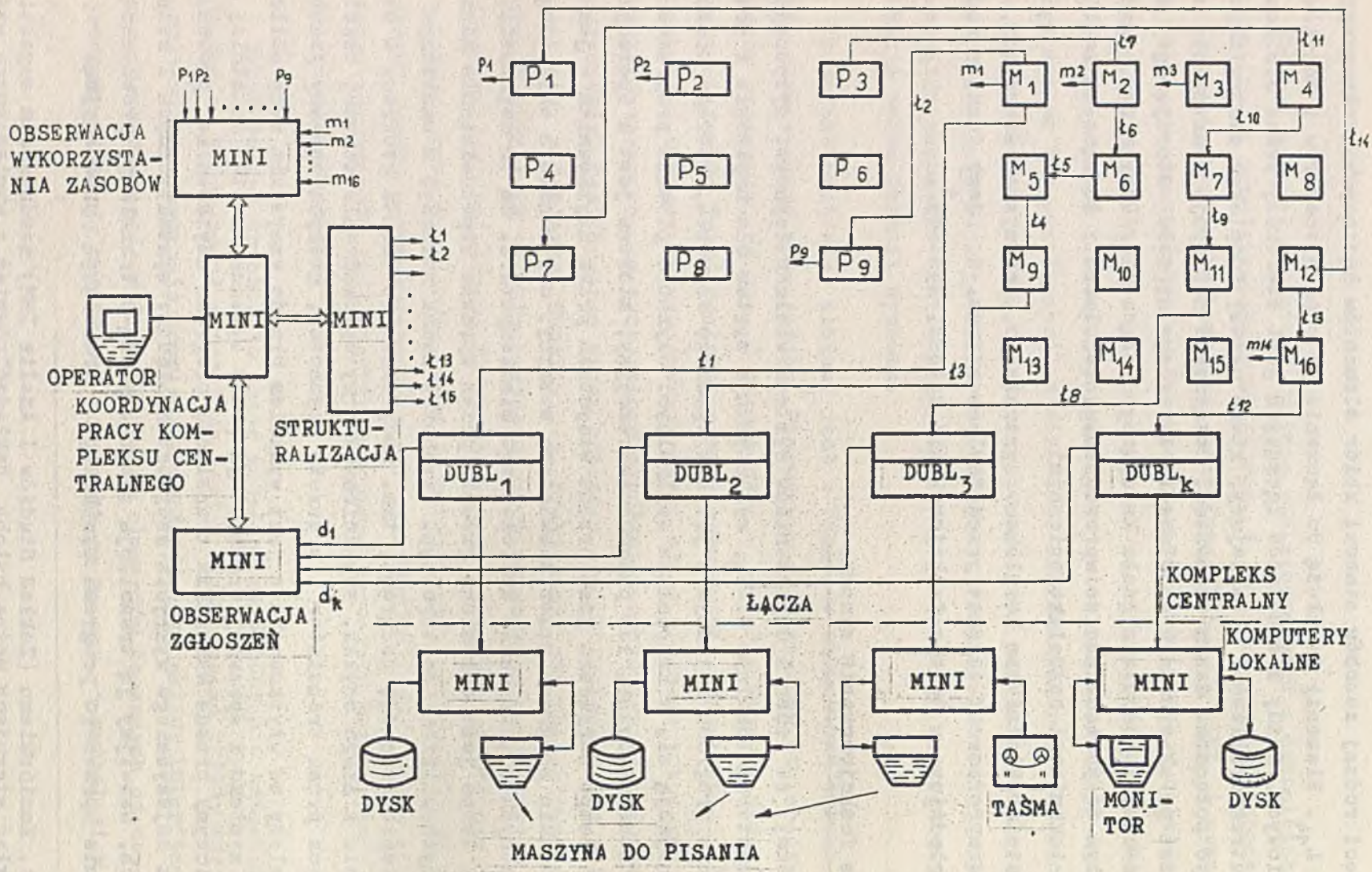
Ostatnim składnikiem kompleksu centralnego jest system sterujący, którego uproszczony schemat przedstawiono na rys. 1. Jego funkcje zostaną uwzględnione w opisie funkcjonowania całości systemu[1].

3. Opis funkcjonowania systemu

Załóżmy, że jeden z użytkowników systemu wielodostępnego, dysponujący minikomputerem lokalnym $MINI_1$, wykorzystuje system dla napisania i uruchomienia programu w jednym z języków wyższego szczebla, którego translatory znajdują się w kompleksie centralnym. Czynność pisania programów w sposób konwersacyjny, lub pseudokonwersacyjny, złożona jest z operacji, których tempo narzucone jest przede wszystkim przez użytkownika - jak np. wprowadzanie programu przez klawiaturę maszyny, czy monitora ekranowego podłączonych bezpośrednio do lokalnego minikomputera. Są to czynności, których tempo jest określone głównie przez sprawne współdziałanie maszyny z użytkownikiem; do tego typu czynności można zaliczyć w omawianym przykładzie redakcję programu, tzn. modyfikowanie tekstu programu w celu usunięcia z niego omyłek, popełnionych przy wprowadzaniu tekstu, błędów, wykrytych przez translator i wreszcie czynności, których sprawny przebieg mało zależy od użytkownika a jest wynikiem przede wszystkim mocy obliczeniowej systemu i sprawnej organizacji pracy systemu. Do tej ostatniej grupy zaliczymy przede wszystkim translację programu użytkownika, prowadzoną dla jak najszybszego wykrycia wszystkich błędów: syntaktycznych i semantycznych. Nazwijmy ją translacją diagnostyczną a translację prowadzącą do uzyskania sprawnego programu wynikowego - translacją produkcyjną.

---

[1] Mgr A. Kamińskiemu (Zakład Studiów i Analiz IMM) zawdzięczam sugestię, że układ sterujący można byłoby "składać" również z wymienionych wyżej zasobów elementów, uzyskując kompletną jednolitość podejścia. Zbadanie konsekwencji takiego rozwiązania nie zostało jeszcze wykonane. Niewątpliwie należałoby rozwiązać zagadnienie "pierwotnej strukturalizacji systemu sterującego" przy uwzględnianiu tej sugestii.



Rys. 1. Struktura systemu wielodostępnego z dynamiczną strukturalizacją

Zgodnie z przyjętą ideą podziału procesów na lokalne i centralne, omówioną już wcześniej, do lokalnych procesów roboczych w naszym przykładzie zaliczymy "obsługę użytkownika" przez lokalny minikomputer (przy wprowadzaniu przez niego programu do pamięci) oraz pomoc przy redakcji programu. Do procesów centralnych zaliczymy natomiast translację diagnostyczną oraz produkcyjną. Samo wykonywanie programu może być natomiast procesem centralnym lub lokalnym - będzie to zależało od wielu czynników,np.objętości programu wynikowego,ilości interakcji z użytkownikiem koniecznych w czasie biegu programu wynikowego "run time" , wymaganej pojemności pamięci dla wyników, czasu pracy itp.

Taki podział funkcji na lokalne i centralne powinien w omawianym przypadku zapewnić właściwe wykorzystanie zasobów systemu przy stosunkowo krótkich czasach reakcji systemu, dostosowanych do oczekiwań użytkownika.

4. Obsługa użytkownika w fazie wprowadzania i redagowania programu

Ogólną zasadą, która powinna być stosowana przy projektowaniu procesów roboczych i sterujących lokalnych, dla realizacji obsługi użytkownika powinno być "ukrywanie" przed użytkownikiem wszystkich szczegółów, które są dla niego nieinteresujące z punktu widzenia jego głównego zadania, tzn. napisania i uruchomienia programu. Wychodząc z tej zasady powiemy, że jedynymi czynnościami, których wykonania system może żądać od użytkownika to:

- podanie nazwiska i/lub numeru identyfikacyjnego,
- podanie języka, w którym będzie pisał program.

Wszystkie inne dane, które są niezbędne systemowi przy dalszej współpracy z użytkownikiem powinny być uzyskiwane od użytkownika przez zadawanie przez system pytań, w logiczny sposób wiążących się z zadaniem, które jest wykonywane. System musi unikać zadawania pytań, których celowość budzi wątpliwości użytkownika, lub podania z góry różnych danych: jak np. wymagana objętość pamięci, estymowany czas trwania translacji lub przybliżona liczba wierszy wydruku, które często są wymagane przez system operacyjny przy pracy wsadowej a ich niewłaściwe określenie oznacza często konieczność powtarzania całego zadania, gdyż zostało ono

przedwcześnie przez system "zlikwidowane" ("killed", "aborted"). Oczywiście, przy systemach konwersacyjnych niebezpieczeństwo takie zwykle użytkownikowi nie grozi, jednak omawiany system jest o tyle szczególny, że praca kompleksu centralnego przebiega w sposób pseudowsadowy; złe zaprojektowanie procesów obsługi użytkownika mogłoby więc znacznie umniejszyć wygodę posługiwania się systemem.

Poprzednio omówiona zasada sprowadza się do zmniejszenia liczby zbędnych pytań kierowanych do użytkownika przez system, względnie unikania zadawania tych pytań w niewłaściwym z punktu widzenia użytkownika czasie. Natomiast przy projektowaniu procesów roboczych obsługi powinna być też stosowana zasada, że wszystkie niezbędne informacje, których użytkownik może sobie życzyć powinny być przez system z wielką gotowością udostępnione. System powinien być maksymalnie "usłużny" w stosunku do użytkownika, jeśli użyć nadmiernie może antropomorfizującego określenia. Do informacji podawanych na życzenie zaliczyć można informację o możliwościach systemu. A więc np. gdy użytkownik rozpoczyna redagowanie programu powinien uzyskać informacje o różnych sposobach identyfikacji zdania, w którym znajduje się błąd. A więc np. przez podanie numeru wiersza tabulogramu lub przez podanie słowa, zaczynającego wiersz, a jeśli mogą być wątpliwości dotyczące jednoznaczności takiego określenia przez podanie błędnego słowa i jego najbliższego kontekstu.

Taki system "usłużnego informowania" szczególnie dogodnie można realizować w przypadku, gdy podstawowym środkiem komunikowania się użytkownika z systemem jest monitor ekranowy. Można wtedy niezależnie od pola ekranu, przeznaczonego na zasadnicze zadanie, wydzielić część ekranu dla przedstawiania przez system "ofert pomocy" w postaci podawania nazw usług, które w danym momencie mogą być przez system świadczone. Taki dynamiczny system oferowania umożliwia uzyskanie bardzo sprawnej komunikacji z użytkownikiem. Np. gdy wykryty zostanie błąd syntaktyczny - system może wyświetlić nie tylko ofertę jak ten błąd usuwać, ale również propozycję opisania na czym on polegał. Ktoś, kto musiał korzystać z systemu, który przy wykryciu błędów syntaktycznych wypisuje tylko mnemotechniczne skróty sygnalizowanych błędów, albo tylko ich kodowe numery - doceni jak tak działający system byłby wygodny. Z drugiej strony jednak użytkownik często korzystający z systemu nauczy się wreszcie znaczeń nawet źle dobranych skrótów mnemotechnicznych i dla niego podawanie długich

objaśnień będzie właśnie odbierane jako niedogodna w pracy "gadatliwość" systemu. Dlatego, aby system można było nazwać ergonomicznie poprawnym musi on tylko proponować różne możliwości informowania, mniej lub bardziej zwięzłe, pozostawiając ich wybór użytkownikowi.

Jednym z warunków umożliwiających realizację omówionej wyżej (w zarysie) obsługi użytkownika w trakcie pisania i redagowania programów jest dysponowanie pojemnymi pamięciami pomocniczymi. Taki werbalny sposób komunikacji systemu z człowiekiem wymaga bowiem przechowywania wielu przygotowanych uprzednio tekstów stanowiących szablony zapytań i odpowiedzi. W tym celu lokalny komputer będzie musiał być wyposażony w odpowiednio pojemną pamięć, np. dyskową.

5. Obsługa użytkownika w fazie translacji i wykonywania programu

Zgodnie z podziałem procesów roboczych na lokalne i centralne, w fazie pisania i redagowania programu użytkownik praktycznie nie kontaktował się z kompleksem centralnym, chociaż system w sposób dla użytkownika niewidoczny podejmował pewne akcje, mające na celu skrócenie czasu reakcji przy równoczesnym możliwie równomiernym obciążeniu linii transmisji danych. Omówimy je krótko, gdyż akcje te stanowią przygotowanie do rozpoczęcia właściwej pracy kompleksu centralnego dla użytkownika, tzn. wykonania translacji diagnostycznej napisanego programu.

W miarę zapisywania programu przez użytkownika w pamięci lokalnego minikomputera - do pamięci dublera, mieszczącego się w kompleksie, przesyłana jest kopia programu użytkownika. Procesem przesyłania zawiaduje dubler, dopasowujący tempo przesyłania bloków informacji z jednej strony do stopnia wykorzystania zajętości linii, z drugiej do tempa, w jakim pojawiają się wprowadzane przez użytkownika wiersze programu i ewentualne poprawki w tekście programu.

W ten sposób w momencie, gdy użytkownik skończył pisać program i zgłasza do kompleksu centralnego zapotrzebowanie na kompilację diagnostyczną - praktycznie nie trzeba tracić czasu na przesłanie informacji - gdyż jest ona już gotowa w pamięci dublera.

Zgłoszenie zapotrzebowania na pracę kompleksu centralnego pojawia się w wydzielonym polu pamięci dublera, przeznaczonym do komunikacji

z kompleksem centralnym. Pole to znajduje się "pod obserwacją" procesu systemowego "obserwacja napływu zgłoszeń obsługi", realizowanego przez wydzielony minikomputer systemu sterującego kompleksu centralnego patrz (rys. 1). Obserwacja ta może być prowadzona na zasadzie "polling" - i w tym przypadku wydzielone pola dublerów mogą być traktowane jako część pamięci (lub rejestrów) tego minikomputera, lub może być rozwiązana przez bardziej tradycyjnie zorganizowany układ przerwań. Przy takiej wersji dubler zgłosiłby przerwanie do minikomputera "obserwacji napływu zgłoszeń obsługi", który pobrałby odpowiednie informacje z pola pamięciowego dublera. Które z tych rozwiązań okaże się korzystniejsze - trudno jest stwierdzić na etapie wstępnego określania funkcji systemu.

Minikomputer "obserwacji zgłoszeń" przekazuje dane uzyskane z dublera do minikomputera "koordynacji pracą kompleksu centralnego" w odpowiedniej, zestandaryzowanej formie. W meldunku tym zawarte są np. dane typu: numer dublera, numer użytkownika, rodzaj (typ) zgłoszenia na obsługę (np. kompilacja diagnostyczna, kompilacja produkcyjna, nazwa języka itp.) Proces "koordynacja pracą kompleksu centralnego" na podstawie tych danych oraz danych przechowywanych w jego pamięci, a dotyczących priorytetów, wynikających z "ważności" danego użytkownika i na podstawie danych przekazanych mu z procesu "obserwacja wykorzystania zasobów" - dokonuje przydziału użytkownikowi zgłaszającemu się przez swojego dublera odpowiedniego procesora roboczego i odpowiedniej do danego zadania liczby modułów pamięciowych. Polecenie to zostaje przekazane minikomputerowi strukturalizacji, który dokonuje odpowiednich połączeń za pomocą elementów łączących. W omawianym przypadku dubler $DUBL_1$ został połączony z modułem pamięciowym M_1 i procesorem P_9 za pomocą elementów łączących $Ł_1$ i $Ł_2$.

Po dokonaniu tych połączeń rozpoczyna się realizacja procesu roboczego, która trwa nieprzerwanie do chwili, gdy pojawi się wewnątrz tego procesu przyczyna, nakazująca jego zakończenie. W przypadku kompilacji diagnostycznej może to być albo zakończenie kompilacji, albo napotkanie takich błędów, które nie mogą być usunięte przez "domniemania" kompilatora. Sygnał o zakończeniu procesu roboczego zostaje przesłany z procesora P_9 drogą p_9 do minikomputera obserwacji wykorzystania zasobów i stąd przez minikomputer koordynacji i strukturalizacji zostaje przesłany sygnał kasujący połączenie $Ł_2$. W ten sposób procesor P_9 zostaje zwolniony i powraca do zbioru zasobów. Moduł pamięciowy M_1 pozostaje związany z du-

blerem $DUBL_1$ dopóki ten nie przekaże wyników tej kompilacji diagnostycznej przez linię transmisyjną do użytkownika, posługującego się lokalnym $MINI_1$. Po zakończeniu transmisji dubler przesyła drogą d_1 sygnał o zakończeniu wykorzystywania zasobów kompleksu centralnego. Wtedy moduł pamięci M_1 oraz element łączący $Ł_1$ również powracają do zbioru zasobów.

W podobny sposób będą przebiegały procesy strukturalizacji zasobów, dające w wyniku wydzielone układy procesorów roboczych, powiązanych z odpowiednią liczbą modułów pamięciowych i dublerami - wywołane zgłoszeniami na obsługę przez kompleks centralny przez innych użytkowników systemu. Procesy te będą przebiegały w sposób uniemożliwiający powstawanie konfliktów polegających na wejściu jednego programu na obszar pamięciowy drugiego; zapewni to kompletna izolacja poszczególnych modułów pamięciowych. Konflikty mogą wynikać wskutek zażądania równoczesnego tego samego typu procesora roboczego przez dwa lub więcej procesów, zgłoszonych przez użytkownika. Konflikty te rozwiązuje proces koordynacji pracą kompleksu centralnego - np. ustawiając te zgłoszenia w kolejkę, z zachowaniem uprzednio określonych priorytetów i przesyłając czekającemu użytkownikowi wiadomość o opóźnieniu w obsłudze, wywołanym ograniczoną mocą przerobową w danej chwili. Inne rozwiązanie, zmniejszające czas oczekiwania w takich sytuacjach, może być uzyskane przez wprowadzenie do kompleksu centralnego pewnej liczby procesorów roboczych uniwersalnych, których specjalizacja może następować przez wymianę ich list mikrorozkazowych w kierunku wykonywania określonego typu przetwarzania.

Opiszemy teraz jeszcze jedną funkcję dublera, która będzie wykonywana równocześnie z przebiegiem procesu kompilacji wykonywanym przez procesor P_9. Otóż o ile w przypadku redagowania programu, z wykorzystaniem procesora lokalnego $MINI_1$ przyjmowaliśmy, że szablony zapytań i odpowiedzi służących do konwersacyjnego informowania użytkownika o błędach i sposobach ich usunięcia przechowywane były w lokalnej pamięci dyskowej - o tyle w tym przypadku racjonalne jest przyjęcie założenia, że szablony tego typu są przesyłane do lokalnego minikomputera z kompleksu centralnego, tylko na czas danej sesji uruchamiania programu. W przeciwnym przypadku musielibyśmy przyjąć, że w lokalnych pamięciach są przechowywane teksty komunikatów diagnostycznych bardzo wielu translatorów, znajdujących się w kompleksie centralnym.

Tak więc przyjmiemy, że w tym samym czasie, kiedy procesor P_9 wykonuje kompilację dla użytkownika $MINI_1$, jego dubler $DUBL_1$ zatrudniony jest przesyłaniem odpowiednich tekstów do pamięci dyskowej $MINI_1$. Mogą one być potrzebne do udzielania wyjaśnień użytkownikowi, w trakcie interpretacji wyników kompilacji diagnostycznej.

Ten przykład może służyć za ilustrację możliwości, które uzyskujemy wprowadzając nadmiarowy sprzęt: staje się możliwe równoczesne wykonywanie czynności różnego typu i sterowanie tymi czynnościami nie musi obciążać nadmiernie układu sterowania kompleksu centralnego. Uwaga: należy zaznaczyć, że na rys.1 nie przedstawiono podsystemu pamięci pomocniczych kompleksu centralnego i systemu sterującego ich pracą, z którymi to układami będzie się w omawianym przykładzie komunikował $DUBL_1$ - przy transmisji danych tekstowych, wiążących się z pracą kompilatora diagnostycznego.

Zupełnie podobnie będzie przebiegała praca przy kompilacji produkcyjnej; różnica będzie polegała na podłączeniu innego procesora roboczego, wyspecjalizowanego w kompilacji produkcyjnej. Program wynikowy, jak już wcześniej powiedzieliśmy, będzie mógł być wykonywany w kompleksie centralnym, lub w minikomputerze lokalnym.

W przypadku, gdy program wynikowy będzie wykonywany przez procesor roboczy kompleksu centralnego przyjmiemy, że wyniki będą przekazywane do minikomputera lokalnego w standardowej postaci blokowej, niezależnej od postaci wydawniczej, którą im nada użytkownik, posługujący się lokalnymi programami redakcji wyników. W zależności od potrzeb będzie on mógł uzyskać te dane zestawione w tabele, odpowiednio opisane, względnie wykresy - wykonane na maszynie do pisania, czy też na monitorze ekranowym, jeśli jego zestaw lokalny będzie wyposażony w tego typu urządzenie. Warto dodać, że najdogodniejszym sposobem redagowania wyników okazać się może sposób interakcyjny; trudno jest bowiem określić a priori formę wydawnictwa najlepiej eksponującą wyniki; wymaga to dużej wyobraźni.

Opisany wyżej przykład wykorzystywania systemu wielodostępnego nie wyczerpuje oczywiście wszystkich możliwych wariantów współpracy użytkownika z kompleksem centralnym. Nowe problemy i nowe możliwości będą występowały np. przy współpracy z minikomputerami wyposażonymi w graficzne mo-

nitory ekranowe. W tym miejscu chcieliśmy przede wszystkim pokazać funkcjonowanie wszystkich elementów systemu z punktu widzenia użytkownika, bez wnikania w sposób realizacji technicznej takiego systemu.

Przedstawiony na rys. 1 zarys struktury systemu wielodostępnego jest wynikiem pierwszej "warstwy" krytyki rozwoju architektury. Przejdziemy teraz do następnej "warstwy" krytyki i w jej wyniku uzupełnimy proponowaną strukturę nowymi szczegółami.

6. Ewolucja struktur pamięciowych

W II części tego opracowania staraliśmy się znaleźć w rozwoju architektury komputerów punkty "rozwidlenia", w których konkretnie przyjęte rozwiązanie było uwarunkowane stanem technologii danego okresu i raz przyjęte determinowało dalszy rozwój, mimo że technologia dostarczała już nowych możliwości zupełnie innych rozwiązań. Podobnie teraz postaramy się wykonać analogiczną analizę w stosunku do rozwoju struktur pamięciowych.

Zakres rozważań ograniczymy tylko do pamięci adresowych, pomijając (przynajmniej na tym etapie) pamięci asocjacyjne i stosowe.

Pierwsze maszyny cyfrowe dysponowały z reguły tylko jednym rodzajem pamięci. Były to pamięci operacyjne o dostępie sekwencyjnym, np. na ultradźwiękowych liniach opóźniających (rtęciowych lub magnetostrykcyjnych) lub pamięci o dostępie swobodnym - na lampach oscyloskopowych, względnie pamięci ferrytowe. Stosunkowo szybko okazało się, że zapotrzebowanie na pojemność pamięci operacyjnej nie może być zaspokojone w sposób ekonomicznie uzasadniony przez powiększanie samej tylko pamięci operacyjnej. Zostały więc wprowadzone pamięci pomocnicze w postaci pamięci bębnowych, taśmowych i znacznie później dyskowych. W związku z tym pojawił się problem przesyłania informacji między poszczególnymi szczeblami hierarchii pamięci maszyny. Zarządzanie przesłaniami informacji zostało pozostawione z reguły użytkownikowi; był to dla niego problem wynikający ze specyfiki narzędzia, którym się posługiwał, odwracający uwagę od rozwiązania zasadniczego problemu. Dopiero ostatnie lata przyniosły szersze wprowadzenie koncepcji pamięci wirtualnej. O zagadnieniu tym była już mowa, poprzestaniemy więc tylko na tej wzmiance. Zajmiemy się nato-

miast innym zagadnieniem, które ilustruje dobrze brak wymiany informacji między projektantami struktury logicznej a projektantami podzespołów komputerowych - w tym przypadku pamięci operacyjnych. Owo niedostateczne poinformowanie wzajemne co kto robi i jak rozwiązuje problemy w swej dziedzinie - spowodowało przyjmowanie rozwiązań, które budzić mogą poważne wątpliwości czy są globalnie ekonomiczne.

Zadanie przekazane konstruktorom pamięci przez projektantów struktury logicznej brzmiało: należy opracować pamięci, w których identyfikacja informacji następuje przez podanie jej adresu; pamięci powinny mieć dużą niezawodność, jak największą pojemność, jak najkrótszy czas dostępu, koszt przechowywania informacji powinien być jak najmniejszy.

Przez cały okres rozwoju maszyn cyfrowych zadanie to nie uległo praktycznie żadnym zmianom. Co więcej - projektanci architektury nie zwrócili wcale uwagi na sposób, w jaki inżynierowie-konstruktorzy pamięci operacyjnych rozwiązują swoje problemy techniczne w miarę, gdy pojemności pamięci budowanych i produkowanych stają się coraz większe. Otóż z wielu względów technicznych wielka pamięć operacyjna, np. na rdzeniach ferrytowych, czy najnowocześniejsza - półprzewodnikowa jest rozbijana na wiele mniejszych podzespołów, które są wyposażane w praktycznie niezależne układy adresowe, układy odczytu i zapisu oraz regeneracji informacji. Wspólnymi dla wszystkich bloków są przeważnie tylko układy sterowania i synchronizacji, stanowiące niewielki procent elektroniki pamięci. Te oddzielne bloki są następnie scalane w jeden większy zespół - co oznacza połączenie układów adresowych i złączenie na wspólną szynę układów przesyłania informacji do pamięci i z pamięci. Stracona zostaje w ten sposób potencjalna możliwość znacznie szybszego przekazywania informacji np. między pamięcią operacyjną a pamięciami pomocniczymi - co mogłoby mieć miejsce w przypadku, gdyby informacje przy takich przesłaniach były odczytywane czy zapisywane do wszystkich podbloków równocześnie. Jednocześnie przez połączenie technicznie oddzielnych układów adresowania w jeden wspólny układ - wynika niebezpieczeństwo interferencji różnych programów, przechowywanych we wspólnej pamięci. Bo zauważmy jak wykorzystywane są takie wielkie pamięci operacyjne, o pojemnościach sięgających setek tysięcy słów w jednym bloku. Rzadko tylko pamięć taka służy jednemu programowi. Przeważnie przechowywane są w niej programy wielu użytkowników. Trzeba je więc chronić przed wzajemnymi zakłóceniami. W tym celu w wielu rozwiąza-

niach zostały wprowadzone dodatkowe pamięci, tzw. pamięci kluczy ochrony zawartości - służące do uniemożliwienia zmiany zawartości części pamięci przez nieuprawnionego do tego użytkownika. Wydawałoby się, że prostszym rozwiązaniem było przyjąć pełne (fizyczne) oddzielenie pamięci poszczególnych użytkowników, przez przydzielenie im w systemach wieloprogramowych odpowiedniej liczby mniejszych, zupełnie oddzielnie (w fizycznym sensie) adresowalnych modułów i oddzielnych układów wprowadzania i odczytywania informacji. Koszty takiego rozwiązania nie powinny być większe; mogłyby się nawet okazać niższe, gdyż wyeliminowana zostałaby pamięć kluczy ochrony. Dlaczego mamy stosować pozorne rozdzielenie użytkowników przez oddawanie im pozornych (wirtualnych) pamięci, skoro oddanie im rzeczywiście oddzielnych pamięci wcale nie musi być rozwiązaniem kosztowniejszym. Te spostrzeżenia były motywem takiego rozwiązania problemu przydziału modułów pamięciowych poszczególnym użytkownikom, korzystającym z kompleksu centralnego jak to przedstawiono na rys. 1 i opisano na poprzednich stronach.

7. Osobliwości komunikacji wewnątrz komputerów

W każdym systemie liczącym można wyróżnić urządzenia i funkcje związane z przekształcaniem, przechowywaniem oraz przesyłaniem informacji. W dotychczasowym rozwoju maszyn najwięcej uwagi zwracano na operacje (realizowane sprzętowo) przekształcania informacji; świadczą o tym listy rozkazowe przeciętnych maszyn złożone z conajmniej kilkudziesięciu operacji. Wydaje się, że najmniej uwagi zwracano na zagadnienia komunikacji wewnątrz maszyny. Spowodowało to ugruntowanie się rozwiązań architektonicznych maszyn cyfrowych o słabo rozwiniętych i łatwo blokujących się drogach przesyłania informacji.

Wyróżnimy kilka przyczyn, które jak się wydaje, mogły zdecydować o takim rozwoju sytuacji, a następnie podamy możliwe kierunki alternatywnych rozwiązań oraz ich zastosowanie w proponowanym systemie wielodostępnym.

8. Pamięć operacyjna jako urządzenie komunikacyjno-przełączające

Omawiając zagadnienia komunikacji wewnątrz komputerów warto zwrócić uwagę na fakt, że pamięć operacyjna spełniała zawsze ważną rolę jako

urządzenie komunikacji między poszczególnymi urządzeniami maszyny cyfrowej. Taka jej rola była oczywista w pierwszych maszynach cyfrowych, gdzie wszystkie dane, programy i wyniki pośrednie były przechowywane w niej właśnie. Jednakże po wprowadzeniu do zestawów maszyn cyfrowych wolniejszych pamięci pomocniczych - pozostawienie pamięci operacyjnej w tej roli mogło już budzić pewne wątpliwości. Zauważmy, że normalną praktyką przy wprowadzaniu danych z urządzeń zewnętrznych jest przekazywanie ich do pamięci pomocniczych nie bezpośrednio, ale właśnie za pośrednictwem pamięci operacyjnej. Podobnie, jeśli mamy przekazać dane np. z pamięci taśmowej do pamięci dyskowej - to również dane te nie będą przekazywane bezpośrednio - lecz przez pamięć operacyjną. Nie powinno to budzić zdziwienia, jeśli zauważymy, że przy przekazywaniu danych wewnątrz maszyny zwykle są wykonywane równocześnie pewne zmiany formatu danych. Urządzeniem wyspecjalizowanym w maszynie w kierunku przetwarzania danych jest arytmometr jednostki centralnej, który może wykonywać swoje operacje przetwarzające tylko na danych, znajdujących się w pamięci operacyjnej, gdyż tylko do nich ma dostęp za pośrednictwem systemu adresowego. Tym niemniej należy równocześnie zauważyć, że to zmonopolizowanie operacji przetwarzających przez arytmometr i silny związek arytmometru z pamięcią operacyjną przez system adresowy jest potencjalnym źródłem przeciążenia pamięci operacyjnej funkcjami transmisji informacji przy rozbudowie systemów liczących. Nie dysponujemy jednak metodami liczbowych ocen umożliwiających wskazanie, od jakiego poziomu strumieni informacji, przepływających przez pamięć operacyjną systemu liczącego, uzasadnione byłoby mówienie o jej przeciążeniu funkcjami transmisji.

Warto podkreślić jednak również zalety wykorzystywania pamięci operacyjnej jako urządzenia komunikacyjno-przełączającego. Polegają one przede wszystkim na tym, że wszystkie informacje przechowywane w pamięci są jednakowo łatwo dostępne przez jej mechanizmy adresowania oraz zapisu i odczytu informacji. Jeśli więc kilka jednostek przetwarzających ma dostęp do tych mechanizmów możliwe jest zupełnie swobodne przekazywanie informacji między tymi jednostkami, bez konieczności fizycznego przenoszenia tej informacji, co byłoby związane ze stratami czasu na transmisję. Zaleta ta jest równocześnie wadą, gdyż korzystanie kilku jednostek ze wspólnego pola pamięciowego niesie ze sobą możliwość szkodliwych interferencji między programami, wykonywanymi przez podłączone jednostki przetwarzające. Spowodowało to konieczność wprowadzania do

pamięci specjalnych mechanizmów ochrony zawartości przed tymi szkodliwymi oddziaływaniami wzajemnymi.

Tym niemniej brak alternatywnych rozwiązań powodował, że zalety takiego wykorzystywania pamięci przeważały nad wadami i w wielu rozwiązaniach dużych komputerów, przeznaczonych dla pracy wieloprogramowej i wieloprocesorowej pamięć operacyjna była "jądrem" systemu liczącego. Za przykład może służyć struktura maszyny CDC 6600, w której bardzo duża pamięć operacyjna jest "otoczona" procesorami specjalizowanymi i peryferyjnymi. Wymiana informacji między nimi przebiega za pośrednictwem pamięci operacyjnej, zajmującej centralną pozycję w systemie.

9. Zwiększanie szerokości dróg transmisyjnych

Jeśli przyjmiemy za uzasadnione wykorzystywanie pamięci operacyjnej jako urządzenia komunikacyjnego-przełączającego oraz istnienie kilku szczebli pamięci pomocniczych - to powinniśmy przyjmować takie metody transmisji między tymi pamięciami, które zmniejszają czas transmisji. Jedna możliwość to zwiększanie szybkości transmisji, która jest jednakże ograniczona szybkościami ruchu nośnika w pamięciach pomocniczych magnetycznych i gęstościami zapisu. Druga możliwość to zwiększanie szerokości dróg transmisji, czyli ich zrównoleglanie. Wymiana informacji między pamięciami operacyjnymi a pomocniczymi dyskowymi czy bębnowymi odbywa się z reguły bajt po bajcie, lub słowo po słowie; transmisja do pamięci taśmowych - zawsze bajt po bajcie. Jednakże liczne pamięci bębnowe, czy dyskowe ze stałymi głowicami mają te głowice zwielokrotnione - dla uzyskania skróconego czasu dostępu. W takich sytuacjach można byłoby, kosztem już niewielkiego rozbudowania układów elektronicznych, wykorzystać te głowice dla zwiększenia szybkości transmisji - zapisując informacje odczytywane z pamięci operacyjnej na kilka odcinków, wydzielonych na obwodzie bębna. Oczywiście, przy tym sposobie zapisu jeden blok informacji, przekazywany z pamięci operacyjnej byłby rozłożony w kilku miejscach na powierzchni nośnika, jednak szybkość przekazywania informacji zwiększyłaby się przy zachowaniu skróconego czasu dostępu. Zostałoby to okupione koniecznością zwiększenia liczby układów elektronicznych zapewniających odpowiednie przekazywanie informacji z zespołów głowic do pamięci operacyjnej, umożliwiające zachowanie skróconego czasu dostępu.

Coraz niższe ceny układów elektronicznych pozwalają sądzić, że zwiększenie kosztu byłoby niewielkie.

Prawdopodobnie można byłoby podać wiele wariantów technicznego rozwiązania problemu zwiększenia szerokości dróg transmisyjnych między poziomami hierarchicznego systemu pamięciowego. Ich wspólną cechą byłby fakt wykorzystywania wprowadzonych już do tych systemów pamięciowych (w pewnym sensie) nadmiarowych elementów konstrukcyjnych - dla celu pierwotnie nie przewidywanego, a mianowicie dla zwiększenia strumienia informacyjnego. Następowałoby to nie przez zwiększenie szybkości przekazywania informacji po pojedynczym torze transmisyjnym (dla jednego bitu), a przez znaczne zrównoleglenie tych torów transmisyjnych. Nazwaliśmy to zwiększaniem szerokości dróg transmisyjnych. Nie będziemy starali się mnożyć pomysłów tego typu. Wydaje się bowiem, że nie byłyby to rozwiązania prawdziwie perspektywiczne. Pojawiająca się nowa technologia przechowywania informacji tj. pamięci holograficzne powinna w perspektywie dostarczyć prawie idealnego rozwiązania problemu. Przedstawiona w 1970 r. przez Jana A. Rajchmana koncepcja pamięci holograficznej, której istotnym składnikiem jest tzw. "latrix", gdy zostanie zrealizowana, będzie rozwiązywała problem krótkiego czasu dostępu do bardzo pojemnej pamięci. Zapewni też bardzo wielką szerokość transmisji informacji między półprzewodnikową pamięcią (operacyjną, buforową) a holograficzną pamięcią pomocniczą o wielkiej pojemności. W przedstawionej koncepcji Rajchman uważa za osiągalne: pojemność całego systemu pamięciowego ok. 10^{10} bitów, zorganizowanych w strony o pojemności 10^5 bitów. Osiągalny czas dostępu do dowolnej informacji w ramach strony stanowiłby ułamek mikrosekundy (gdyż byłby to dostęp do pamięci typu półprzewodnikowego), czas dostępu do dowolnej strony wyniósłby pojedyncze mikrosekundy, gdyż byłby to dostęp metodami optycznymi.

Już krótka charakterystyka techniczna pozwala stwierdzić, że pamięć taka, a właściwie dwupoziomowy hierarchiczny system pamięciowy jest z punktu widzenia systemowego rewelacyjnym rozwiązaniem zagadnienia przechowywania informacji. W perspektywie może on wyeliminować wszystkie dotychczasowe rozwiązania systemów pamięciowych, oparte na stosowaniu dysków czy taśm magnetycznych, jako pamięci pomocniczych w systemie. Eliminuje on bowiem całkowicie ruchome części mechaniczne, zapewnia wielką niezawodność działania (niskie stopy przekłamań) wynikające z przyjęcia

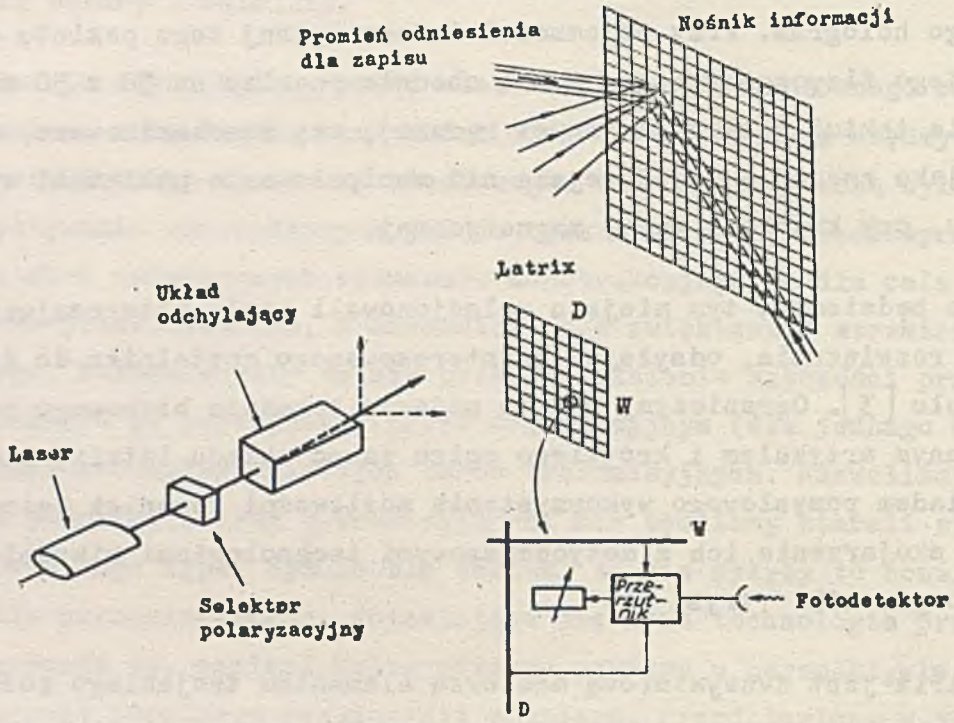
holograficznej zasady przechowywania informacji, umożliwia dalsze zwiększanie pojemności przez wprowadzenie możliwości wymiany "pakietu" mieszczącego hologram. Przy pojemności informacyjnej tego pakietu ok. 10^{10} bitów, jego fizyczne wymiary można obecnie oceniać na 50 x 50 cm [1]. Manipulowanie takimi pakietami (nawet ręczne), czy zmechanizowane, można oceniać jako znacznie wygodniejsze niż manipulowanie pakietami wymiennych dysków, czy krążkami taśmy magnetycznej.

Nie będziemy w tym miejscu relacjonowali wielu interesujących szczegółów rozwiązania, odsyłając zainteresowanego czytelnika do źródłowego artykułu [3]. Ograniczymy się do podania schematu blokowego pamięci za cytowanym artykułem i krótkiego opisu zasad układu latrix, który jest przykładem pomysłowego wykorzystania możliwości techniki holograficznej, przez skojarzenie ich z dotychczasowymi technologiami mikroelektroniki i optocyfroniki (rys. 2).

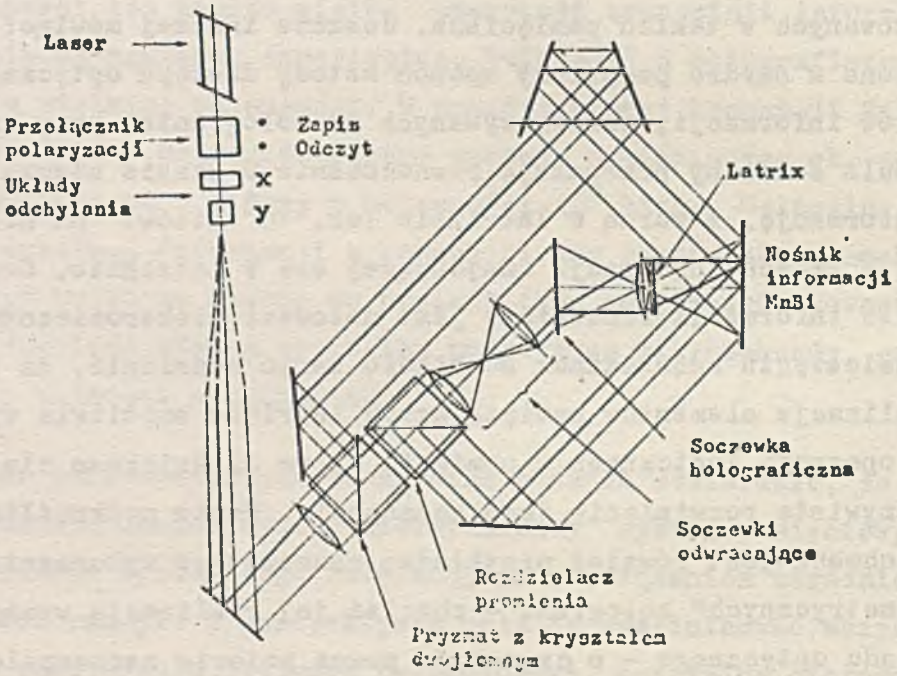
Latrix jest dwuwymiarową macierzą elementów trojakiego rodzaju: modulujących światło np. przez zmianę przepuszczalności ciekłego kryształu , fotoczułych i bistabilnych przerzutników. W ten sposób jedna macierz tych elementów tworzy zarazem tzw. twornik stronicy [4], układ odczytujący hologram i pamięć buforową lub główną-operacyjną , dostępną za pomocą metod stosowanych w takich pamięciach. Jeszcze inaczej mówiąc: w latriksie są skojarzone w bardzo pomysłowy sposób metody dostępu optycznego do dużych zbiorów informacji, przechowywanych na hologramie, przy czym pojedynczy impuls świetlny przekazuje równocześnie w czasie ułamka mikrosekundy całą informację, zawartą w latriksie (ok. 10^5 bitów) na hologram lub odwrotnie. W ramach informacji znajdującej się w latriksie, dostęp do dowolnego bitu informacji zapewniony jest metodami elektronicznymi w czasie ok. kilkudziesięciu nanosekund. Dodatkowo warto nadmienić, że półprzewodnikowa realizacja elementów pamiętających latriksu umożliwia wykonywanie złożonych operacji logicznych w pamięci. O tym J. Rajchman nie pisze, lecz to jest oczywiste rozwinięcie jego koncepcji . Warto podkreślić, że koncepcja Rajchmana jest również przykładem znakomitego wykorzystania własności "geometrycznych" hologramu - chociaż jej realizacja wymaga dość złożonego układu optycznego - o czym daje pewne pojęcie zaczerpnięty z artykułu Rajchmana rys. 3.

---

[1] Informacja uzyskana od mgr A. Sikorskiego z Zakładu Optocyfroniki IMM.



Rys. 2. Zasada działania pamięci holograficznej wg koncepcji J. Rajchmana



Rys. 3. Schemat konstrukcyjny pamięci holograficznej wg koncepcji J. Rajchmana

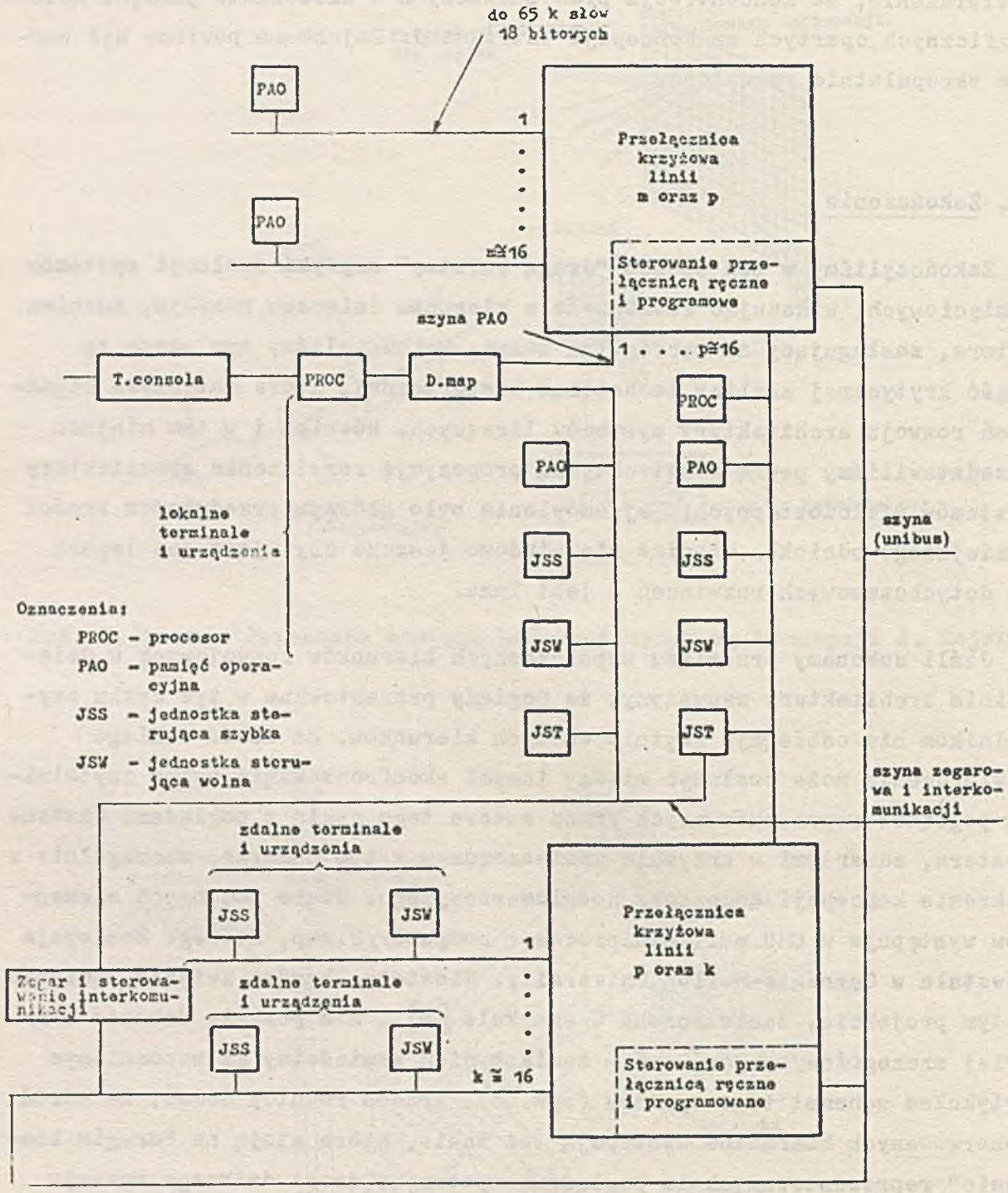
Na podstawie tego co wyżej powiedziano, wydaje się mało ryzykowne stwierdzenie, że koncentracja prac badawczych w dziedzinie pamięci holograficznych opartych na koncepcji latriksa J. Rajchmana powinna być bardzo skrupulatnie rozważona.

10. Zakończenie

Zakończyliśmy w ten sposób "drugą warstwę" krytyki ewolucji systemów pamięciowych, wskazując równocześnie kierunek dalszego rozwoju, zdaniem autora, zasługujący na szczególną uwagę. Zakończyliśmy tym samym tę część krytycznej analizy techniczno-historycznej, która dotyczyła zagadnień rozwoju architektury systemów liczących. Również i w tym miejscu przedstawiliśmy pewną konstruktywną propozycję rozwiązania architektury systemów wielodostępnych; jej omówienie było głównym przedmiotem treści niniejszego odcinka. Chociaż nie wiadomo jeszcze czy jest ona lepsza od dotychczasowych rozwiązań - jest inna.

Jeśli dokonamy przeglądu współczesnych kierunków rozwojowych w dziedzinie architektury zauważymy, że poglądy prezentowane w tym cyklu czytelnikom nie odbiegają zbytnio od tych kierunków. Za dowód takiego stwierdzenia może posłużyć między innymi skonfrontowanie przez czytelnika poglądów reprezentowanych przez autora tego cyklu z poglądami Caxtona Fostera, zawartymi w artykule zamieszczonym w tym numerze, szczególnie w zakresie koncepcji komputera konglomeracyjnego. Wiele podobnych elementów występuje w CMU multiminiprocesor computer/C.mmp, którego koncepcja powstała w Carnegie-Mellon University. Niestety, bardzo zwięzła wzmianka o tym projekcie, zamieszczona w artykule [5] , nie pozwala dokonać bardziej szczegółowych porównań - zamiast nich zamieścimy za wspomnianym artykułem schemat tego systemu (rys. 4). Trzeba również dodać, że wśród obserwowanych kierunków występują też takie, które stoją na "drugim biegunie" reprezentowanych tu poglądów, upatrując drogi dalszego rozwoju w jeszcze dalszym "eksploatowaniu" zasady przełączania procesora. Znakomitym przykładem tej drogi jest propozycja M.J. Flynna i A. Podvina - An Unconventional Computer Architecture: Shared Resource Multiprocessing [6].

Pozostawiając czytelnikowi dokonanie porównań i wyciągnięcie wniosków, chcielibyśmy podkreślić, że dość istotny wydaje się fakt następujący. Zamieszczona w niniejszym odcinku propozycja innej organizacji syste-



Rys. 4. Uproszczony schemat blokowy systemu wieloprocesorowego Uniwersytetu Carnegie-Mellon

mów liczących wyniknęła na drodze systematycznego postępowania (analizy techniczno-historycznej), które prawie w całości zostało przytoczone w kolejnych odcinkach. W ten sposób podlega ono weryfikacji, przynajmniej w zakresie sprawdzenia poprawności rozumowania i zasadności przyjmowanych założeń. W wyniku powstała koncepcja, która ma wiele cech wspólnych z koncepcjami innych autorów. Nawet uwzględniając wszystkie wpływy, którym autor tego cyklu był poddany, studiując literaturę przedmiotu i pracując w tej dziedzinie dość długi okres czasu - wydaje się, że względnie niezależne otrzymanie podobnego wyniku jest jakąś wstępną weryfikacją płodności drogi postępowania, jaka tu została przyjęta.

Biorąc to pod uwagę należałoby wspomniane metody, a w szczególności "analizę techniczno-historyczną" zastosować do innych sfer informatyki w nadziei, że w wyniku tego mogą powstać inne koncepcje. Zamiar taki zostanie w najbliższym czasie zrealizowany w odniesieniu do dziedziny programowania. Przeznaczony na te zagadnienia zostanie odrębny cykl, wiążący się z niniejszym zastosowaniem analogicznej metody postępowania. Prawdopodobnie będzie to równocześnie kolejna "warstwa krytyki". Po jej przeprowadzeniu można będzie ponownie powrócić do prób przedstawienia szczegółów koncepcji, przedstawionej na rys. 1.

Równocześnie, aby móc zgromadzić argumenty innego typu niż werbalne, w ewentualnej dyskusji nad zaletami przedstawionej tu propozycji systemu wielodostępnego i aby sprecyzować opis tej koncepcji - został sporządzony przez autora tego cyklu opis struktury proponowanego systemu wielodostępnego w języku SIMULA 67. Po przeprowadzeniu symulacji na maszynie CYBER 70 zarówno opis, jak wyniki symulacji zostaną opublikowane w Pracach IMM.

Na zakończenie kilka słów wyjaśnienia. Struktura artykułów tego cyklu nie pozwoliła w zaplanowanej objętości pomieścić wszystkich zagadnień sygnalizowanych w pierwszym odcinku, w szczególności poglądów autora dotyczących niektórych warunków realizacji dużych przedsięwzięć informatycznych w Polsce. Częściowo, ale tylko częściowo zostało to wykonane przez opublikowanie w Przeglądzie Technicznym nr 20/1973 artykułu publicystyczno-polemicznego: "Grzechy główne - czy tylko informatyki?" Charakter spraw, jakie należałoby poruszyć w tej dziedzinie skłonił mnie do wyłączenia ich z niniejszego cyklu. Uważam natomiast, że warto oddzielnie spróbować zestawić metody, jakie mogą być pomocne w ukierunkowywaniu

przyszłych działań w dziedzinie informatyki. Artykuł na ten temat zostanie umieszczony w jednym z następnych numerów ETO Nowości, co będzie stanowiło realizację pozostałych zapowiedzi dotyczących problemów poruszonych w tym cyklu, umieszczonych we wstępie do odcinka I niniejszego opracowania (ETO NOWCŚCI 4/1972).

Literatura

[1] FOSTER C. Caxton: A View of Computer Architecture. Comm. ACM, 15, 1972, nr 7, s. 557-565.

[2] DAŃDA J., MROZIEWICZ B.: Przełączane łącze z półprzewodnikowym źródłem promieniowania. Urząd Patentowy PRL, patent nr 59479 data zgłoszenia: 30.X.1968.

[3] RAJCHMAN J.A.: Promise of Optical Memories. J. App. Phys. 41, 1970, nr 3, s. 1376-1383.

[4] WRZESZCZ Z.: Kierunki realizacji pamięci optycznych swobodnego dostępu. ETO Nowości, 1972, nr 1, s. 3-27.

[5] BELL C.G., CHEN R., REGE S.: Effect of Technology on Near Term Computer Structures. Computer, 1972, marzec-kwiecień, s. 29-38.

[6] FLYNN M.J., PODVIN A.: Shared Resource Multiprocessing, ibid., s. 20-28.

[7] FOSTER C.C.: The Next Three Generation, ibid., s. 39-42.

ETO NOWOŚCI nr 1/1974

681.322.001.6

ZAGADNIENIA ROZWOJU MASZYN MATEMATYCZNYCH

Dyskusja w Instytucie Maszyn Matematycznych

WPROWADZENIE

W ramach normalnego w takiej sytuacji wprowadzenia do dyskusji chciałbym podać motywy, którymi kierowała się redakcja naszego czasopisma decydując się, po pierwsze na zamieszczenie w kolejnych numerach kilku wybranych tłumaczeń artykułów z jubileuszowego, z okazji 25-lecia organizacji amerykańskiej Association for Computing Machinery, wydanego numeru Communications of the ACM, a po drugie, zorganizowania i opublikowania zapisu dyskusji na temat tych artykułów. Chcę mówić o tym dlatego między innymi, że pozwoli to, mam nadzieję, na takie ukierunkowanie dyskusji, które nie będzie stało w sprzeczności z zamierzeniami redakcji, co z kolei, może pomóc tak ukształtować naszą dyskusję, aby nasi Czytelnicy mogli odnieść możliwie duże korzyści czytając jej prawie pełny zapis.

Pierwszy motyw wynika z naszego podstawowego obowiązku, którym jest informowanie naszych Czytelników o najnowszych osiągnięciach w dziedzinie informatyki, zarówno w zakresie sprzętu, jak i oprogramowania. Zazwyczaj zadanie to realizujemy przez publikowanie opracowań noszących charakter kompilacji, bądź bardziej samodzielnych opracowań, omawiających najnowsze tendencje w informatyce. Dotychczas pisane one były w przeważającej mierze przez autorów, rekrutujących się spośród pracowników Instytutu Maszyn Matematycznych, chociaż bardzo często i bardzo chętnie drukowaliśmy opracowania z innych ośrodków. Mieliśmy jednak trudności z uzyskaniem materiału, który odpowiadałby w pełni założeniom naszego pisma.

Postanowiliśmy wydać kilka najbliższych numerów ETO NOWOŚCI o ściśle sprecyzowanym profilu, dostosowanym nie tyle może do współczesnych trendów rozwojowych informatyki światowej, ile do tej ich części, która zdaniem naszego zespołu redakcyjnego, będzie miała w najbliższym okresie kilku lat znaczenie największe dla informatyki w Polsce. Będą to kierunki: organizacja wielkich, wielomaszynowych systemów elektronicznego przetwarzania danych, porównanie dojrzałych systemów operacyjnych znanych producentów sprzętu, nowe kierunki w organizacji procesów przetwarzania informacji np. oparte na możliwościach stwarzanych przez optocyfronikę), sformułowanie zasad, którymi powinien być kierowany proces petryfikacji software'u tj. przejmowania tradycyjnych funkcji oprogramowania przez sprzęt), porównanie zdalnego przetwarzania danych (teleprocessing) z przetwarzaniem wielodostępnym, postępy w dziedzinie oprogramowania monitorów ekranowych alfanumerycznych i graficznych.

Jak łatwo zauważyć, wymieniona przeze mnie tematyka pokrywa się w niewielkim tylko stopniu z tematami artykułów, o których mamy dyskutować. Nie jest to jednak , zdaniem redakcji, wielki niedostatek z dwóch względów. Po pierwsze, jest to tylko część artykułów, które w najbliższym okresie zamierzamy publikować, a po drugie - artykuły z Communications of the ACM wydają się tak ciekawe i w gruncie rzeczy powiązane z poprzednio omówionymi naszymi zamierzeniami, że ich zreferowanie może dobrze uzupełnić, pierwotnie może zbyt wąsko określony profil przeszłych numerów naszego czasopisma.

Mamy nadzieję, że przedyskutowane szczególnie zostaną zagadnienia produkcji oprogramowania i nieodłącznie z nią związane zagadnienia warsztatu, czy narzędzi programowych - problem niesłychanie aktualny w Polsce, a zwłaszcza w Instytucie Maszyn Matematycznych.

Nie potrzebuję chyba podawać zalet, ogólnie uznanych, które ma dyskusja w kształtowaniu poglądu na sprawy dla naszej informatyki istotne, i przewagi nad metodą, która mogłaby polegać na opublikowaniu wielu oddzielnych artykułów przeglądowych, dotyczących poszczególnych zagadnień. Po-

Pomysł zorganizowania dyskusji podsunął naszej redakcji mgr inż. Włodzimierz Mardal.

Artykuły, które są przedmiotem naszej dyskusji tylko w swej części dotyczą spraw, które są uniwersalnie ważne; wydaje się jednak, że zawierają one sporo stwierdzeń trafnych i praktycznych; dlatego też zostały włączone jako materiał dyskusyjny.

Jerzy Dańda

Z inicjatywy redakcji naszego czasopisma odbyła się w lutym 1973 r. dyskusja na temat zagadnień rozwoju maszyn matematycznych. Udział w dyskusji wzięły następujące osoby:

mgr inż. Jerzy Dańda	- redaktor naczelny "ETO-Nowości"
doc. dr hab. inż. Andrzej Janicki	- zastępca dyrektora ds naukowych Instytutu Maszyn Matematycznych
mgr Thanasis Kamburelis	- WZE-ELWRO Ośrodek Badawczo-Rozwojowy Maszyn Cyfrowych
mgr inż. Bohdan Kasierski	- WZE-ELWRO OBR MC
Władysław Klepacz	- IMM
mgr inż. Andrzej Kojemski	- IMM
doc. dr hab. inż. Stanisław Majerski	- IMM
doc, mgr inż. Romuald Marczyński	- CO PAN
mgr inż. Włodzimierz Mardal	- IMM

Ponadto obecne były następujące osoby:

mgr Magdalena Godyńska	- IMM
mgr Zygmunt Hauser	- IMM
dr inż. Tadeusz Jankowski	- IMM
mgr Janusz Janowski	- IMM

mgr inż. Maria Kowalewska - IMM
mgr inż. Jolanta Krauze - ZWPP "ERA"
mgr inż. Wojciech Kubera - IMM
dr Jacek Olszewski - IMM
inż. Eugeniusz Szydłowski - IMM
mgr inż. Włodzimierz Zapendowski - IMM

Podstawą do dyskusji były artykuły, które ukazały się w Communications of the ACM 1972, nr 7:
As the Industry Sees It, s. 508-517
Foster Caxton C.: A View of Computer Architecture, s. 557-565
Fosdick Lloyd D.: The Production of Better Mathematical Software, s. 611-617
Salton G.: Dynamic Document Processing, s. 658
Lynch W.C.: Operating System Performance, s. 579

Redakcja "ETO-Nowości", starając się o przedstawianie na łamach swego czasopisma najważniejszych kierunków rozwoju systemów komputerowych, postanowiła opublikować tłumaczenia niektórych z tych artykułów. W czasie dyskusji omawiano szczegółowo artykuł C.C. Fostera "Review of Computer Architecture"[1]. Redakcja "ETO-Nowości" przypuszcza, że przebieg dyskusji może być interesujący dla Czytelników i dlatego poniżej przedstawia ją z pewnymi skrótami.

Dyskusję zagaił redaktor naczelny "ETO-Nowości" mgr inż. Jerzy Dańda, prosząc przede wszystkim o wypowiedzenie się w sprawie sposobu udostępnienia artykułów z Communications of the ACM czytelnikowi polskiemu.

R. M a r c z y ń s k i

"Artykuły te powinny być przetłumaczone na język polski z tego względu, że są artykułami dyskusyjnymi, które należy czytać szybko, bez trudności językowych. Oczywiście jest w Polsce kilka osób, pracujących w naszej dziedzinie, które języki obce mają znakomicie opanowane. Chodzi tu jednak o znacznie większe grono. Wydanie tych artykułów po polsku byłoby bardzo korzystne, lecz jak je wydać, aby umożliwić szerszy rynek zbytu?

---

[1] Foster C.C.: Perspektywy rozwoju architektury komputerów. Oprac. A. Kojemski, s. 57

"ETO-Nowości" są takimi nowościami, których się na ogół nie czyta. Ja osobiście czytuję to czasopismo, ale jestem chyba osobliwością w mojej instytucji. Dobrze byłoby, gdyby te artykuły dotarły szeroko do społeczeństwa, a przede wszystkim do osób stojących wysoko w hierarchii społecznej, do osób, do których należy decyzja. Nie wiem, jak to zrobić. Można by to wydać w postaci książkowej, ale znowu powstaje obawa, że książka ta trafi tylko do wybranych osób. Kwestia samego opublikowania to jeszcze nie wszystko, ważne jest jak i gdzie to jest wydane, jak wydrukowane".

W. K l e p a c z

"Wymienione artykuły są pisane suchym językiem, dostęp do tego rodzaju publikacji musi być z natury rzeczy ograniczony".

J. D a ń d a

"W zespole redakcyjnym "ETO-Nowości" zastanowimy się nad tym i ewentualnie przedyskutujemy tę sprawę oddzielnie. Może jednak poczytność "ETO-Nowości" nie jest tak mała i opublikowanie artykułów z Communications of the ACM przez nas wywoła jakiś pozytywny skutek.

W tej chwili proponuję przystąpić do omawiania treści artykułów. Moim zdaniem, wnioski wypływające z tych artykułów w znacznie szerszym zakresie odnoszą się do naszej sytuacji krajowej, niżby to mogło wydawać się na pierwszy rzut oka. Foster w swoim artykule powiada, że w latach osiemdziesiątych komputer czy mikrokomputer, którego szybkość działania określa na powyżej 1 miliona operacji na sekundę, a pojemność pamięci - na około 64 K bajtów, będzie kosztował od 1 do 2 dolarów. Podstawowym warunkiem takiego ustalenia ceny jest oczywiście stworzenie masowego rynku zbytu, którego poszukuje w skomputeryzowanych kalendarzykach, kartach kredytowych i podobnych zastosowaniach. Z drugiej strony Foster podaje bardzo ciekawą koncepcję tzw. konglomeracyjnego systemu przetwarzania, złożonego z dziesiątków czy setek bardzo małych komputerów. Myślę, że właśnie ten typ wykorzystania może być u nas preferowany. Hasło: "Komputer dla wszystkich", jak to głoszą Amerykanie - polegające na tym, że każda gospodyni będzie miała książkę kucharską w komputerze, który będzie też sterował piekarnikiem i jednocześnie pilnował dziecka -

jest dla mnie jakąś mrzonką i niepoważnym podejściem do sprawy. Ale jeśli mowimy o perspektywie 20-25 lat, to trudno odżegnywać się nawet od takich rozwiązań, które wynikają z nadmiernego rozszerzania i pobudzania rynku konsumpcyjnego. Istnieje tu jakaś kontrowersja i może na ten temat warto dyskutować".

R. M a r c z y ń s k i

"Artykuł Fostera przeczytałem jeszcze w zeszłym roku, grubo wcześniej niż mi redakcja "ETO Nowości" go przysłała. Wykorzystałem go nawet do pewnego opracowania. Jeśli chodzi o moje osobiste zdanie, to zgadzałem się z Fosterem jeszcze przed tym, zanim on ten artykuł napisał. Uważam za prawidłowy kierunek budowę małych maszyn z układów scalonych. Widzę wielkie możliwości seryjnego produkowania takich maszyn. Nie chodzi tu jednak o maszyny, które będą robiły ciasto; wątpię aby kobiety-gospodynie ich używały, gdyż kobiety są oporne, jeśli chodzi o używanie bardzo nowoczesnych urządzeń w domu. Mam doświadczenie na przykładzie żony i jeszcze kilku innych znajomych pań.

Poruszę problem zastosowania mikrokomputerów w telewizji kolorowej, w której - jak wiadomo - istnieją trzy systemy. Nie jest tu istotne, czym one się różnią. Wydaje mi się jednak prawdopodobna realizacja rozpakowywania i pakowania informacji telewizyjnych za pomocą minikomputerów. Sądzę, że można by w pewien sposób różne systemy telewizyjne sprowadzić do jednego, który mógłby być modyfikowany w zależności od kraju. Problem ten mógłby mieć później wtórnie wpływ i na naszą dziedzinę".

A. K o j e m s k i

"Spróbuję krótko przedstawić niektóre zagadnienia i ciekawsze przykłady zastosowań mikrokomputerów poruszone w omawianym artykule C. Fostera. Sądzę, że jest to celowe również z tego względu, że są wśród nas osoby, które go nie zdążyły przeczytać.

Uwaga redakcji

Wypowiedź A. Kojemskiego podajemy w dużym skrócie, odsyłając Czytelnika do tłumaczenia artykułu na stronie 57

1) Realizacja procesora komputerowego (mikrokomputera) w postaci mikroukładu (chip) przy pozostawieniu rozwiązań urządzeń zewnętrznych na poziomie bliskim obecnemu. Z tego wynika, że za 25 lat procesor w maszynie cyfrowej w zasadzie przestanie się liczyć, natomiast główna uwaga konstruktora i koszty sprzętu będą skoncentrowane na urządzeniach zewnętrznych. Może być tak, że każde urządzenie zewnętrzne będzie samodzielnym urządzeniem przetwarzającym przez dołożenie do niego procesora w postaci mikroukładu.
2) Za około 25 lat cena takich mikrokomputerów będzie wynosiła około 1 dolara i żeby je masowo produkować, trzeba znaleźć dla nich szerokie zastosowanie. Ze względu na masowość, bardzo istotny będzie wybór rozwiązania, gdyż wyrób ten powinien być standaryzowany i dopasowany do wielu zastosowań.
3) Zastosowanie mikrokomputerów w dziedzinie usług osobistych, z przeznaczeniem dla przeciętnego obywatela, można zilustrować takimi przykładami jak: "długoterminowy budzik", karty identyfikujące posiadacza służące np. do porozumiewania się w celach handlowych, przy podejmowaniu pieniędzy, do otwierania zamków elektronicznych. (Ciekawe jest założenie Fostera, że w latach dwutysięcznych domy i samochody też powinny być zamykane, z tym, że na bardziej nowoczesne zamki np. elektroniczne). Do innych ciekawszych przykładów można zaliczyć maszynę do pisania z mikrokomputerem w postaci mikroukładu, pełniącego funkcje sekretarki ze znajomością słownika ortograficznego, urządzenia do rozpoznawania głosu stanowiące indywidualny "adapter" dla poszczególnych osób do komunikacji z różnymi urządzeniami i skomputeryzowane gry dla wypełnienia zwiększającej się ilości wolnego czasu.
4) Zastosowanie mikrokomputerów do funkcji obliczeniowych. Foster opisuje między innymi znane już z innych publikacji systemy sieciowe, w których węzłach będą zainstalowane pojedyncze mikrokomputery. Systemy te mają służyć do przetwarzania bardzo dużych ilości informacji.
5) Pomimo bardzo szerokiego potraktowania spraw wykorzystania mikrokomputerów autor pominął niektóre dziedziny, nawet dość istotne, jak np. motoryzacja".

J. D a ń d a

"Czy pan sądzi, że te kierunki u nas w Polsce będą się podobnie rozwijały?"

A. K o j e m s k i

"Uważam, że tak. Jeżeli zgodzimy się wszyscy, że tak będzie, to musimy sobie odpowiedzieć na pytanie, co trzeba czynić. Jesteśmy krajem opóźnionym w rozwoju omawianej dziedziny. Jeśli chcemy osiągnąć pewien wybrany poziom - to być może najlepiej metodą obejścia lub przeskakiwania pewnych etapów. Bardzo istotny jest rozwój technologii, umożliwiający posiadanie omawianych minikomputerów. Jeśli nawet nie będziemy w stanie zapewnić sobie ich produkcji, to możemy działać szeroko w dziedzinie zastosowań, opracowywania nowych rozwiązań ideowych, stosując np. metody modelowania na urządzeniach mniej doskonałych technicznie. Przykładem może tu być telewizja i inne wspomniane dziedziny. Ale czy można w tym ustalić naszą specjalizację krajową i czy to rokowałoby powodzenie? Trudno przewidzieć. Istotna jest ciągle sprawa dystansu dzielącego nasz kraj od krajów wyprzedzających nas w rozwoju sprzętu".

W. M a r d a l

"Punktem wyjścia do rozważań omawianego artykułu jest stwierdzenie, że szybki postęp odbywa się w dziedzinie miniaturyzacji elementów technicznych służących do budowy maszyn; elementy te tanieją, co wynika z masowości produkcji, opanowania wydajnych procesów technologicznych itp. Ta miniaturyzacja wraz ze znaczną obniżką cen powoduje - w pewnym sensie - dwa skutki. Po pierwsze, maszyny, które dziś nazywamy minikomputerami a niedługo może będziemy nazywali piko, a w każdym razie mikrokomputerami, będą uzyskiwały wymiary umożliwiające dziś nawet nie uświadamiane sobie przez nas kierunki zastosowań. Minikomputer z pewnością będzie w coraz większym stopniu wchodził jako dodatek usprawniający do różnorodnego sprzętu. Nie wykluczałbym dzisiaj żadnych, nawet wydających się dzisiaj może śmiesznymi, zastosowań takich, jak sterowanie piekarnikiem. Ale na pewno zupełnie realne jest sterowanie tramwajem czy samochodem, optymalizacja funkcjonowania każdego tego typu urządzenia, z którym będziemy mieli na co dzień do czynienia. W związku z tym będzie to chyba masowe zastosowanie komputera jako pewnego elementu innego sprzętu, nie tylko sprzętu tego typu, jak przyrządy pomiarowe czy produkcyjne, ale również sprzętu powszechnego użytku, wspomagającego człowieka w życiu codziennym. Prawdopodobnie tego rodzaju zastosowania - już to chyba

można powiedzieć - nie będą stawiały wielkich wymagań co do rozbudowanej komunikacji człowiek-maszyna. Wobec tego, w takich zastosowaniach nie istnieje problem pokonania bariery miniaturyzacji sprzętu do wprowadzania i wyprowadzania informacji. W większości przypadków będzie to komunikacja maszyny nie z człowiekiem, lecz z urządzeniem kontrolowanym bądź sterowanym.

Drugi skutek miniaturyzacji polega na tym, że w rozsądnych wymiarach geometrycznych, uzyskujemy ogromny "skok" mocy obliczeniowej, pamięci, pojemności maszyny i możemy rozwiązywać wielokrotnie trudniejsze,bardziej złożone problemy obliczeniowe w tym samym czasie co dzisiaj.Wydaje mi się, że ten kierunek będzie miał duże znaczenie dla sterowania wielkimi obiektami, również całymi organizacjami, przedsiębiorstwami - w kierunku zoptymalizowania ich ekonomicznego działania. Nie zgodziłbym się jednak z tezą, że maszyny dużej mocy, geometrycznie porównywalne z tymi maszynami, które dzisiaj uważamy za do przyjęcia w sensie wymiarowym, mogłyby być oparte na zasadach zespołów minikomputerów, mikrokomputerów czy pikokomputerów. Mam wątpliwość, czy to jest kierunek, który rzeczywiście nabierze znaczenia, czy w ogóle wejdzie w życie. Wydaje się, że pozostaniemy chyba jednak przez długi okres czasu przy koncepcji budowy zwartych maszyn, może o nowszej, lepszej organizacji, ale nie w formie sieci komunikujących się ze sobą mikrokomputerów. Byłoby to chyba jakieś naruszenie sensownych proporcji, bowiem w maszynach dużej mocy, o podanym przeze mnie profilu zastosowań, problemy komunikacji człowiek-maszyna będą odgrywały, w odróżnieniu od pierwszej kategorii - znacznie większą rolę. Nadal pozostanie chyba w praktyce posługiwanie się dokumentami w jakiejś formie pisanymi. Nawet komunikacja głosem będzie wymagała dość złożonych urządzeń analizy i syntezy mowy i nie bardzo byłoby sensowne, przy takich bardzo rozbudowanych urządzeniach wprowadzania i wyprowadzania informacji, szerzej mówiąc - komunikacji między człowiekiem a maszyną, budować samą część liczącą, opierającą się na bardzo znacznie zminiaturyzowanych maszynach".

A. J a n i c k i

"Osobiście sądzę,że artykuł C.C.Fostera nie nadaje się do poważniejszej dyskusji o charakterze naukowym, czy nawet o charakterze naukowo-technicznym. Raczej byłbym skłonny traktować jego treść jako pewien zbiór dość ciekawych myśli, interesujących spostrzeżeń,które pobudzają

do dyskusji, do wymiany poglądów i podkreślenia wartości wszelkiego rodzaju prognoz czy wszelkiego rodzaju futurologicznych przemyśleń. I chyba nic więcej. Co prawda sam też nie czuję się dzisiaj wystaczająco przygotowany do tego, żeby powiedzieć coś bardziej sensownego na temat przyszłości. Chciałbym jedynie poddać pod rozwagę kilka spostrzeżeń, które odnoszą się do trzech grup zagadnień.

Pierwsza grupa zagadnień wynikać będzie z pewnego pojęcia użyteczności tego, co czynimy i czynić zamierzamy. Tu właśnie chciałbym przyjąć inny punkt widzenia niż autor omawianego artykułu. Sądzę,że oparł się on raczej na pewnym przedłużeniu znanego z literatury rozumowania i wyciągnięciu wniosków z faktów, które dzieją się w dziedzinie maszyn cyfrowych na gruncie technologii. Wiadomo - miniaturyzacja, wiadomo - niesłychane możliwości zamieszczenia nawet bardzo złożonych funkcji w bardzo ograniczonym wycinku przestrzeni, w małej objętości. Pobudza to naturalnie do wielu pomysłów aplikacyjnych.Pomysły przedstawione przez C.C.Fostera mają charakter trochę peryferyjny,co nie znaczy,że peryferie te nie mogą czasami dać znakomitych efektów,o ile będą stosowane w skali masowej. Sądzę, że mnie państwo rozumiecie, nie chcę się dalej rozwodzić.

Co oznacza przyjęcie punktu widzenia użyteczności? Co ono nam daje w sensie jakichś szerszych kryteriów? Pytania istotne, bo pomysłowość ludzka będzie bardzo pobudzona, jeśli będzie miała za przysłowiowego dolara różne rzeczy do dyspozycji. Można będzie te układy za "dolara" stosować prawie wszędzie, a nawet cała propaganda, wzory i styl życia mogą być pod ten cel formowane. Nie chciałbym naturalnie sprowadzać tego problemu do kategorii problemów ekonomiczno-politycznych. Nadmieniłem tylko, że przy masowości uruchamiane są i inne motywy niż tylko rozwój danej dziedziny oparty na celowym jej aplikowaniu. Czy omawiany artykuł daje odpowiedzi na wspomniane pytania? Nie sądzę, aby z tego artykułu wynikały jakieś poważniejsze wskazówki co do celowości stosowania takich czy innych architektur. Jeśli chodzi o koncepcje architektury systemów cyfrowych - to nie ma w tym artykule nic nowego, co zresztą już podkreślił pan Kojemski. To, co autor podaje jest znane już z poważnych źródeł i co więcej - jest badane. Nawet pewne rozwiązanie sieciowe czy karuzelowe metody obsługi wynikają nie z tych prognoz rozwojowych, o których mówi Foster, lecz z poważnych badań, dawno zapoczątkowanych w dziedzinie masowej obsługi czy teorii kolejek. Może być tu jedynie mowa o pewnej subtelnej modyfikacji dobrze znanego podejścia. Z badań tych

wynika, co jest optymalną obsługą klienta, zgłoszeń lub procesu, a jeśli procesu, to dobrze zdefiniowanego i celowego. Muszą więc istnieć kryteria oceny jakości danego procesu. Innymi słowy trzeba dobrze wiedzieć to, o co nam chodzi oraz to, co mamy w końcu uzyskać w sensie zastosowań.

Przechodzę teraz do drugiej grupy luźnych uwag. Przypuśćmy, że zbudujemy wielkie sieci komputerowe. Strukturami sieciowymi już wiele lat zajmuje się nauka i technika, nie tylko z punktu widzenia możliwości zastosowania miniaturowej płytki w węźle sieci, ale nawet prostrzymi sytuacjami kiedy w węźle sieci znajduje się minikomputer w dzisiejszym tego słowa znaczeniu. Takie rzeczy są badane i rozważane. Dlaczego właśnie przeszedłem od procesu do tego miejsca? No bo nieźle już wiadomo, co to znaczy problematyka algorytmizacji i oprogramowania takiej sieci. Otóż zaryzykuję twierdzenie, że już dzisiaj można realnie myśleć o stworzeniu sieci złożonej nawet z tysiąca minikomputerów. I takie prace są prowadzone w świecie. Natomiast nadal istnieją problemy z rozdzieleniem procesu zrównaleglaniem zadań; musi być dobrze zdefiniowane wejście oraz wyjście muszą być podane funkcje, kryteria oceny jakości tego, o co nam chodzi, analiza, koszt, efektywność itd. bo przecież cała ta sieć czemuś ma służyć. Powstaje tu dodatkowe zagadnienie: jakie funkcje mają być zrealizowane w tej sieci, jak podzielić zadania między poszczególne węzły, jak sterować całością itd.itd. Widać cały ogrom prac podstawowych w tej dziedzinie, które to prace jak dotąd jeszcze nie dają konkretnych rezultatów przy liczbie węzłów większej niż 10. Nawet poniżej dziesięciu są naprawdę poważne trudności matematyczne i programowe. Naturalnie jest to niesłychanie interesująca, poważna problematyka, ale wymaga ona zaawansowanych badań. Nie chodzi o to, czy sprzęt ma być mniejszy czy większy, bo w pewnych zagadnieniach nie to decyduje. Oczywiście można mówić o kieszonkowych zamkach, ale nie wiem, czy w tej chwili, w najbliższym 10-leciu przynajmniej, najważniejszym problemem społecznym jest zamiana zamków na elektronowe. Natomiast sieć telekomunikacyjna, jak dobrze już dzisiaj wiadomo (zresztą wcale nie z pracy Fostera), powinna być jednolita, niezależnie od tego czy przekazujemy dane między komputerami, czy będzie to wymiana informacji, lub wiadomości między dwiema osobami. Powinna być standaryzowana i scentralizowana w sterowaniu czyli w obsłudze. Taki na przykład adapter głosu ludzkiego - śmiem w tej chwili twierdzić - nie jest potrzebny z punktu widzenia architektury maszyn, natomiast istnienie jego ułatwiać będzie w przyszłości rozwój zastosowań sys-

temów cyfrowych.

Chciałbym się odciąć od dyskusji na takie tematy, jak adapterek lub klucz, czy powiedzmy jeszcze jakiś bufor ze sterowaniem przy urządzeniu peryferyjnym, bo jest dla mnie oczywiste, że takie sprawy rozwiązywane są, będą i być winny, niezależnie od rozmyślań o architekturze systemów. Wracam do centralnego zagadnienia: architektura maszyn i w ogóle maszyny cyfrowe. Otóż myślę, że właśnie tutaj nie może być mowy o jakimś spontanicznym generowaniu zupełnie dowolnych struktur tych mikroukładów. Raczej musi to być mniej lub więcej spójna całościowa koncepcja. Zgodziłbym się w tym miejscu z tym, że zamiana jakichś poważniejszych bloków przetwarzania w jakichś okreolonych procesach na bliżej nieokreślone sieci mikroukładów w sensie zastosowań sieci neuronalnych - w dającej się obecnie przez nas przewidzieć perspektywie - nie jest alternatywą do poważnych badań w tym kierunku całościowych koncepcji. Nie oznacza to jednak, że ten kierunek nie powinien być badany także z pozycji absolutnej modularności, zarówno sprzętu jak i oprogramowania. Wracam do tezy, że każdy element systemu winien być rozumiany jako coś, co obsługuje dobrze określony fragment procesu, czyli znowu unia funkcji, sprzętu i oprogramowania przy optymalnie rozbitych modułach, przy czym dekompozycja ta musi być dokonana harmonijnie, zarówno z punktu widzenia jakości tego procesu jak i zwykłych elementów technologicznych. Pewne standardy, aczkolwiek nie optymalne dla tych procesów, jednak łatwiejsze w technologii i tańsze, będą oczywiście rzutowały na to, że optymalne struktury staną się quasi optymalnymi z innych względów. Jednak istniejące metody syntezy, w moim rozumieniu, nie tracą na aktualności nawet w świetle uwag Fostera.

I wreszcie ostatnia grupa moich spostrzeżeń odnosi się do strategii doganiania. Wydaje mi się, że jeśli strategia postępu i rozwoju w naszej dziedzinie byłaby jedyna i to prostoliniowa, to dopędzenie przodujących ośrodków nie byłoby realizowalne żadną metodą. Bo w naszych warunkach nie jest możliwe ani nadmierne przyspieszenie etapów rozwoju, ani przeskakiwanie etapów.Na ogół, jeśli istnieje już dobrze określona strategia, to przeskakiwanie etapów rozwoju może tylko pogorszyć a nie polepszyć rezultat. Sądzę, że istota rzeczy polega na czym innym. Strategie rozwoju w praktyce wcale nie są linią prostą, zazwyczaj istnieje jakaś krzywa reprezentująca daną strategię postępu i rozwoju i niesłychanie ważne jest nie przeskakiwanie etapów porządnej roboty (bo powiedziałem, że tego nie powinno się zrobić) , lecz właściwy wybór celu działania.

Na przykład z teorii pościgu wiemy, że z grubsza biorąc są tylko trzy podejścia. Jedno - zwane psią krzywą pogoni - jest mało efektywną metodą i w naszej dziedzinie, jak to już dało się nawet pokazać, nie doprowadza do celu. Drugim - jest strategia bezpośredniego wyprzedzenia, która sprowadza się do hipotetycznego ustalenia po jakiej krzywej konkurencyjny proces zmierza do określonego celu i wybranie własnej skrótowej drogi na wprost w celu ominięcia pewnych etapów rozwoju. Byłoby to najefektywniejsze, ale dobrze wiadomo, że tego rodzaju procesy nie są gładkie, a ponadto mają przegięcia o niewiadomych znakach. Wobec tego nietrudno się pomylić w wyborze własnej strategii i w ogóle nie doprowadzić do zbieżności tych strategii w żądanym punkcie.

Istnieje trzecia też dobrze określona metoda częściowego wyprzedzenia wraz z manewrem korygującym własną strategię w zależności od tego, co się dzieje z systematyczne badanym i studiowanym procesem rozwoju uznanym za wzorcowy. Jest to metoda chyba najlepsza. Jeśli tak, to nie obawiałbym się stwierdzenia, że nawet w sytuacji, kiedy ma miejsce dystans technologiczny, będzie można uzyskać o wiele lepsze rezultaty bez posiadania dostępu do pełnej technologii w sensie Fostera , niż przy jej posiadaniu lecz przy nieprecyzyjnie określonych celach i przy nieprzyjęciu punktu widzenia użyteczności całego działania. Nawet bym tu poczynił założenie, że pewien dystans technologiczny winien w naszych warunkach istnieć zawsze i nie opłaca się nam pokonywać tego dystansu bardziej niż do jakiejś ypsylonowej wartości. Oczywiście niezbyt odlegle bo nie konkurencyjnie ale i nie za blisko - bo za drogo. Była to ostatnia z grup - zastrzegam się jeszcze raz - zupełnie luźnych uwag, które mają jedynie charakter przyczynkowy i mają zachęcić do dalszej dyskusji".

T. K a m b u r e l i s

"Czuję się trochę sprowokowany do wypowiedzenia się na temat, jaka będzie przyszłość: czy będą to monokomputerowe czy też wielokomputerowe systemy. Uważam, że będą przeważały wieloprocesorowe czy wielokomputerowe systemy. Chyba będzie okazja sprostować to, że dotychczasowa "wielość" poszła w kierunku monokomputerowych systemów, w sensie wykonywania wielu procesów jednocześnie, np. wielu programów czy też obsługi wielu użytkowników. Jak wiadomo argumentacja historyczna była taka, że mamy powolne urządzenia zewnętrzne zaś szybkie procesory; wobec tego nakładano na nie wiele funkcji, wiele procesów. Powstają jednak wielkie straty czasowe właśnie przez to przechodzenie z jednego procesu na drugi.

Chyba w związku z potanieniem procesorów będzie okazja je rozdzielić i tym samym zlikwidować tradycyjną wieloprogramowość. Praktycznie wieloprocesorowy system wykonuje jeden proces danego użytkownika tak długo dopóki jest to możliwe. Inną sprawą jest potrzeba dokonania rzeczywiście szybkiego obliczenia w ramach jednego procesu. Dzisiaj się tworzy superwielkie komputery chyba po to, aby z olbrzymią szybkością rozwiązywać jakiś jeden konkretny program, jedno zadanie. Wydaje się, że byłaby możliwość skupienia wielu procesorów nad wykonaniem jednego programu, w bardzo krótkim czasie. Jasne jest, że muszą tutaj powstać odpowiednio efektywne programy czy też metody programowania, które pozwalałyby na równoległe zaprogramowanie i praktycznie równoległe wykonywanie takiego jedynego zadania.

Można sobie wyobrazić procesory o elastycznej organizacji, które w miarę potrzeby raz skupialibyśmy na wykonywaniu jednego dużego zadania, a innym razem na wykonywaniu wielu prostych zadań. Wydaje mi się, że praktycznie nie byłoby kłopotów, żeby to logicznie zrealizować. Między innymi taka organizacja komputera, która pozwalałaby na szybkie przerzucanie funkcji rozwiązywałaby też sprawę przeniesienia istniejącego oprogramowania, którego dzisiaj się dokonuje metodami emulacji. Można by mieć procesory specjalnie przeznaczone do wykonywania programów komputera o innej architekturze logicznej, tak długo jak to jest potrzebne, a potem wykorzystywać je do innych architektur".

A. J a n i c k i

"Czy tu nie wkradło się nieporozumienie? To co usłyszeliśmy jest chyba już dzisiaj bezsporne, leży to u podstaw modularności tych rozwiązań, o których mowa. Natomiast, jak mi się wydawało, myślą przewodnią u Fostera jest nie to, co tkwi już w wynikach prac w ramach - nazwijmy to - czwartej generacji. Wieloprocesorowość w ramach jednej konstrukcji, czy też w ramach modułów rozrzuconych przestrzennie, jest już ideą zaakceptowaną i rozwiązywaną. Czy więc tu chodzi o tę modularność? Tylko czy wtedy oznacza to dopasowanie pamięci i sterowań, i procesorów, i wszelkich perceptorów i elementów współpracy z człowiekiem? Architektura musi być konsekwentna. Stanowi ona część wyodrębnioną w procesie rozwoju, czy więc chodzi tutaj o sprawy sieciowe, tak jak u Fostera, gdzie właściwie koncepcja każdego elementu jest ściśle związana z miejscem i nawet czasem? Człowiek może przecież przebywać w różnych miejscach w określonym czasie z jego kieszonkowym urządzeniem bez konieczności np. przyłączania się do sieci".

T. K a m b u r e l i s

"Muszę tu przede wszystkim wyjaśnić, że należę do grona osób, które nie czytały artykułu Fostera, więc moje uwagi nie dotyczyły tylko tego artykułu. Natomiast to, co powiedziałem o koncepcji wieloprogramowości i wieloprocesorowości, tzn. przeznaczenia jednego procesora do wykonywania wielu procesów, czy też odwrotnie - było tylko luźną uwagą na ten temat".

A. J a n i c k i

"Dziękuję. To wyjaśnia".

R. M a r c z y ń s k i

"Dyskusja nad tym artykułem jest bardzo ciekawa. Na początku mówiłem o rzeczach niemerytorycznych, podałem tylko jedno zastosowanie, które - jak mi się wydawało, może poprzeć koncepcję Fostera bardzo taniej produkcji prostego a jednak bardzo wydajnego sprzętu. W tym samym artykule zresztą Foster podaje inne możliwe rozwiązania, o których tutaj nie chciałem mówić, bo to by nas zaprowadziło za daleko. Oczywiście dyskutowanie o wszystkich rzeczach byłoby właściwie metasystemem, a my tutaj zebraliśmy się po prostu po to - jak to zrozumiałem - żeby wyrazić jakiś pogląd, co należy zrobić, aby ewentualnie myślami z tych artykułów zapłodnić więcej ludzi.Organizatorom dyskusji nie zależało chyba na tym, aby każdy z obecnych przedstawił własne stanowisko, bo mnie się wydaje, że dwie godziny na to by absolutnie nie wystarczyło, gdyż jest to materiał na pół roku ciężkiej pracy przygotowywania. Wypowiadanie własnych poglądów jest teraz bardzo trudne. Wracając do tego, co mówił pan dyrektor Janicki w sprawie produkcji systemów wielomaszynowych: oczywiście są takie prace, można między innymi podać taki projekt, który nazywa się "Minimachines-computer". Te prace są prowadzone na kilku uniwersytetach, gdzie wykonuje się sieci minimaszyn i bada różnorodną problematykę. Poza tym wiadomo, że w systemie ARPA pewne elementy łączy się za pomocą małych maszyn.

Chciałbym jeszcze dodać jedno. Nie zgodziłbym się z pewnymi wypowiedziami, które padły tutaj. Mam na myśli to, że my chcemy wszystko standaryzować, ujednolicać, wszystko porządkować. Uważam, że przyszłość w

ogóle nie tkwi w standaryzacji. Dlaczego? Dlatego, że człowiek powinien sobie ułatwiać życie a nie komplikować. Wiadomo, że na początku była Wieża Babel, później ludzie się rozeszli, do dnia dzisiejszego mówimy różnymi językami i jakoś nie potrafimy się porozumieć. Mnie się wydaje, że kierunek powinien być taki (podkreślam, że to jest moje osobiste zdanie): w zasadzie urządzenia powinny nam pomagać w porozumiewaniu się, w przekształcaniu jednej klasy informacji na drugą i raczej tego oczekujemy od przyszłości. Jeślibyśmy zgodzili się z tym podejściem, to istnieje tu znowu pewien aspekt praktyczny, który już oczywiście zrealizowano, np. w systemie ARPA. Po prostu mamy różne, pracujące w różnych językach maszyny, które są łączone za pomocą urządzeń przetwarzających jeden język na drugi. Jeśliby więc istniały bardzo małe i tanie urządzenia i gdybyśmy pokonali problemy tłumaczenia, to w zasadzie problem tłumaczenia z jednego języka naturalnego na drugi też może się okazać możliwy do zrealizowania. Mówię o tym dlatego,że wszyscy dążymy do standaryzacji,porządkowania, a jednocześnie wszyscy od tej standaryzacji odchodzimy i coraz bardziej się jej opieramy. Czy nie warto więc założyć, że nie można standaryzować, lecz tylko dopasowywać przez urządzenia pośredniczące. Myślę tu nie tylko o standaryzacji języków czy maszyn, lecz też o standaryzacji pewnych pojęć czy informacji za pomocą urządzeń przetwarzających. Powiedzmy sobie, że jest to taka myśl zupełnie "nieuczesana", ale w każdym razie właśnie to mam na myśli".

J. D a ń d a

"Chciałbym dodać do tego swoje zdanie. Jesteśmy rzeczywiście na etapie standaryzowania czego się da i jak tylko się da. Szczególnie uważamy za duże osiągnięcie ostatnich lat na przykład Standard Interface. Otóż to rozwiązanie, jeśli próbować zastosować tę ogólną zasadę, którą docent Marczyński podał, można określić jako właśnie bardzo tradycyjne. Jeśli mielibyśmy bardzo tani mikrokomputer, którego koszt byłby znikomy, to można w ogóle zlikwidować kwestię standardowego interface wkładając takie mikrokomputery[1] między urządzeniem a jakimiś częściami wszędzie tam, gdzie ma być coś standardowego. W miarę rozbudowania standardów interface'owych, już nie tylko standaryzujemy poziomy sygnałów w jednej chwi-

[1] I.M. Barron przedstawia podobną koncepcję mówiąc o "blister computers" na str. 326 artykułu "Network Systems: the Economic Solution for the Fourth Generation User", opublikowanym w zbiorze "The Fourth Generation" INFOTECH Ltd, Maidenhead, Berkshire, 1971.

li czasowej, ale zaczynamy standaryzować sekwencje sygnałów. Albo więc musimy kwestiom standaryzacji poświęcić olbrzymią uwagę, żeby wymyślić uniwersalny interface dla wszelkich możliwych urządzeń, albo też w urządzenia musimy pakować wiele zbędnego sprzętu.

Przechodzę teraz do spraw ogólniejszych. Chciałbym polemizować tutaj ze zdaniem Pana Mardala, sceptycznie odnoszącego się do tego, że sieci złożone ze względnie standardowych elementow (w tym przypadku będą to maszyny), zupełnie wyprą maszyny duże. Otóż wydaje mi się, mimo założenia wspaniałego postępu w automatyzacji projektowania, że możemy dojść do takiego punktu, kiedy po prostu dużej maszyny nawet bardzo zwarty zespół nie potrafi zaprojektować. Właściwie już maszyny wielodostępne jako urządzenia pojedyncze osiągnęły rzeczywiście olbrzymi stopień komplikacji. I co się okazało? Takie urządzenie nie mogło być zaprojektowane od razu dobrze. Zaniedbano przy tym jedną z podstawowych zasad dobrej sztuki inżynierskiej: trzeba robić szybko, następnie trzeba sobie stworzyć możliwość obserwacji tego co się zrobiło i szybko robić następny poprawiony model.

Oczywiście, po jakimś czasie pojawił się duży nacisk na programowe środki zbierania danych o pracy systemów wielodostępnych. Ale okazało się, że wprowadzenie programu, który zbiera statystyczne dane o pracy systemu, tak zakłóca w wielu przypadkach jego pracę, że uzyskuje się wyniki zniekształcone obserwacją. Powoływano się nawet w tym kontekście na zasadę nieoznaczoności Heisenberga, prawdopodobnie humorystycznie. Producenci dużych systemów liczących zaczęli stosować specjalne urządzenia cyfrowe, które przyłączano do maszyn wielodostępnych w celu zbierania pewnych informacji, a nawet niektóre firmy minikomputerowe wyspecjalizowały się w produkcji takich przystawek do obserwacji pracy systemu.

Rzeczywistość zmusiła do przyjęcia pewnych rozwiązań, które - gdyby lepiej pomyśleć - można było od razu wbudowywać. Drugi przykład. Jest paradoksem, że interesujemy się pracą systemów wtedy tylko, gdy dany system znajdzie się w sytuacji awaryjnej, natomiast nie interesujemy się pracą systemu, kiedy on działa dobrze. Dobry inżynier największą ilość czasu poświęca na pozornie bezproduktywne obserwowanie jak urządzenie działa. Natomiast jeśli weźmiemy pod uwagę grupę programistów to wysuwam hipotezę, że programista interesuje się pracą swojego programu tylko wtedy, kiedy ten program działa niezgodnie z jego założeniami. Dlaczego? Dlatego, że nie ma nawet takich narzędzi, jakie ma inżynier elektronik, żeby zobaczyć co się dzieje. Czy jest to uzasadniona sytuacja? Nie, i

jeszcze raz nie. Już 15 lat temu stworzono podstawy "computer-graphics", były to między innymi prace Licklidera z 1962 roku, w których właśnie zwrócono uwagę na wykorzystanie monitorów graficznych w programowaniu. Jeśli przejrzymy ogólne czy specjalistyczne czasopisma, to niewiele znajdzie się artykułów na temat stosowania "computer-graphics" w maszynach. Z tego, jak mi się wydaje, tylko 1% dotyczy zastosowania "computer-graphics" jako narzędzia pisania programów.

Teraz chciałbym przejść do jeszcze ogólniejszych zasad projektowania inżynierskiego. Weźmy pod uwagę podejście Nadlera (metoda Nadlera była referowana w "Informatyce"[1]). Zwykle jeżeli coś chcemy skonstruować, to na ogół poszukujemy nowego rozwiązania i jakoś je modyfikujemy, zaś Nadler proponuje żeby nie zastanawiać się nad tym, co już mamy i jak to ulepszyć, tylko zastanowić się, czy możemy sobie stworzyć idealny obraz tego, co nam jest naprawdę potrzebne. Wymaga to oczywiście pewnego treningu myślowego, żeby pozbyć się ograniczeń, które wynikają z pewnego rodzaju nawyków myślowych. Wprawdzie Nadler mówi na początku, że taką naprawdę dobrą zasadą jest: zastanów się, jeżeli powstaje problem, czy tego problemu w ogóle nie możesz zlikwidować, może go w ogóle nie ma. Żartem mówiąc w naszym przypadku, może powinniśmy byli rozejść się przed rozpoczęciem dyskusji.

Spróbujmy jednak zastosować metodę nadlerowską do problemu, o którym mówi Foster. Otóż można byłoby zacząć poszukiwanie jakiegoś idealnego rozwiązania. Ostatnio jest to dosyć modne i nawet wyrosły całe gałęzie nauk interdyscyplinarnych oparte na takim założeniu, że w pewnym sensie idealnymi są rozwiązania ewolucyjne. Na tej zasadzie rozwinęła się m.in. bionika jako nauka czysto aplikacyjna, która chciała podpowiadać inżynierom, jak konstruować pewne układy. Nie ulega chyba prawie żadnej wątpliwości, że to podejście, które próbowało szukać analogii aż na poziomie neuronalnym, chyba nie jest specjalnie płodne; wydaje mi się, że większość z Państwa także tak uważa. Ale takie osiągnięcia, jak choćby rozwój teorii perceptronów, który był bardzo ściśle oparty na pewnych faktach podstawowych z dziedziny sieci neuronalnych wskazuje, że to nie jest aż tak bezpłodne zajęcie. Mimo że nie ma dotąd perceptronów jako urządzeń,to

---

[1] Kijak Z.: Metoda Nadlera w projektowaniu systemu EPD. Informatyka, 1972, nr 9, s. 6-8.

przynajmniej powstała bardzo ciekawa gałąź informatyki teoretycznej[1].

Wydaje się, że prawie całe podejście perceptronowe i jego praktyczne znaczenie zostanie zupełnie obalone albo ograniczone do kwestii teoretycznych, ze względu na to, że wykorzystanie światła koherentnego otwiera zupełnie nowe możliwości w dziedzinie rozpoznawania pewnych względnie zestandaryzowanych obiektów, takich jak np. litery i to litery dość poważnie zniekształcone.

Wracając do zagadnienia kontroli pracy urządzenia, które nie jest w stanie awarii: spójrzmy na pewne nowe osiągnięcia teorii, nie na poziomie neuronalnym ale na poziomie psychologicznym, na osiągnięcia teorii emocji. Okaże się, że są duże analogie. To, co psychologowie już odkryli w dziedzinie teorii emocji i budowy systemu emocjonalnego ma jakąś przedziwną analogię z tym, co wydaje się, w najbliższym okresie trzeba będzie wbudowywać do maszyn. Jednak nie chcę dalej tego rozwijać w tej chwili - może będą inne okazje.

Zgadzam się ze zdaniem doktora A. Janickiego, że artykuł Fostera sugeruje takie rozwiązania, które u nas wydają się zupełnie nieuzasadnione. Ale z drugiej strony, ponieważ autor popuścił wodze fantazji, to i nas skłania do podobnie swobodnego myślenia. Zacytuję z wypowiedzi Lyncha[2] coś, co mi się niesłychanie spodobało. Mówi on, że podstawowa zasada inżynierska jest taka: jeśli nie rozumiesz tego, co masz zaprojektować, zaprojektuj to, co rozumiesz.

Dlaczego mówię to w związku z artykułem Fostera: dotąd nie wiemy, jak powinniśmy budować olbrzymie scalone systemy przetwarzania informacji, a równocześnie jako inżynierowie (a nie naukowcy) musimy coś zrobić. W związku z tym projektujmy to, co potrafimy zaprojektować, składajmy, obserwujmy i udoskonalajmy. Na drodze zestawiania dużych systemów minikomputerów cel wydaje się łatwiejszy do osiągnięcia".

---

[1] Marvin Minsky, Seymour Papert: "Perceptrons. An Introduction to Computational Geometry". Cambridge, Massachusetts and London 1969, The MIT Press.

[2] Lynch W.C.: Operating System Performance. Comm. ACM, 1972, t. 15, nr 7, s. 579; patrz również opracowanie tego artykułu: ETO Nowości, 1973, nr 2, s. 55.

A. J a n i c k i

"Kolega Marczyński wypowiedział o problemie standaryzacji zdanie, które jak sądzę wymaga wypowiedzenia jeszcze innego zdania, po to, aby problem lepiej przemyśleć i być może innym razem poważnie porozmawiać. Od czasów Homera - jak ktoś pięknie powiedział - człowiek zmienił się niewiele. Nie tylko humanistyczne aspekty naszego życia, aktualne hasła o ochronie środowiska, ale po prostu właśnie teoria użyteczności, która już ma i swój aparat i konkretne wyniki, mówi wyraźnie, że wszystko co czynimy ma być dla nas użyteczne, czyli nie powinniśmy padać ofiarą tego molocha, który tworzymy. Dlatego też dla mnie bezsporny jest wygłoszony tutaj punkt widzenia, że jeśli standaryzacja miałaby być naszym celem, to byłoby to straszne. Wydaje mi się jednak, że cokolwiek by nie mówić i pięknymi słowami ubierać - nie wyjdziemy poza pewne wyniki, które nam zaprezentowały porządne teorie. Mam na myśli np. teorię systemów, która wyrosła nie na gruncie maszyn cyfrowych, bo te początkowo były sztuką a nie inżynierią i nie miały z nauką zbyt wiele wspólnego, dopiero teraz tworzy się podwaliny naukowe. Teoria systemów wyrosła na gruncie teorii komunikacji, gdzie jak dobrze wiadomo, w każdym modelu można przyjąć jeden z dwu punktów widzenia: strony nadawczej i strony odbiorczej. Naturalnie, jeśli wszystko jest dobrze zrobione, to można sobie przeliczyć wielkości charakteryzujące te punkty widzenia jeden na drugi i dojść do wzajemnie przeliczalnych określeń czy wniosków (np. prawdopodobieństw á priori, jeśli się z jednej strony patrzy i prawdopodobieństwa á posteriori, jeśli się patrzy z drugiej strony).

Otóż jeżeli popatrzymy z punktu widzenia nas - jako użytkowników, którym to wszystko co wymyślimy ma służyć - to precz ze standaryzacją. Przykładowo będzie zatem dla wszystkich oczywisty następujący aspekt czwartej generacji systemów komputerowych, aczkolwiek nie jest ona jeszcze dobrze zdefiniowana. Tak jak w pierwszej generacji mieliśmy jako wejście tylko monitor dalekopisowy czy też drukarkę, czy inne urządzenie, jak mamy w drugiej generacji już rozbudowane wejścia łącznie z monitorami ekranowymi, jak w trzeciej generacji istnieją grafoskopy, wyjścia graficzne właśnie typu zobrazowań a można podać liczne prace, gdzie się mówi o zupełnie innym, nietradycyjnym lecz również elektronowym charakterze zobrazowania - tak w czwartej generacji konieczne jest sprzężenie foniczne. Czyli do czego zmierzamy? Czy nie do odejścia od standaryzacji?

Ale jeśli przyjęć punkt widzenia systemu, który ma realizować funkcje przez nas oczekiwane, to standaryzacja staje się niesłychanie pomocnym narzędziem. Wobec tego z pewnością skazani będziemy na adapterki podłączane wszędzie tam, gdzie kończy się system, który ma nam służyć, a my się do niego przyłączamy. Naturalnie tam jest przede wszystkim miejsce na swobodę wynikającą z naszej psychiki, z naszych upodobań i z celu naszego życia. Natomiast chciałbym podkreślić, że świat komputerów i wszystko to co dalej w głąb systemu od miejsca, gdzie podłączony został adapter, powinno być objęte standardami nie tylko ze względów technologicznych, lecz również i dlatego, abyśmy nad tym światem komputerów i w przyszłości panować mogli".

R. M a r c z y ń s k i

"Moje zdanie nie było zaprzeczeniem standaryzacji, podałem tylko pewną filozofię. A teraz pozwolę sobie przeczytać niewielki ustęp z opracowania, które wykonałem dla Komitetu Informatyki w związku z II Kongresem Nauki Polskiej.

Rozwój informatyki z jednej strony jest uwarunkowany stanem techniki i technologii, a z drugiej strony - stanem wiedzy i informacji społeczeństwa oraz jego kulturą techniczną. Stosunkowo łatwo jest przenieść do danego społeczeństwa technikę przez jakiś zakup licencji, znacznie trudniej zmienić nawyki u ludzi. Doszukując się analogii z maszynami cyfrowymi, można powiedzieć, że rozwój jest ograniczony brakiem dopasowania między dwoma niejednolitymi układami, maszynami o różnej strukturze i czasie życia, ludźmi a komputerami. Człowiek ma stałą strukturę i można przyjąć, że od początku naszej cywilizacji a może i wcześniej - niezmienną. Okres życia człowieka wynosi 60-70 lat. Komputer jest urządzeniem szybko zmiennym. W ciągu 25 lat rozwoju tej dziedziny byliśmy świadkami co najmniej czterech różnych struktur coraz bardziej wyrafinowanych. Okres życia komputera szacuje się średnio na około 5 lat. Ale dany komputer można całkowicie zmienić przez zmianę programu. Jak widać z tego, łatwiej jest dopasować komputer do człowieka niż odwrotnie. Pozostaje też inne zjawisko: 25 lat maszyn cyfrowych nie pozostało bez wpływu na człowieka, który również powoli dopasowuje się do komputera. Jednak zbyt szybkie zmiany maszyn prowadzą do różnych napięć, do kryzysów, których w miarę możliwości należałoby uniknąć. Tematyka badawcza takiego niedopasowania należy raczej do psychologii. Zagadnienie to wysunąłem dlatego, że ciągła pogoń za nowoczesną technologią, którą na przestrzeni lat obserwujemy, ciągle

idzie dalej, chyba że podejmiemy takie środki, które ten rozdział zlikwidują, a jeśli nie zlikwidują, to zmniejszą, a tym samym zmniejszą straty społeczne. Znalezienie środków dla rozwiązania tego problemu uważam za jedno z ważniejszych zadań nauki".

W. K l e p a c z

"Człowiek jest niewolnikiem pewnych pojęć, które przejmuje od świata zewnętrznego. To samo jest w zastosowaniach. Przewijało się to w wypowiedziach przedstawicieli producentów, o tym także mówi Foster - jest to marzenie o jakichś nowych zastosowaniach. Ale nie wiadomo, czy już dojrzeliśmy do tych nowych zastosowań. I w pewnym momencie chyba w wypowiedziach producentów wychwyciłem takie przypuszczenie, że dopiero ten rozwój zastosowań będzie szerszy, jeśli komputery wejdą właśnie do sfery prywatnej. Silnik elektryczny kiedyś był zupełnie poza sferą prywatną, a teraz stał się narzędziem w domu, w gospodarstwie. Telefon kiedyś znajdował się tylko w sferze organizacji, w przedsiębiorstwach. W naszej dziedzinie też musi nastąpić przełom. Osobiście wierzę, że podobnie jak mamy magnetofony kasetowe, tak samo każdy prawdopodobnie będzie miał w przyszłości jakąś taką przystawkę, infotekę, z której będzie mógł otrzymać informacje nagrane na jakichś kasetach, albo też będzie mógł się przyłączyć do jakiegoś systemu. Sądzę, że te sprawy chyba dość szybko się pojawią. Wydaje się, że będzie tu właśnie działał nacisk tego szalonego postępu technicznego, miniaturyzacji, potanienia. Wszędzie podkreśla się, że te płytki będą coraz tańsze, co spowoduje powszechną ich dostępność. Jest tylko problem, który znów wszędzie się przewija, że chyba człowiek jeszcze nie potrafi organizować tego w całość. Pojawiają się jednak już obecnie konkretne propozycje osiągnięcia celu powszechności przez standaryzację bardzo prostych urządzeń końcowych, podobnych do rozwiązań automatów, do których wrzucamy pieniądze. I to będzie chyba pierwsza droga wejścia człowieka do tych wielkich systemów informacyjnych, którymi będzie mógł wspierać swoją działalność i zawodową i ściśle osobistą".

J. D a ń d a

"Chciałem jeszcze podać dwie ciekawostki, które wiążą się z tymi zagadnieniami. Pierwsze ilustruje to, co zresztą już omówił szeroko docent Janicki, że my nie możemy bezkrytycznie przejmować pewnych rzeczy z ka-

pitalizmu. Producenci minikalkulatorów bardzo się starali o obniżenie ceny poniżej 100 dolarów. Jest to niewątpliwie jakiś efekt psychologiczny, gdyż chodziło o rzecz, która wynikała z dokładnego zbadania tradycji amerykańskich. Japończycy chcieli wejść na rynek amerykański i firmy, które z tego żyją, firmy konsultacyjne, wyszperały zwyczaj, zresztą powszechnie znany w USA, że dzieciom do lat 14 czy 18 nie daje się podarków z okazji świąt Bożego Narodzenia w cenie powyżej 100 dolarów. W związku z tym ocenili - i chyba trafnie - że jeżeli obniżą kalkulatory poniżej 100 dolarów, to się otworzy na nie olbrzymi rynek. I nie omylili się. Właśnie w okresie świąt Bożego Narodzenia skoczyły niesłychanie zakupy kalkulatorów.

Taki mechanizm u nas raczej nie będzie działał. Chociaż - zwróćmy uwagę, że na magnetofony Grundiga był początkowo bardzo mały popyt, a gdy obniżono cenę poniżej pewnej magicznej kwoty, którą człowiek jest przyzwyczajony wydać bez wielkiego zastanawiania się, od razu Grundigów zabrakło w sklepach. Nie można więc powiedzieć, żebyśmy się tak zupełnie inaczej zachowywali.

Druga sprawa, która - muszę powiedzieć - mnie bardzo zaskoczyła. Często jako konstruktorzy ubolewamy nad tym, jeżeli jakieś urządzenie nie wchodzi do produkcji - uważamy, że stało się źle. Chyba rzeczywiście w warunkach naszego rynku konstrukcji i rynku możliwości produkcyjnych, powinniśmy dążyć do wykorzystywania każdej konstrukcji, bo nie mamy zbyt wielu konstruktorów. Wydawałoby się, że szczególnie na tę stronę zagadnienia będą zwracały uwagę duże firmy światowe. Tymczasem tam uważają, że jest bardzo źle: nie jest zrównoważony zasób ludzki w firmie, jeżeli np. 30% opracowań nie idzie na półkę. Uważają, że tylko wtedy mogą zdobyć rynek, jeśli wyprodukują nadmiar pewnych pomysłów, z których wiele nie będzie realizowanych.

Proszę zwrócić uwagę: to jest informacja z "Transactions on Engineering Management"[1] , takie rzeczy się bada. Wydaje mi się, że jeśli tylko krytycznie na taką rzecz spojrzeć (ja nie potrafię w tej chwili ustosunkować się, raczej podaję to pod osąd państwa), to być może okaże się, że niektóre nasze stereotypy myślowe są fałszywe. Sam byłem osobiście prze-

---

[1] N.R. Baker, J. Siegman, J. Larson: "The Relationship Between Certain Characteristics of Industrial Research Proposals and Their Subsequent Disposition", IEEE Trans. on Engineering Management, t. EM-18, 1971, nr 4, s. 118-124.

konany, póki tego nie przeczytałem, że jeśli coś z opracowanych konstrukcji poszło "na półkę", to było bardzo źle. W tej chwili ten artykuł wzbudził u mnie jakiś proces myślowy".

A. K o j e m s k i

"Chodzi chyba o uświadomienie pracownikom, że jeżeli coś jest opracowywane w równoległych zespołach to po to, aby wybrać z kilku rozwiązań najlepsze i wykorzystać je do produkcji. Gdybyśmy oceniali jakość instytucji po tym, ile jej się udało odłożyć "na półkę", to byłoby to chyba spojrzenie od odwrotnej strony niż normalnie powinno się patrzeć. Dobra firma powinna być tak silna, aby była w stanie skonstruować pięć modeli samochodów, a jeden tylko wprowadzić na rynek. Firma, która tego nie może zrobić, produkuje model odpowiadający tylko jednemu wypróbowanemu rozwiązaniu i robi go dlatego takim, że zwykle nie może konkurować z innymi. Na tym polega słabość tej firmy".

J. D a ń d a

"Mogę dorzucić, że prawdopodobnie firmy chcą mieć jak najmniej składanych na tę "półkę" konstrukcji. W gruncie rzeczy wskaźnik mówiący o odkładaniu na półkę można traktować jako jeden z tych globalnych wskaźników, które mówią coś o przedsiębiorstwie. Były to badania wykonywane w licznych firmach; badano korelacje czy firmy, które mają mało "na półce", odnoszą sukcesy czy nie. Okazało się, że jest pewna liczba "półkowania", pewien procent tych pomysłów "półkowanych", który charakteryzuje firmy odnoszące sukcesy. W ten sposób zostało to w tym artykule przedstawione".

W. K l e p a c z

"Jeśli zbyt duży odsetek jest przeznaczony na półkę, to niestety grozi bardzo smutnymi konsekwencjami dla takiej firmy, tzn. likwidacją, przejęciem przez kogoś zręczniejszego. Oczywiście, jest to pewna część wszystkich kosztów działalności, bo często są pomysły bardzo piękne ale nie do realizacji, ze względów ekonomicznych czy innych".

J. D a ń d a (zwraca się do mgr Kamburelisa)

"Czy mógłbyś - bo jesteś jednak przedstawicielem potężnej, jak na skalę Polski firmy w naszym gronie - coś ze swoich doświadczeń powiedzieć na ten temat?"

T. K a m b u r e l i s

"W ELWRO około 95% konstrukcji idzie do produkcji, a około 5% na półkę. My także mamy wątpliwości, czy to jest dobrze czy źle".

J. D a ń d a

"Ale jak wygląda sprawa tych nieuniknionych problemów psychicznych, które wynikają dla twórców "zapółkowanego" tematu. Czy jakoś sobie z tym radzą?"

T. K a m b u r e l i s

"Sądzę, że nawet bywają dramaty. Może podam tu konkretny przykład: ostatnio powstało u nas urządzenie własnej produkcji, moim zdaniem technicznie udane. Natomiast administracja stwierdziła, że nie jest ono ekonomiczne. I mimo pozytywnej próby zaniechano tego opracowania, co skończyło się szumnym wyniesieniem się twórców tego opracowania z "ELWRO", szukaniem kontaktów w celu ulokowania swojego opracowania i wreszcie znalezieniem poparcia gdzie indziej. Tam właśnie trwają próby uruchomienia tego wyrobu, którego opracowanie powstało w ELWRO i tu zostało przebadane. Twórcy to jakoś przeżyli, do dzisiaj nie poddali się, i być może jest szansa, że wygrają ten mecz. Chyba ten przykład jest dość charakterystyczny".

B. K a s i e r s k i

"Mnie się zawsze wydawało, że sytuacja kiedy 100% opracowań idzie do produkcji jest niekorzystna, chociażby z ekonomicznego punktu widzenia. Wiadomo przecież, że inercja przemysłu, szczególnie u nas, jest tak duża, że potem jakakolwiek zmiana już po wprowadzeniu do produkcji jest prawie niemożliwa. Trudne też jest doskonalenie. W stosunku do kosztów

seryjnej produkcji, koszt opracowania jest dużo mniejszy, powinno się więc w wielu sytuacjach nawet równolegle opracowywać projekty. Z organizacyjnego punktu widzenia trudno mi sobie wyobrazić zupełne dublowanie prac; również z punktu widzenia ambicjonalnego. Powstają potem sytuacje konfliktowe - trzeba przyjąć jeden lub drugi projekt. Ale jestem przekonany, że dublowanie każdej pracy byłoby ze wszech miar korzystne, w każdej sytuacji".

W. M a r d a l

"Dlaczego konstruktorzy przeżywają dramat? Chyba dlatego, że wychowaliśmy ich od początku kariery zawodowej w takim przekonaniu, iż tylko produkcyjne wykorzystanie ich opracowania jest jedynym źródłem satysfakcji autorów. Żaden inny obrót sprawy więc nie przyniesie im satysfakcji zawodowej. I to jest chyba nieprawidłowe.

Są dwa pierwiastki każdej pracy: jeden to względy ekonomiczne, które mogą spowodować zaniechanie dalszych prac; drugi pierwiastek - to jest wkład, jaki ten zespół badawczy czy konstrukcyjny wnosi w ogóle w postęp danej dziedziny. Często prace prowadzone nad jakimś urządzeniem mogą być wykorzystane w innej dziedzinie, jak również wnioski i doświadczenia, które z tego wynikają. To, że ludzie się wykształcili mogą służyć radą przy rozwiązywaniu wielu problemów, powinno złagodzić upadek, spowodowany niepowodzeniem produkcyjnym. Lądowanie powinno być "miękkie" a nie takie brutalne, że ludzie rozstają się z instytucją, zmieniają region, pewnie mają potem problemy z rodzinami, z czym wiążą się inne dalsze poważne problemy. Jest tu chyba jakieś złe nastawienie, a atmosfera wokół podejmowanych nowych prac i pokazywanie źródeł satysfakcji jest chyba niewłaściwe!"

S. M a j e r s k i

"To jest chyba sprawa organizacyjnego ustawienia prac i to od samego początku. Jeżeli jakaś grupa osób zajmuje się danym zagadnieniem, to trzeba aby te osoby wyraźnie nastawiły się, że będzie to jedna z wersji, którą później rozpatrzy się celem dokonania wyboru. Jeśli opracowane urządzenie będzie porównywane z innym produkowanym już czy opracowywanym i parametry będą gorsze, to potem inaczej wygląda ta chwila,

kiedy konstruktorzy dowiadują się, że produkt ich pracy został odrzucony. Często jednak bywa sytuacja, że przemysł jest nieprzystosowany do produkcji lub odgrywają rolę akurat inne jakieś uboczne czynniki. Przypomniałem sobie czasy, kiedy jeszcze u nas, w Instytucie Maszyn Matematycznych, opracowano maszynę ZAM-2. Były opracowywane co najmniej dwie lub trzy wersje. W rezultacie dyskusji, przyjęto jedną z wersji i jak dzisiaj wiadomo, była to decyzja pozytywna, jako że ta maszyna została wyprodukowana w kilkunastu egzemplarzach. Równolegle z tą bardzo prostą maszyną była opracowana wersja, bardziej skomplikowana, która mimo iż była bardziej perspektywiczna od pierwszej, to jednak przy ówczesnym stanie niezawodności elementów i ograniczeniu możliwości urządzeniowych musiała być odrzucona. Obyło się wtedy bez kłopotów dlatego, że od samego początku było wiadomo, że wiele czynników będzie decydować o tym, która wersja będzie przyjęta".

J. D a ń d a

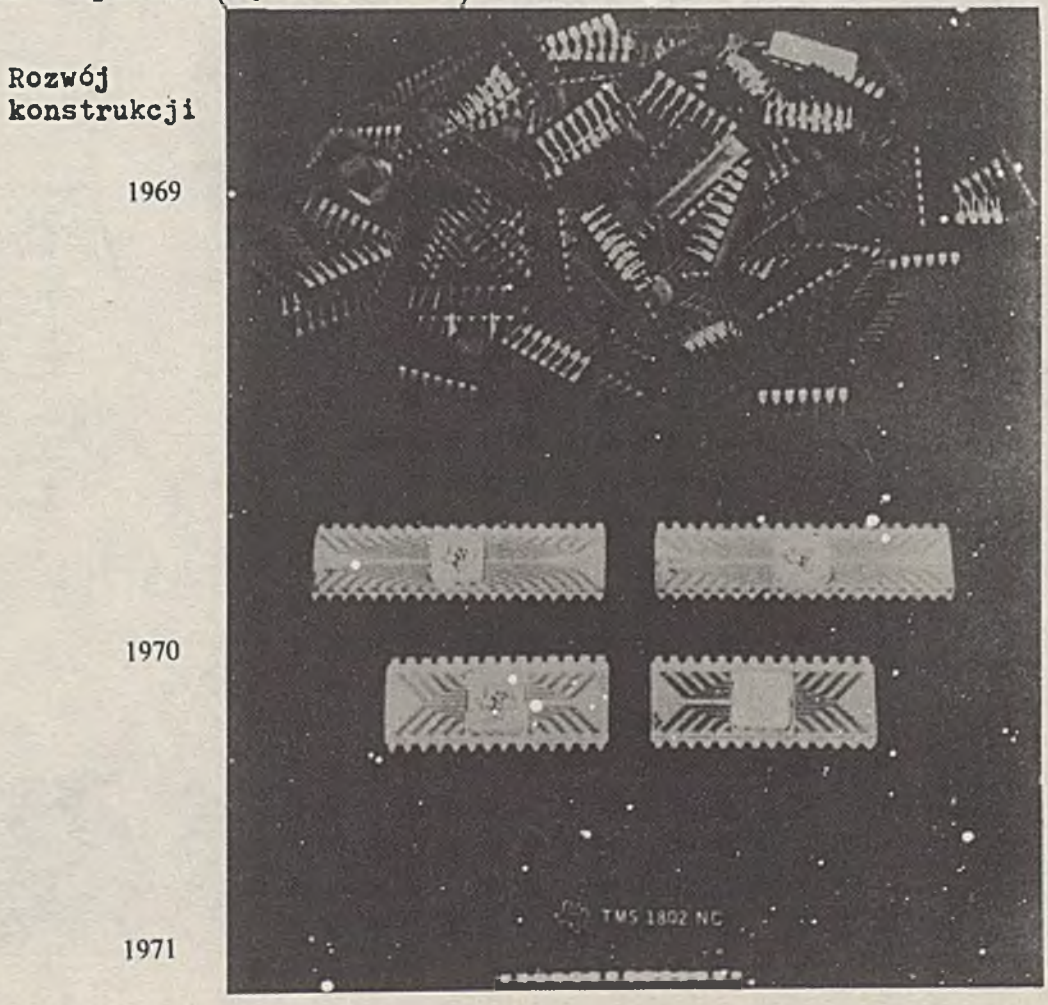
"Tak więc z problematyki perspektywicznej, wróciliśmy do historii, co oznacza, że dyskusję należy na tym przerwać. Dziękuję wszystkim".

Na podstawie stenogramu
opracowała Dorota Prawdzic

PERSPEKTYWY ROZWOJU ARCHITEKTURY KOMPUTERÓW[1]

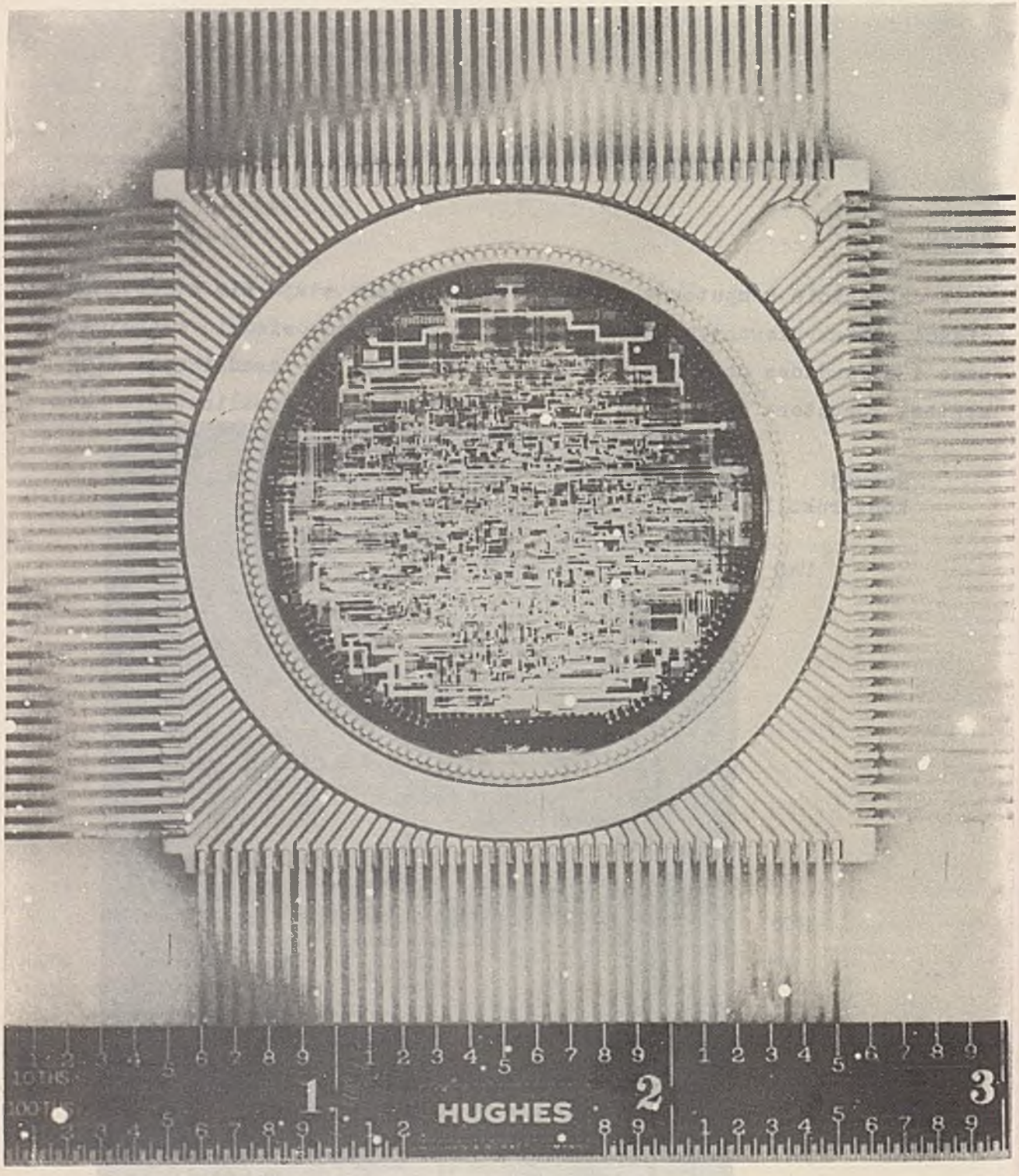
Wstęp

W dziedzinie komputerów coraz powszechniejsze staje się stosowanie części prefabrykowanych. Zaczynają być dostępne odpowiednio zaprojektowane i sprawdzone podzespoły funkcjonalne, które są zasadniczymi fragmentami komputera (rys. 1, 2 i 3). Możliwe jest już zrealizowanie kom-

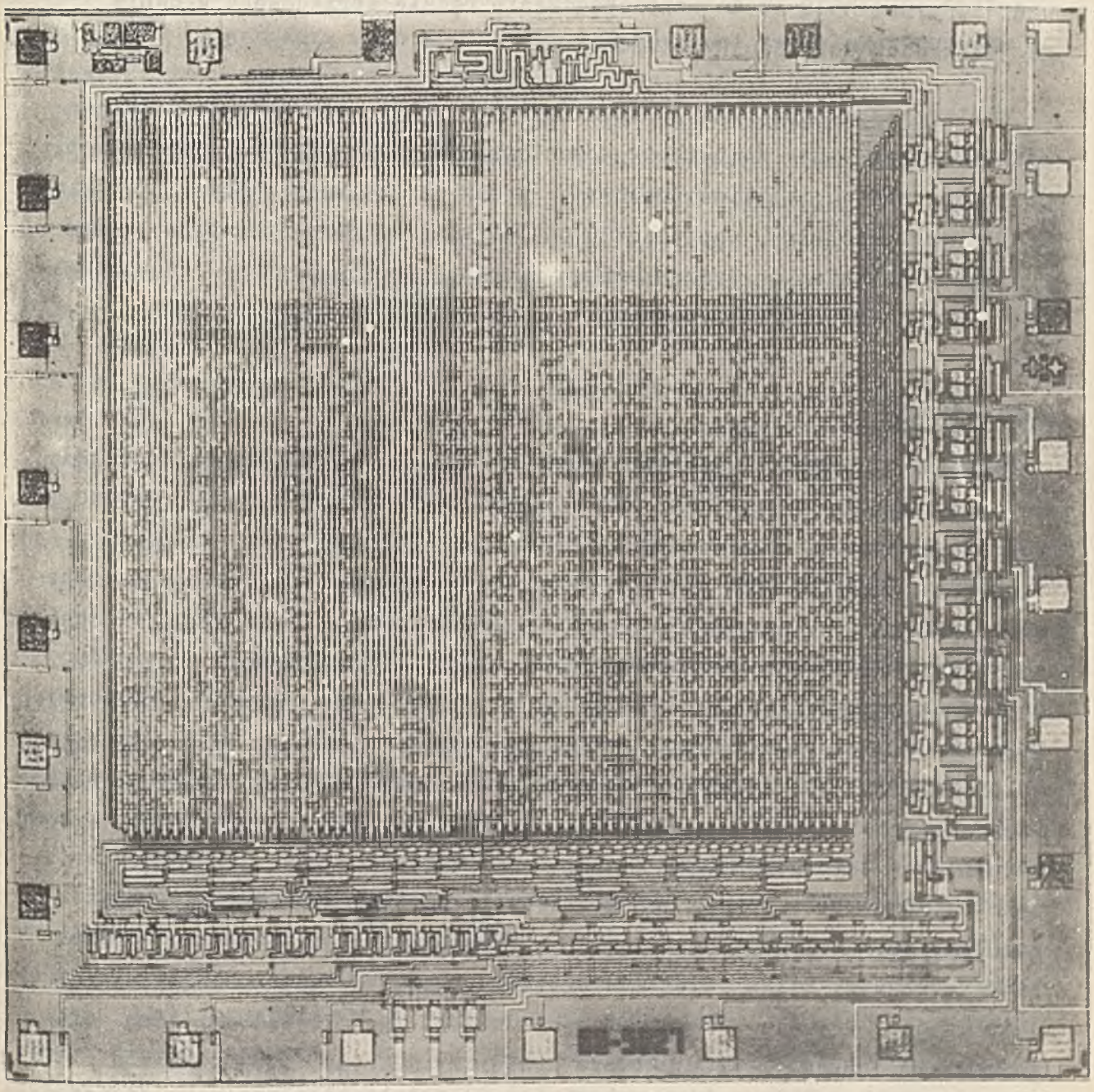


Rys. 1. Płytki niezbędne do wykonania kalkulatora

[1] FOSTER CAXTON C.: A View of Computer Architecture. Communications of the ACM, 1972, nr 7, s. 557-565. Opracowanie: mgr inż. Andrzej Kojemski.



Rys.2. Dynamiczna pamięć MOS typu ROM o 12228 bitach firmy Electronic Arrays. Niezbędny jest zegar wejściowy tylko o jednej fazie, gdyż układy umieszczone na płytce dostarczają dwóch faz wymaganych przez pamięć



Rys.3. Ośmiobitowy układ mnożący LSI zawierający 950 elementarnych układów na płytce

pletnego komputera w postaci płytki[1]. Dostępne są już minikomputery z pamięcią o pojemności kilku kilobajtów, na małych płytkach mieszczących się w dłoni. Nie ma podstaw, aby przypuszczać, że tendencje rozwoju konstrukcji pokazane na rys. 1 nie będą się utrzymywały, tak że przed 1975 r., to co nazwiemy mikrokomputerem będzie zbudowane tylko na jednej płytce i będzie kosztowało ok. 100 dolarów. Można przewidywać, iż przed 1980 r. to samo urządzenie będzie sprzedawane w cenie od jednego do dziesięciu dolarów.

W niniejszym opracowaniu rozważamy jaki wpływ może mieć ta tendencja na architekturę komputerów w następnym ćwierćwieczu. Obecnie nawet prognozowanie na najbliższy kwartał, w tak szybko rozwijającej się dziedzinie jest niełatwe, natomiast w odniesieniu do okresu 25 lat - można przyjąć założenie, że przedstawione prognozy będą zapomniane dużo wcześniej, przed upływem tego czasu.

Mikrokomputer

Najpierw spróbujmy określić jak może wyglądać mikrokomputer. Dokładne przewidywanie wewnętrznej struktury mikrokomputera w 1997 r. jest obecnie daremną próbą. Przypuszczalnie będzie ona odpowiadala obecnej architekturze, ale może też być inna. Jeśli miałaby być podobna do obecnej struktury, wówczas nie należałoby przewidywać żadnych zmian w architekturze komputerów przez następne 25 lat, co jest zupełnie nieprawdopodobne. Jeśli natomiast przedstawiona struktura różniłaby się od obecnej, wówczas powinna być wyraźnie lepsza od stosowanej teraz. Gdyby tak było, to producenci wykorzystaliby tę propozycję natychmiast i struktura przewidywana przez nas byłaby strukturą komputerów 1975 r. a nie 1997 r. Niezależnie od tego powinniśmy określić niektóre parametry mikrokomputerów, tak aby można było ocenić ich możliwości. Dlatego w naszych rozważaniach założymy 16000 słów 32 bitowych, umiarkowaną, lecz odpowiednio dobraną listę rozkazów, kilka rejestrów ogólnego przeznaczenia i dołączenie urządzeń wejścia-wyjścia do conajmniej jednego z tych rejestrów. Jako długość słowa naszego mikrokomputera wybieramy 32 bity. To zapewnia 8 bitów dla kodu operacji, 8 dla modyfikacji i 16 dla wyboru jednego z 65 kilobajtów pamięci lokalnej. Ta pamięć będzie służyła do przechowywania zarów-

[1] "Chip" - mikroukład, płytka np. krzemowa, z wbudowaną strukturą określonego układu (tłum.).

no programu jak i danych. Nie przewiduje się autonomicznego procesora dla urządzeń wejścia-wyjścia. Ponieważ jest to mikrokomputer, więc on sam będzie służył jako procesor urządzeń wejścia-wyjścia dla innych większych maszyn.

Spróbujmy określić jakiej szybkości możemy oczekiwać od tej maszyny. Układy przełączające wysokiej jakości pozwalają dzisiaj uzyskiwać opóźnienia 1-10 ns na jedną warstwę. Jeżeli weźmiemy pod uwagę 10-20 warstw na realizację rozkazu, możemy przewidywać czas wykonywania rozkazu równy 25 ns, co odpowiada szybkości 40 MIPS (million instructions per second - milion operacji na sekundę). Powinniśmy pamiętać, że cały komputer jest na jednej płytce, a więc opóźnienia wynikające z propagacji wzdłuż powierzchni 1 mm^2 można pominąć. Szybkość układów logicznych procesora nie jest jednak jedynym czynnikiem określającym szybkość komputera. Ważny jest również czas cyklu pamięci.

Mniej pewne jest przewidywanie, że pamięć będzie w stanie pracować z taką samą szybkością. Zakładając maksymalne wartości współczynników wzmocnienia logicznego na wejściu (fan in) i na wyjściu (fan out) równe 8, otrzymujemy wymagania na około 6 poziomów ($8^6 = 2^{18}$) układów dostępu i podobną "głębokość" drzewa wyjściowego przy adresacji 65 kilobajtów. Dwanaście poziomów po 2 ns na poziom daje 25 ns lub około 40 milionów/sekundę odwołań do pamięci. Uwzględniając jedno odwołanie się do pamięci dla wybrania rozkazu i jedno odwołanie się dla realizacji rozkazu otrzymujemy szybkość działania równą 20 MIPS. Bądźmy ostrożni i podzielmy tę liczbę przez dwa. Stąd przewidujemy, że mikrokomputer będzie w stanie wykonywać 10 MIPS. Będzie to odpowiednik modelu 7090 (bez urządzeń peryferyjnych) w postaci jednej płytki w cenie około 1 dolara[1].

Teraz powinniśmy rozpatrzeć sprawę wielkości kosztów. House i Henzel [1] zauważają, że w latach sześćdziesiątych koszt minikomputera spadał 10 razy w ciągu 10 lat. W 1960 r. IBM był w stanie produkować model 7090 za około 3 miliony dolarów, z czego połowa przypada na procesor i główną pamięć. Przy założeniu, że wnioski House'a i Henzela mogą być ekstrapolowane oraz uwzględniając odstęp 40 lat między rokiem 1960 i 1997 otrzymujemy cenę 1.500.000 dolarów/10^4 czyli 150 dolarów. Jeśli ten współ-

[1] Model 7090 posiada szybkość ok. 0,2 MIPS, lecz ma rozbudowaną listę rozkazów i jeden lub więcej autonomicznych kanałów danych; przyjmujemy te właściwości za równoważne.

czynnik zmiany kosztów w czasie wykorzystamy do modelu 960A Texas Instruments z 1971 r. z pamięcią 16k, który sprzedawany będzie za 7350 dolarów przez następne 25 lat, wówczas otrzymamy 7350/300 = 25 dolarów. Ponieważ rzeczy zwykle zmieniają się szybciej niż to można oczekiwać, więc wydaje się, że można przewidywać koszt 1 dolara za mikrokomputer. Jeśli to oszacowanie jest nawet z dziesięciokrotnym błędemza duże lub za małe, to istotne pozostaje stwierdzenie, że wydajny procesor łącznie z pamięcią będzie dostępny za niewielką cenę.

Coury [2] wskazuje, że jeśli koszt procesora będzie można pominąć w porównaniu do kosztów urządzeń peryferyjnych to koszt całej instalacji określą te właśnie urządzenia peryferyjne. Dlatego należy wnikliwie poszukać takich dziedzin, w których minikomputery mogą być z pożytkiem zastosowane.

Można przypuszczać, że mikrokomputery będą wykorzystywane głównie w dwóch dziedzinach, tj.

- do zwiększania możliwości działań logicznych w sprzęcie nie związanym z komputerami,
- do zwiększania możliwości bardziej lub mniej standardowych instalacji komputerowych.

W pozostałej części tego opracowania będą kolejno rozpatrywane te dwie dziedziny.

Zastosowania mikrokomputerów w urządzeniach nie wykonujących obliczeń

Przeciętny "szary człowiek" nie jest zainteresowany obliczeniami za pomocą komputerów. Zwraca on uwagę tylko na to, co komputery mogą zrobić dla niego a nie na parametry urządzenia, wybieranie liczb pierwszych, automat skończony lub pomysłowe rozwiązania szyn urządzeń peryferyjnych. Poza tym jest on nastawiony do wydawania swych pieniędzy tylko wtedy, gdy widzi w tym swój interes. Jeśli zatem ma dojść do masowej produkcji mikrokomputerów, aby uzyskać przewidywaną przez nas cenę 1 dolara za sztukę, to powinniśmy znaleźć zastosowania przemawiające do "szarego człowieka".

Gdy mikrokomputery będą już dostępne w postaci płytek - nie może być mowy o tym, aby architekt komputera decydował o licz-

bie bitów w słowie. Lecz zanim zestandaryzujemy, na podstawie określonych danych, rozwiązanie płytek, dobrze byłoby rozpatrzeć dziedziny zastosowań mikrokomputerów. Przedstawimy kilka możliwych masowych zastosowań mikrokomputerów sądząc, że znajomość celu ułatwi ich projektowanie. Niezależnie od tego, gdy dostępne już będą standardowe elementy, wówczas specjaliści zajmujący się architekturą komputerów będą w stanie zwrócić uwagę na szersze możliwości wykorzystania produktów swojej pracy.

Usługi

Rozpatrzmy jako pierwszy przykład długoterminowy budzik. Budzik jest ustawiony na 7^{30} (lecz nie na 19^{30}) od poniedziałku do piątku, na 8^{30} w soboty i wyłączony na niedziele. Łatwo można zaprogramować do wypełniania tych funkcji mikrokomputer, z procesorem o dokładności nie większej niż cztery lub pięć miejsc dziesiętnych, pamiętający informacje zarówno o roku przestępnym jak i ruchomych świętach.

Możemy również wyobrazić sobie zegarek na rękę z pewnym rodzajem odczytu cyfrowego. Może on być wyposażony w "rejestr łaskoczący". Budzik będzie wyświetlał informację na tarczy zegarka, np.: "masz umówioną wizytę u dentysty za 47,3 minuty".

Nałóżmy płytkę na elastyczną kartę (mieszczącą się w portfelu), którą wkłada się do handlowego urządzenia końcowego (merchant's terminal). Uzyskuje się natychmiastowe potwierdzenie konta, uzupełnione potwierdzeniem tożsamości o stopniu złożoności odpowiadającym pamięci typu ROM - stanowiącej część płytki. (Trzeba dość poważnych nakładów, obróbki w piecach próżniowych, masek itp. aby podrobić taką kartę).

Dodajmy mikrokomputer do mechanicznego zamka; wówczas możemy stosować naszą kartę przechowywaną w portfelu do wchodzenia do domu, uruchamiania samochodu lub wstępu do ulubionego teatru lub też zgodnego z "kluczem" klubu. Komputer związany z zamkiem będzie pamiętał, kto jest upoważniony do otwierania drzwi. Będzie można nosić tylko jedną kartę w portfelu zamiast paczki oddzielnych kart do potwierdzania i pęku kluczy.

Dodajmy mikrokomputer do elektrycznej maszyny do pisania. Zestawiajmy i wypisujmy listy w końcowej poprawionej i bezbłędnej wersji. Zmagazynujmy adresy przyjaciół i stowarzyszeń w pamięci mikrokomputera. Dodajmy

listę słów, w których często występują błędy. Zaczynamy zbliżać się do obsługi równoważnej dobrej sekretarce-maszynistce. Pamięć o pojemności 64 kilobajtów będzie magazynować 52 strony tekstu, przy założeniu 250 słów pięcioznakowych na stronę. Taka pamięć będzie też mogła magazynować 325 informacji zawierających nazwiska z adresami, z których każda odpowiada 200 znakom, lub też magazynować 6500 słów dziesięcioznakowych.

Rozpoznawanie słów mówionych za pomocą komputera napotyka na główną trudność wynikającą z tego, że żadna para ludzi nie posiada identycznej wymowy. Przypuśćmy, że próbujemy uzyskać komputer rozpoznający głos określonej osoby. Ktoś podchodzi do jakiegoś piszącego urządzenia, wkłada swoją prywatną płytkę z rozpoznawaczem głosu, bierze mikrofon do ręki i zaczyna dyktować. Można sobie wyobrazić gniazdko z boku telefonu (do którego oczywiście dołączony jest mikrokomputer do przechowywania często używanych numerów itp.) służące do tego, aby po ułożeniu zgodnego z życzeniem listu, piszący wywołał lokalny urząd pocztowy i przesłał list do komputera przechowującego i przekazującego informacje. Późną nocą, gdy ludzie nie wykorzystują linii telefonicznych dalekiego zasięgu, maszyna w urzędzie pocztowym wywołuję maszynę piszącą adresata i dyktuje list, aby w czasie śniadania można go było przeczytać.

Skomputeryzowana telewizja

Rozpatrzmy zastosowanie mikrokomputera do przetwarzania obrazu telewizyjnego. Zgodnie z ustalonymi normami każdy kanał TV ma szerokość pasma 4,5 MHz i raster 540-liniowy kreślony 30 razy na sekundę[1]. Przyjmując 512 linii z 512 elementami na jedną linię, 32 ramki na sekundę i cztery możliwe sygnały (wyłączone, czerwony, niebieski i żółty - co umożliwia odtwarzanie odcieni) otrzymujemy 2^{25} bitów/s. Jest to bliskie 32 milionów bitów/s. Nasz mikrokomputer z 10 MIPS i słowem 32-bitowym łatwo może sterować wyświetlaniem takiego obrazu. Każda ramka ma $2^9 x 2^9 x 2^2$ czyli 2^{20} bitów. Stąd nasze 2^{16} słów 2^5 bitowych może przechowywać dwie ramki lub jedną ramkę i instrukcje. Bardzo istotne jest, że przy komputerowym wyświetlaniu obrazu nie potrzebujemy transmitować i retransmitować informacji redundacyjnej, tj. części obrazu, które pozostają takie same od ramki do ramki. Biorąc pod uwagę to, że dużo powierzchni obrazu jest w

[1] Są to normy obowiązujące w USA (tłum.).

tym samym kolorze i to, że ruch od ramki do ramki jest dość wolny i ciągły, możemy szacować, że tylko ok. 1% informacji zmienia się między kolejnymi ramkami. Oznacza to, że możemy zmniejszyć szerokość pasma wizyjnego 100 razy i że wszystkie kanały obecnie wyznaczone mogą być umieszczone w paśmie zarezerwowanym dla kanału 2. Da to kilka istotnych efektów: jakość obrazu będzie wyższa niż obecnie, zasięg transmisji na stosunkowo niskiej częstotliwości kanału 2 będzie dużo większy niż na wyższych częstotliwościach, a praktycznie wyeliminowany będzie słaby obecnie odbiór w obszarach wcięć charakterystyki promieniowania. Przy dekodowaniu komputerowym istotnie zmniejszymy koszt odbiorników w porównaniu do obecnych. Najważniejsze jest to, że zastosowanie komputerów zwolni dla innych użytkowników prawie 400 MHz z zakresu fal radiowych - zasób, który w niedługim czasie będzie na wyczerpaniu.

Skomputeryzowane gry

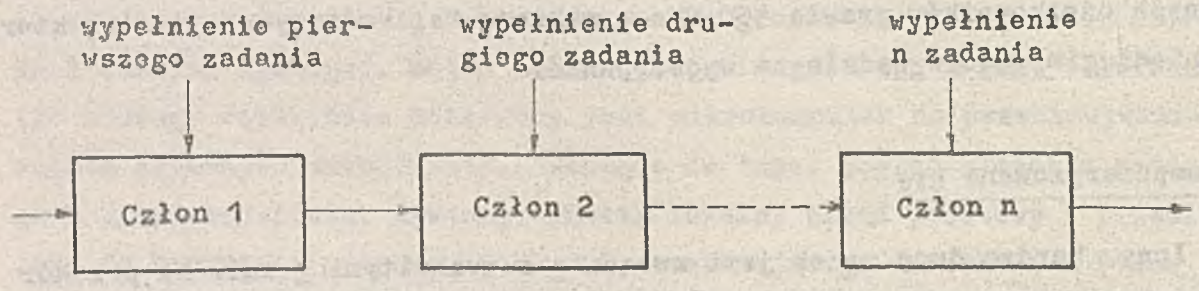
Inny, bardzo duży rynek jest związany z rozmaitymi grami. Na przekór oczekiwaniom Martina i Normana [3], wątpię w to aby wielu ludzi traktowało programowanie komputera jako hobby. Natomiast obecne tendencje wskazują, że komputery można łatwo przystosować do udziału w grach. Gry: "Wojna kosmiczna", "NIM", "KALACH", szachy i "GO" są przykładami gier już skomputeryzowanych. Komputery służą jako plansze do gry, urządzenia działające losowo i jako przeciwnik z regulowanym poziomem umiejętności. Można przewidywać rozwój przemysłu dostarczającego "grę tygodnia" w postaci płytki, którą wkłada się do skomputeryzowanego telewizora albo piszącej maszyny lub też do obydwu tych urządzeń. Takie urządzenie potrafi zastąpić czwartego do brydża lub umożliwi graczowi udział generała R.E. Lee lub U.S. Granta w skomputeryzowanej "bitwie pod Gettysburgiem".

Zastosowanie mikrokomputerów do obliczeń

Niezależnie od uwag zawartych w poprzedniej części, ludzie w 1997 r. będą jeszcze zajmowali się obliczeniami, a ściślej mówiąc, będą to robiły za nich komputery. Jak zwykle można przewidzieć dwie przeciwstawne tendencje. Są to zastosowania związane z mikrokomputerami i makrokomputerami.

mują wszystkich spraw związanych z projektowaniem. Są to: linia (pipeline), z przeplataniem instrukcji (interlace), procesory sieciowe oraz metody z podziałem pracy.

Najlepszym przykładem organizacji komputera typu liniowego jest IBM System 370/195 i CDC - STAR, w których wiele członów jest łączonych szeregowo do wypełniania działań. Każdy człon w ciągu realizuje część pracy związaną z problemem, potem przekazuje ten problem do następnego członu w ciągu i z kolei zwraca się do poprzedzającego członu po przyjęcie nowego problemu (rys. 4). Jednostka mnożenia w zmiennym przecinku może mieć aż 30 członów.



Rys. 4. Organizacja typu "rurociąg" lub linia zbiorcza. Wiele członów wypełnia elementarne operacje. Szybkość na wyjściu jest wysoka, chociaż może występować znaczne opóźnienie między końcami linii

Takie rozwiązanie ma na celu uzyskanie wysokiej wydajności (nowe instrukcje mogą być wprowadzone co 25 ns) i jednocześnie zapewnienie dużego wykorzystania sprzętu przez przeznaczenie każdego członu do wypełniania tylko swojego specjalnego zadania. Zakłada się przy tym wstępnie, że będzie dostateczna liczba niezależnych instrukcji, która pozwoli na utrzymanie linii przez dłuższy okres czasu w stanie bliskim pełnej zajętości. Krytykować takie rozwiązanie można z dwóch głównych powodów. Po pierwsze oceniono, że 40% (1,5 miliona dolarów) kosztów jednostki centralnej 370/195 związane było z jednostką mnożenia w zmiennym przecinku. Lecz wg dobrze znanego zestawu Gibsona tylko 3,8% z wszystkich wykonywanych instrukcji jest typu mnożenia w zmiennym przecinku. Po drugie, Tjaden i Flynn [4] wykazali, że największa równoległość jaką można uzyskać z pojedynczego ciągu instrukcji, jest mniejsza niż dwie instrukcje wykonywane jednocześnie. W toku podobnych studiów na uniwersytecie w Massachusetts stwierdzono, że nawet przy nieograniczonej liczbie zasobów (rejestry, pomocnicze pamięci dla instrukcji, jednostki funkcjonalnej itp.) możliwa do uzyskania równoległość jest mniejsza niż dwa. Jeśli

Jednostki centralne minikomputerow są już fizycznie mniejsze i tańsze niż niezbędne dla nich zestawy urządzeń peryferyjnych, a wprowadzenie mikrokomputerów będzie uwydatniało tę sytuację. Dlatego nie będziemy dalej dyskutować nad zagadnieniami związanymi z "mini", lecz zwrócimy naszą uwagę na problemy wynikające z projektowania dużych i bardzo dużych instalacji komputerowych.

Urządzenia końcowe

Rozpocznijmy od urządzenia, które łączy użytkownika z instalacją. Przeciętnie człowiek czyta mniej niż 300 słów na minutę. Przypuśćmy, że szybko czytający przekracza dwa razy tę szybkość, tj. czyta 600 słów na minutę, czyli 10 słów lub 50 znaków na sekundę. Szybka maszynistka może pisać około 120 słów na minutę czyli 10 znaków na sekundę. Dla zachowania symetrii weźmy pod uwagę monitor ekranowy z klawiaturą, o szybkości 60 znaków/s w obu kierunkach. Urządzenie to nie pozwala uzyskiwać trwałego zapisu (hard copy), lecz jeśli przyjmiemy, że nasz system komputerowy pozwala dołączać dodatkowe informacje i otrzymywać dokumenty na podstawie odsyłaczy, to potrzeba takiego trwałego zapisu nie jest uzasadniona.

Wielu ludziom może wydawać się, że ich dane są zbyt obszerne dla takich powolnych urządzeń, lecz są to właśnie te sytuacje, w których komputerowe przetwarzanie i "monitorowanie" danych jest najbardziej owocne.

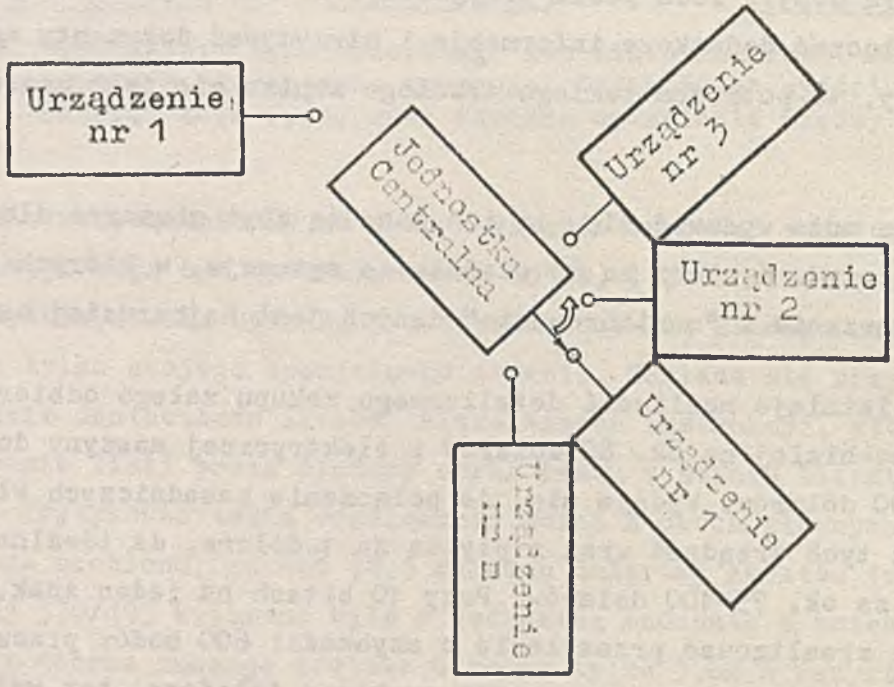
Powszechnie istnieje możliwość detalicznego zakupu małego odbiornika telewizji czarno-białej za ok. 80 dolarów i elektrycznej maszyny do pisania za ok. 100 dolarów. Wydaje się, że połączenie zasadniczych właściwości każdego z tych urządzeń wraz z płytką za 1 dolara, da idealne urządzenie końcowe za ok. 75-100 dolarów. Przy 10 bitach na jeden znak, będziemy w stanie zrealizować przesyłanie o szybkości 600 bodów pracujące w reżimie "dane poniżej głosu" za pomocą zwykłego telefonu, bez wpływu na jego normalne funkcjonowanie.

Instalacja centralna

W tej chwili można rozróżnić cztery rodzaje rozwiązań przy projektowaniu wielkich komputerów. Niektóre z nich zastosowano już w praktyce, inne są na etapie studiow. Nie wyłączają się one wzajemnie, ani nie obej-

naprawdę tak jest, wówczas posiadanie 30 członów w jednostce mnożenia dla komputera o jednym ciągu instrukcji jest absurdem. Dużo lepsze może być posiadanie osobnego mikrokomputera do mnożenia w zmiennym przecinku, przez przeszukiwanie tablicy zawartej w jego własnej pamięci.

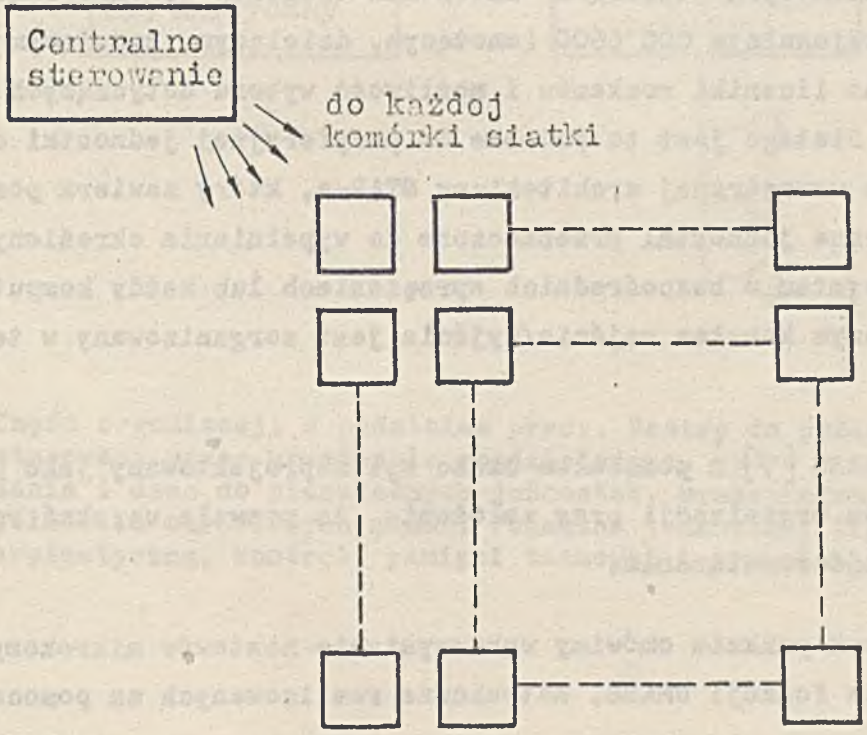
Drugie rozwiązanie, z przeplataniem wielu ciągów instrukcji, może być również nazwane podejściem karuzelowym (rys. 5). Przykładami mogą być tu: projekt peryferyjnego procesora CDC - 6600, Honeywell 800, procesor Flynna o wielu ciągach instrukcji [5] oraz edynburski projekt Fostera [6]. Tutaj kilka niezależnych programów jest wypełnianych w tym samym czasie. Najpierw instrukcja jest pobierana z ciągu A, potem B, potem C itd. Ponieważ kolejne instrukcje pochodzą z niezależnych programów, więc mogą być traktowane jako wzajemnie niezależne i stąd potencjalnie może być uzyskiwana wydajność odpowiadająca jednostce centralnej typu liniowego. Podstawową sprawą przy takim rozwiązaniu jest



Rys. 5. Organizacja typu karuzelowego. Pojedyncza jednostka centralna wypełnia jedną operację dla zadania A, potem dla B, potem dla C i tak dalej

zasadnicze założenie, że jednostka centralna jest dużo szybsza niż pamięć. Aby pokonać tę trudność, stosuje się kilka bloków pamięci realizujących instrukcje wybierania, a dane i te instrukcje są zaraz dostarczane do szybkiej jednostki centralnej. Takie założenie było uzasadnione w końcu lat sześćdziesiątych, lecz my postulowaliśmy powyżej pamięć scaloną z jednostką centralną, a więc o porównywalnej szybkości. To oczywiście jest przeciwstawne racjonalizacji wynikającej z takiego rozwiązania.

Przykładami siatkowej organizacji procesora są Illiac IV i STARAN Goodyear (rys. 6). Tutaj duża liczba identycznych komórek wykonuje to samo dla dużej siatki danych. Gdy występują operacje tego właśnie typu, wówczas procesor typu siatkowego jest tyle razy szybszy od zwykłego, ile komórek znajduje się w maszynie siatkowej.



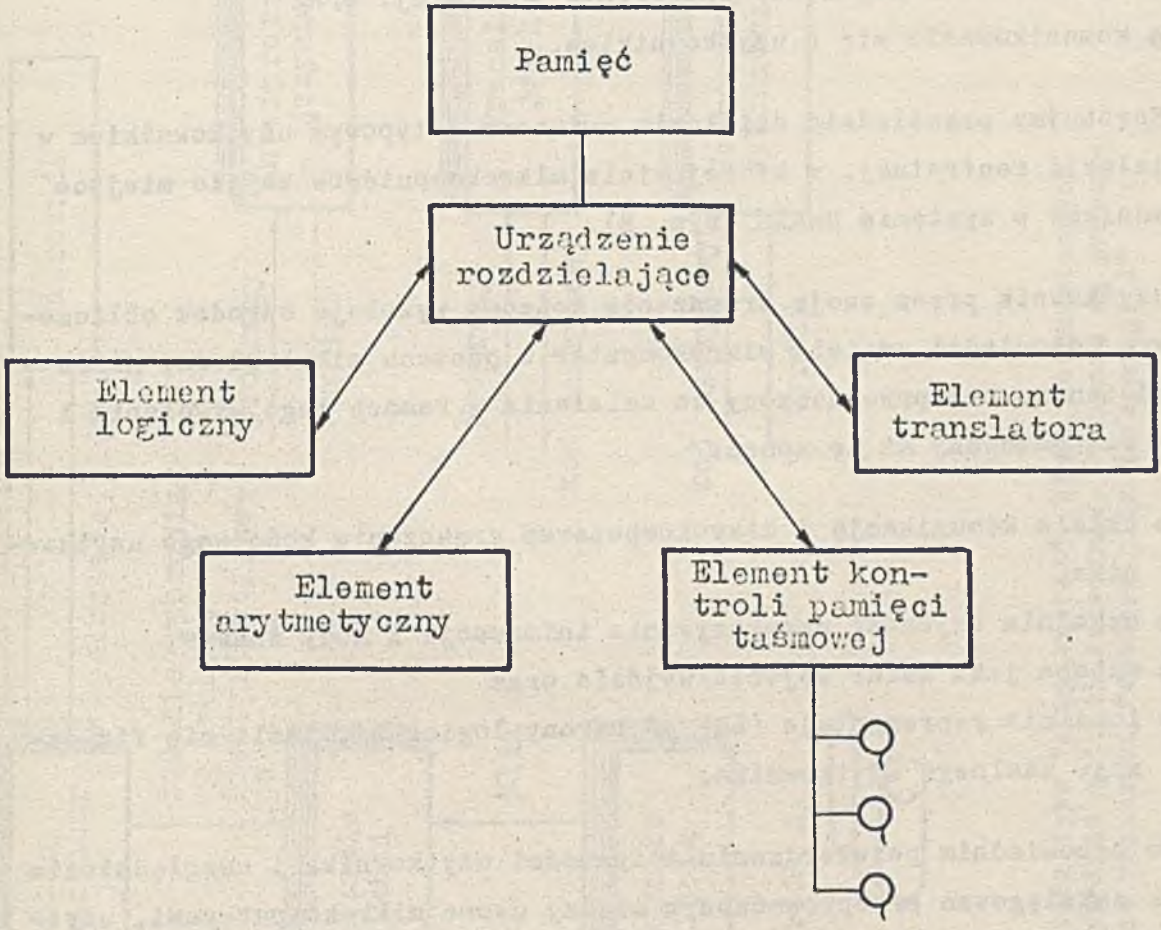
Rys. 6. Organizacja siatkowa. Centralne sterowanie wydaje rozkazy, które są wykonywane przez każdy element siatki

W czasie wykonywania działań innych niż siatkowe, maszyna siatkowa nie może być szybsza niż konwencjonalna. Oczywiście autorzy propozycji tego rozwiązania zakładają, że działania siatkowe przeważają. Jest bardzo prawdopodobne, że jednostka centralna będzie mieć siatkowy procesor o "inteligentnych" komórkach, które mogą być konstruowane z mikrokomputerów, jako zasadniczych elementów składowych. Taka jednostka może rozwiązywać problemy systemowego przetwarzania plików informacji (file).

Omówimy teraz metodę z podziałem pracy (rys. 7). Podstawowym prototypem tego rozwiązania była Gamma 60. Miała ona aż 128 elementów, z których każdy wykonywał niezależne, lecz wzajemnie powiązane programy. Było tam urządzenie sterujące dyskiem w formie jednostki arytmetycznej dającej na podstawie pary wektorów wynik w postaci trzeciego wektora (patrz również CDC - STAR - "String Array" procesor). Miało ono "inteligentne" taśmowe urządzenia sterujące, translatory kodu, bloki przełączników przekaźnikowych itd. Co do znaczenia elementy te odpowiadają jednostkom funkcjonalnym CDC 6600 (mnożącym, dzielącym, sumującym itp.), lecz każdy ma liczniki rozkazów i możliwość wyboru dotyczących go instrukcji. Dlatego jest to podobne do peryferyjnej jednostki centralnej 6600 lub wewnętrznej architektury STAR-a, który zawiera poszczególne autonomiczne jednostki przeznaczone do wypełniania określonych zadań. Każdy system o bezpośrednich sprzężeniach lub każdy komputer z autonomicznym kanałem wejścia/wyjścia jest zorganizowany w ten sposób.

System UMASS [7] z podziałem czasu był zaprojektowany jako ulepszenie tego typu organizacji przy założeniu, że pozwala uzyskać rzeczywistą modularność rozwiązania.

W następnym punkcie omówimy wykorzystanie zestawów mikrokomputerów do spełniania funkcji UMASS, dotychczas realizowanych za pomocą oprogramowania.



Rys. 7. Część organizacji z podziałem pracy. Dostęp do pamięci jest sterowany przez urządzenie rozdzielające, które przekazuje zadania i dane do niezależnych jednostek, przeznaczonych do wypełniania określonych zadań. Pokazano jednostki: logiczną, arytmetyczną, kontroli pamięci taśmowej i translacji

Organizacja komputera typu "konglomarat"

UMASS został zorganizowany jako zestawienie składników, z których każdy spełnia określoną część zadania dla użytkownika. Składniki te realizują

- zgłaszanie,
- przydzielanie lub przekazywanie przydzielonych zasobów,

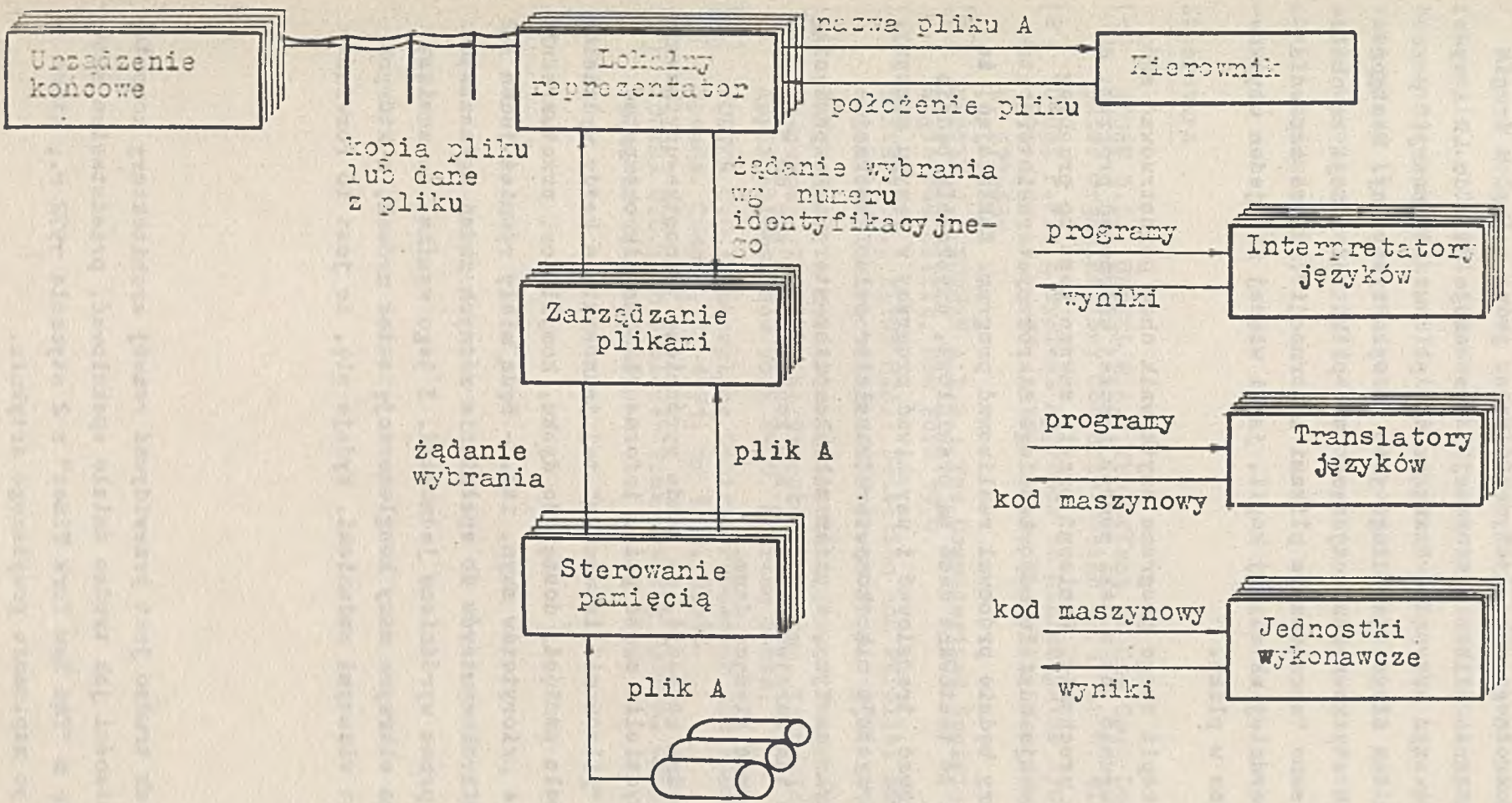
- wypełnianie takich działań przetwarzających, jak wydawanie tekstu,
- wypełnianie programów,
- wybieranie i magazynowanie plików informacji oraz
- komunikowanie się z użytkownikiem.

Spróbujmy prześledzić działania związane z typowym użytkownikiem w instalacji centralnej, z której wiele mikrokomputerów zajęło miejsce składników w systemie UMASS (rys. 8).

Użytkownik przez swoje urządzenie końcowe wywołuje ośrodek obliczeniowy. Odpowiedzi udziela mikrokomputer w postacu mikroukładu. Mikroukład ten został przeznaczony do działania w ramach tego wywołania i jest zaangażowany aż do końca:

- ustala komunikację z mikrokomputerem urządzenia końcowego użytkownika,
- uzgadnia szybkość przekazywania informacji i kody znaków,
- działa jako bufor wejścia/wyjścia oraz
- lokalnie reprezentuje (LR) od strony logicznej, jeśli nie fizycznej, zdalnego użytkownika.

Po odpowiednim potwierdzeniu tożsamości użytkownika i uwzględnieniu stanu zaksięgowań przeprowadzonym między dwoma mikrokomputerami, użytkownik wprowadza swój pierwszy rozkaz, np. w celu otrzymania kopii programu, którego używał wczoraj. Znając numer identyfikacyjny użytkownika i nazwę pliku informacji LR zwraca się do mikrokomputera-kierownika w sprawie ustalenia położenia tego pliku informacji. Mikrokomputer-kierownik na pewno będzie mieć siatkowy typ organizacji (aby mógł przetłumaczyć numer identyfikacyjny użytkownika i nazwę pliku informacji na określony adres pamięci zewnętrznej) Potem sterowanie przechodzi do mikrokomputera-zarządzanie plikami informacji, który następnie zwraca się do mikrokomputera-sterowanie pamięcią zewnętrzną, aby uzyskać (w pamięci mikrokomputera-zarządzanie plikami informacji) kopię pożądanego programu. Po zbadaniu nagłówka programu "zarządzanie plikami informacji" w pierwszej kolejności ustala, czy użytkownik jest upoważniony do dostępu do tego pliku informacji, a jeśli tak, to jakiego rodzaju jest ten dostęp. Zakładając, że to sprawdzenie kończy się wynikiem potwierdzającym "zarządzanie plikami informacji" patrzy czy jest to plik wielokrotnego dostępu, czy też są ogra-



Rys. 8. Organizacja komputera typu "konglomerat". Wiele autonomicznych mikrokomputerów wypełnia oddzielne zadanie. Pokazane są przesyłania odpowiadające rozkazowi "wybierz plik informacji A"

niczenia do jednokrotnego wykorzystywania przez jedną osobę. W drugim przypadku "zarządzanie plikami informacji" przekazuje plik do LR i wyłącza się. W pierwszym przypadku "zarządzanie plikami informacji" przechowuje ten plik tak długo, jak długo jest on wykorzystywany. Następne zgłoszenia innych użytkowników dotyczące tego pliku informacji są obsługiwane przez to samo "zarządzanie plikami informacji", które zapewnia uzyskiwanie najbardziej aktualnej kopii, jeśli więcej niż jeden użytkownik dokonuje zmian w pliku.

Po uzyskaniu kopii swego programu użytkownik chce poinstruować swój LR co dodać, co usunąć lub w jaki sposób inaczej uformować program, a potem usiłuje go uruchomić. Ponieważ prawie zawsze jest to przebieg próbny, LR potrzebuje udziału odpowiedniego mikrokomputera-interpretatora języka, który będzie próbował realizować program. Zakładając, że nie ma błędów i że testowane dane są prawidłowe, użytkownik będzie chciał optymalizować, translować i uzyskiwać programy w języku maszyny. LR najpierw wykorzystuje mikrokomputer-translator celem generacji instrukcji w języku maszyny, a potem mikrokomputer-sterowanie pamięcią zewnętrzną celem ich zarejestrowania, a w końcu zwraca się do "kierownika" aby wprowadzić bieżące dane.

Najogólniej można zauważyć, że każda wyróżniona jednostka-użytkownik lub dająca się wydzielić część pliku informacji - ma mikrokomputer przeznaczony do "pilnowania interesów" tej jednostki, a każdy wydzielony zasób - jak pole pamięci, dostęp do dysku, kompilator, przetwarzanie liczb, określanie priorytetów szyn, itp. - będą miały również jeden lub kilka własnych mikrokomputerów do spełniania różnych zadań zgłaszanych do tych zasobów przez wyróżnione jednostki. Z tego wynika, że zamiast jednego spoistego olbrzyma mamy konglomerację setek mrówek, z których każda zajęta jest własnymi zadaniami. Wydaje się, że jest to rozwiązanie przyszłości.

Zakończenie

Aby pokazać jak trudno jest przewidywać rozwój architektury komputerów, a w szczególności jak trudno śmiało spekulować, przedstawiam następującą informację z "The New York Times" z 2 stycznia 1972 r., która ukazała się już po napisaniu powyższego artykułu.

"Innym rodzajem usług (pocztowych USA) są usługi pocztowe typu faksymile... Dzięki usłudze, która została wymyślona przez biuro wyrobów pocztowych, faksymile listu, wykresu, rysunku itp. jest transmitowane przez linie telefoniczne do poczty odbiorczej, gdzie jest umieszczane w kopercie i wysyłane do adresata w ciągu czterech godzin od chwili przekazania".

Literatura

[1] HOUSE D.L., HENZEL R.A.: The Effect of Low Cost Logic on Minicomputer Organization. Computer Design 1971, nr 1, s. 97-100.

[2] COURY F.F.: A Systems Approach to Minicomputer I/O. Proc. AFIPS 1970 SJCC, vol. 36, AFIPS Press, Montvale, New York 1970, s. 677-681.

[3] MARTIN J., NORMAN A.R.D.: The Computerized Society. Prentice-Hall, Englewood Cliffs, New York 1970.

[4] TJADEN G.S., FLYNN M.J.: Detection and Parallel Execution of Independent Instructions. IEEE Trans Comp. 1970, nr 10, s. 889-895.

[5] FLYNN M.J.: A Multiple-Instruction Stream Processor with Shared Resources. In Parallel Processor Systems, Technologies and Applications, New York 1970, Spartan Books.

[6] FOSTER C.C.: Uncoupling Central Processor and Storage Device Speeds. Comput. J. 1971, nr 1, s. 45-48.

[7] FOSTER C.C.: An Unclever Time-sharing System. Computing Surveys 1971, nr 1, s. 23-48.

ETO NOWOŚCI nr 1/1974

Mgr inż. Tomasz RAWIŃSKI
Spółdzielnia Pracy Inżynierów Geodetów
Zespół Zastosowań EMC

681.322.001.13

ROZSZERZENIE KONCEPCJI JEDNOLITEGO SYSTEMU ELEKTRONICZNYCH MASZYN CYFROWYCH. WPROWADZENIE EMC ODRA DO JS EMC

OD REDAKCJI

Koncepcja emulacji, która zakłada istnienie środków programowych (tj. programu Emulator pracującego pod kontrolą systemu operacyjnego) i środków technicznych usprawniających wykonanie programu starego komputera w nowym komputerze, jest kompromisem technicznym między możliwością efektywnego przenoszenia oprogramowania starego komputera (np. ODRA 1300) na nowy komputer (np. JS EMC) a kosztem realizacji tej możliwości. Przy czym w koncepcji emulacji, naszkicowanej w artykule Th. Kamburelisa pt. "Komputery ODRA 1300 a JS EMC" (Informatyka 1973, nr 7), można przenosić oprogramowanie EMC serii ODRA 1300 także na jednoprocesorowe systemy JS EMC. Zatem nie musi być spełnione, bardzo kosztowne w praktyce, wymaganie istnienia systemu wieloprocesorowego. Praktyczna realizacja tego wymagania w JS EMC mogłaby kosztować około 3 mln zł (dodatkowy sprzęt) zamiast około 100 tys. zł w przypadku emulacji. Oczywiście opracowanie programu Emulator jest dodatkowym kosztem (np. około 2 mln zł) lecz koszt ten rozkłada się na całą serię nowych komputerów. Również w "systemie wielojęzykowym" duże będą koszty opracowań programowych związanych z włączeniem programów serii ODRA 1300 do JS EMC (chociażby z powodu braku urządzeń zewnętrznych serii ODRA 1300 w takim "systemie wielojęzykowym").

1. Porównanie obecnej i proponowanej koncepcji JS EMC. Jednojęzykowy a wielojęzykowy JS EMC

Obecnie wprowadzaną w życie koncepcję JS EMC można ująć następująco: zbudujmy ciąg zgodnych pod względem języka wewnętrznego jednostek centralnych, czy procesorów oraz pewien zestaw urządzeń zewnętrznych takich, że każde z nich może współpracować z każdą jednostką centralną z wymienionego ciągu.

W ramach JS EMC opartego na takiej koncepcji możliwe jest wykonywanie programów użytkowych w jednym tylko języku wewnętrznym. Dlatego taki JS EMC nazwiemy jednojęzykowym JS EMC (JJ SEMC).

Inną możliwą koncepcją JS EMC jest koncepcja następująca: zbudujmy zestaw urządzeń zawierający procesory o różnych językach wewnętrznych, urządzenia zewnętrzne i inne niezbędne urządzenia tak, aby w ramach każdego systemu liczącego, złożonego z tych składników i podlegającego jednolitemu kierownictwu (jednolitej gospodarce zasobami) były wykonywalne programy użytkowe w różnych językach wewnętrznych i aby w każdym takim systemie liczącym mogło uczestniczyć dowolne urządzenie należące do zestawu.

W ramach takiego JS EMC są wykonywalne programy użytkowe w różnych językach wewnętrznych, przy czym zdolność do wykonywania programów w różnych językach ma każdy system liczący złożony ze składników JS EMC i dlatego opisany JS EMC nazwiemy wielojęzykowym JS EMC (WJ SEMC).

Rozpatrzymy teraz zagadnienie w jakich stosunkach wzajemnych mogą pozostawać jednojęzykowe i wielojęzykowe JS EMC. Niech będzie dany pewien JJ SEMC i WJ SEMC, w którym:

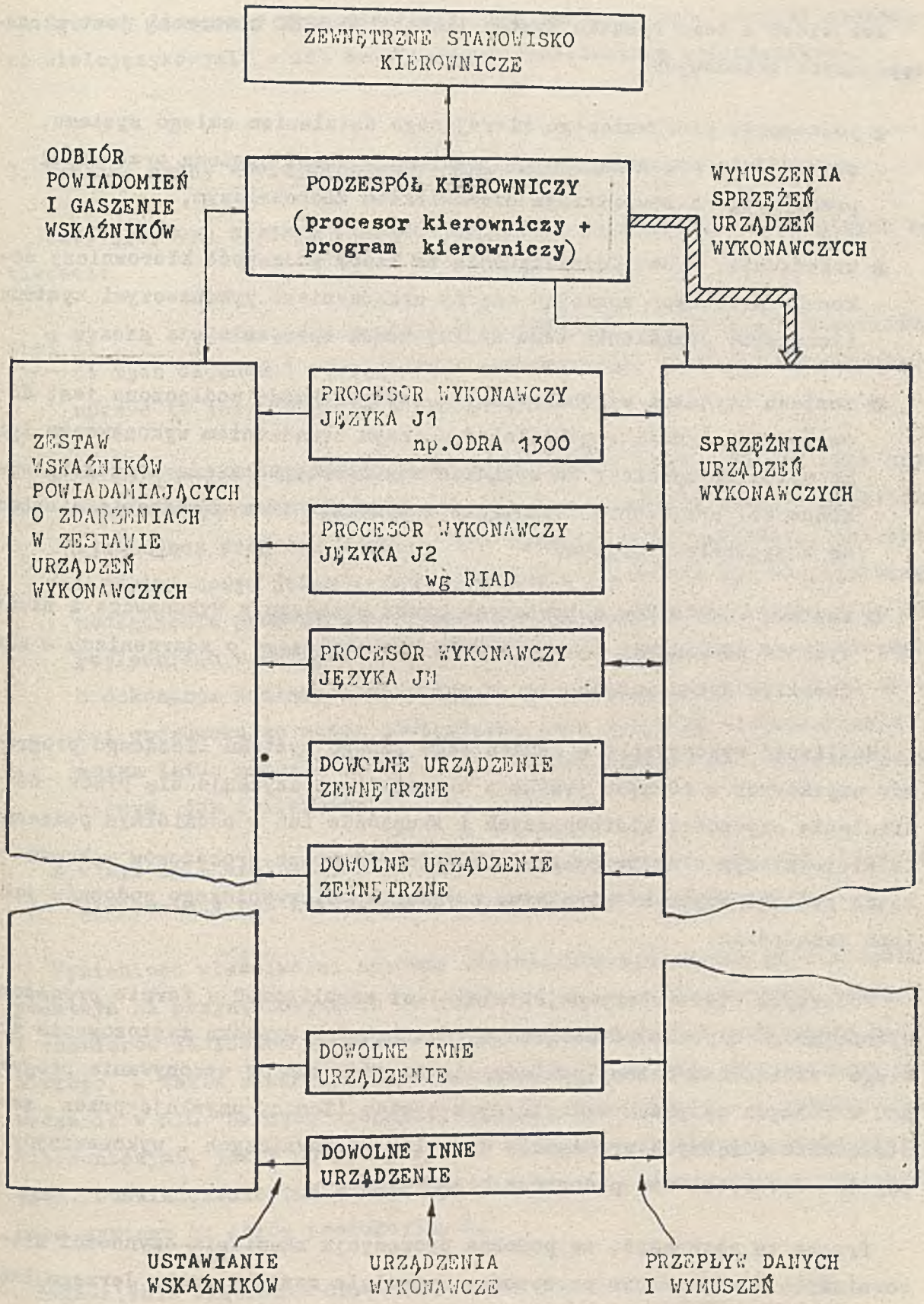
- wykonywalne są programy użytkowe w języku wewnętrznym JJ SEMC,
- mogą uczestniczyć urządzenia zewnętrzne należące do JJ SEMC.

Opisany WJ SEMC nazwiemy rozszerzeniem JJ SEMC.

Z określenia wielojęzykowego JS EMC, będącego rozszerzeniem jakiegoś jednojęzykowego JS EMC, wynika przede wszystkim, że w ramach rozszerzenia daje się wykorzystać cały istniejący dorobek związany z jednojęzykowym JS EMC, którego rozszerzenia dokonujemy.

2. Architektura systemów liczących wielojęzykowego JS EMC

Wprowadzenie w życie koncepcji WJ SEMC może nastąpić dopiero po odpowiedzeniu sobie na pytanie: jaką formę, jaką architekturę muszą mieć systemy liczące WJ SEMC, aby mogły charakteryzować się wymaganymi od nich właściwościami? Architekturę, ustrój systemu liczącego spełniającego odpowiednie wymagania przedstawia rys. 1.



Rys. 1. Architektura wielojęzykowego systemu liczącego

Jak widać z tego rysunku system liczący WJ SEMC utworzony jest z następujących składowych:

- podzespołu kierowniczego kierującego działaniem całego systemu, szczególnie prowadzącego gospodarkę zasobami systemu oraz współpracującego z zewnętrznym stanowiskiem kierowniczym,
- urządzenia, przez oddziaływanie na które podzespół kierowniczy dokonuje dowolnych sprzężeń między urządzeniami wykonawczymi systemu liczącego; urządzeniu temu nadamy nazwę sprzężnicy,
- zestawu urządzeń wykonawczych, z których każde podłączone jest do sprzężnicy i może współdziałać z innym urządzeniem wykonawczym tylko przez sprzężnicę; do urządzeń wykonawczych należą procesory wykonawcze, urządzenia zewnętrzne i wszelkie inne urządzenia niezbędne w systemie liczącym,
- zestawu wskaźników ustawianych przez urządzenia wykonawcze i służących do powiadamiania podzespołu kierowniczego o zdarzeniach w urządzeniach wykonawczych.

Możliwość wykonywania w ramach tego samego systemu liczącego programów użytkowych w różnych językach wewnętrznych uzyskuje się przez oddzielenie czynności kierowniczych i skupienie ich w oddzielnym podzespole kierowniczym oraz przez zastosowanie odrębnych procesorów wykonawczych podlegających kierownictwu podzespołu kierowniczego podobnie jak inne urządzenia.

Podzespół kierowniczy najłatwiej jest zrealizować w formie procesora wykonującego odpowiedni program kierowniczy. W wypadku zastosowania takiego rozwiązania możemy powiedzieć, że zdolność do wykonywania programów w różnych językach wewnętrznych system liczący uzyskuje przez zastosowanie odrębnych procesorów do celów kierowniczych i wykonawczych, czy też wyodrębnienie procesora kierowniczego.

Trzeba tu zaznaczyć, że podobna propozycja skupienia czynności kierowniczych w oddzielnym procesorze wysunięta została przez Jerzego Dańdę [1] jako sposób podniesienia sprawności czynności kierowniczych w systemach liczących.

Systemy liczące o przedstawionej architekturze będę nazywał systemami wielojęzykowymi, a ich architekturę architekturą wielojęzykową.

3. Istotne cechy wielojęzykowych systemów liczących

Wielojęzykowe systemy liczące mają wiele pożądanych i korzystnych właściwości:

- wysoką skuteczność czynności kierowniczych wynikającą z niezależności tych czynności od czynności wykonawczych; bardziej wyczerpująco sprawa ta jest omówiona w pracy [1],
- do systemu wielojęzykowego w każdej chwili można włączyć (bez naruszania tego, co dotąd istniało w systemie) nowy procesor wykonawczy wykonujący programy w języku już występującym w systemie, lub też w jakimś nowym języku; włączenie takie sprowadza się do fizycznego podłączenia procesora do odpowiedniego gniazda sprzężnicy i do odpowiedniego wskaźnika oraz powiadomienia podzespołu kierowniczego o dokonanym podłączeniu; podobnie można odłączyć procesor od systemu; podsumowując można stwierdzić, że w systemie wielojęzykowym można łatwo zmienić zdolność przerobową zarówno pod względem jakościowym, jak i ilościowym,
- w systemie wielojęzykowym można w każdej chwili wymienić podzespół kierowniczy bez żadnych zmian w pozostałej części systemu.

Wymienione właściwości systemu wielojęzykowego czynią go szczególnie podatnym na przystosowywanie do dowolnych wymagań dotyczących jakości i rozmiarów zdolności przerobowej czy też właściwości podzespołu kierowniczego, a także stwarzają dogodne warunki do rozwoju systemu oraz prowadzenia w nim "na żywo" rozmaitych doświadczeń zarówno z podzespołami kierowniczymi, jak i nowymi językami wewnętrznymi dla programów użytkowych. Doświadczenia takie mogą być prowadzone równocześnie z wykorzystaniem systemu do celów produkcyjnych.

4. Zagadnienie włączenia ODRY 1300 do JS EMC a koncepcja WJ SEMC

Wyłoniło się ostatnio zagadnienie jak wykorzystać oprogramowanie EMC ODRA 1300 w JS EMC. Opierając się na wprowadzonych pojęciach chciałbym

zaproponować pewne rozwiązanie tego zagadnienia.

Wydaje się, że rozwiązaniem najwłaściwszym byłoby zbudowanie wielojęzykowego systemu liczącego wyposażonego w dwa procesory wykonawcze (jeden wykonujący programy w języku maszyny ODRA 1300 i drugi w języku JS EMC) oraz przystosowanego do współpracy z wszelkimi innymi urządzeniami JS EMC. Oczywiście mamy na myśli obecnie realizowany jednojęzykowy JS EMC. Zbudowanie takiego systemu liczącego byłoby równocześnie zapoczątkowaniem rozwoju wielojęzykowego JS EMC, będącego rozszerzeniem, w sensie wcześniej ustalonym, obecnie budowanego jednojęzykowego JS EMC.

Dla pełnej zgodności z obecnie budowanym JS EMC podzespół kierowniczy takiego wielojęzykowego systemu liczącego musiałby zostać zbudowany w sposób umożliwiający przyjmowanie przez system opisów prac (Job Description) w języku opisów prac (JDL) JS EMC, jak również EMC ODRA 1300. Jest to najtrudniejsza część całego przedsięwzięcia.

Na zakończenie konieczne wydaje się określenie stosunku przedstawionego rozwiązania zagadnienia włączenia maszyny ODRA do JS EMC do metody emulacji zaproponowanej przez T. Kamburelisa w pracy [2].

Na ten temat chciałbym dodać, że nadanie jednostce centralnej JS EMC zdolności emulowania poszczególnych instrukcji maszyny ODRA 1300 wcale nie zapewnia wykonalności programów EMC ODRA 1300 w systemach liczących JS EMC. Zapewnienie takie nastąpiłoby dopiero w przypadku stworzenia możliwości współpracy tych programów z systemami operacyjnymi JS EMC, co wymaga dokonania odpowiednich przeróbek tych systemów operacyjnych. Przeróbki takie wydają się przedstawiać podobne trudności jak budowa nowego podzespołu kierowniczego dla systemu wielojęzykowego, zaś korzyści wynikłe ze zbudowania podzespołu kierowniczego systemu wielojęzykowego przekraczają niewątpliwie korzyści wynikające z dokonania przeróbek systemów operacyjnych JS EMC.

Literatura

[1] DAŃDA J.: Niektóre problemy następnych generacji komputerów. Cz. II. ETO Nowości 1973, nr 1.

[2] KAMBURELIS T.: Komputery ODRA 1300 a JS EMC. Informatyka 1973, nr 7.

JAK PRACUJE TRANSLATOR*

Mgr Tomasz KRAWCZYK
Instytut Maszyn Matematycznych

681.322.06:800.92

METODY ANALIZY SKŁADNI OPARTE NA RELACJACH PIERWSZEŃSTWA

Wstęp

Po analizie leksykalnej, w ramach której z tłumaczonego programu wydziela się podstawowe jego elementy (np. identyfikatory, słowa kluczowe, liczby), następuje analiza składniowa. Część translatora, która dokonuje analizy składniowej programu będziemy dalej nazywali analizatorem składni (ang. parser). Analizator składni wykonuje dwie zasadnicze funkcje:

a) sprawdza czy program jest składniowo poprawny,

b) dokonuje rozbioru składniowego programu.

Ogólnie, metody analizy składniowej można podzielić na dwie grupy: analizę typu redukcyjnego (ang. "bottom-up") oraz analizę typu generacyjnego (ang. "top-down").

W tym opracowaniu przedstawiono metody typu redukcyjnego oparte na relacjach pierwszeństwa. W punkcie 1 podany jest materiał teoretyczny, który wykorzystuje się w następnych punktach opisujących konkretne metody analizy.

1. Podstawowe pojęcia i definicje

1.1. Pojęcia i definicje z lingwistyki matematycznej

Alfabetem nazywamy skończony, niepusty zbiór, ktorego elementy nazywać będziemy symbolami. Słowem nad alfabetem nazywamy dowolny skończony ciąg symboli tego alfabetu. Zbiór wszystkich słów nad alfabetem V (ze słowem pustym ε) oznaczamy przez V^*. Często wygodnie jest pisać $\beta \ldots$ zamiast $\beta\gamma(\ldots\gamma$ zamiast $\beta\gamma$) jeśli nie interesuje nas dalsza (poprzednia)część słowa.

*Poprzednie artykuły z tego cyklu zamieściliśmy w numerach 3 i 4/73 ETO NOWOŚCI.

Gramatyką bezkontekstową nazywać będziemy uporządkowaną czwórkę (V_N, V_T, P, S), gdzie V_N jest to zbiór symboli nieterminalnych, V_T - zbiór symboli terminalnych, $V = V_N \cup V_T$ - alfabet gramatyki, $P \subseteq V_N \times V^*$ - skończony zbiór par zwanych produkcjami, S - wyróżniony symbol należący do zbioru V_N. Produkcje gramatyki zapisywać będziemy w postaci $A \longrightarrow \alpha$, $A \in V_N, \alpha \in V^*$. Jeżeli $\beta, \gamma \in V^* - \{\varepsilon\}$ i istnieją słowa δ, τ oraz produkcja $A \longrightarrow \alpha$, takie że $\beta = \delta A \tau$ i $\gamma = \delta \alpha \tau$ to mówimy, że γ jest bezpośrednim wyprowadzeniem β, a β bezpośrednią redukcją γ i zapisujemy to w postaci: $\beta \Longrightarrow \gamma$. Jeżeli $\tau \in V_T^*$, to takie bezpośrednie wyprowadzenie nazywamy prawostronnym. Jeżeli $\beta, \gamma \in V^*$ i istnieje ciąg bezpośrednich wyprowadzeń $\beta = \alpha_1 \Longrightarrow \alpha_2 \Longrightarrow \ldots \Rightarrow \alpha_n = \gamma$, to mówimy, że słowo γ jest wyprowadzalne ze słowa β i ciąg $\alpha_1, \ldots, \alpha_n$ nazywamy wyprowadzeniem γ z β, a ciąg $\alpha_n, \ldots, \alpha_1$ redukcją słowa γ do słowa β, co oznaczamy przez $\beta \overset{+}{\Longrightarrow} \gamma$. Jeżeli $\beta \overset{+}{\Longrightarrow} \gamma$ lub $\beta = \gamma$ to fakt ten oznaczamy $\beta \overset{*}{\Longrightarrow} \gamma$.

Wyprowadzenie nazywamy prawostronnym, jeśli każde bezpośrednie wyprowadzenie zawarte w nim jest prawostronne. W przypadku prawostronnego wyprowadzenia stosujemy odpowiednio oznaczenia: $\beta \overset{r}{\Longrightarrow} \gamma$, $\beta \overset{+r}{\Longrightarrow} \gamma$, $\beta \overset{*r}{\Longrightarrow} \gamma$.

Niech G będzie gramatyką bezkontekstową. Jeżeli słowo γ jest wyprowadzalne z S, to γ nazywamy formą zdaniową $(S \overset{*}{\Longrightarrow} \gamma)$, zdaniem nazywać będziemy formę zdaniową składającą się tylko z symboli terminalnych.

Językiem L (G), generowanym przez gramatykę G nazywamy zbiór wszystkich zdań:

$$L(G) = \left\{ x : S \overset{*}{\Longrightarrow} x \quad \text{i} \quad x \in V_T^* \right\}.$$

Niech $\gamma = \beta \alpha \delta$ będzie formą zdaniową gramatyki G, jeśli $S \overset{*}{\Longrightarrow} \beta A \delta$ i $A \overset{+}{\Longrightarrow} \alpha$, to α nazywamy frazą.

Słowo β jest rdzeniem (uchwytem) formy zdaniowej α, jeżeli istnieje $A \in V_N$, $w \in V_T^*$, $\gamma \in V^*$ takie, że $\alpha = \gamma \beta w$ oraz $S \overset{*r}{\Longrightarrow} \gamma A w \Longrightarrow \gamma \beta w$.

Niech x będzie formą zdaniową. Tworzymy dla niej drzewo wywodu w następujący sposób: wierzchołkiem drzewa jest symbol wyróżniony gramatyki a pozostałe węzły są symbolami pojawiającymi się w wyprowadzeniu tej formy, z tym, że dla każdego rozgałęzienia w drzewie istnieje pro-

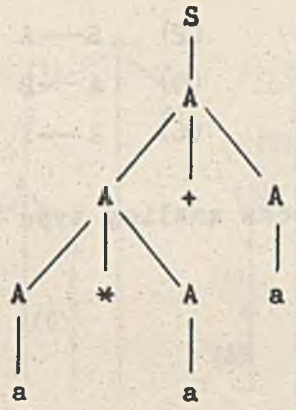
dukcja, której lewa strona jest punktem rozgałęzienia, a prawa jest słowem złożonym z węzłów tego rozgałęzienia.

Przykład

Dla gramatyki $G = (\{S, A\}, \{a, +, *\}, \{S \to A,\ A \to A + A,\ A \to A * A,\ A \to a\}, S)$ drzewo wywodu dla wyprowadzenia:

$$S \Longrightarrow A \Longrightarrow A + A \Longrightarrow A * A + A \Longrightarrow a * A + A \Longrightarrow a * a + A \Longrightarrow a * a + a$$

jest następujące:



Zdanie gramatyki jest niejednoznaczne, jeśli istnieją dla niego co najmniej dwa drzewa wywodu.

Gramatykę nazywamy niejednoznaczną, jeśli zawiera zdanie niejednoznaczne. W przeciwnym wypadku gramatyka jest jednoznaczna.

1.2. Analiza składniowa

Niech $G = (V_N, V_T, P, S)$ będzie gramatyką bezkontekstową. Przez analizę składniową rozumiemy rozpoznanie, czy dowolne słowo $x \in V_T$ jest zdaniem języka $L(G)$, czy też nie, oraz gdy słowo $x \in L(G)$ podanie jego struktury składniowej w gramatyce G, tzn. zbudowanie drzewa wywodu, bądź podanie ciągu produkcji użytych w wywodzie zdania x.

Możemy wyróżnić dwa główne rodzaje analizy: analizę typu redukcyjnego oraz analizę typu generacyjnego. Analizę typu redukcyjnego przeprowadzamy zaczynając redukcję od słowa terminalnego x aż do otrzymania

symbolu wyróżnionego gramatyki. Proces analizy typu generacyjnego rozpoczynamy od symbolu wyróżnionego i kolejno stosując produkcje gramatyki dochodzimy do danego słowa terminalnego. Będziemy się zajmowali deterministycznymi metodami analizy, tzn. takimi metodami, które w zależności od rodzaju analizy pozwalają w każdym kroku na jednoznaczną redukcję, czy też na zastosowanie jednoznacznie określonej produkcji.

Przykład

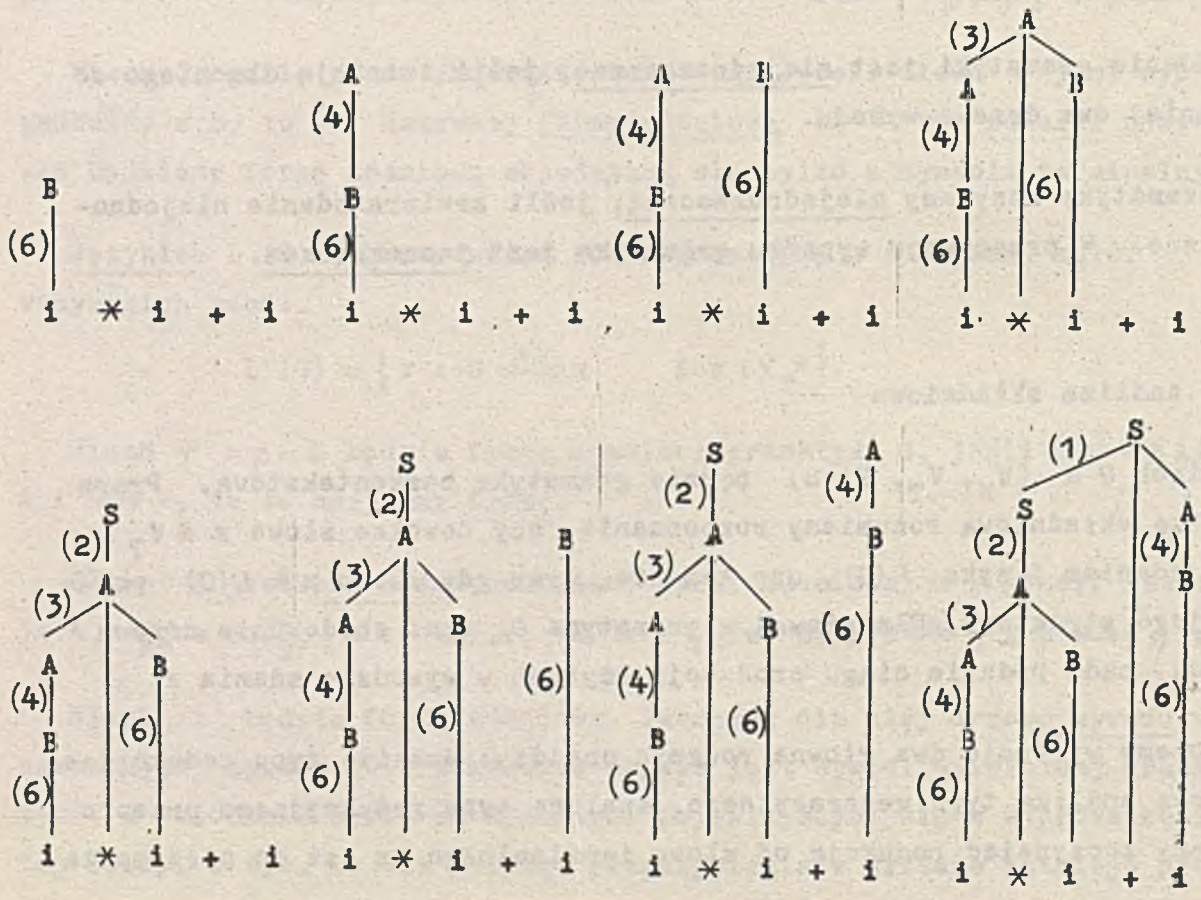
Rozpatrzmy gramatykę $G = (\{S, A, B\}, \{i, +, *, (,)\}, P, S)$, gdzie

P : (1) $S \to S + A$ (2) $S \to A$

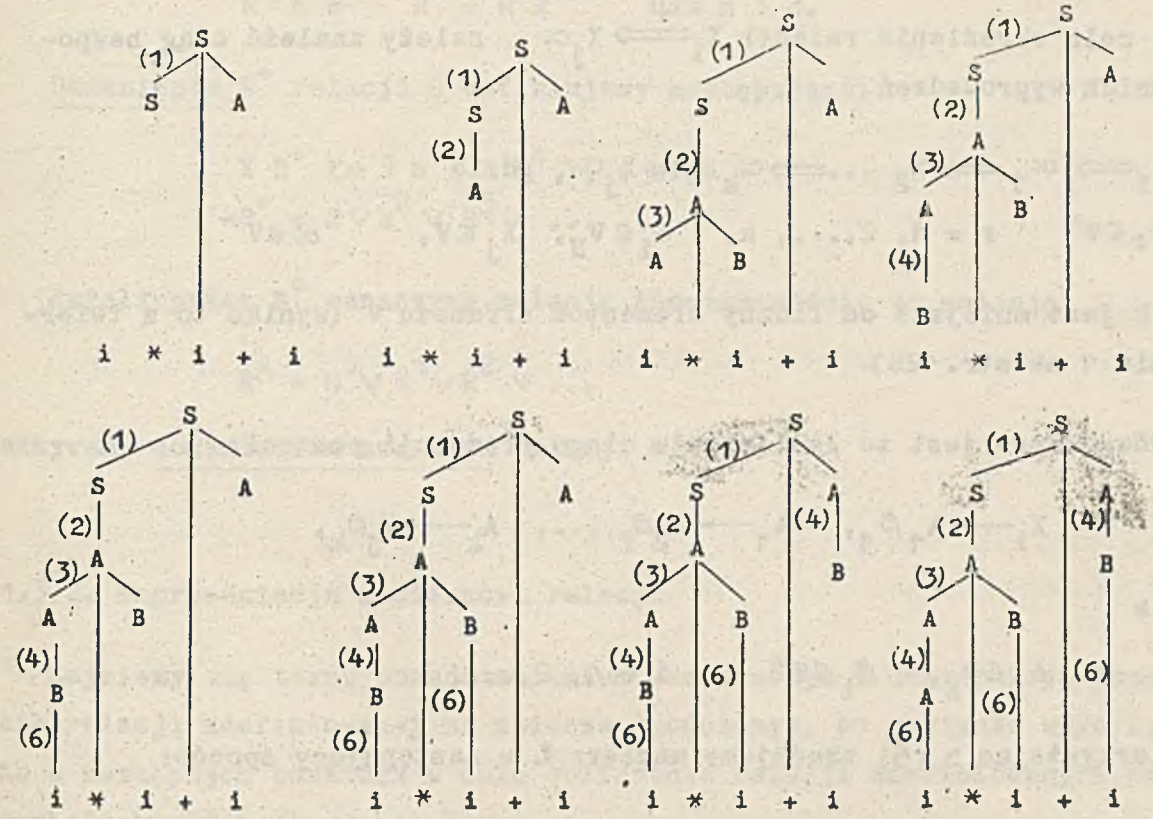
(3) $A \to A * B$ (4) $A \to B$

(5) $B \to (S)$ (6) $B \to i$

Poniżej scharakteryzowano proces analizy typu redukcyjnego dla słowa terminalnego i * i + i.



W czasie analizy typu generacyjnego drzewo wywodu budowane jest od symbolu wyróżnionego S:



1.3. Relacje związane z gramatykami i ich reprezentacja macierzowa

1.3.1. Relacje związane z gramatykami

Jeżeli V jest zbiorem, to każdy podzbiór $R \subseteq V \times V$ nazywamy <u>relacją binarną</u> na zbiorze V. Jeżeli symbole X i Y należą do zbioru V i uporządkowana para (X, Y) należy do R to mówimy, że X jest w relacji R z Y ($X\ R\ Y$). Relację przeciwną do relacji R na zbiorze V definiujemy następująco:

$$R^c \overset{Df}{=} \{ (Y, X) \mid (X, Y) \in R \}, \quad X, Y \in V$$

Niech R_1 i R_2 będą relacjami binarnymi na zbiorze V.

<u>Suma</u> relacji R_1 i R_2 : $R_1 \vee R_2$ jest sumą teoriomnogościową zbiorów R_1 i R_2 : $R_1 \vee R_2 = R_1 \cup R_2$.

<u>Złożeniem</u> lub <u>iloczynem</u> relacji R_1 i R_2 nazywamę relację R_1R_2 zdefiniowaną następująco:

$X\ (R_1R_2)\ Y \equiv \exists\ Z \in V$ takie, że $X\ R_1\ Z \wedge Z\ R_2\ Y$, dla $X, Y \in V$.

Pokażemy konstrukcję macierzy **L** reprezentującej relację $X_i \overset{+}{\Longrightarrow} X_j\alpha$; macierz R możemy obliczyć w analogiczny sposób.

W celu określenia relacji $X_i \overset{+}{\Longrightarrow} X_j\alpha$ należy znaleźć ciąg bezpośrednich wyprowadzeń

$$X_i \Longrightarrow \alpha_1 \Longrightarrow \alpha_2 \ldots \Longrightarrow \alpha_k \Longrightarrow X_j\alpha \text{ , gdzie}$$

$$\alpha_r \in V^* \qquad r = 1, 2, \ldots, k, \quad X_i \in V_N, \quad X_j \in V, \quad \alpha \in V^*$$

zaś k jest mniejsze od liczby elementów alfabetu V (wynika to z twierdzenia 1 na str. 128).

Równoważne jest to znalezieniu ciągu produkcji postaci:

$$(5) \qquad X_i \longrightarrow A_1\beta_1, \quad A_1 \longrightarrow A_2\beta_2, \ldots, \quad A_k \longrightarrow X_j\beta_k,$$

gdzie

$$A_i \in V_N, \quad \beta_i \in V^*, \qquad i = 1, 2, \ldots, k.$$

Korzystając z (5) zbudujemy macierz **L** w następujący sposób:

Niech L_o będzie macierzą boolowską taką, że dla $X_i \in V_N$, $X_j \in V$, $\alpha \in V^*$:

$l_{ij} = 1 \equiv$ istnieje produkcja postaci $X_i \longrightarrow X_j\alpha$

$l_{ij} = 0 \equiv$ w przeciwnym wypadku

Ponieważ domknięciem relacji $X_i \longrightarrow X_j\alpha$ jest $X_i \overset{+}{\Longrightarrow} X_j\beta$ czyli ciąg produkcji (1), zatem macierz $L = L_o^+$ możemy obliczyć korzystając z algorytmu Warshalla.

Przykład 1

Rozpatrzmy gramatykę $G = (V_N, V_T, P, S)$,
gdzie $V_N = \{S, A\}$, $V_T = \{a, b\}$,
P : $S \longrightarrow Aa$
$A \longrightarrow b$
$A \longrightarrow AS$

Niech $X_1 = S$, $X_2 = A$, $X_3 = a$, $X_4 = b$, wtedy

Zdefiniujemy indukcyjnie n-tą potęgę relacji R:

$$R^1 = R \qquad R^n = R\,R^{n-1} \quad \text{dla } n > 1.$$

Domknięcie R^+ relacji R definiujemy następująco:

$$X\,R^+\,Y \equiv \exists\, n : X\,R^n\,Y, \text{ a więc}$$
$$R^+ = R \vee R^2 \vee R^3 \vee \ldots$$

Jeżeli przez R^o oznaczymy relację identyczności, to relację

$$R^* = R^o \vee R^1 \vee R^2 \vee \ldots$$

nazywamy domknięciem zwrotnym relacji R.

1.3.2. Reprezentacja macierzowa relacji

Zajmiemy się teraz przedstawieniem macierzowym i konstrukcją domknięcia relacji zdefiniowanej na zbiorze skończonym, co zostanie wykorzystane w następnych punktach w celu obliczenia relacji zdefiniowanych na symbolach alfabetu gramatyki.

Twierdzenie 1

Niech V będzie alfabetem składającym się z n symboli i niech R będzie dowolną relacją binarną na zbiorze V. Jeśli dla dwóch symboli $X, Y \in V$ zachodzi relacja: $X\,R^+Y$, to istnieje wtedy liczba naturalna $p \leqslant n$ taka, że $X\,R^p\,Y$.

Relacje binarne można reprezentować również w maszynie cyfrowej jako macierze boolowskie. Macierz boolowska B jest to taka macierz, w której elementy b_{ij} mogą przyjmować wartości 1 lub 0.

Macierz B jest reprezentacją relacji binarnej R nad alfabetem $V = \{X_1, \ldots, X_n\}$ jeżeli spełniony jest warunek:

$$X_i\,R\,X_j \equiv b_{ij} = 1 \qquad (i, j = 1, \ldots, n)$$

Sumę macierzy boolowskich $D = B + C$ definiuje się następująco:

$$d_{ij} = b_{ij} \vee c_{ij} \qquad (\vee\text{- suma logiczna})$$

Mnożenie $D = B \cdot C$ definiujemy jako:

$$d_{ij} = \bigvee_k (b_{ik} \wedge c_{kj}) \qquad (\wedge - \text{iloczyn logiczny}).$$

Podamy teraz cztery twierdzenia dotyczące reprezentowania relacji przez macierze boolowskie.

Twierdzenie 2

Suma dwóch relacji nad tym samym alfabetem reprezentowana jest przez sumę macierzy boolowskich reprezentujących te relacje.

Twierdzenie 3

Relacja odwrotna do relacji R reprezentowana jest przez macierz transponowaną do macierzy reprezentującej relację R.

Twierdzenie 4

Iloczyn dwóch relacji nad tym samym alfabetem reprezentowany jest przez iloczyn macierzy boolowskich reprezentujących te relacje.

Twierdzenie 5

Niech B będzie boolowską macierzą n x n reprezentującą relację R nad alfabetem V. Wtedy macierz B^+ zdefiniowana następująco:

$$B^+ = B + B \cdot B + \ldots + \underbrace{B \cdot B \cdot \ldots \cdot B}_{n \text{ razy}}$$

reprezentuje domknięcie R^+ relacji R.

W celu obliczenia macierzy B^+ można wykorzystać algorytm podany przez Warshalla [8].

- Przeglądamy od góry do dołu kolejne kolumny macierzy B, poczynając od skrajnie lewej. W momencie napotkania jedynki przechodzimy do czynności opisanych w punkcie poniższym.

- Przypuśćmy, że w trakcie przeglądania j-tej kolumny napotkamy jedynkę w i-tym wierszu. Należy wówczas znaleźć j-ty wiersz i w i-tym wierszu wstawić jedynki wszędzie tam, gdzie występują jedynki w

j-tym wierszu. Pozostałe elementy i-tego wiersza pozostają niezmienione.

Następnie przechodzimy znowu do punktu pierwszego przeglądając dalsze wiersze danej kolumny, lub przechodząc, w przypadku końca tej kolumny do następnej. Cały proces kończymy z chwilą zakończenia przeglądania ostatniej kolumny. Otrzymana w ten sposób macierz jest szukaną macierzą B^+.

Powyższy algorytm można prosto zapisać w ALGOL-u.

Rozważmy macierz B [m, n] z m wierszami i n kolumnami.

```
for     j:= 1 step 1 until n do
for     i:= 1 step 1 until m do
if      b_ij = 1 ∧ i ≠ j then
for     k:= 1 step 1 until n do
b_ik := b_ik ∨ b_jk
```

Przykład

Rozważmy gramatykę $G = (V_N, V_T, P, S)$:

$V_N = \{S, A, B\}$

$V_T = \{a, b, +, =\}$

P : $S \longrightarrow A = B$

$B \longrightarrow A$

$B \longrightarrow B + A$

$A \longrightarrow a$

$A \longrightarrow b$

Rozważmy relację L, zdefiniowaną następująco:

$X_i \; L \; X_j \equiv X_i \longrightarrow X_j \ldots$

Macierz B reprezentująca tę relację w gramatyce G, ma następującą postać:

		1	2	3	4	5 = n
	i/j	S	A	B	a	b
1	S	0	1	0	0	0
2	A	0	0	0	1	1
3	B	0	1	1	0	0
4	a	0	0	0	0	0
m = 5	b	0	0	0	0	0

Korzystając z powyższego algorytmu otrzymujemy macierz B^+:

	S	A	B	a	b
S	0	1	0	1	1
A	0	0	0	1	1
B	0	1	1	1	1
a	0	0	0	0	0
b	0	0	0	0	0

Macierz B^+ reprezentuje relację $X_i \overset{+}{\Longrightarrow} X_j \ldots$.

1.4. Praktyczne ograniczenia na gramatyki

Podamy teraz dwa ograniczenia na gramatyki wynikające z praktyki ich stosowania. Ograniczenia te nie zawężają klasy języków generowanych przez te gramatyki, ale ułatwiają analizę tych języków. Po pierwsze: gramatyka nie może dopuszczać wyprowadzeń postaci $A \overset{+}{\Longrightarrow} A$. Takie wyprowadzenia czynią gramatykę niejednoznaczną. Po drugie: każdy symbol nieterminalny A w gramatyce musi spełniać trzy warunki:

1) musi występować w pewnej formie zdaniowej $\alpha A \beta$ takiej, że

$$S \overset{*}{\Longrightarrow} \alpha A \beta, \qquad \alpha, \beta \in V^*$$

2) można wyprowadzić z niego słowo terminalne

$$A \overset{+}{\Longrightarrow} x, \qquad x \in V_T^*$$

3) nie można wyprowadzić z niego słowa pustego, tzn. nie istnieje produkcja postaci: $A \longrightarrow \epsilon$

Warunki 1) i 2) mówią nam o tym, że gramatyka nie zawiera produkcji "bezużytecznej" tj. takiej, że w języku nie istnieje zdanie, w wypro-

wadzeniu którego występowałaby ta produkcja. Istnieją proste algorytmy [5] eliminujące symbole nieterminalne nie spełniające warunków 1-3.

Gramatykę, w której każdy symbol nieterminalny spełnia dwa pierwsze warunki nazywamy zredukowaną.

W dalszej części będziemy zakładać, że gramatyka jest gramatyką zredukowaną oraz nie zawiera wyprowadzeń postaci $A \overset{+}{\Longrightarrow} A$ i produkcji $A \longrightarrow \epsilon$.

2. Gramatyki z prostym pierwszeństwem
(z relacjami pierwszeństwa typu (1,1))

W redukcyjnej metodzie analizowania problem polega na znalezieniu "rdzenia" w aktualnie rozpatrywanej formie zdaniowej i zredukowanie go do symbolu nieterminalnego. W tym rozdziale rozwiążemy ten problem dla pewnej klasy gramatyk tzw. gramatyk z prostym pierwszeństwem. Ponieważ formę zdaniową analizujemy "od lewej do prawej", to zarówno wszystkie formy zdaniowe jak i wszystkie wyprowadzenia będą prawostronne. W dalszej treści "forma zdaniowa" oznaczać będzie "prawostronną formę zdaniową".

Jeżeli przed analizą uda nam się dla każdej pary symboli gramatyki zdefiniować jednoznaczne relacje, za pomocą których możemy określić początek i koniec rdzenia, to rdzeń danej formy zdaniowej możemy znaleźć, np.przeglądając ją od lewej do prawej i rozpatrując określone uprzednio relacje na dwóch sąsiednich symbolach aż znajdziemy koniec rdzenia, a następnie,cofając się symbol po symbolu,znajdujemy w podobny sposób początek rdzenia.

Relacje na symbolach gramatyki określamy nieformalnie w poniższy sposób.

Rozważmy dwa dowolne symbole X i $Y \in V$, dla których istnieje forma zdaniowa postaci $\alpha X Y \beta$, gdzie α, β są słowami nad alfabetem V.

Mogą zajść trzy przypadki:

1) X jest częścią rdzenia a Y nie jest; relację tę oznaczamy $X \doteq Y$;

X ma pierwszeństwo nad Y, ponieważ musi być redukowane przed Y; w gramatyce musi zatem istnieć produkcja postaci $A \longrightarrow \ldots X$;

2) obydwa symbole X i Y występują w rdzeniu ($X \doteq Y$); symbole X i Y mają to samo pierwszeństwo (muszą być redukowane równocześnie); w gramatyce musi więc istnieć produkcja $A \longrightarrow \gamma X Y \delta$; $\gamma, \delta \in V^*$;

3) Y jest częścią rdzenia, lecz X nie jest; mowimy, że X jest mniejsze od Y ($X \lessdot Y$); w gramatyce musi istnieć produkcja postaci: $A \longrightarrow Y \ldots$.

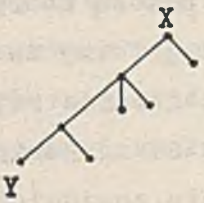
Gdy dla symboli X, Y nie istnieje forma zdaniowa postaci $\alpha X Y \beta$ wtedy mówimy, że między X i Y nie zachodzą relacje pierwszeństwa.

2.1. Definicja i konstrukcja relacji pierwszeństwa

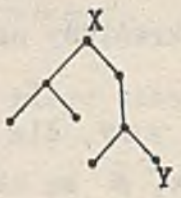
Niech G będzie zredukowaną gramatyką bezkontekstową o alfabecie V. Z definicji relacji $\Longrightarrow$, $\overset{+}{\Longrightarrow}$, $\overset{*}{\Longrightarrow}$ wynika, że relacja $\overset{+}{\Longrightarrow}$ jest domknięciem relacji $\Longrightarrow$, a relacja $\overset{*}{\Longrightarrow}$ jest domknięciem zwrotnym relacji $\Longrightarrow$.

Zdefiniujemy następujące trzy relacje na symbolach gramatyki:

$X\ (G_L)\ Y \equiv$ istnieje wyprowadzenie postaci $X \overset{+}{\Longrightarrow} Y \ldots$



$X\ (G_R)\ Y \equiv$ istnieje wyprowadzenie postaci $X \overset{+}{\Longrightarrow} \ldots . Y$



$X\ (G_L^*)\ Y \equiv$ istnieje wyprowadzenie postaci $X \overset{*}{\Longrightarrow} Y \ldots$

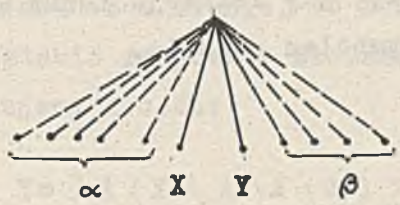
Podamy teraz formalną definicję relacji pierwszeństwa.

Definicja

Dla danej gramatyki G definiujemy między symbolami alfabetu V trzy relacje pierwszeństwa w następujący sposób:

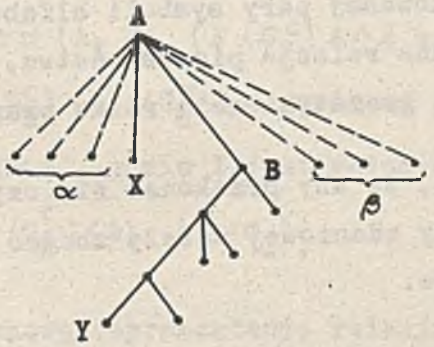
1) $X \doteq Y \equiv$ istnieje w gramatyce G produkcja

$A \longrightarrow \alpha X Y \beta$, $\alpha, \beta \in V^*$

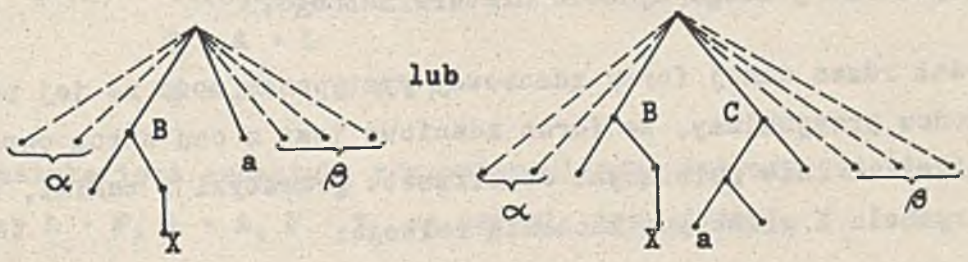


2) $X \lessdot Y \equiv$ istnieje w gramatyce G produkcja

$A \longrightarrow \alpha X B \beta$ taka, że zachodzi relacja $B (G_L) Y$ tzn. $B \overset{+}{\Longrightarrow} Y...$



3) $X \gtrdot a \equiv$ a jest symbolem terminalnym i istnieje w gramatyce G produkcja $A \longrightarrow \alpha B C \beta$ taka, że zachodzą relacje $B (G_R) X$ i $C (G_L^*) a$ tzn. $B \overset{+}{\Longrightarrow} ... X$ i $C \overset{*}{\Longrightarrow} a...$



Podamy twierdzenia dotyczące związku relacji pierwszeństwa z rdzeniem formy zdaniowej.

Twierdzenie 1

$X \doteq Y$ wtedy i tylko wtedy, gdy słowo X Y jest częścią rdzenia pewnej formy zdaniowej.

Twierdzenie 2

X $\dot{>}$ a wtedy i tylko wtedy, gdy istnieje forma zdaniowa α X a β, w której X jest ostatnim symbolem rdzenia.

Twierdzenie 3

X $\dot{<}$ Y wtedy i tylko wtedy, gdy istnieje forma zdaniowa α X Y β, w której Y jest pierwszym symbolem rdzenia.

Definicja

Gramatyka G jest gramatyką z prostym pierwszeństwem lub gramatyką z pierwszeństwem typu (1,1), jeżeli:

- dla każdej uporządkowanej pary symboli alfabetu gramatyki G zachodzi co najwyżej jedna relacja pierwszeństwa,
- wszystkie produkcje gramatyki mają różne prawe strony.

(1,1) wskazuje na to, że aby przekonać się czy pewien ciąg symboli jest rdzeniem danej formy zdaniowej należy zbadać tylko jeden symbol po każdej stronie tego ciągu.

Jeżeli gramatyka G jest gramatyką z prostym pierwszeństwem, to relacje $\dot{<}$, $\doteq$, $\dot{>}$ określają sposób w jaki można znaleźć rdzeń każdej formy zdaniowej. Drugi warunek gwarantuje, że rdzeń ten można zredukować tylko do jednego symbolu nieterminalnego.

Ponieważ rdzeń danej formy zdaniowej występować może na jej początku lub końcu przyjmujemy, że forma zdaniowa jest z obu stron ograniczona symbolami $\perp$ nie należącymi do alfabetu gramatyki i takimi, że dla każdego symbolu X gramatyki zachodzą relacje:

$$\perp \dot{<} X, \quad X \dot{>} \perp$$

Twierdzenie 4

Gramatyka z prostym pierwszeństwem jest jednoznaczna. Ponadto rdzeniem dowolnej formy zdaniowej $X_1 \ldots X_n$ jest położone najbardziej po lewej stronie słowo $X_j \ldots X_i$ takie, że

(1) $$X_{j-1} \lessdot X_j \doteq X_{j+1} \doteq \ldots \doteq X_i \gtrdot X_{i+1}$$

Konstrukcja relacji pierwszeństwa

Konstrukcja relacji $\doteq$ nie nastręcza żadnych kłopotów. Wystarczy przejrzeć wszystkie prawe strony produkcji i jeśli symbole X oraz Y występują w produkcjach obok siebie, to $X \doteq Y$. Relacje pierwszeństwa $\lessdot$ i $\gtrdot$ można przedstawić za pomocą złożenia relacji $\doteq$ z relacjami G_L, G_R, G_L^* w następujący sposób:

$X \lessdot Y \equiv \exists A\,(X \doteq A \wedge A\,(G_L)\,Y)$, czyli

(1) $\lessdot = (\doteq)\,(G_L)$

$X \gtrdot Y \equiv \exists (A, B)(A\,(G_R)X \wedge A \doteq B \wedge B\,(G_L^*)\,Y)$, czyli

$X \gtrdot Y \equiv \exists (A, B)\ (X\,(G_R^c)\,A \wedge A \doteq B \wedge B\,(G_L^*)\,Y)$, więc

$\gtrdot = (G_R^c)\,(\doteq)\,(G_L^*)$, a ponieważ

$G_L^* = I \vee G_L$, gdzie I jest relacją identyczności, więc

(2) $\gtrdot = (G_R^c)\,(\doteq)\ (I \vee G_L)$.

Zastosujemy macierzową reprezentację relacji w celu obliczenia relacji $\lessdot$, $\gtrdot$. Niech macierze **A**, **L**, **R**, **P** i **S** będą kolejno reprezentacjami relacji: $\doteq$, G_L, G_R, $\lessdot$ i $\gtrdot$.

Z (1) i (2) otrzymujemy:

(3) $$\mathbf{P} = \mathbf{A} \cdot \mathbf{L}$$

(4) $$\mathbf{S} = \mathbf{R}^T \cdot (\mathbf{A} + \mathbf{P})$$

Gramatyka jest gramatyką z prostym pierwszeństwem, jeżeli iloczyny macierzy $\mathbf{A} \cdot \mathbf{P}$, $\mathbf{A} \cdot \mathbf{S}$, $\mathbf{P} \cdot \mathbf{S}$ są zerami, tzn. jeżeli:

$\mathbf{A} \cdot \mathbf{P} = \mathbf{A} \cdot \mathbf{S} = \mathbf{P} \cdot \mathbf{S} = [0]$ - macierz zerowa

Aby obliczyć macierze **L** i **R** najpierw należy obliczyć relacje postaci

$$X_i \overset{+}{\Longrightarrow} X_j\alpha \quad \text{i} \quad X_i \overset{+}{\Longrightarrow} \alpha X_j$$

dla wszystkich $X_i \in V_N$, $X_j \in V$, $\alpha \in V^*$

$$A = \begin{array}{c|cccc} X_i \backslash X_j & S & A & a & b \\ \hline S & 0 & 0 & 0 & 0 \\ A & 1 & 0 & 1 & 0 \\ a & 0 & 0 & 0 & 0 \\ b & 0 & 0 & 0 & 0 \end{array} \qquad L_o = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad R_o = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad R^T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$P = A \cdot L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad S = R^T \cdot (A+P) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Ponieważ $A \cdot P = A \cdot S = P \cdot S = [0]$, więc G jest gramatyką z prostym pierwszeństwem.

2.2. Algorytm analizowania

Podamy teraz algorytm analizowania (rozpoznawania) zdań korzystający z relacji pierwszeństwa.

Jeżeli mamy gramatykę z prostym pierwszeństwem, to relacje pierwszeństwa możemy reprezentować jako macierz B o wartościach:

$b_{ij} = 0$ jeśli nie istnieje relacja pierwszeństwa między symbolami X_i, X_j

$b_{ij} = 1$ jeśli $X_i \lessdot X_j$

$b_{ij} = 2$ jeśli $X_i \doteq X_j$

$b_{ij} = 3$ jeśli $X_i \gtrdot X_j$

Jeżeli macierze P, A, S reprezentują odpowiednie relacje $\lessdot$, $\doteq$, $\gtrdot$, to macierz B możemy skonstruować w następujący sposób:

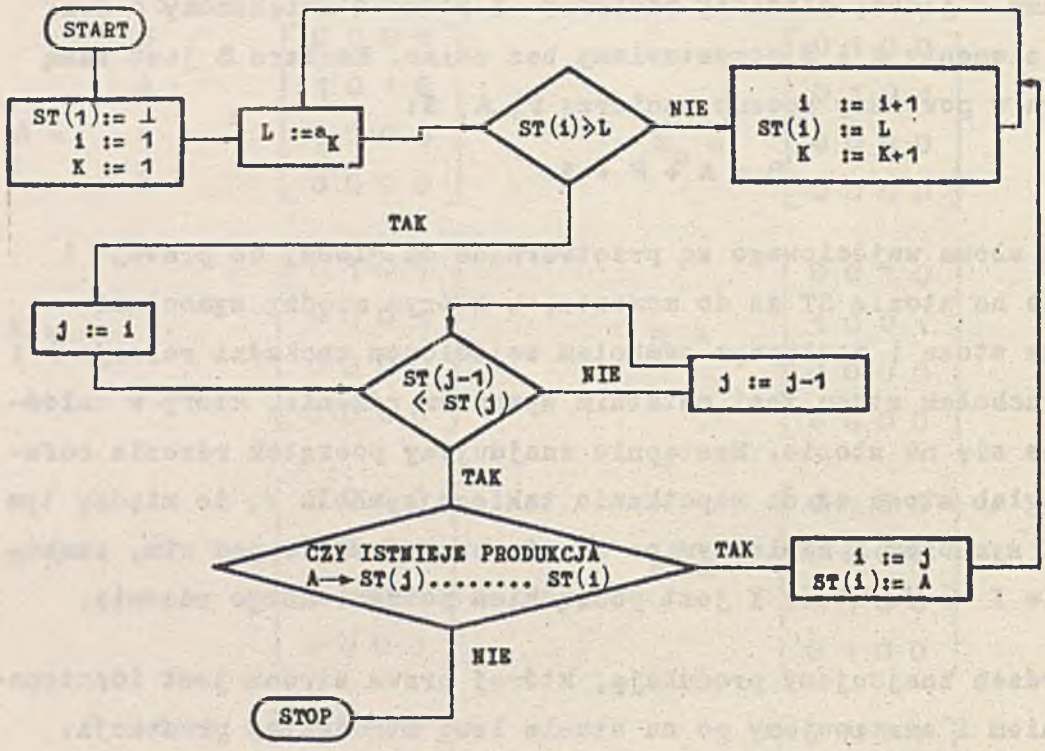
elementy macierzy P pozostawiamy bez zmian; elementy macierzy A równe 1 zwiększamy o jeden; elementy macierzy S równe 1 zwiększamy o dwa; pozostałe elementy A i S pozostawiamy bez zmian. Macierz B jest sumą zmienionych w powyższy sposób macierzy P, A, S:

$$B = A + P + S$$

Symbole słowa wejściowego są przetwarzane od "lewej do prawej" i umieszczane na stosie ST aż do momentu, w którym między symbolami wierzchołka stosu i następnym symbolem wejściowym zachodzi relacja $\doteq$. Wtedy wierzchołek stosu jest ostatnim symbolem rdzenia, który w całości znajduje się na stosie. Następnie znajdujemy początek rdzenia cofając się w głąb stosu aż do napotkania takiego symbolu Y, że między tym symbolem a symbolem X zapisanym na stosie bezpośrednio pod nim, zachodzi relacja $X \lessdot Y$. Wtedy Y jest początkiem poszukiwanego rdzenia.

Mając rdzeń znajdujemy produkcję, której prawa strona jest identyczna z rdzeniem i zastępujemy go na stosie lewą stroną tej produkcji. Proces powyższy powtarzamy aż do momentu, gdy na stosie będzie tylko symbol S, a następnym symbolem $\perp$.

Poniżej podajemy dokładny schemat działania tego algorytmu. W schemacie tym ST oznacza stos, zaś i, j oznaczają indeksy pomocnicze stosu. Słowo $a_1 a_2 \ldots a_n$ jest analizowanym zdaniem. Analizę zaczynamy z ogranicznikiem $\perp$ na stosie, a ponadto zakładamy, że symbol $a_{n+1} = \perp$. Q i L są rejestrami pomocniczymi używanymi do przechowywania symboli w czasie analizy. K jest licznikiem symboli w analizowanym słowie.



$a_1 \ldots a_n$ jest zdaniem wtedy i tylko wtedy, gdy po zakończeniu pracy algorytmu i = 2, ST(2) = S, L = ⊥.

Przykład 2

Zbadamy czy słowo bbaa należy do języka generowanego przez gramatykę z przykładu 1.

Macierz B dla tej gramatyki ma postać:

$$B = \begin{bmatrix} 3 & 3 & 3 & 3 \\ 2 & 1 & 2 & 1 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

Pokażemy działanie analizatora podając w każdym kroku zawartość stosu ST, następny symbol wejściowy L, relację między wierzchołkiem stosu ST i symbolem L oraz pozostałą część zdania:

Krok	S(1)	S(2) ...	relacja	L	a_k ...
0	⊥		$\lessdot$	b	b a a ⊥
1	⊥	b	$\gtrdot$	b	a a ⊥
2	⊥	A	$\lessdot$	b	a a ⊥
3	⊥	A b	$\gtrdot$	a	a ⊥
4	⊥	A A	$\doteq$	a	a ⊥
5	⊥	A A a	$\gtrdot$	a	⊥
6	⊥	A S	$\gtrdot$	a	⊥
7	⊥	A	$\doteq$	a	⊥
8	⊥	A a	$\lessdot$	⊥	
9	⊥	S	$\gtrdot$	⊥	

Słowo bbaa jest zdaniem należącym do języka L (G).

2.3. Funkcje pierwszeństwa

Macierz pierwszeństwa dla gramatyki zajmuje dużo miejsca w pamięci maszyny: dla języka o 100 symbolach macierz pierwszeństwa składa się ze 100 x 100 co najmniej dwubitowych elementów. Często więc do reprezentowania relacji zamiast macierzy używa się dwóch funkcji zwanych funkcjami pierwszeństwa, takich, że dla wszystkich symboli gramatyki:

$$
\begin{aligned}
& X \doteq Y \supset f(X) = g(Y) \\
(1) \qquad & X \lessdot Y \supset f(X) < g(Y) \\
& X \gtrdot Y \supset f(X) > g(Y)
\end{aligned}
$$

Zauważmy, że dla danej macierzy pierwszeństwa może istnieć wiele funkcji f i g spełniających podane wyżej warunki. Z drugiej jednak strony dla pewnych macierzy pierwszeństwa funkcje takie w ogóle nie istnieją.

Na przykład dla macierzy

	X_1	X_2
X_1	$\doteq$	$\gtrdot$
X_2	$\doteq$	$\doteq$

funkcje f i g nie istnieją, ponieważ z (1)

$f(X_1) > g(X_2)$, $g(X_2) = f(X_2)$
$f(X_2) = g(X_1)$, $g(X_1) = f(X_1)$, czyli

otrzymaliśmy sprzeczność: $f(X_1) > f(X_1)$

Zanim przedstawimy metodę konstrukcji funkcji pierwszeństwa, podamy parę podstawowych definicji i twierdzeń z teorii grafów skierowanych.

Definicja

Graf skierowany jest to para zbiorów $\mathfrak{D} = (X, M)$, gdzie X jest skończonym zbiorem węzłów, a $M \subseteq X \times X$ jest zbiorem gałęzi (linii skierowanych łączących węzły).

Jeżeli x i y są dwoma węzłami grafu skierowanego i jeżeli istnieje droga z x do y złożona z gałęzi o niezerowej długości, gdzie długość jest ilością gałęzi występujących w tej drodze, wtedy y nazywamy następnikiem x, jeżeli droga ma długość jeden, wtedy y jest bezpośrednim następnikiem x. Zbiór następników węzła x oznaczamy przez $\mathfrak{v}(x)$, a zbiór bezpośrednich następników przez $\mathfrak{v}_1(x)$. Jeżeli S jest zbiorem, to $|S|$ oznacza liczbę jego elementów tego zbioru. Cyklem $C = (x_1, x_2, \ldots, x_k)$ nazywamy uporządkowaną listę węzłów takich, że istnieją gałęzie: $(x_1, x_2), \ldots, (x_{k-1}, x_k), (x_k, x_1)$

Graf skierowany $\mathfrak{D} = (X, M)$ może być reprezentowany przez macierz boolowską B o wymiarach $|X|$, w której $b_{ij} = 1$ wtedy i tylko wtedy, gdy $(x_i, x_j) \in M$ i $b_{ij} = 0$ w przeciwnym przypadku.

Macierz B odpowiadająca grafowi $\mathfrak{D}$ reprezentuje również pewną relację binarną.

Relacja domknięcia D^+ grafu skierowanego $\mathfrak{D}$ tworzona jest przez dodanie gałęzi (x, y) wszędzie tam, gdzie $y \in \mathfrak{v}(x)$. Odpowiadającą jej macierz B^+ można obliczyć korzystając z algorytmu Warshalla [8]. Tak więc $b^+_{ij} = 1 \equiv x_j \in \mathfrak{v}(x_i)$. Niech $\mathfrak{D} = (X, M)$ będzie grafem skierowanym.

Definicja 1

Funkcję $N : X \to \{0, 1, 2, \ldots, |X|\}$ definiujemy jako

$$N_{\mathfrak{D}}(x) = |\mathfrak{v}(x)|$$

Definicja 2

Niech $N : X \longrightarrow \{0, 1, \ldots, |X|\}$ będzie dowolną funkcją. N nazwiemy funkcją porządkującą wierzchołki grafu, gdy dla wszystkich x, y $\in X$ jeżeli y $\in \Gamma(x)$, to $N(x) > N(y)$.

Twierdzenie 1

$N_{\mathcal{D}}$ jest funkcją porządkującą wierzchołki wtedy i tylko wtedy, gdy graf $\mathcal{D}$ jest niecykliczny.

Podamy teraz algorytm tworzenia funkcji pierwszeństwa na podstawie zadanej macierzy pierwszeństwa.

Algorytm ten jest pewną modyfikacją algorytmu Bella [1]; znajduje on funkcje pierwszeństwa, o ile takie funkcje istnieją.

Krok A. Tworzymy graf skierowany $\mathcal{G}$ z węzłami $f_1, \ldots, f_n, g_1, \ldots, g_n$ i gałęziami (f_i, g_j) jeśli $Y_i > Y_j$ i (g_j, f_i) jeśli $Y_i < Y_j$.

Krok B. Jeżeli $Y_i \doteq Y_j$, to przez poprowadzenie gałęzi z węzła f_i do wszystkich węzłów będących bezpośrednimi następnikami węzła g_j oraz z węzła g_j do wszystkich węzłów będących bezpośrednimi następnikami węzła f_i zrównujemy zbiory $\Gamma(f_i)$ i $\Gamma(g_j)$

Krok C. Każdemu węzłowi x grafu skonstruowanego w kroku A oraz kroku B przypisujemy liczbę $|\Gamma(x)| = N_{\mathcal{D}}(x)$.

Twierdzenie 2

Niech G będzie gramatyką z prostym pierwszeństwem i niech $\mathcal{G}$ będzie grafem skierowanym, skonstruowanym w krokach A oraz B. Funkcje pierwszeństwa istnieją wtedy i tylko wtedy, gdy graf $\mathcal{D}$ nie jest cykliczny i wówczas funkcje f i g zdefiniowane następująco:

$$f(Y_i) = N_{\mathcal{D}}(f_i) \quad \text{i} \quad g(Y_i) = N_{\mathcal{G}}(g_i)$$

są funkcjami pierwszeństwa.

Algorytm konstruowania funkcji pierwszeństwa można sformułować w terminach macierzy boolowskich.

Niech A, P i S będą macierzowymi reprezentacjami relacji $\doteq$, $\lessdot$ i $\gtrdot$

krok A´

Reprezentacją grafu skierowanego z kroku A jest macierz M_o

$$M_o = \begin{bmatrix} O & S \\ P & O \end{bmatrix}$$

krok B´

Należy obliczyć macierz boolowską $M = C \cdot M_o$, gdzie

$$C = I + \begin{bmatrix} O & A \\ A^T & O \end{bmatrix}^+ = \begin{bmatrix} I & A \\ A^T & I \end{bmatrix}^+$$

gdzie I jest macierzą jednostkową

krok C´

Obliczamy macierz M^+. Przyporządkowujemy każdemu węzłowi x liczbę równą ilości jedynek w x-tym wierszu macierzy M^+. Funkcje pierwszeństwa istnieją wtedy i tylko wtedy, gdy graf nie jest cykliczny, tzn. na głównej przekątnej macierzy M^+ nie pojawiają się jedynki.

Przykład

Rozważmy gramatykę $G = (V_N, V_T, P, S)$

$V_N = \{S, A\}$

$V_T = \{\lambda, (,)\}$

P :

$S \longrightarrow A$)

$A \longrightarrow ($

$A \longrightarrow A\lambda$

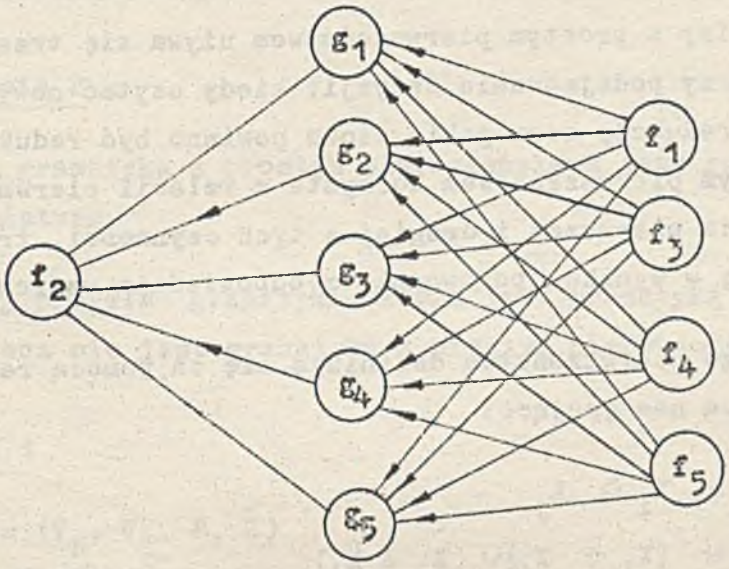
$A \longrightarrow A\ S$

Macierz pierwszeństwa B dla tej gramatyki jest następująca:

$$B = \begin{array}{c} \\ S \\ A \\ \lambda \\ (\\ ) \end{array} \begin{array}{c} \begin{array}{ccccc} S & A & \lambda & (&) \end{array} \\ \begin{bmatrix} \gtrdot & \gtrdot & \gtrdot & \gtrdot & \gtrdot \\ \doteq & \lessdot & \doteq & \lessdot & \doteq \\ \gtrdot & \gtrdot & \gtrdot & \gtrdot & \gtrdot \\ \gtrdot & \gtrdot & \gtrdot & \gtrdot & \gtrdot \\ \gtrdot & \gtrdot & \gtrdot & \gtrdot & \gtrdot \end{bmatrix} \end{array}$$

Korzystając z algorytmu zbudujemy graf skierowany odpowiadający tej macierzy pierwszeństwa. Aby uniknąć niepotrzebnego zamazania grafu

przez gałęzie, węzły, których bezpośrednie następniki były zmieniane w kroku B są połączone linią bez strzałki. Jeżeli będziemy wizualnie poszukiwać cykli, takie gałęzie powinny być rozważane jako linie mające strzałki w dwóch kierunkach.



Graf ten nie jest cykliczny, a więc na mocy twierdzenia 2 funkcje pierwszeństwa istnieją.

Odpowiednie macierze boolowskie mają postać:

$$M = M_0 = \begin{bmatrix} & & 11111 \\ & & 00000 \\ & 0 & 11111 \\ & & 11111 \\ & & 11111 \\ 00000 & & \\ 01000 & & \\ 00000 & & 0 \\ 01000 & & \\ 00000 & & \end{bmatrix} \qquad M^+ = \begin{bmatrix} 01000 & 11111 \\ 00000 & 00000 \\ 01000 & 11111 \\ 01000 & 11111 \\ 01000 & 11111 \\ 00000 & 00000 \\ 01000 & 00000 \\ 00000 & 00000 \\ 01000 & 00000 \\ 00000 & 00000 \end{bmatrix}$$

Ponieważ w macierzy M^+ na głównej przekątnej nie występują jedynki, funkcje pierwszeństwa istnieją i są równe:

$$(f_1 \cdots f_5 \; g_1 \cdots g_5) = (6066601010)$$

3. Inne metody analizy korzystające z relacji pierwszeństwa

3.1. Gramatyki ze słabym pierwszeństwem

Koncepcja słabego pierwszeństwa jest uogólnieniem prostego pierwszeństwa opisanego w poprzednim punkcie.

W czasie analizy z prostym pierwszeństwem używa się trzech relacji $\lessdot$, $\doteq$, $\gtrdot$ przy podejmowaniu decyzji: kiedy czytać nowy symbol, kiedy wykonywać redukcję oraz jakie słowo powinno być redukowane. Analizator ze słabym pierwszeństwem korzysta z relacji pierwszeństwa tylko do wykonania pierwszej i drugiej z tych czynności, trzecią czynność wykonuje się w wyniku porównania z odpowiednim wzorcem.

Relacje słabego pierwszeństwa definiuje się za pomocą relacji prostego pierwszeństwa następująco:

$$X_i > X_j = X_i \gtrdot X_j$$
$$X_i < X_j = (X_i \lessdot X_j) \vee (X_i \doteq X_j)$$

Definicja

Gramatykę G nazywamy gramatyką ze słabym pierwszeństwem jeżeli:

- między każdymi dwoma symbolami gramatyki zachodzi co najwyżej jedna relacja słabego pierwszeństwa;
- wszystkie produkcje gramatyki mają różne prawe strony;
- jeżeli istnieją dwie produkcje postaci
  $A_1 \longrightarrow \alpha X \beta$ $(\alpha \in V^*, \beta \in V^*)$,
  $A_2 \longrightarrow \beta$,
  to między symbolami X i A_2 nie zachodzą relacje słabego pierwszeństwa, tzn. w gramatyce G nie istnieje forma zdaniowa
  $\ldots X A_2 \ldots$

Twierdzenie 1

Gramatyka ze słabym pierwszeństwem jest jednoznaczna. Ponadto rdzeniem dowolnej formy zdaniowej $X_1, \ldots, X_n$ tej gramatyki jest położone najbardziej po lewej stronie tej formy zdaniowej słowo $X_j \ldots X_i$ takie, że

1) $X_{j-1} < X_j < \ldots < X_i > X_{i+1}$,
2) w gramatyce istnieje produkcja postaci: $A \rightarrow X_j \ldots X_i$,
3) jeżeli słowo $X_1 \ldots X_i$ spełnia warunki 1 i 2, to $j < 1$.

Warunek 3 mówi nam o tym, że rdzeń jest najdłuższym podsłowem formy zdaniowej, spełniającym warunki 1 i 2.

Twierdzenie 2

Każda gramatyka z prostym pierwszeństwem jest gramatyką ze słabym pierwszeństwem.

Podamy przykład gramatyki, która jest gramatyką ze słabym pierwszeństwem, lecz nie jest gramatyką z prostym pierwszeństwem.

Przykład 1

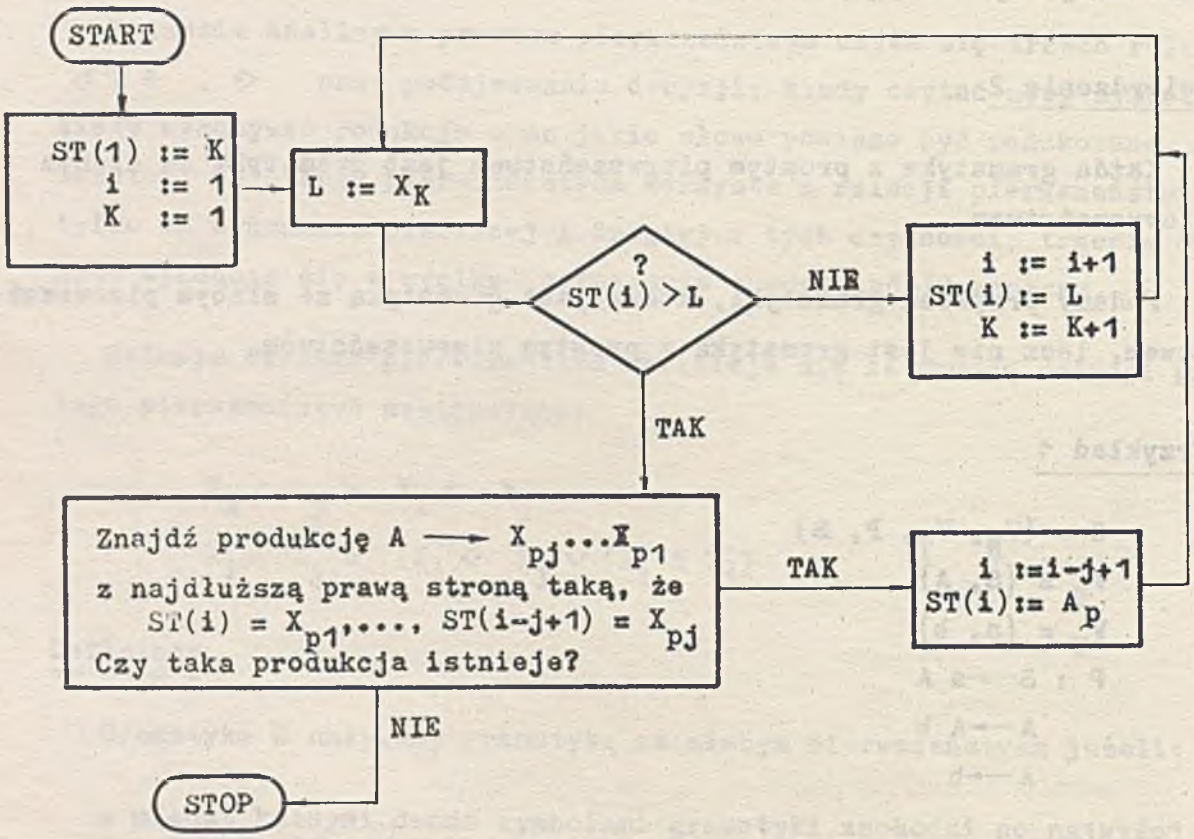
$G = (V_N, V_T, P, S)$
$V_N = \{S, A\}$
$V_T = \{a, b\}$
$P : S \rightarrow a\ A$
$A \rightarrow A\ b$
$A \rightarrow b$

Macierz słabego pierwszeństwa dla tej gramatyki ma postać:

$$B = \begin{array}{c} \\ S \\ A \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cccc} S & A & a & b \end{array} \\ \begin{bmatrix} & & & < \\ & & < & < \\ & & & > \end{bmatrix} \end{array}$$

Wszystkie produkcje tej gramatyki mają różne prawe strony. Dla produkcji: $A \rightarrow A\ b$, $A \rightarrow b$, A nie jest w relacji z A, więc pkt 3 definicji jest również spełniony. Gramatyka G jest zatem gramatyką ze słabym pierwszeństwem. Dla symboli a i A zachodzą dwie relacje pierwszeństwa $a \doteq A$ oraz $a \lessdot A$, a więc G nie jest gramatyką z prostym pierwszeństwem.

Podamy teraz analizator dla języków generowanych przez gramatyki ze słabym pierwszeństwem (oznaczenia są takie same jak w opisie analizatora z pkt 2).



$X_1 \ldots X_n$ jest zdaniem języka
jeśli i = 2, ST(i) = S oraz L = $\perp$

Jeżeli symbole lewostronnie rekursywne pojawiają się w prawych stronach produkcji nie tylko jako symbole najbardziej lewostronne (tak jak np. symbol A w podanym wyżej przykładzie), dochodzi wtedy do kolizji symboli będących w relacjach prostego pierwszeństwa. Między takimi symbolami może istnieć co najwyżej jedna relacja słabego pierwszeństwa. Fakt, że w gramatykach ze słabym pierwszeństwem symbole lewostronnie rekursywne nie powodują kolizji jest istotny. Lewostronna rekursja w produkcjach jest wygodna w czasie analizy redukcyjnej wykonywanej od lewej do prawej, gdyż dla rekursji tego typu rdzeń w formie zdaniowej będzie można znaleźć wcześniej niż dla rekursji prawostronnej.

3.2. Gramatyki z rozszerzonym pierwszeństwem (z pierwszeństwem typu (m, n))

Gdy zawodzą metody analizy z prostym jak i słabym pierwszeństwem wówczas w celu określenia jednoznacznych relacji można przeglądać większy kontekst dookoła dwóch rozpatrywanych symboli.

Niech $G = (V_N, V_T, P, S)$ będzie gramatyką bezkontekstową i symbole $\not\in, \not\$ \notin V$

Dla wszystkich αA, $B\beta$ takich, że $\alpha A \in \{\not\in\}^* V^+$, $B\beta \in V^+\{\not\$\}^*$ rozszerzone relacje pierwszeństwa (relacje pierwszeństwa typu (m, n)) definiujemy następująco:

niech $|\alpha A| = m$, $|B\beta| = n$, $m, n > 0$,

$\tau_2 \in \{\not\in\}^* V^*$, $x_2 \in V_T^* \{\not\$\}^*$, $\tau_1, \tau \in V^*$

a) $\alpha A \doteq B\beta$, jeżeli istnieje wyprowadzenie prawostronne $\not\in^m S \not\$^n \Longrightarrow \tau_2 C x_2$ i produkcja w P postaci $C \longrightarrow \tau_1 A B \tau_1$, gdzie $\tau_2\tau_1 = \ldots \alpha$, $\tau_1 x_2 = \beta \ldots$

b) $\alpha A \lessdot B\beta$, jeżeli istnieje wyprowadzenie prawostronne $\not\in^m S \not\$^n \overset{*}{\Longrightarrow} \ldots \alpha A C x_2$ i produkcja w P postaci: $C \longrightarrow B \tau_1$, $\tau_1 x_2 = \beta \ldots$

c) $\alpha A \gtrdot B\beta$, jeżeli istnieje wyprowadzenie prawostronne $\not\in^m S \not\$^n \overset{*}{\Longrightarrow} \tau_2 C B\beta \ldots$ i produkcja w P postaci $C \longrightarrow \tau_1 A$, gdzie $\tau_2\tau_1 = \ldots \alpha$ i $B\beta \in V_T^+\{\not\$\}^*$

Definicja

Gramatyka G jest gramatyką z rozszerzonym pierwszeństwem (z pierwszeństwem typu (m, n)), jeżeli zachodzą dwa warunki:

- dla wszystkich słów $\alpha A \in \{\not\in\}^* V^+$, $B\beta \in V^+\{\not\$\}^*$ takich, że $|\alpha A| = m$, $|B\beta| = n$ zachodzi co najwyżej jedna relacja rozszerzonego pierwszeństwa;
- wszystkie produkcje mają różne prawe strony.

Można udowodnić, że tak zdefiniowana gramatyka jest gramatyką jednoznaczną, że rdzeń każdej jej formy zdaniowej jest określony jednoznacznie i sformułować pozostałe twierdzenia analogiczne do twierdzeń dotyczących relacji pierwszeństwa niższego typu.

4. Gramatyki z operatorowym pierwszeństwem

W analizie z prostym pierwszeństwem definiowaliśmy relacje między wszystkimi, zarówno terminalnymi jak i nieterminalnymi, symbolami gramatyki. W analizie z operatorowym pierwszeństwem definiujemy relacje tylko między symbolami terminalnymi gramatyki.

4.1. Definicja gramatyki z operatorowym pierwszeństwem

Będziemy rozpatrywali gramatyki, w których w produkcjach po prawych stronach nie występują obok siebie dwa symbole nieterminalne.

Definicja 1

Gramatyka G jest gramatyką operatorową, jeżeli żadna produkcja nie jest postaci $A \longrightarrow \ldots B\ C \ldots$, gdzie B, C są symbolami nieterminalnymi.

Można udowodnić, że dla gramatyk operatorowych w żadnej formie zdaniowej, nie występują obok siebie dwa symbole nieterminalne.

Definicja 2

Niech G będzie gramatyką operatorową oraz a i b będą dwoma symbolami terminalnymi, wtedy

zapis	oznacza, że
1) $a \doteq b$	istnieje produkcja $A \longrightarrow \ldots a\ b \ldots$
	lub $A \longrightarrow \ldots a\ C\ b \ldots$
2) $a \lessdot b$	istnieje produkcja $A \longrightarrow \ldots a\ B \ldots$
	gdzie $B \overset{+}{\Longrightarrow} b \ldots$ lub $B \overset{+}{\Longrightarrow} C\ b \ldots$
3) $a \gtrdot b$	istnieje produkcja $A \longrightarrow \ldots B\ b \ldots$
	gdzie $B \overset{+}{\Longrightarrow} \ldots a$ lub $B \overset{+}{\Longrightarrow} \ldots a\ C$ dla pewnych B, C nieterminalnych

Zdefiniowane wyżej relacje operatorowego pierwszeństwa są podobne do relacji prostego pierwszeństwa. Różnica między nimi jest tylko taka, że relacje operatorowego pierwszeństwa definiuje się tylko dla symboli terminalnych.

Definicja

Gramatyka operatorowa jest gramatyką z operatorowym pierwszeństwem, jeżeli między każdymi dwoma symbolami terminalnymi zachodzi co najwyżej jedna relacja operatorowego pierwszeństwa.

4.2. Konstrukcja relacji

Aby skonstruować relację $\doteq$ wystarczy przejrzeć prawe strony produkcji i wyszukać słowa postaci aCb lub ab. Przy konstrukcji relacji $\lessdot$, dla każdego B trzeba znaleźć zbiór symboli terminalnych a_i taki, że $B \overset{+}{\Longrightarrow} a_i \ldots$ lub $B \overset{+}{\Longrightarrow} Ca_i \ldots$, gdzie C jest symbolem nieterminalnym. Podobnie postępuje się przy konstrukcji relacji $\gtrdot$

Zdefiniujemy dwie nowe relacje.

Definicja

Niech G będzie gramatyką operatorową i niech a będzie symbolem terminalnym, wtedy

$A\ (G_{LT})\ a$ istnieje produkcja $A \longrightarrow a \ldots$ lub $A \longrightarrow Ca \ldots$

$A\ (G_{RT})\ a$ istnieje produkcja $A \longrightarrow \ldots a$ lub $A \longrightarrow \ldots aC$,

gdzie C jest dowolnym symbolem nieterminalnym.

Relacje G_{LT} i G_{RT} można łatwo wyznaczyć. W tym celu wystarczy przejrzeć prawe strony produkcji.

Relacje operatorowego pierwszeństwa można obliczyć podobnie jak w rozdziale 2.1.

Relacja $\lessdot$ jest złożeniem trzech relacji $\doteq$, $I + G_L$ i G_{LT}, czyli $\lessdot = (\doteq)\ (I + G_L)\ (G_{LT})$. Podobnie $\gtrdot = ((G_R + I)\ (G_{RT}))^C\ (\doteq)$.

Relacje $\doteq$, G_L, G_R są relacjami z pkt 2.1, a I jest relacją identyczności.

Konstruując macierze dla relacji $\doteq$, $(I + G_L)$, (G_{LT}), $(G_R + I)$, $(G_{RT})^C$ (patrz 2.1 konstrukcja relacji pierwszeństwa) i mnożąc je odpowiednio przez siebie otrzymamy relację $\lessdot$ oraz $\gtrdot$.

4.3. Fraza pierwotna i analizator

Rozpatrzmy gramatykę $G = (\{S, A, B\}, \{i, +, *, (,)\},$

$\{S \longrightarrow S + A,\quad S \longrightarrow A,\quad A \longrightarrow A * B,\quad A \longrightarrow B,\quad B \longrightarrow (S),\quad B \longrightarrow i\}, S)$

Macierz operatorowego pierwszeństwa dla tej gramatyki ma postać:

$$B = \begin{array}{c} \\ + \\ * \\ (\\ ) \\ i \end{array} \begin{array}{c} \begin{array}{ccccc} + & * & (&) & i \end{array} \\ \begin{bmatrix} \gtrdot & \lessdot & \lessdot & \gtrdot & \lessdot \\ \gtrdot & \gtrdot & \lessdot & \gtrdot & \lessdot \\ \lessdot & \lessdot & \lessdot & \doteq & \lessdot \\ \gtrdot & \gtrdot & & \gtrdot & \\ \gtrdot & \gtrdot & & \gtrdot & \end{bmatrix} \end{array}$$

Weźmy formę zdaniową S + B, oraz jej wyprowadzenie:

$$S \Longrightarrow S + A \Longrightarrow S + B$$

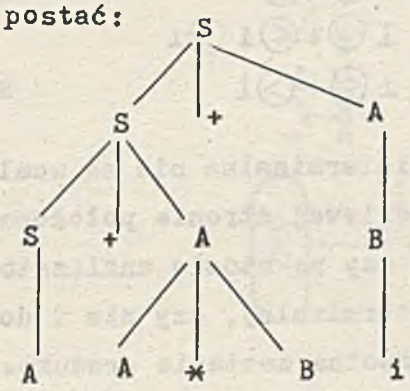
Rdzeniem tej formy jest B. Przyjmując, tak jak poprzednio, że forma zdaniowa jest ograniczona symbolem $\perp$ i że dla wszystkich symboli terminalnych zachodzi: $\perp \lessdot a$ oraz $a \gtrdot \perp$, mamy, że $\perp \lessdot +$ oraz $+ \gtrdot \perp$, a zatem relacji operatorowego pierwszeństwa nie można wykorzystać do znajdowania rdzenia składającego się z pojedynczego symbolu nieterminalnego. Tak więc posługując się relacjami operatorowego pierwszeństwa nie można skonstruować takiego jak uprzednio analizatora. Analizator będzie w każdym kroku rozpoznawał i redukował nie rdzeń, lecz tzw. frazę pierwotną.

Definicja

Frazą pierwotną formy zdaniowej nazywa się fraza zawierająca co najmniej jeden symbol terminalny, lecz nie zawierająca żadnej innej frazy pierwotnej.

Przykład

Rozpatrzmy formę zdaniową A + A * B + i gramatyki G. Drzewo wywodu tej formy zdaniowej ma postać:



Frazami są słowa: A + A * B + i, A + A * B, i, A oraz A * B. Z tych słów frazami pierwotnymi nie mogą być frazy A + A * B + i, A + A * B (zawierają one inne frazy pierwotne) ani też A (nie zawiera symbolu terminalnego). Zatem frazami pierwotnymi są słowa A * B oraz i.

Twierdzenie

Fraza pierwotna położona najbardziej na lewo w formie zdaniowej gramatyki z operatorowym pierwszeństwem jest pierwszym z lewej podsłowem

$$A_j\ a_j \ldots A_i\ a_i\ A_{i+1}$$

tej formy takim, że

$$a_{j-1} \lessdot a_j,\ a_j \doteq a_{j+1}, \ldots, a_{i-1} \doteq a_i,\ a_i \gtrdot a_{i+1},$$

gdzie a_k - symbole terminalne
A_k - symbole nieterminalne

Zauważmy, że symbol nieterminalny pojawiający się na lewo od a_j lub na prawo od a_i zawsze należy do frazy pierwotnej.

Przykład

Posługując się powyższym twierdzeniem przeprowadzimy analizę formy zdaniowej A + A * B + i, korzystając z macierzy pierwszeństwa operatorowego:

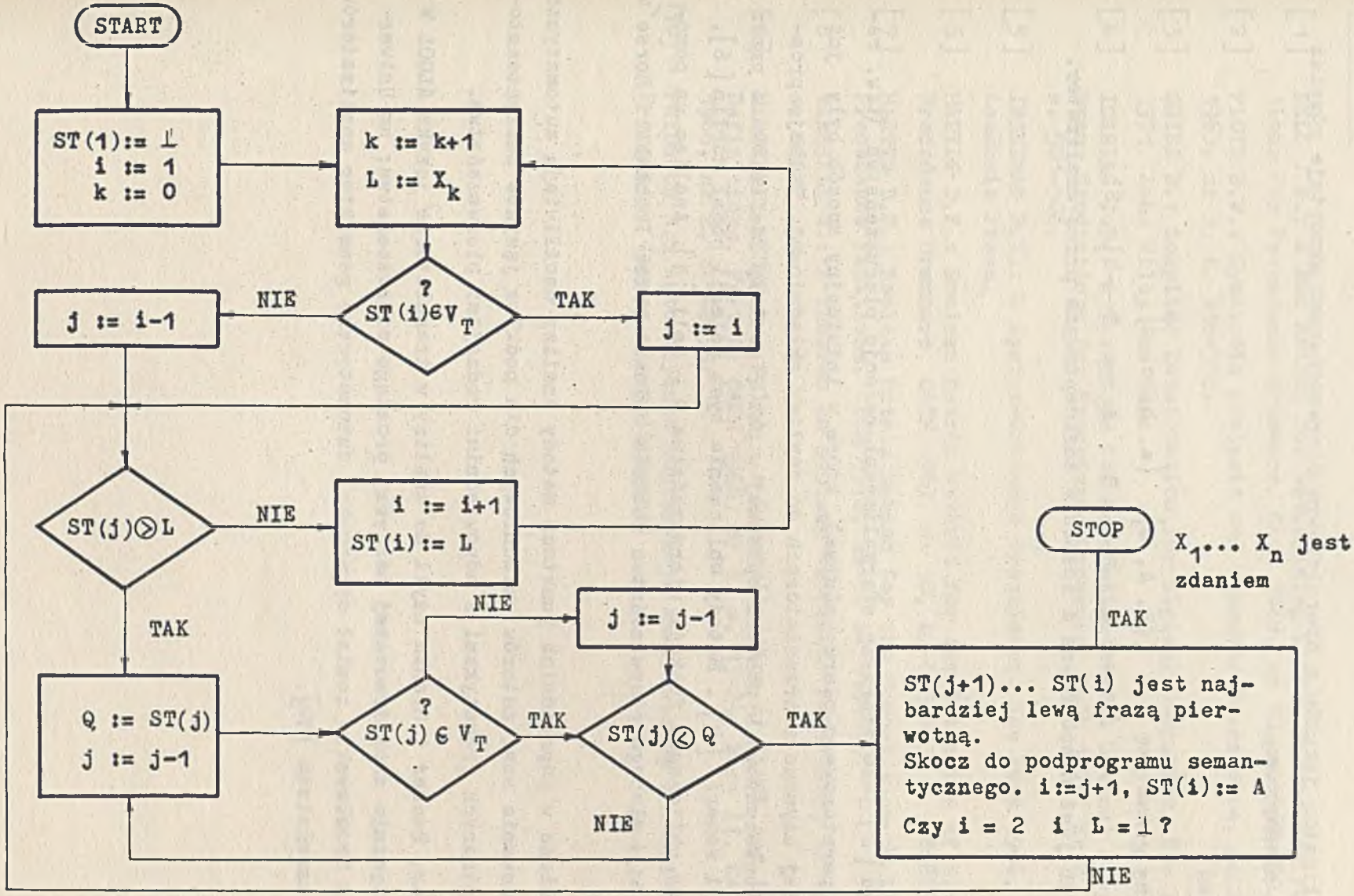
krok	forma zdaniowa	relacje	fraza pierwotna	redukuj do
1	⊥ A + A * B + i ⊥	⊥ ⋖ + ⋖ * ⋗ +	A * B	A
2	⊥ A + A + i ⊥	⊥ ⋖ + ⋗ +	A + A	S
3	⊥ S + i ⊥	⊥ ⋖ + ⋖ i ⋗ ⊥	i	B
4	⊥ S + B ⊥	⊥ ⋖ + ⋗ ⊥	S + B	S

Zauważmy, że symbole nieterminalne nie są wcale sprawdzane przy znajdowaniu najbardziej po lewej stronie położonej frazy pierwotnej. Dlatego też jest obojętne czy na stosie analizatora zostanie umieszczony poprawny symbol nieterminalny, czy nie i do jakiego symbolu nieterminalnego ta fraza pierwotna zostanie zredukowana.

W związku z tym analizator korzystający z pierwszeństwa operatorowego trzeba uzupełnić dodatkowymi podprogramami, tzw. akcjami semantycznymi, których zadaniem jest kontrola symboli nieterminalnych. Nie było to potrzebne w metodzie z relacjami pierwszeństwa. Zyskuje się tu jednak na wydajności analizatora (mniejsza jest macierz pierwszeństwa, zawiera ona bowiem tylko relacje dla symboli terminalnych, a analizator nie redukuje fraz postaci $A \longrightarrow B$, ponieważ symbol nieterminalny nie może być frazą pierwotną).

Zanalizujemy zdanie i * i + i generowane przez gramatykę G z ostatniego przykładu. (Sieć działań analizatora podajemy na stronie następnej).

krok	ST(1) ...	relacja	L	X_k ...
0	⊥	⋖	i	* (i + i) ⊥
1	⊥ i	⋗	*	(i + i) ⊥
2	⊥ A	⋖	*	(i + i) ⊥
3	⊥ A *	⋖	(	i + i) ⊥
4	⊥ A * (	⋖	i	+ i) ⊥
5	⊥ A * (i	⋗	+	i) ⊥
6	⊥ A * (A	⋖	+	i) ⊥
7	⊥ A * (A +	⋖	i	) ⊥
8	⊥ A * (A + i	⋗	)	⊥
9	⊥ A * (A + A	⋗	)	⊥
10	⊥ A * (A	=	)	⊥
11	⊥ A * (A)	⋗	⊥	
12	⊥ A * A	⋗	⊥	
13	⊥ A	STOP		



Algorytm analizatora opartego na relacjach operatorowego pierwszeństwa

Uwaga

Analizator języków z operatorowym pierwszeństwem akceptuje również zdania niepoprawne.

Przykład

Weźmy gramatykę $G = (\{S, A, B\}, \{a, b, c, d\}, \{S \to c\,A,\ S \to d\,B,\ A \to a,\ B \to b\}, S)$

Słowo cb jest niepoprawne a zostanie zaakceptowane przez analizator.

Zakończenie

Floyd [2] jako pierwszy sformalizował relacje pierwszeństwa (tzw. relacje operatorowego pierwszeństwa), które w intuicyjny sposób były już wcześniej używane w translatorach do analizy składniowej. Relacje prostego pierwszeństwa w postaci opisanej w pkt 2 zostały zdefiniowane przez Wirtha i Webera [9]. Metody obliczania tych relacji podał Martin [6], a metody obliczania funkcji pierwszeństwa 9 - Bell [1]. Analiza za pomocą relacji słabego pierwszeństwa została opisana przez Ichbiaha i Morse'a [4].

Omówione w poprzednich punktach metody analizy umożliwiają automatyczne konstruowanie analizatorów składniowych dla podklas języków bezkontekstowych opisanych gramatykami z odpowiednimi rodzajami pierwszeństwa.

Bauer, Becker i Graham użyli do analizy w translatorze języka ALGOL W automatycznie skonstruowanej macierzy prostego pierwszeństwa; na Uniwersytecie Wrocławskim został opracowany automatyczny generator analizatorów z pierwszeństwem [10].

Literatura

[1] BELL J.R.: A New Method for Determining Linear Precedence Functions for Precedence Grammars. CACM 1969, nr 10, s. 567-569.

[2] FLOYD R.W.: Syntactic Analysis and Operator Precedence. JACM 1963, nr 3, s. 316-333.

[3] GRIES D.: Compiler Construction for Digital Computers. New York 1971. John Wiley and Sons.

[4] ICHBIAH J.D.: A Technique for Generating Almost Optimal Floyd - Evans Productions for Precedence Grammars. CACM 1970, nr 8, s. 501-508.

[5] INGERMAN P.Z.: A Syntax-Oriented Translator. New York 1966, Academic Press.

[6] MARTIN D.F.: Boolean Matrix Methods for the Detection of Simple Precedence Grammars. CACM 1968, nr 10, s. 685-687.

[7] MARTIN D.F.: Boolean Matrix Method for the Computation of Linear Precedence Functions. CACM 1972, nr 6, s. 443-454.

[8] WARSHALL S.: A Theorem on Boolean Matrices. JACM 1962, nr 1, s. 11-12.

[9] WIRTH M., WEBER H.: Euler - a Generalization of ALGOL and its Definitions. Part I. CACM 1966, nr 1, s. 13-24; Part II. CACM 1966, nr 2, s. 89-99.

[10] FRYDRYCH B.: Automatyczna konstrukcja analizatorów składni dla P-gramatyk (praca niepublikowana)

WYDAWNICTWA PRZEMYSŁU MASZYNOWEGO "WEMA"
oferują usługi wydawnicze

W Warszawie działa specjalne wydawnictwo resortowe powołane do świadczenia usług wydawniczych na rzecz jednostek organizacyjnych resortu przemysłu maszynowego.

Do szczególnych zadań Wydawnictw Przemysłu Maszynowego "WEMA" należy:
- prowadzenie działalności wydawniczej zgodnie z potrzebami resortu,
- koordynacja działalności wydawniczej w jednostkach organizacyjnych resortu,
- koordynacja i nadzór nad prawidłowym wykorzystaniem maszyn i urządzeń poligraficznych,
- prowadzenie własnego ośrodka poligraficznego,
- prowadzenie ośrodka informacji wydawniczej.

Od ubiegłego roku Wydawnictwo znacznie rozszerzyło zakres usług i obecnie wydaje:
- katalogi branżowe i karty katalogowe

oraz na zlecenie przedsiębiorstw przemysłowych różnego rodzaju literaturę firmową, jak:
- katalogi zakładowe,
- katalogi części wymiennych,
- informatory techniczno-handlowe,
- dokumentacje techniczno-ruchowe, instrukcje obsługi i instrukcje naprawcze,
- dokumentacje techniczne kapitalnych remontów,
- wydawnictwo reklamowe, jak prospekty, foldery, ulotki itp.

Katalogi branżowe wydaje się w porozumieniu i we współpracy z właściwymi gestyjnie zjednoczeniami.

Sprzedażą katalogów WPM "WEMA" zajmują się następujące księgarnie:
Księgarnie "WSPÓLNEJ SPRAWY":
Warszawa, ul. Marszałkowska 28, tel. 21-66-60
Warszawa, ul. Marchlewskiego 35, tel. 20-49-69

"DOM KSIĄŻKI":
Główna Księgarnia Techniczna, Warszawa, ul. Świętokrzyska 14, tel. 26-63-38.

Księgarnie te prowadzą sprzedaż odręczną i wysyłkową.

Literaturę firmową WPM "WEMA" wykonują na konkretne zamówienie przedsiębiorstw przemysłowych.

WPM "WEMA" znacznie skróciły cykle wydawnicze i zapewniają obecnie terminową realizację zamówień.

Wszelkich informacji na temat warunków przyjmowania i realizacji zamówień wydawniczych udziela Sekretariat Wydawnictwa, Warszawa, ul. Daniłowiczowska 18, pokój nr 7, tel. 27-49-47, skr. poczt. 90.