

IITiS

AMOK

ANALIZATOR
MODELI KOLEJKOWYCH

Wersja 2.xx

POLSKA AKADEMIA NAUK
© INSTYTUT INFORMATYKI
TEORETYCZNEJ I STOSOWANEJ
ul. Bałycka 5, 44-100 GLIWICE

dr hab. inż. Tadeusz Czachórski
mgr inż. Marek Kowalczyk
mgr inż. Zdzisław Szczerbiński
mgr inż. Joanna Tomasiak
mgr inż. Andrzej Wilk
mgr inż. Waldemar Wołowiec

Edycja marzec 1989

Oznaczenie 2.xx dotyczy handlowej wersji opracowanego w IITiS-PAN pakietu programowego o nazwie InterAMOK.

Oznaczenie 1.xx dotyczy poprzedniej, wsadowej wersji tego pakietu.

Oznaczenie 3.xx przewidziano dla opracowywanego w latach 1989-90 pakietu programowego SynchroAMOK przeznaczonego do analizy sieci kolejek zawierającej elementy synchronizacji i obejmującego również moduł symulacji.

Spis treści

1 Wstęp	15
2 Podstawowe pojęcia modeli kolejkowych	21
3 Struktura pakietu AMOK	27
4 Język opisu sieci kolejek	33
4.1 Podstawowe zasady	35
4.2 Obiekty	36
4.3 Parametry obiektów	37
4.4 Lista rodzajów klientów	40
4.5 Definicje parametrów obiektów	40
4.5.1 Definicja parametru NAME	40
4.5.2 Definicja parametru COPY	41

4.5.3	Definicja AS (nazwa obiektu)	42
4.5.4	Definicja parametru SCHEDULE	42
4.5.5	Definicja parametru SERVICE	43
4.5.6	Definicja parametru TRANSITION	46
4.6	Parametry liczbowe	47
4.6.1	Liczby całkowite	47
4.6.2	Liczby rzeczywiste	48
4.7	Przykładowe sieci kolejek i ich zapisy w języku AMOKu	49
5	Program konwersji CONV	55
5.1	Komunikaty analizy syntaktycznej	57
5.2	Komunikaty o błędach - ERROR	58
5.3	Ostrzeżenia — WARNING	63
5.4	Komunikaty uzupełniające — MESSAGE	63
5.5	Komunikaty FATAL o wadliwym działaniu programu CONV . . .	66
5.6	Inne komunikaty	66
6	Sposób korzystania z pakietu	69
6.1	Menu główne	71

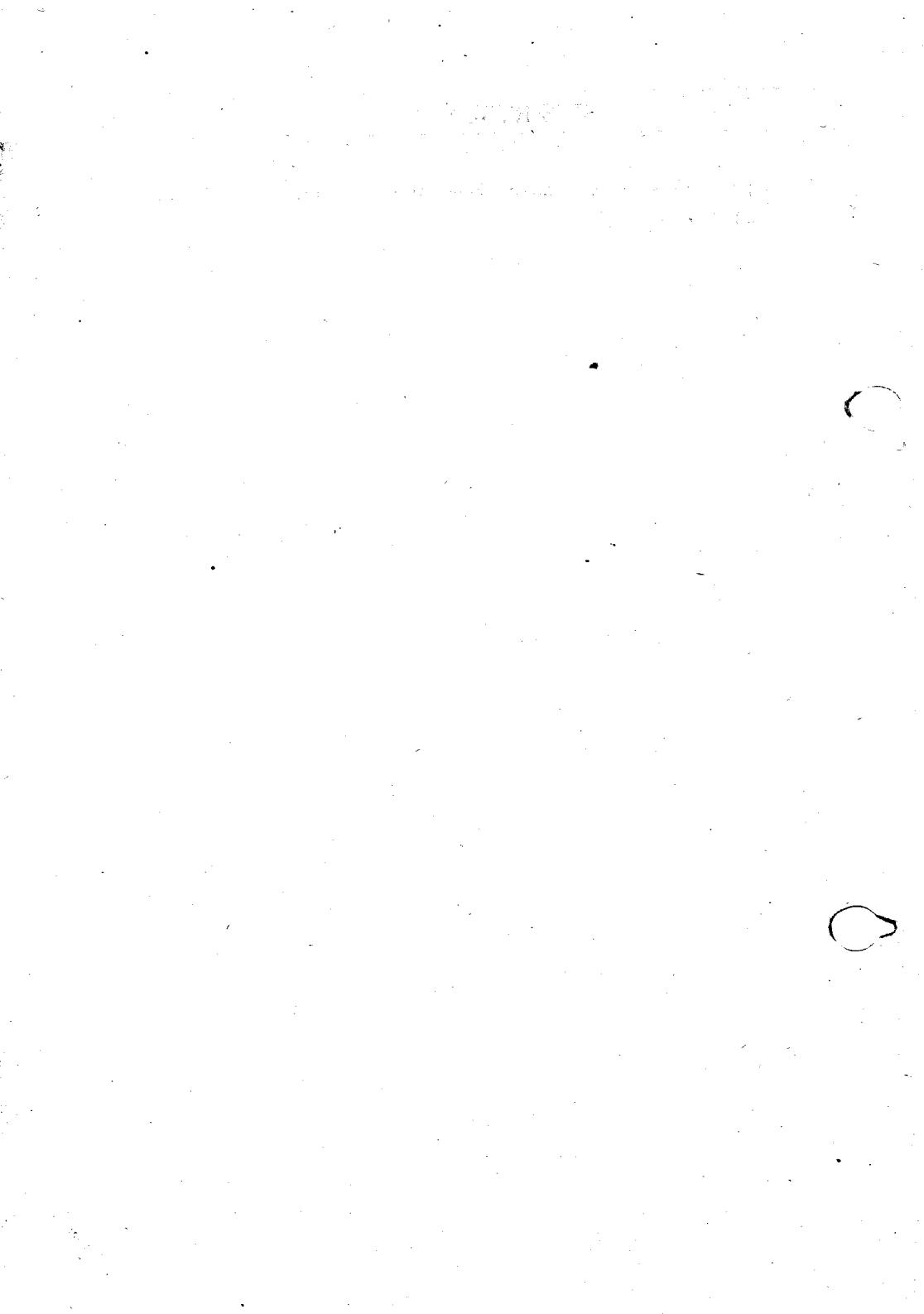
6.2	Menu DATA BASE	77
6.3	Definiowanie sieci kolejek	78
6.3.1	Menu definiowania obiektów kolejkowych	79
6.3.2	Definiowanie klas klientów	83
6.3.3	Definiowanie parametrów	85
6.3.4	Wykonywanie obliczeń	86
6.4	Prezentacja wyników	88
7	Przykłady wykorzystania pakietu	103
7.1	Model systemu transakcyjnego	105
7.2	Czas czekania na remont maszyn	113
7.3	Liczba łóżek w gabinecie zabiegowym pogotowia ratunkowego	116
7.4	Model systemu komputerowego zarządzającego bazą danych	118
7.5	Analiza przepustowości przystanku przesiadkowego podmiejskiej linii kolejowej	124
8	Jądro pakietu AMOK	137
8.1	Przegląd metod analizy	139
8.2	Literatura	143

9 Program JACKSON	145
9.1 Opis metody	147
9.2 Uwagi o programie	148
9.3 Literatura	149
10 Program DIF	151
10.1 Opis metody	153
10.1.1 Aproksymacja dyfuzyjna pojedynczego stanowiska obsługi	154
10.1.2 Aproksymacja sieci stanowisk	157
10.2 Uwagi o programie	158
10.3 Literatura	159
11 Program MVA	161
11.1 Opis metody	163
11.2 Uwagi o programie	165
11.3 Literatura	166
12 Program HMVA	167
12.1 Opis metody	169

SPIS TREŚCI

5

12.2 Uwagi o programie	170
12.3 Literatura	170
13 Program EPF	171
13.1 Opis metody	173
13.2 Uwagi o programie	176
13.3 Literatura	176
14 Program ITERM	177
14.1 Opis metody	179
14.2 Uwagi o programie	183
14.3 Literatura	183
15 Instalacja pakietu AMOK	185



Spis rysunków

2.1	Ogólna postać sieci stanowisk obsługi	24
3.1	Struktura pakietu AMOK	30
4.1	Rozkład Coxa	45
4.2	Rozkład hiperwykładniczy	45
4.3	Przykładowa otwarta sieć stanowisk obsługi	49
4.4	Zapis otwartej sieci stanowisk obsługi	51
4.5	Przykładowa zamknięta sieć stanowisk obsługi	53
4.6	Zapis zamkniętej sieci stanowisk obsługi	54
6.1	Kopia ekranu z widocznym na nim głównym menu	71
6.2	Struktura menu głównego oraz powiązanych z nim menu zmiany pliku z bazą i menu definiowania metody	72
6.3	Kopia ekranu z menu definiowania metody	73

6.4 Kopia ekranu podczas wyświetlania pliku z opisem modelu	74
6.5 Kopia ekranu podczas zmiany pliku z bazą danych	75
6.6 Kopia ekranu podczas wyświetlania ostatniego komunikatu	75
6.7 Kopia ekranu zmiany bazy danych	78
6.8 Struktura menu definiowania sieci, klas i obiektów	80
6.9 Struktura menu definiowania parametrów	81
6.10 Kopia ekranu z menu definiowania obiektów kolejkowych	81
6.11 Kopia ekranu podczas definiowania typu i nazwy obiektu	82
6.12 Kopia ekranu podczas definiowania typu i nazwy klasy	84
6.13 Kopia ekranu z menu definiowania parametrów	85
6.14 Wielowymiarowa tablica wyników	88
6.15 Zbiór wyników dla jednego obiektu i jednej klasy klientów	89
6.16 Kopia ekranu z podstawowymi wynikami obliczeń	90
6.17 Menu prezentacji wyników oraz powiązanych z nim menu wyboru	91
6.18 Przykład ekranu z rozwiniętą listą obiektów	93
6.19 Przykład ekranu z rozwiniętą listą klas	93
6.20 Kopia ekranu z wyborem elementu obiektu	94

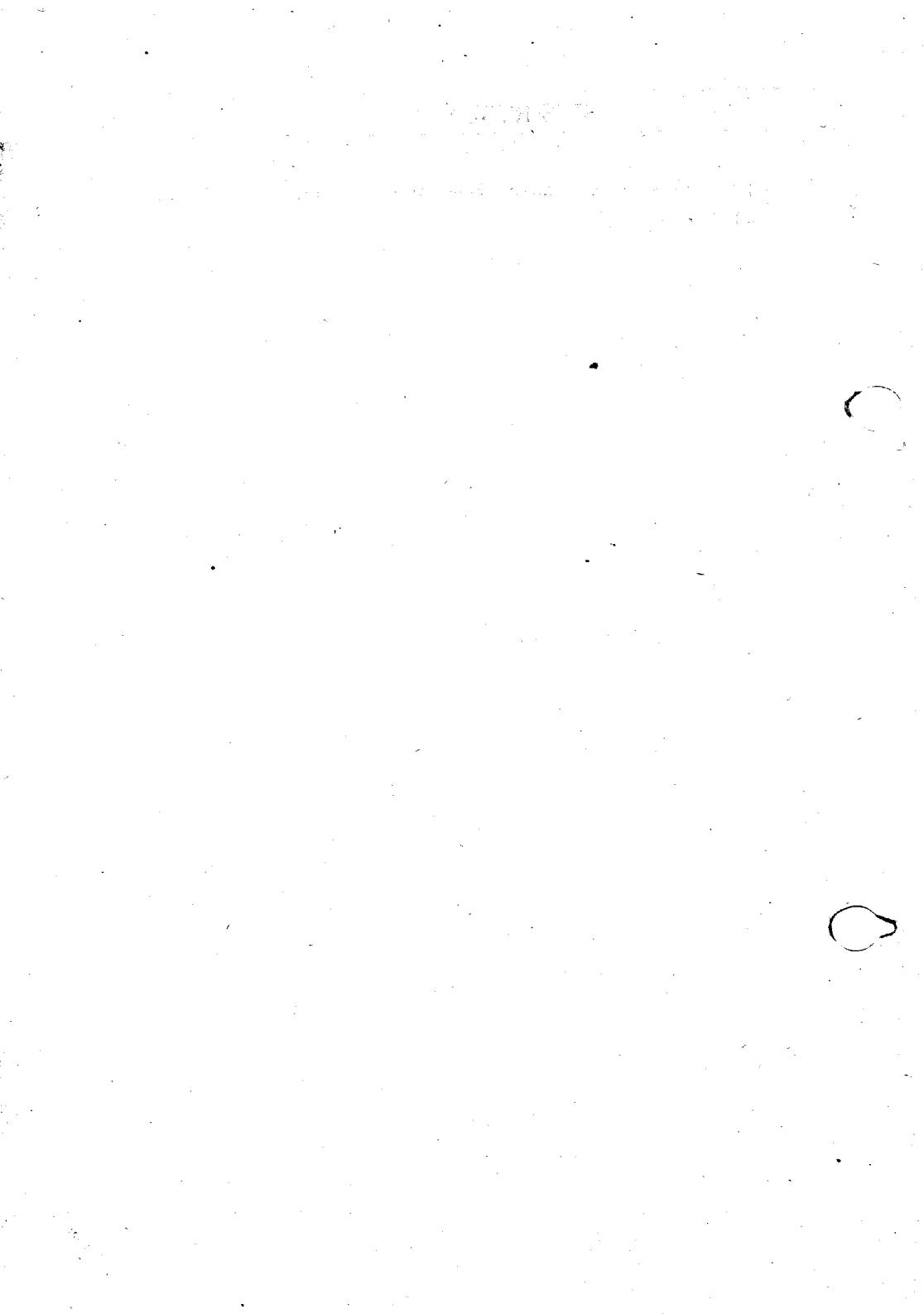
6.21 Kopia ekranu z wyborem parametru obiektu	94
6.22 Kopia ekranu z wynikami obliczeń czasów przejścia	95
6.23 Kopia ekranu z wynikami obliczeń śr. liczby klientów w kolejce	95
6.24 Kopia ekranu z wynikami obliczeń wykorzystania	96
6.25 Kopia ekranu z wynikami obliczeń przepustowości	96
6.26 Kopia ekranu z wyborem formy prezentacji	97
6.27 Kopia ekranu z wynikami obliczeń w formie tabeli liczb	98
6.28 Kopia ekranu z wynikami obliczeń w formie histogramu	98
6.29 Kopia ekranu z wynikami obliczeń w formie wykresu	99
6.30 Kopia ekranu z wynikami obliczeń w formie krzywej Kiviata	99
6.31 Przykład wydruku ekranu z wynikami obliczeń	101
6.32 Przykład wydruku wszystkich wyników obliczeń	102
7.1 Analizowany model systemu transakcyjnego	106
7.2 Opis modelu transakcyjnego	108
7.3 Wyniki obliczeń modelu systemu transakcyjnego	110
7.4 Wyniki obliczeń modelu systemu transakcyjnego bez transakcji administratora	111

7.5 Wyniki obliczeń modelu systemu transakcyjnego z ograniczoną liczbą transakcji	112
7.6 Opis modelu zakładu remontu maszyn	114
7.7 Wyniki obliczeń modelu zakładu remontowego	115
7.8 Opis modelu gabinetu zabiegowego	117
7.9 Wyniki obliczeń modelu gabinetu zabiegowego	117
7.10 Model systemu komputerowego	119
7.11 Zapis modelu systemu komputerowego w języku AMOKu	121
7.12 Zapis modelu systemu komputerowego w języku AMOKu (dokonczenie)	122
7.13 Średnie czasy reakcji systemu i poszczególnych stanowisk dla transakcji zdalnych w funkcji intensywności nadjęć tych transakcji	123
7.14 Ruch pasażerów na przystanku przesiadkowym	124
7.15 Model kolejowy ruchu pasażerów na przystanku przesiadkowym	128
7.16 Tandem dwóch stanowisk o średnim czasie obsługi t_x i przepustowości μ_x	129
7.17 Opis modelu przystanku kolejowego w języku AMOKu	131
7.18 Opis modelu przystanku kolejowego w języku AMOKu (kontynuacja)	132

SPIS RYSUNKÓW

11

- 7.19 Opis modelu przystanku kolejowego w języku AMOKu
(dokończenie) 133



Spis tabel

2.1 Typy sieci analizowane przez AMOK	26
2.2 Maksymalne rozmiary sieci analizowanych przez wersję 2.10 AMOKu	26
4.1 Parametry sieci	50
5.1 Komunikaty o błędach	60
5.2 Komunikaty o błędach (kontynuacja)	61
5.3 Komunikaty o błędach (dokończenie)	62
5.4 Ostrzeżenia	63
5.5 Komunikaty uzupełniające	64
5.6 Komunikaty uzupełniające (dokończenie)	65
7.1 Zależność czasu czekania w kolejce do automatu i liczby oczekujących pasażerów od liczby automatów	134

7.2 Zależność czasu oczekiwania na autobus i liczby oczekujących pasażerów od częstotliwości kursowania autobusów	135
7.3 Średnie czasy oczekiwania, średnie liczby pasażerów, stopnie zajętości (autobus co 2 minuty, 3 automaty)	135

Rozdział 1

Wstęp



AMOK jest pakietem programowym, umożliwiającym praktyczne stosowanie modeli teorii masowej obsługi, a w szczególności modelowanie systemów komputerowych. Został on stworzony do opisu i oceny efektywności takich systemów, lecz może znaleźć zastosowanie wszędzie tam, gdzie stosuje się teorię masowej obsługi: model w postaci sieci stanowisk obsługi, między którymi krążą klienci, ustawiani w razie potrzeby w kolejki, może odnosić się do wielu sytuacji i obiektów.

W modelach systemów komputerowych stanowiskami obsługi są elementy tych systemów: procesory, dyski pamięci, linie transmisyjne, terminale, a klientami są wykonywane programy; w modelach sieci telekomunikacyjnych stanowiskami obsługi są linie transmisyjne pomiędzy połączonymi w sieć komputerami, klientami są komutowane (przesypane w tej sieci) pakiety informacji, ustawiane w kolejki w komputerach — węzłach sieci.

W modelach systemów transportowych stanowiskami obsługi są np. pociągi, autobusy, ciężarówki, kasy biletowe, perony, przejścia między peronami, a klientami są pasażerowie i przewożone towary.

Modelując pracę portu towarowego przyjmiemy, że stanowiskami obsługi są nabrzeża portowe, klientami są statki, a czas obsługi to czas załadowania lub wyładowania towarów.

Modelując ruch ludzi w domu towarowym przyjmiemy, że stanowiska obsługi to wejścia, schody, punkty sprzedaży, kasy.

Modelując funkcjonowanie zbiornika wodnego przyjmiemy, że strumienie nadchodzących klientów to zmierzony w jednostkach objętości przepływ wody w poszczególnych rzekach i strumieniach wpadających do tego zbiornika, a przepustowość stanowiska to odpływ wody ze zbiornika; obliczona liczba klientów w stanowisku obsługi odpowiada objętości wody zawartej w zbiorniku.

Analizując pracę przedsiębiorstwa taksówkowego i obliczając średni czas czekania na taksówkę przyjmiemy, że jest to wielokanałowe stanowisko obsługi o liczbie równoległych kanałów odpowiadającej liczbie samochodów; czas obsługi to czas jednego kursu.

Opisując pracę szpitala w kategoriach masowej obsługi przyjmiemy, że szpital jest siecią stanowisk (stanowiska odpowiadają salom szpitalnym) o tymu kanałach obsługi, ile w danej sali jest łóżek, czas pobytu pacjenta w sali jest czasem obsługi.

W modelu pracy skrzyżowania ulicznego czasem obsługi jest czas przejazdu przez skrzyżowanie, liczba kanałów obsługi odpowiada liczbie pasów ruchu.

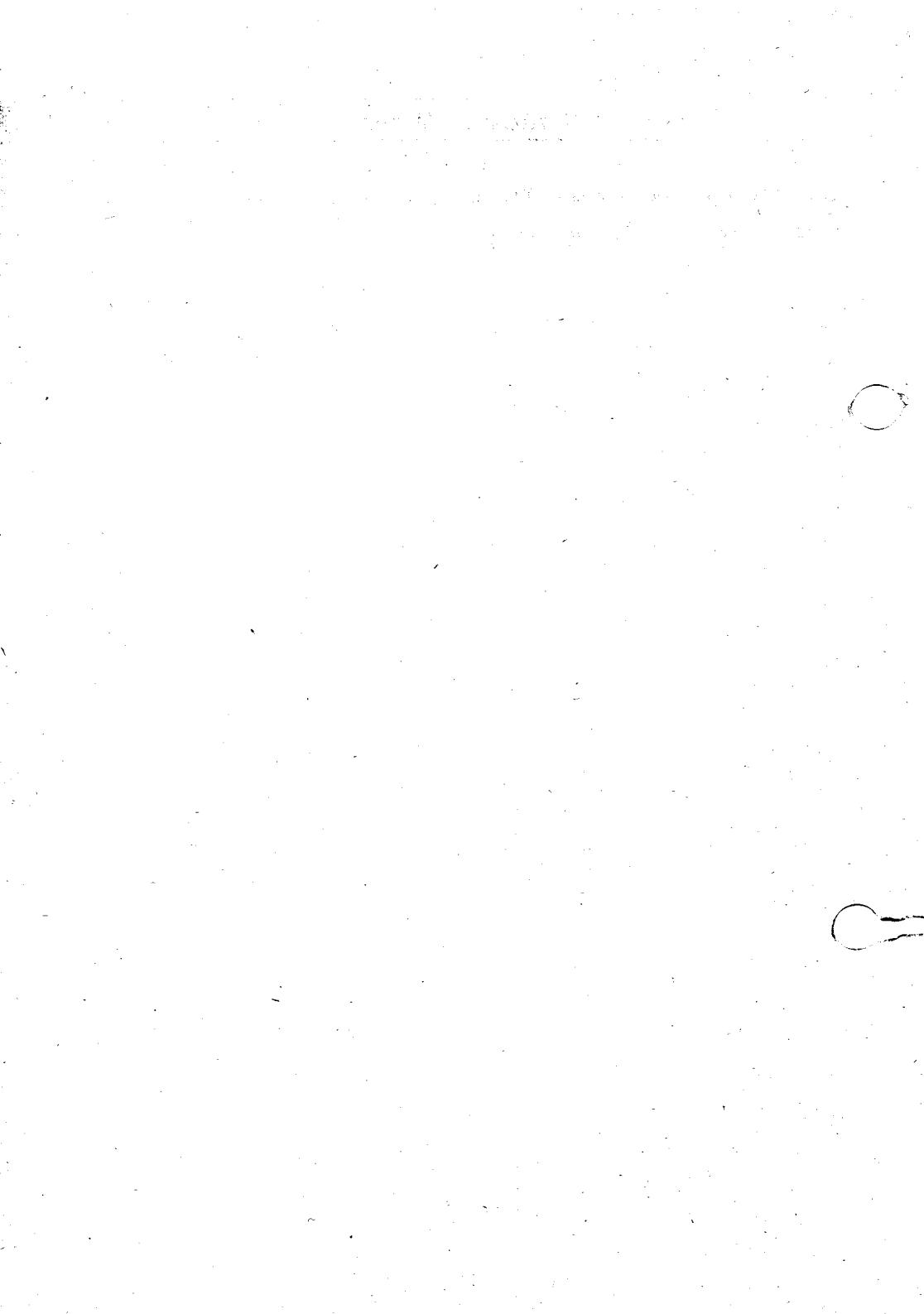
Przykładów takich można podać więcej, jako że badania operacyjne i teoria masowej obsługi są już dobrze zdefiniowaną i stosowaną od kilkudziesięciu lat dziedziną wiedzy. Brak odpowiednich możliwości obliczeniowych stanowił często przeszkodę w jej stosowaniu. AMOK — jak wiele innych istniejących w świecie systemów, np. opracowany w IBM pakiet RESQ czy francuski QNAP, jest programowym produktem ułatwiającym wykorzystanie złożonych nierzaz obliczeniowo modeli.

AMOK przeznaczony jest dla mikrokomputerów IBM PC/XT(AT) i wszystkich mikrokomputerów kompatybilnych z IBM PC na poziomie systemu operacyjnego MS-DOS. Wymaga on 640 KB pamięci RAM i 3 MB pamięci na twardym dysku. Istnieje wersja dla systemów z podłączonym koprocesorem numerycznym 8087. Programy pakietu napisane zostały w językach Fortran i Pascal.

Obsługa pakietu jest prosta i nie powinna stwarzać żadnych trudności. Diagnostyka błędów, dokonywana wewnętrz programów, ułatwia znacznie pracę użytkownika.

Przewidywany jest stały rozwój AMOKu. Będzie on uwzględniał, między innymi, analityczne modele opisujące synchronizację zadań w sieci stanowisk obsługi, symulację (wraz z oceną wiarygodności rezultatów symulacji) problemów, dla

których brak jest rozwiązania analitycznego, rozwój grafiki systemu poprzez stworzenie graficznego języka opisu modeli.



Rozdział 2

Podstawowe pojęcia modeli kolejkowych

Przyjęte w badaniach operacyjnych, opisujące różne obiekty, modele masowej obsługi (teorii kolejek), mają następującą postać:

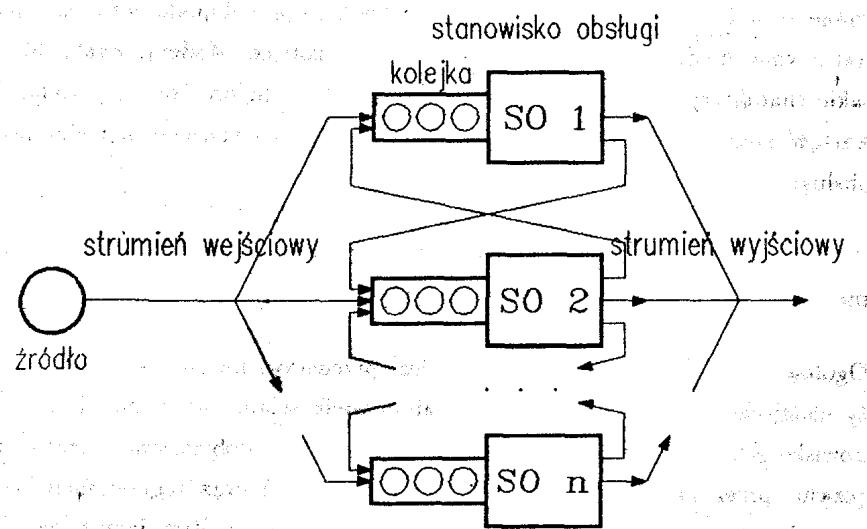
Klienci nadchodzą do stanowiska z określona częstotliwością. Jeśli stanowisko jest wolne, następuje obsługa klienta, który przebywa w stanowisku przez czas zwany czasem obsługi, po czym klient opuszcza stanowisko. Jeżeli stanowisko jest zajęte, klienci ustawiają się przed nim czekając na swoją kolej. Odstępy czasu między nadaniem klientów i czas obsługi klientów przyjmują się w modelu jako wielkości znane — są to zmienne losowe, opisane rozkładami prawdopodobieństwa. Znany jest również mechanizm szeregowania klientów w kolejce. Model pozwala obliczyć takie charakterystyczne dla systemu wielkości, jak długość kolejki (rozkład lub wartość średnia), czas czekania na usługę, stopień wykorzystania stanowiska obsługi.

Stanowiska mogą być połączone w sieć, a droga klienta w sieci jest określona przez macierz prawdopodobieństw przejść pomiędzy stanowiskami.

Ogólna postać sieci stanowisk obsługi jest przedstawiona na rys. 2.1. Momen-ty nadania klientów i czas ich obsługi są wielkościami losowymi. Każde sta-nowisko scharakteryzowane jest rozkładem prawdopodobieństwa czasu obsługi (czasu, przez który klient przebywa w stanowisku) oraz regulaminem szere-gowania klientów w kolejce (np. według kolejności napływających zgłoszeń, w kolejności odwrotnej, wszyscy klienci obsługiwani są jednocześnie itp.).

Opis sieci uzupełnia rozkład odstępów czasu między klientami w strumieniu wejściowym oraz prawdopodobieństwa przejść klientów między stanowiskami.

Sieć może być otwarta (istnieje strumień wejściowy i wyjściowy klientów z sieci) lub zamknięta (w sieci krążą ciągle ci sami klienci). Klienci mogą dzielić się na klasy, różniące się czasem obsługi i drogą w sieci.



Rys. 2.1: Ogólna postać sieci stanowisk obsługi

Należy obliczyć "miary pracy" stanowisk sieci:

No of clients

- rozkład (lub wartość średnia) liczby klientów obecnych w poszczególnych stanowiskach,

Passage time

- średni czas reakcji (średni czas czekania w kolejce powiększony o średni czas obsługi) poszczególnych stanowisk.

Throughput

- przepustowość (średnią liczbę klientów przechodzących w jednostce czasu przez stanowisko),

Utilisation

- współczynnik wykorzystania stanowiska (stosunek czasu, w którym stanowisko było zajęte do czasu obserwacji).

Ponadto, dla sieci otwartej oblicza się czas reakcji sieci (czas pobytu klienta w sieci — od momentu wyjścia ze źródła i wejścia do sieci, do momentu wyjścia z sieci), zaś dla sieci zamkniętej średni czas obiegu klienta, pomiędzy kolejnymi wizytami w wybranym stanowisku.

Nie ma obecnie jednej ogólnej metody rozwiązania sieci. Przyjęty sposób postępowania zależy od szczegółowych założeń, dotyczących typu rozkładów czasu obsługi, regulaminów kolejek, a także od rozmiaru sieci — liczby stanowisk, liczby klientów i ich klas. Wynika stąd potrzeba wielu algorytmów obliczeniowych — wielu modułów pakietu AMOK. Typy sieci analizowane przez AMOK podaje skrótnie tablica 2.1.

Maksymalne rozmiary sieci analizowanych przez wersję 2.10 pakietu AMOK podaje tablica 2.2.

26 Rozdział 2. Podstawowe pojęcia modeli kolejkowych

Moduł	typ sieci		liczba klas		typ stanowisk	
	otwarta	zamknięta	jedna	wiele	BCMP	G/G/1
JACKSON	+			+	⊕	
MVA		+		+	⊕	⊗
HMVA		+		+	⊗	⊗
EPF		+	+		⊕	⊗
ITERM		+	+		⊕	⊗
DIF	+			+	⊗	⊗

⊕ — obliczenia dają wyniki dokładne

⊗ — obliczenia dają wyniki przybliżone

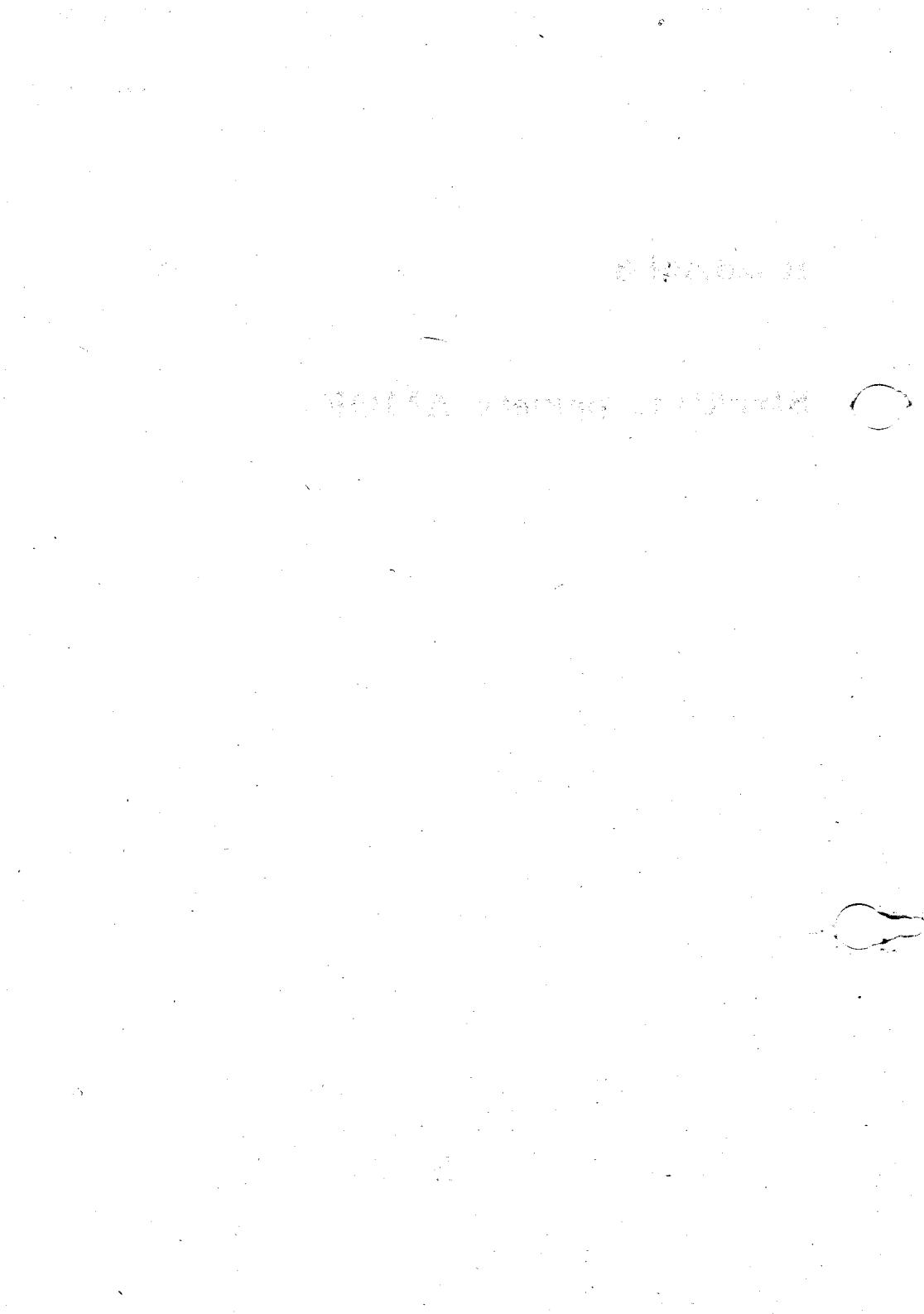
Tab. 2.1: Typy sieci analizowane przez AMOK

Moduł	Liczba źródeł	Liczba stanowisk	Liczba klas	Liczba klientów
JACKSON	30	30	8	—
DIF	30	30	5	—
MVA	—	30	10	∞
HMVA	—	30	10	∞
EPF	—	30	1	50
ITERM	—	10	1	50

Tab. 2.2: Maksymalne rozmiary sieci analizowanych przez wersję 2.10 AMOKu

Rozdział 3

Struktura pakietu AMOK

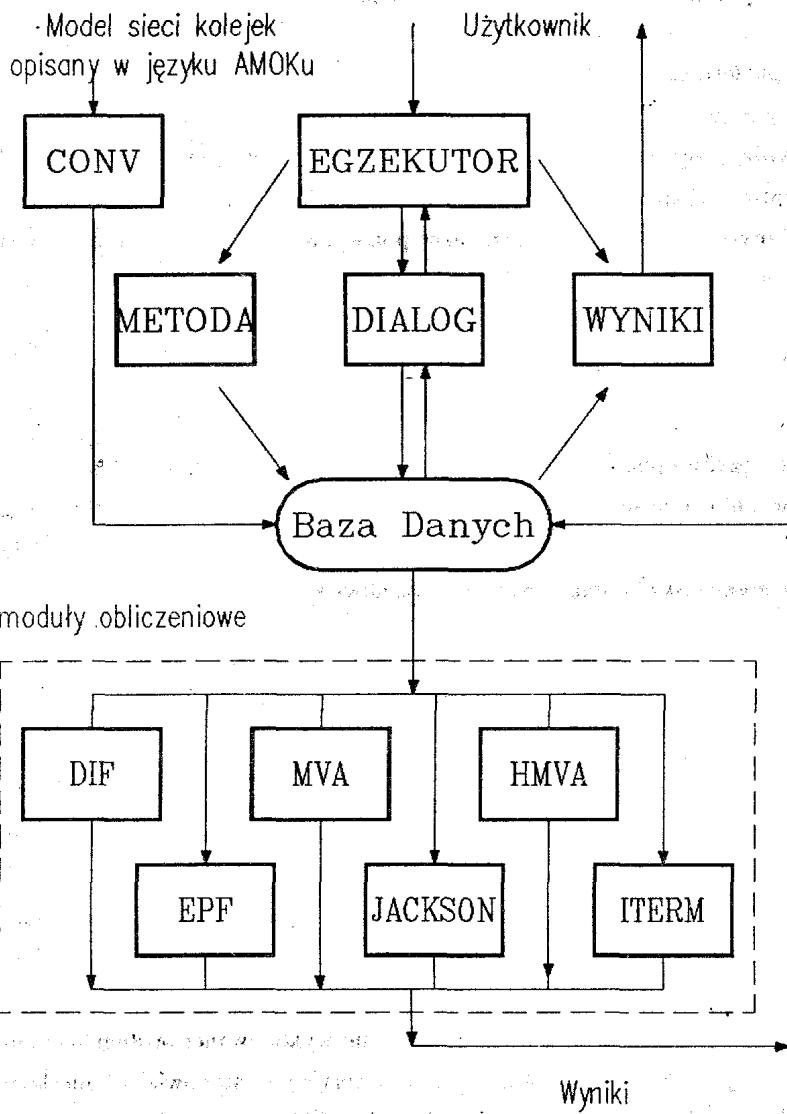


Strukturę pakietu AMOK przedstawia rys.3.1.

Jądro pakietu stanowią napisane w Fortranie moduły obliczeniowe, rozwiązuające różne wersje modeli kolejkowych. Pozostałe programy, napisane w Pascalu, umożliwiają użytkownikowi kontakt z systemem — zapis modelu i odczyt danych. Elementem pośredniczącym między tymi dwiema kategoriami programów jest baza danych — programy obliczeniowe pobierają z niej dane o modelu i składają w niej wyniki.

Definiowanie modelu może przebiegać w dwojakim sposobie:

- w języku opisu modeli — model zapisany w tym języku umieszczony jest w pliku i analizowany przez program konwertatora CONV, który wypełnia bazę danych. Należy podać elementy sieci: *źródła*, które generują klientów i *stanowiska obsługi*. Każdy ze stanowisk musi posiadać nazwę, zdefiniowany rozkład czasu obsługi, regulamin kolejki oraz prawdopodobieństwa przejść dla klientów, którzy zakończyli obsługę. Przewidziano następujące rozkłady: wykładowiczy, stały, równomierny, Erlanga n -tego rzędu, hiper-wykładowiczy, Coxa z n wykładowiczymi fazami (po fazie i , $i = 1, \dots, n-1$, następuje faza $i+1$ z prawdopodobieństwem a ; lub też faza i jest, z prawdopodobieństwem $1-a$, fazą ostatnią), rozkład dowolny o podanej średniej i wariancji. Regulamin kolejki można określić jako: FIFO (first-in-first-out) — naturalny, tj. w kolejności obsługi zgodnej z kolejnością nadania klientów, LIFO (last-in-first-out) — w kolejności odwrotnej do kolejności nadania klientów, LIFO-PR (last-in-first-out-preemptive-resume) — regulamin LIFO z przerwaniem aktualnie wykonywanej obsługi w momencie nadania nowego klienta, IS (infinite server) — stanowisko z nieskończoną liczbą równoległych kanałów obsługi, PS (processor sharing) — obsługa z podziałem czasu procesora — wszyscy obecni w stanowisku klienci są obsługiwani jednocześnie, czas ich obsługi rośnie proporcjonalnie do liczby obsługiwanych klientów.



Rys. 3.1: Struktura pakietu AMOK

Jeżeli regulamin nie jest zdefiniowany, przyjmuje się, że regulamin jest typu FIFO. Dokładny opis języka zawiera rozdział 4. Opis konwertora zamieszczony jest w rozdziale 5.

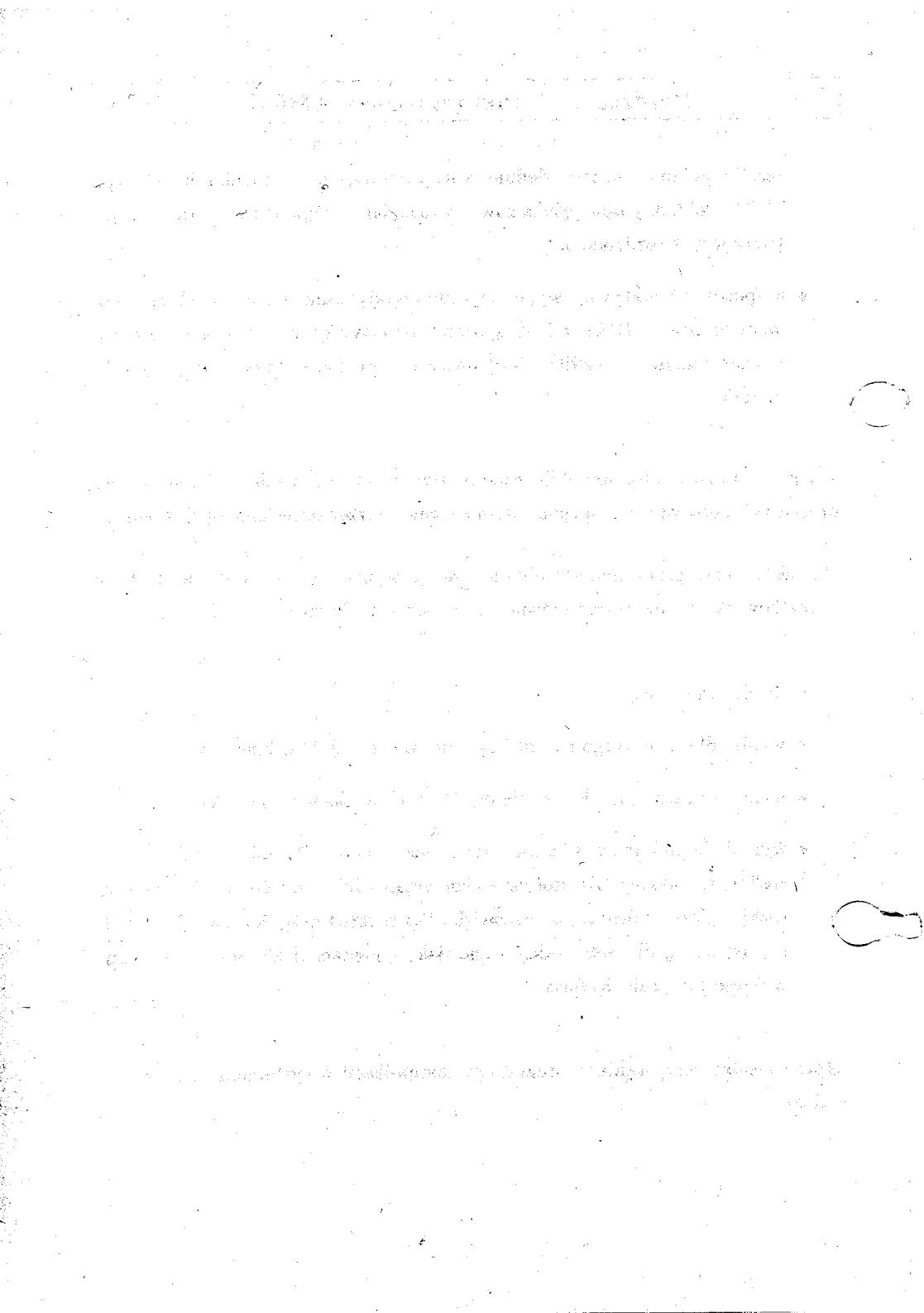
- w sposób interaktywny, wypełniając lub modyfikując bazę danych przy pomocy programu DIALOG. W sposób interaktywny można wygodnie wprowadzać zmiany w modelu zdefiniowanym wcześniej przy pomocy języka AMOKu.

Wybór metody, sterowanie obliczeniami, wybór zakresu wyników i sposobu ich prezentacji odbywa się w sposób interaktywny poprzez określenie opcji menu.

Po zakończeniu pracy modułu obliczeniowego wyniki są wprowadzane do bazy, a użytkownik określa zakres i formę ich prezentacji. Można wyświetlić:

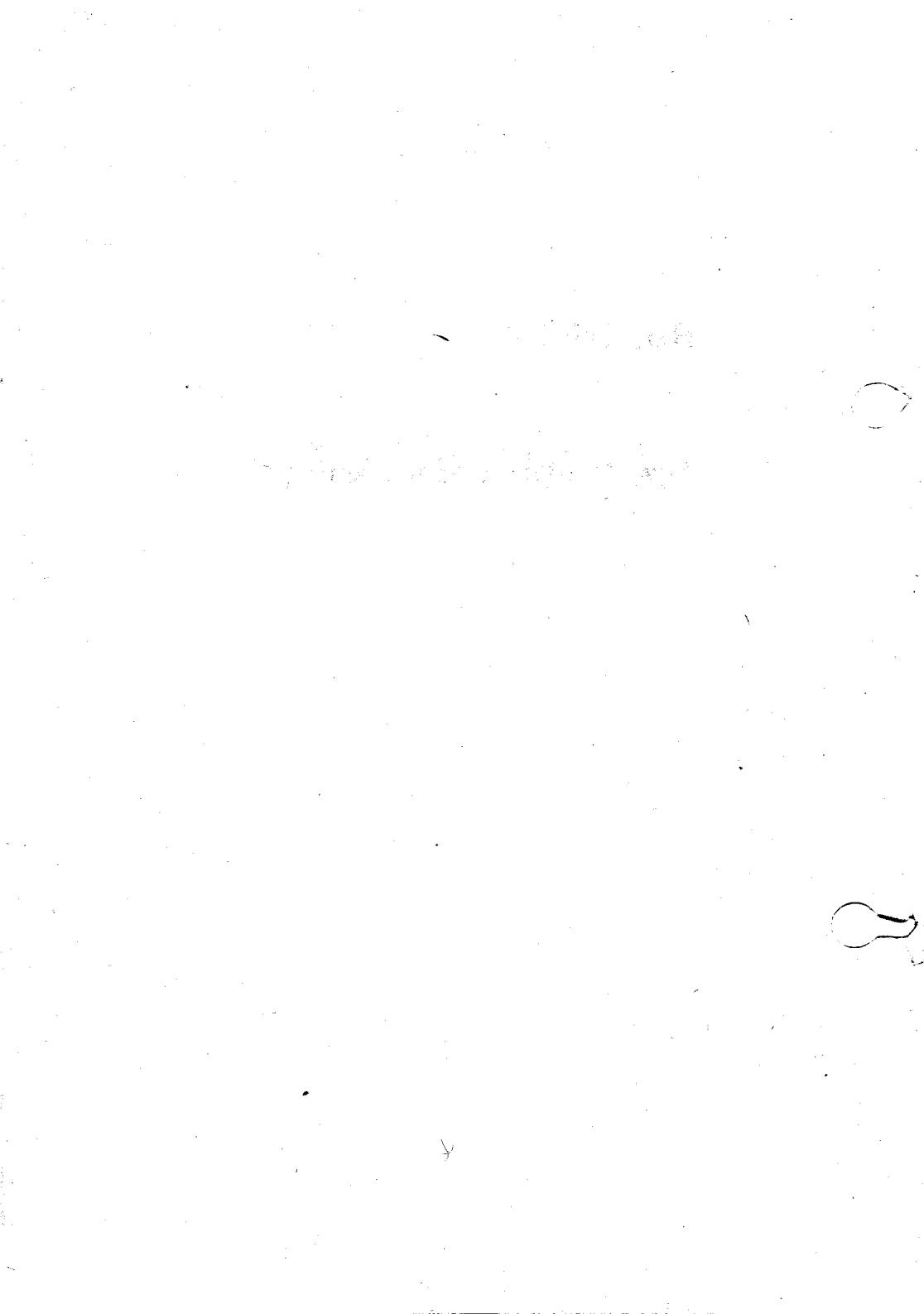
- wyniki całej sieci,
- wyniki dla wybranego stanowiska (dla wszystkich klas klientów),
- wyniki dla wybranej klasy klientów (dla wszystkich stanowisk),
- wyniki dla pojedynczego parametru - można wybrać średni czas odpowiedzi, średnią liczbę klientów, stopień wykorzystania lub przepustowość stanowisk, rozkład liczby klientów w stanowisku lub rozkład długości kolejki. Wyniki podawane są dla wszystkich stanowisk w postaci liczbowej, histogramu, wykresu lub grafu Kiviata.

Opis sposobu korzystania z programów komunikacji z systemem zawiera rozdział 6.



Rozdział 4.

Język opisu sieci kolejek



Rozdział ten prezentuje język opisu sieci kolejek umożliwiający sformalizowany zapis analizowanego modelu kolejkowego w sposób czytelny i zrozumiały dla użytkownika. Zapis w tym języku jest analizowany przez program konwertatora, który wypełnia bazę danych pakietu.

4.1 Podstawowe zasady

Cały opis umieszczony jest w wierszach, z których każdy liczy maksimum 80 znaków.

Wiersz zerowy jest wierszem wyróżnionym i służy do identyfikacji zbioru zawierającego opis. Pierwsze cztery znaki tego wiersza muszą zawierać tekst SOUR. Pozostałe znaki mogą zawierać dowolny tekst wprowadzony przez użytkownika. Wiersze następne zawierają kolejne elementy opisu w języku AMOKu, pisane zgodnie z regułami gramatycznymi tego języka.

Podstawowym elementem języka jest jednostka leksykalna. Jednostką leksykalną jest:

- | | |
|-----------------------|---|
| nazwa | — ciąg liter, cyfr i znaku “-”; zaczynający się od litery; maksymalnie 16 znaków, |
| liczba | — ciąg cyfr, maksymalnie 20 cyfr, |
| znak specjalny | — znaki różne od liter i cyfr. |

Obowiązują następujące zasady ogólne podczas pisania elementów języka:

1. Każda jednostka leksykalna musi w całości mieścić się w jednym wierszu, tzn. nie może być dzielona i jej kontynuacja przenoszona do następnego wiersza.
2. Spacje są całkowicie pomijane.
3. Pomijane są całkowicie wszystkie znaki umieszczone między dwoma kolejnymi znakami wykrzyknika lub między wykrzyknikiem i końcem wiersza. Pomijane są również znaki wykrzyknika. Właściwość tę wykorzystuje się do umieszczania komentarzy.
4. Wszystkie konstrukcje języka mogą być rozmieszczone w dowolny sposób w jednym lub kilku wierszach. Wymagane jest tylko zachowanie porządku jednostek leksykalnych ustalonego przez reguły gramatyczne języka.

4.2 Obiekty

1. Obiekty są podstawowymi elementami opisywanej sieci kolejek, pomiędzy którymi krążą klienci.
2. Obiektami sieci kolejek (typami obiektów) są:
 - (a) źródło - generuje klientów;
 - (b) stanowisko obsługi - obsługuje klientów przez określony czas.
3. Opis każdego obiektu rozpoczyna identyfikator typu obiektu, składający się ze standardowej nazwy typu poprzedzonej i zakończonej znakiem "/".
Identyfikatorami typu są:

- /SOURCE/ — dla źródła;
/STATION/ — dla stanowiska obsługi.

4. Po identyfikatorze typu obiektu występują parametry obiektu. Większość parametrów jest opcjonalna, przyjmuje się wtedy ich standardową definicję.
5. Koniec opisu obiektu wyznacza identyfikator następnego obiektu lub koniec opisu sieci.

4.3 Parametry obiektów

1. Parametry obiektu występują po identyfikatorze typu obiektu.
2. Zapis każdego parametru rozpoczyna się od nazwy parametru będącej słowem kluczowym języka, po której występuje definicja parametru poprzedzona znakiem “=” i zakończona znakiem średnika.

<nazwa parametru> = <definicja parametru>;

Niektoře parametry tzw. zależne od rodzaju klienta mogą mieć definicje dotyczące niektórych wybranych rodzajów klientów. W tym wypadku pomiędzy słowem kluczowym i znakiem “=” może wystąpić ujęta w nawiasy okrągłe lista rodzajów klientów, których ta definicja dotyczy:

<nazwa param. > (<lista rodz.klientów>) = <definicja parametru>;

3. Nazwa parametru określa jego rodzaj i może być jedną z poniższego zestawu:

NAME	— definiuje nazwę obiektu;
COPY	— powoduje skopiowanie definicji wszystkich pominiętych, a niezbędnych dla tego obiektu parametrów z obiektu, którego nazwę parametr COPY wskazuje;
SCHEDULE	— definiuje sposób szeregowania klientów w obiekcie;
SERVICE	— definiuje sposób obsługi klientów w obiekcie;
TRANSITION	— definiuje sposób wyboru kierunku przesyłu klienta do następnego obiektu.

4. Parametr NAME, jeśli występuje, musi być pierwszym parametrem opisu obiektu. Pozostałe parametry wchodzące w skład podzbioru parametrów danego typu obiektu mogą występować w dowolnej kolejności. Niektóre z tych parametrów mogą być opcjonalne, przyjmuje się wtedy ich standar-dową definicję.
5. Parametry: NAME, COPY i SCHEDULE są parametrami, których definicje nie zależą od rodzajów przechodzących przez obiekt klientów. Każdy z tych parametrów w opisie jednego obiektu może występować co najwy-żej jeden raz i tylko wtedy, gdy wchodzi w skład podzbioru parametrów opisujących ten obiekt.
6. Parametry: SERVICE I TRANSITION są parametrami, których definicje mogą dotyczyć tylko pewnych rodzajów klientów. Dlatego też w opisie jednego obiektu każdy z tych parametrów może być zastąpiony odpowiednim zestawem parametrów o identycznym słowie kluczowym. W tym przypadku parametry te oprócz definicji zawierają listę rodzajów klientów, których ta definicja dotyczy.
7. Jeżeli w zestawie parametrów o identycznym słowie kluczowym występuje parametr bez listy rodzajów klientów, to oznacza on, że definicja w nim

zawarta dotyczy wszystkich tych rodzajów klientów, którzy nie zostali wymienieni w listach klientów pozostałych parametrów zestawu. Parametr bez listy rodzajów klientów może wystąpić w opisie obiektu co najwyżej jeden raz i musi być umieszczony jako ostatni w zestawie.

Przykład:

```
SERVICE (:K1) = <definicja obsługi klientów z klasy K1>;  
SERVICE (:K2) = <definicja obsługi klientów z klasy K2>;  
SERVICE (:K3,:K4) = <definicja obsługi klientów z klas K3 i K4>;  
SERVICE = <definicja obsługi wszystkich pozostałych klientów>;
```

UWAGA: Każdy z parametrów występujących w opisie obiektu typu SOURCE może wystąpić w nim co najwyżej jeden raz i żaden z nich nie może zawierać listy rodzajów klientów. W obiekcie tym definicje parametrów zależne od rodzaju klientów dotyczą wszystkich wygenerowanych w nim klientów.

8. Na końcu definicji parametru może występować opcjonalny znak średnika. Znak ten jest obowiązkowy, gdy służy do rozdzielenia dwóch nazw jako odrębnych jednostek leksykalnych. Sytuacja taka powstaje np. wtedy, gdy definicja parametru kończy się nazwą, po której w tej samej linii występuje słowo kluczowe następnego parametru.
9. Każda z metod analizy sieci obowiązuje zasady wnoszące pewne ograniczenia przy definiowaniu obiektów. Zatem obiekt może być poprawnie opisany, ale użyte parametry lub ich definicje nie pozwalają na zastosowanie danej metody obliczeń.

4.4 Lista rodzajów klientów

1. Listy rodzajów klientów ujęte w nawiasy okrągłe występują pomiędzy słowem kluczowym i znakiem “=” parametrów, których definicje są różne dla różnych rodzajów klientów oraz w definicjach parametru SCHEDULE.
2. Lista rodzajów klientów składa się z nazw klas klientów oddzielonych od siebie znakiem przecinka.
3. Każdy klient może należeć tylko do jednej klasy klientów. Przypisanie klienta do danej klasy dokonuje się w definicji parametru TRANSITION obiektu SOURCE.

4.5 Definicje parametrów obiektów

4.5.1 Definicja parametru NAME

1. Za pomocą parametru NAME definiuje się nazwę obiektu. Z wyjątkiem SOURCE parametr ten jest obowiązkowy dla wszystkich pozostałych obiektów.
2. Nazwę obiektu umieszcza się za znakiem “=”, jej długość nie może przekraczać 16 znaków.
3. Nazwa każdego obiektu musi być unikalna i jej pierwszych osiem znaków nie może być identycznych z pierwszymi ośmioma znakami nazwy innego obiektu.

4. Parametr NAME musi być pierwszym parametrem obiektu.

Przykład:

NAME = PROCESOR;

4.5.2 Definicja parametru COPY

1. Za pomocą parametru COPY można skopiować do opisywanego obiektu wszystkie parametry wcześniej zdefiniowane w innym obiekcie.
2. Definicję parametru COPY stanowi nazwa obiektu, z którego parametry mają być kopowane.
3. W przypadku obiektów o różnych typach kopowane są tylko te parametry, które w danym obiekcie są dozwolone.
4. Parametry, które nie występują we wskazanym przez COPY obiekcie lub których definicje mają być inne od skopiowanych umieszcza się w opisie w zwykły sposób.
5. Za pomocą parametru COPY nie jest kopowana nazwa obiektu. Definicję parametru COPY nie może być definicja AS.
6. Obiekt wskazywany przez parametr COPY musi być obiektem wcześniej zdefiniowanym.

Przykład:

COPY = DYSK-1;

4.5.3 Definicja AS (nazwa obiektu)

1. Z wyjątkiem parametru NAME oraz COPY definicje pozostałych parametrów mogą być zastąpione definicją AS.
2. Definicja AS oznacza, że definicja danego parametru jest identyczna z definicją tegoż parametru umieszczonego w obiekcie o podanej nazwie.
3. We wskazanym obiekcie parametr ten musi być parametrem dozwolonym dla tego obiektu, nie może być w nim pominięty, jak również sam nie może być zdefiniowany za pomocą AS.
4. Parametr zawierający definicję AS nie może zawierać list klientów, jak również nie może występować w postaci zestawu parametrów o identycznym, słowie kluczowym.

Przykład:

SERVICE = AS(CPU)

4.5.4 Definicja parametru SCHEDULE

1. Parametr SCHEDULE definiuje sposób szeregowania klientów w obiektach typu STATION.
2. Parametr SCHEDULE jest opcjonalny. Gdy jest pominięty przyjmuje się, że klienci są szeregowani w kolejkach według regulaminu FIFO.
3. Definicja parametru SCHEDULE składa się z nazwy regulaminu szeregowania i występującej po niej, na ogół opcjonalnej, ujętej w nawiasy okrągłe, listy rodzajów szeregowanych klientów.

4. Nazwy regulaminów szeregowania klientów są następujące:

- | | |
|------------------|---|
| FIFO, FCFS | — naturalny; |
| LIFO, LCFS | — w kolejności odwrotnej do przybycia (stosowy), bez wywłaszczenia; |
| PS | — z podziałem procesora; |
| IS | — nieskończona liczba stanowisk obsługi; |
| LIFO-PR, LCFS-PR | — stosowy z wywłaszczeniem |

4.5.5 Definicja parametru SERVICE

1. Znaczenie definicji parametru SERVICE zależy od rodzaju obiektu w opisie którego została użyta. W obiekcie typu SOURCE parametr SERVICE definiuje standardowy rozkład losowy odstępów czasu pomiędzy generacją kolejnych klientów. W obiekcie typu STATION parametr SERVICE definiuje standardowy rozkład losowy czasów obsługi klientów na stanowisku obsługi.
2. Parametr SERVICE jest parametrem obowiązkowym przy opisie obiektów.

Definicje standardowych rozkładów losowych.

Definicję standardowego rozkładu losowego stanowi nazwa rozkładu prawdopodobieństwa z występującą po niej listą parametrów liczbowych ujętą w nawiasy kwadratowe. Parametrem liczbowym jest tutaj liczba stałoprzecinkowa bez znaku.

- Rozkład deterministyczny

DET [*<wartość czasu>*]

lub

<wartość czasu>

Słowo DET oraz otaczające parametr nawiasy kwadratowe są opcjonalne.

- Rozkład równomierny

UNI [*<min. wartość czasu>*, *<maks. wartość czasu>*]



- Rozkład wykładniczy

EXP [*<średnia wartość czasu>*]

- Rozkład Coxa

COX [*<prawdopodobieństwo p_1 >*, *<wartość czasu s_1 >* / ... /

<prawdopodobieństwo p_F >, *<wartość czasu s_F >*]

Znak "/" może być zastąpiony znakiem przecinka. Znaczenie parametrów rozkładu Coxa ilustruje rysunek 4.1.

- Rozkład Erlanga

ERL [*<n>*, *<wartość czasu s>*]

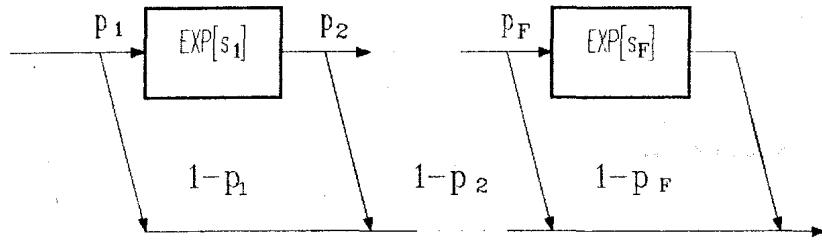
Rozkład Erlanga jest szczególnym przypadkiem rozkładu Cox'a, w którym wszystkie $p_i = 1$. Wartość czasu "s" jest tutaj sumą $\sum_{i=1}^n s_i$.

- Rozkład hiperwykładniczy

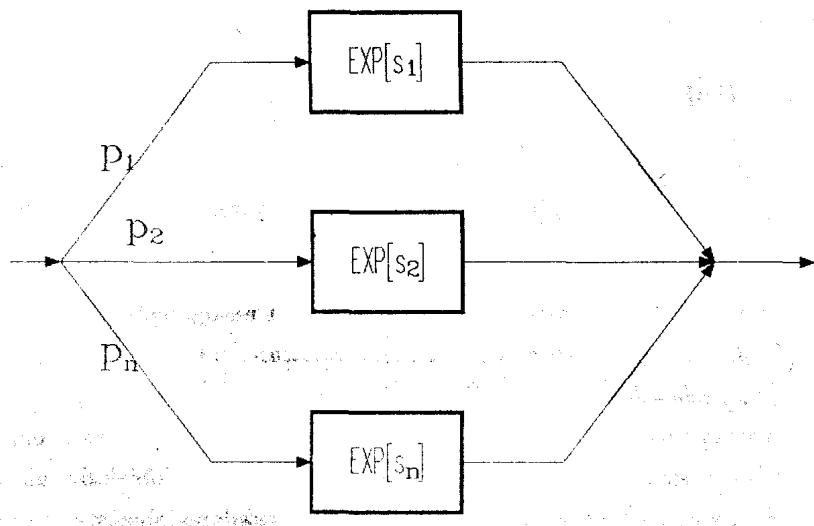
HIP [*<prawdopodobieństwo p_1 >*, *<wartość czasu s_1 >* / ... /

<prawdopodobieństwo p_n >, *<wartość czasu s_n >*]

Suma prawdopodobieństw równa jest jedności. Znak "/" może być zastąpiony znakiem przecinka. Znaczenie parametrów rozkładu hiperwykładniczego ilustruje rys. 4.2.



Rys. 4.1: Rozkład Coxa



Rys. 4.2: Rozkład hiperwykładniczy

- Rozkład ogólny

$G [<\text{wartość średnia czasu}>, <\text{współczynnik zmienności}>]$

Współczynnik zmienności jest ilorazem odchylenia standardowego i wartości średniej: $C_x = \frac{\sigma_x}{E(x)}$.

Przykłady:

```

SERVICE      = 3.0;
SERVICE      = EXP [0.5];
SERVICE      = ERL [5, 0.333];
SERVICE      = HIP [0.6, 5.0/0.4, 10.0];
SERVICE (:K1) = EXP [1.0];
SERVICE (:K2) = UNI [0.5, 2.5];

```

4.5.6 Definicja parametru TRANSITION

- Parametr TRANSITION definiuje wybór kierunku przesłania klienta do następnego obiektu sieci.
- Postać definicji parametru TRANSITION jest następująca:
 $[<\text{prawdopodobieństwo } p_1>] <\text{nazwa obiektu}>: <\text{nazwa klasy}>, \dots,$
 $[<\text{prawdopodobieństwo } p_n>] <\text{nazwa obiektu}>: <\text{nazwa klasy}>$
 Według powyższej definicji klient udaje się z prawdopodobieństwem p_i do obiektu sieci o_i stając się klientem klasy k_i . Nazwy obiektów nie mogą obejmować obiektów typu SOURCE. Suma prawdopodobieństw równa jest jedności.
- Jeżeli klienci przesyłani są tylko do jednego obiektu sieci, to prawdopodobieństwo wraz z otaczającymi nawiasami kwadratowymi można pominąć.

4. Jeżeli klienci mają opuścić sieć kolejek to zaznacza się to wstawiając zamiast nazwy obiektu, znaku dwukropka i nazwy klasy słowo OUT. OUT nie jest słowem zastrzeżonym języka i jeśli użytkownik użył go jako nazwy obiektu sieci, to klienci zostaną skierowani do tego obiektu.
5. Definicja TRANSITION użyta przy opisie obiektu typu SOURCE określa początkowe klasy klientów. Jeżeli w definicji tej pominięto nazwę klasy, to wygenerowani z tego obiektu klienci należą do klasy bez nazwy.

Przykłady:

```
TRANSITION      =  CPU;  
TRANSITION      =  [0.5] DISC-1:KL1, [0.5] DISC-2;  
TRANSITION (:K1) =  OUT;  
TRANSITION (:K2) =  [0.5] PROC-WE-WY, [0.5] OUT;
```

4.6 Parametry liczbowe

4.6.1 Liczby całkowite

Liczba całkowita w konstrukcjach języka AMOKu jest zapisywana jako ciąg cyfr bez znaku. Wszystkie wartości liczbowe występujące w konstrukcjach języka AMOKu są dodatnie, dlatego też przyjęto, że będą one zapisywane bez znaku liczby.

4.6.2 Liczby rzeczywiste

Liczba rzeczywista może być zapisana w jednej z poniższych postaci:

n.m

n.

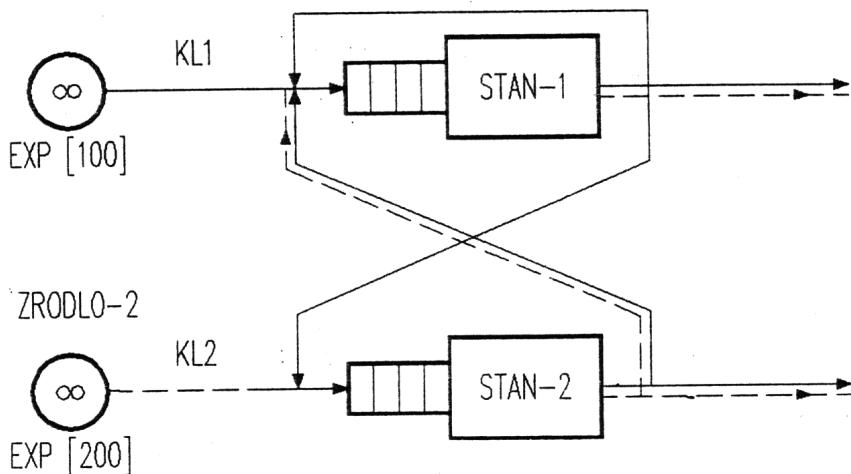
.m

i

gdzie:

- n, m — ciągi cyfr. Łączna liczba cyfr obu ciągów nie może być większa od 10;
- i — liczba całkowita.

Podobnie jak w przypadku liczb całkowitych, nie jest dozwolone używanie znaku liczby.



Rys. 4.3: Przykładowa otwarta sieć stanowisk obsługi

4.7 Przykładowe sieci kolejek i ich zapisy w języku AMOKu

Przykład 1

Rys. 4.3 przedstawia sieć otwartą, zawierającą dwa źródła, które generują klientów w odstępach czasu o rozkładzie wykładniczym o średnich odpowiednio 100 i 200.

Źródło pierwsze (nie nazwane), generuje klientów klasy KL1, źródło drugie, nazwane ZRODLO-2, generuje klientów klasy KL2.

		KL1	KL2
obsługa	STAN-1	rozkład Erlanga drugiego rzędu o średniej 20	rozkład hiperwykładniczy drugiego rzędu $p_1 = 0.3, s_1 = 30,$ $p_2 = 0.7, s_2 = 40$
	STAN-2	rozkład wykładniczy o średniej 30	rozkład deterministyczny o średniej 20
prawdopodobieństwa przejść	STAN-1	0.1 – powrót do STAN-1 0.5 – wyjście z sieci 0.4 – przejście do STAN-2	1.0 – wyjście z sieci
	STAN-2	0.5 – przejście do STAN-1 0.5 – wyjście z sieci	0.7 – przejście do STAN-1 0.3 – wyjście z sieci

Tab. 4.1: Parametry sieci

Droga klientów w sieci uwidoczniona jest w postaci linii — dla klientów klasy KL1 — ciągłych, dla KL2 — przerywanych. Regulamin szeregowania w obu stanowiskach obsługi, nazwanych STAN-1 i STAN-2, jest ten sam — naturalny. Klienci obsługiwani są z różnymi rozkładami, o różnych średnich i wariancjacach, w zależności od klasy, do której przynależą. Podobnie, prawdopodobieństwa przejść zależne są od klasy klientów — patrz tabela 4.1

Zapis powyższej sieci przedstawia rys. 4.4.

SOUR PRZYKŁAD SIECI OTWARTEJ

DESCRIPTION

```
/SOURCE/    SERVICE = EXP [100]          !ZRODŁO BEZ NAZWY!
            TRANSIT = STAN-1: KL1;

/SOURCE/    NAME      = ZRODŁO - 2
            SERVICE   = EXP [200]
            TRANSIT   = STAN-2: KL2

/STATION/   NAME      = STAN-1;
            SCHEDULE = FIFO           !MOŻNA POMINIĘĆ!
            SERVICE  (:KL1) = ERL [2,20]
            SERVICE  (:KL2) = HIP [0,3, 30/0,7, 40]
            TRANSIT  (:KL1) = [0.1] STAN-1,
                           [0.5] OUT,
                           [0.4] STAN-2;
            TRANSIT  (:KL2) = OUT;

/STATION/   NAME      = STAN-2;
            TRANSIT  (:KL2) = [0.7] STAN-1, [0.3] OUT
            TRANSIT = [0.5] STAN-1, [0.5] OUT      !DLA KL1!
            SERVICE  (:KL1) = EXP [30]; SERVICE = DET [20] !DLA KL2!
```

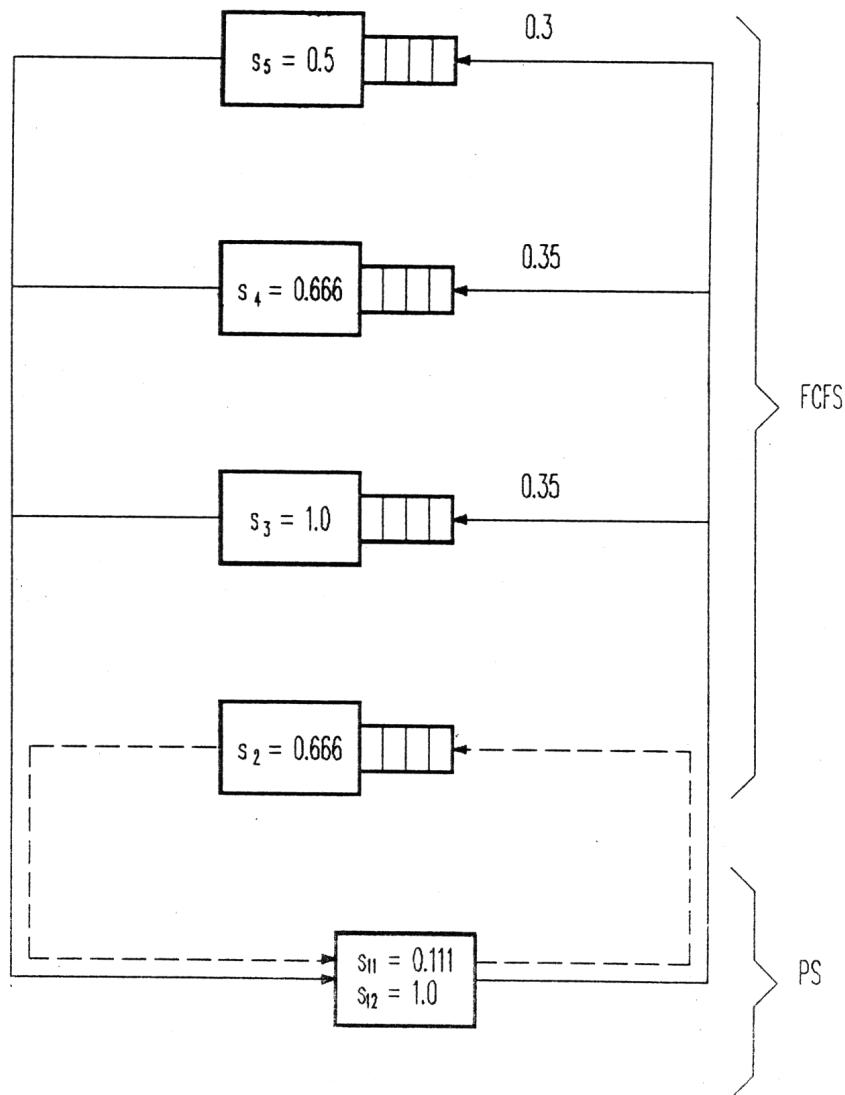
END

Rys. 4.4: Zapis otwartej sieci stanowisk obsługi

Przykład 2.

Sieć kolejek (rys. 4.5) jest zamknięta, z pięcioma stanowiskami i dwiema klasami klientów. Stanowiska 2,3,4,5, mają kolejkę o regulaminie naturalnym, stanowisko 1 dzieli swój czas pomiędzy obsługiwanych klientów. Sieć ta może być modelem wieloprogramowego systemu komputerowego, gdzie stanowisko 1 reprezentuje jednostkę centralną, a pozostałe stanowiska — urządzenia we/wy.

Opis sieci w języku AMOKu przedstawia rys. 4.6.



Rys. 4.5: Przykładowa zamknięta sieć stanowisk obsługi

SOUR PRZYKŁAD SIECI ZAMKNIETEJ

DESCRIPTION

```
/STATION/      NAME = STAT 1
                  SCHEDULE = PS
                  SERVICE (:CLASS1) = EXP [0.111]
                  SERVICE (:CLASS2) = EXP [1.0]
                  TRANSIT (:CLASS1) = [0.35] STAT 3,
                                         [0.35] STAT 4,
                                         [0.3]  STAT 5
                  TRANSIT (:CLASS2) = STAT 2

/STATION/      NAME = STAT 2
                  SERVICE = EXP [0.666]
                  TRANSIT = STAT 1

/STATION/      NAME = STAT 3
                  SERVICE = EXP [1.0]
                  TRANSIT = AS (STAT 2)

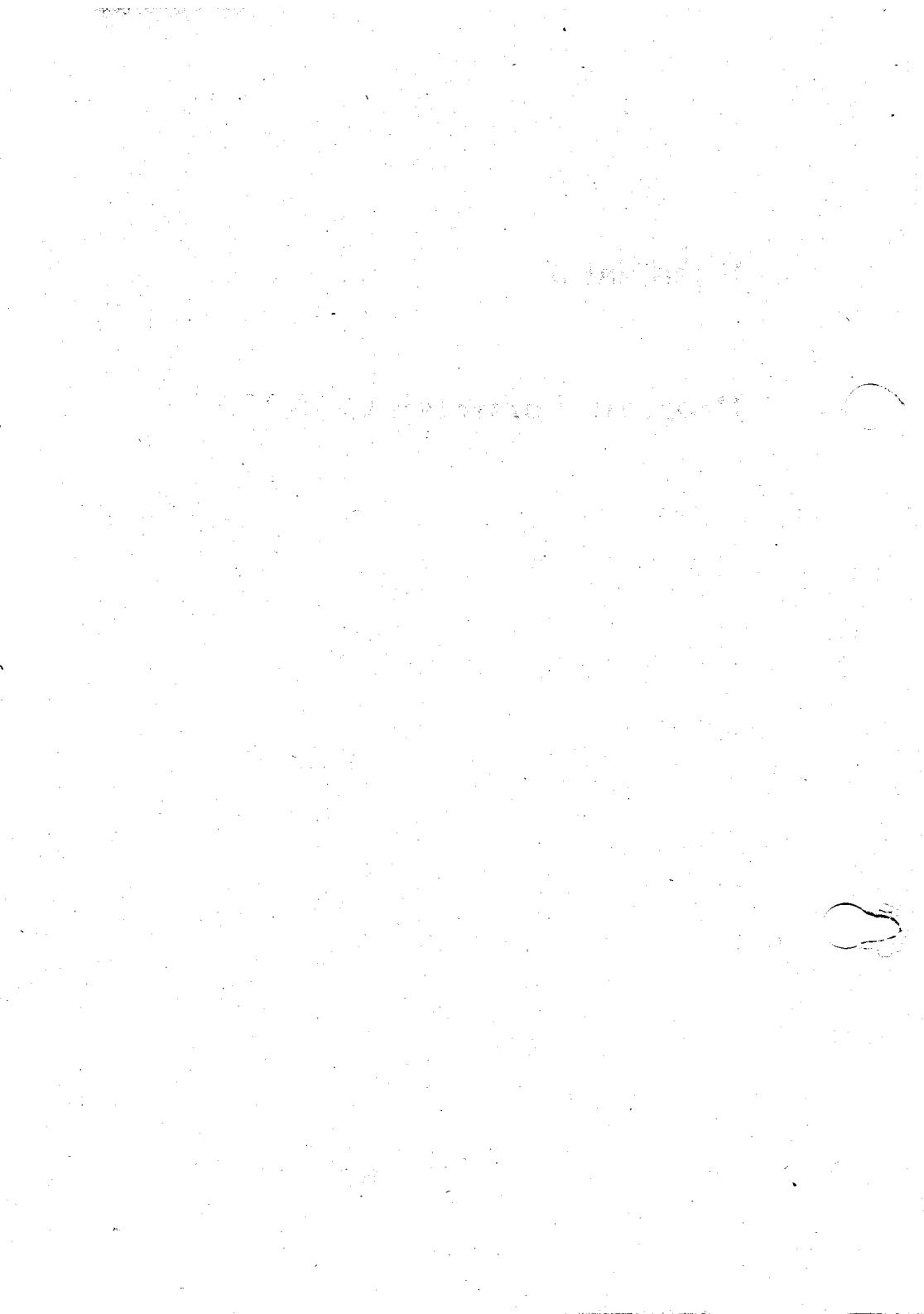
/STATION/      NAME = STAT 4
                  COPY = STAT 2

/STATION/      NAME = STAT 5
                  SERVICE = EXP [0.5]
                  TRANSIT = STAT 1
```

END

Rozdział 5

Program konwersji CONV



CONV jest programem pakietu AMOK, który na podstawie opisu sieci kolejek w języku AMOKu tworzy plik danych wejściowych dla innych modułów pakietu, tj. programów analitycznego rozwiązywania sieci. Podczas tej konwersji dokonywana jest analiza poprawności opisu sieci. Zwykle w wyniku pracy programu CONV, zostają wygenerowane dwa pliki. Pierwszy z nich powstaje w wyniku analizy opisu i zawiera informacje o przebiegu konwersji, ewentualnie wykrytych błędach i informacje dodatkowe. Drugi powstaje tylko w przypadku, gdy konwersja została zakończona poprawnie i jest wspomnianym plikiem danych wejściowych dla innych modułów pakietu AMOK, nazywanym również bazą danych.

5.1 Komunikaty analizy syntaktycznej

1. Program konwersji CONV podczas analizy syntaktycznej generuje cztery rodzaje komunikatów:
 - a) Komunikaty o błędach — ERROR
 - b) Ostrzeżenia — WARNING
 - c) Wiadomości — MESSAGE
 - d) Komunikaty o wadliwym działaniu programu CONV — FATAL
2. Każdy z powyższych rodzajów komunikatów jest wyprowadzany w następującym formacie:

CHAR <numer znaku> <typ błędu> <numer błędu> <informacja objaśniająca>
gdzie:

- numer znaku* — jest numerem znaku, od którego zaczyna się jednostka leksykalna, której komunikat dotyczy. Znaki liczone są od 1. W skład liczonych znaków nie wchodzą spacje oraz znaki komentarza wraz z otaczającymi je znakami wykrywnika,
- typ błędu* — ERROR, WARNING, MESSAGE lub FATAL,
- numer błędu* — numer komunikatu w ramach rodzaju błędu,
- informacja objaśniająca* — jest słowną informacją o przyczynie wyprowadzania komunikatu.

3. Komunikaty są drukowane po linii, której dotyczą. Pomiędzy analizowaną linią a pierwszym komunikatem odnoszącym się do tej linii wstawiany jest wiersz ze strzałkami. Strzałki, podobnie jak numery znaków, wskazują początkowe znaki jednostek leksykalnych, których komunikaty dotyczą.
4. Wyprowadzanych jest nie więcej niż 10 komunikatów odnoszących się do jednej linii opisu. Pozostałe komunikaty są tracone. Jest wyprowadzona jedynie informacja o liczbie straconych komunikatów w następującej postaci:
 # <liczba straconych komunikatów> REPORTS LOST.

5.2 Komunikaty o błędach - ERROR

1. Komunikat o błędzie ERROR (patrz tablice 5.1, 5.2, 5.3) zostaje wyprowadzony wtedy, gdy w danej konstrukcji języka AMOKu zostanie napotkana jednostka leksykalna, której występowanie nie było przewidziane zgodnie

z przyjętymi regułami gramatycznymi. Przyczyną wyprowadzenia komunikatu o błędzie może być również występowanie jednostki leksykalnej o nieodpowiedniej postaci.

2. Komunikatowi o błędzie czasami towarzyszy komunikat typu MESSAGE (patrz tablice 5.5, 5.6), który podaje uzupełniające informacje związane z wykrytym błędem.
3. Wykrycie błędu zazwyczaj przerywa dalszą analizę syntaktyczną opisu. Od tego miejsca kontroluje się wszystkie kolejne nazwy, aż do napotkania pierwszego słowa kluczowego spośród wybranych słów kluczowych języka AMOKu. Począwszy od tego słowa następuje ponowne uruchomienie analizy syntaktycznej.
4. Słowa kluczowe oraz ich skróty nie są słowami zastrzeżonymi, zatem użytkownik może wprowadzić do opisu nazwy własne identyczne ze słowami kluczowymi. Nie zaleca się jednak zbyt częstego stosowania takich nazw jak również skrótów niektórych słów kluczowych, gdyż w przypadku popełnienia błędu może to doprowadzić do powstania tzw. błędów pochodnych.
5. W przypadku wykrycia błędu syntaktycznego nie zostaje wygenerowana baza danych wejściowych, jak również nie jest przeprowadzona analiza semantyczna opisu.

nr błędu	komunikat
1	nieoczekiwana nazwa
2	nieoczekiwana liczba
3	nieoczekiwany znak
4	powtórzona nazwa (nazwa nie jest unikalna)
5	oczekiwany był kwadratowy nawias zamykający
6	oczekiwany był okrągły nawias otwierający
7	oczekiwany był okrągły nawias zamykający
8	oczekiwany był okrągły nawias zamykający lub przecinek
9	oczekiwany był znak "="
10	oczekiwany był kwadratowy nawias otwierający
11	oczekiwany był znak ">" lub przecinek
12	oczekiwany był znak kropki
13	nie jest używany
14	oczekiwany był znak "/"
15	oczekiwany był znak gwiazdki "*"
16	nie jest używany
17	pominięto poprzedzający znak gwiazdki "*"
18	pominięto poprzedzający znak "/"
19	nie jest używany
20	nie jest używany
21	liczba ma za dużo cyfr (ponad 20 cyfr)
22	część całkowita liczby poza zakresem INTEGER
23	oczekiwana była liczba
24	kontynuacja liczby w nowym wierszu ?
25	wykładnik poza zakresem
26	prawdopodobieństwo większe od 1 ?

Tab. 5.1: Komunikaty o błędach

nr błędu	komunikat
27	nie jest używany
28	nie jest używany
29	nie jest używany
30	nie jest używany
31	nieoczekiwany znak — oczekiwano liczby, nazwy lub znaku “?”
32	nie jest używany
33	niesparowane nawiasy
34-40	nie są używane
41	oczekiwana była nazwa klasy klientów
42	oczekiwany był znak — lub lista rodzajów klientów
43	nieoczekiwana liczba — oczekiwano definicji rodzaju klientów
44	nieoczekiwany znak — oczekiwano definicji rodzaju klientów
45	nie jest używany
46	nie jest używany
47	nie jest używany
48	AS nie może być definicją parametru występującego w zestawie
49	definicja AS nie jest dozwolona dla parametru z listą klientów
50	definicja AS musi być jedyną definicją parametru SERVICE
51	nierozpoznana nazwa regulaminu szeregowania klientów
52	regulamin szeregowania o tej nazwie nie jest dozwolony dla danego obiektu
53	nierozpoznana definicja parametru SERVICE
54-59	nie są używane
60	wskazany obiekt zawiera parametr COPY
61	nierozpoznana nazwa parametru obiektu

Tab. 5.2: Komunikaty o błędach (kontynuacja)

nr błędu	komunikat
62	niedozwolony parametr dla analizowanego obiektu
63	ten parametr już wystąpił w analizowanym obiekcie
64	analizowany parametr nie jest dozwolony w obiekcie wskazanym za pomocą definicji AS
65	parametr ten już wystąpił z definicją AS
66	oczekiwany był parametr NAME
67	parametr NAME musi być pierwszym parametrem obiektu
68	parametr ten już wystąpił bez listy rodzaju klientów (zatem parametr bez listy klientów nie jest ostatnim w zestawie)
69	parametr z definicją AS nie został zdefiniowany we wskazanym przez AS obiekcie
70	parametr we wskazanym obiekcie był już kopowany za pomocą AS
71	oczekiwana była nazwa obiektu
72	nie używany
73	obiekt o tej nazwie nie został zdefiniowany
74	oczekiwana była nazwa typu obiektu.

Tab. 5.3: Komunikaty o błędach (dokończenie)

Nr ostrzeżenia	komunikat
1	liczba cyfr znaczących będzie zmniejszona (do 10 cyfr)
2	za dłuża nazwa zmiennej — skrócono do 6 znaków
3	nie używany
4	nazwa za dłuża — użyto pierwszych 16 znaków
powyżej 4	nie jest używany

Tab. 5.4: Ostrzeżenia

5.3 Ostrzeżenia — WARNING

- Ostrzeżenia WARNING (patrz tablica 5.4) są drukowane w przypadku wykrycia błędu, który może być usunięty przez sam program CONV. Usunięcie błędu wiąże się zwykle z niewielką zmianą analizowanego opisu i o niej również informuje komunikat ostrzeżenia.
- Ostrzeżenia nie wstrzymują generowania bazy danych wejściowych, jak również nie wstrzymują przeprowadzania analizy semantycznej opisu.

5.4 Komunikaty uzupełniające — MESSAGE

Komunikaty tego typu towarzyszą komunikatom o błędach i zawierają dodatkowe, uzupełniające informacje (patrz tablice 5.5, 5.6).

numer wiadomości	numer błędu któremu wiadomość towarzyszy	komunikat
1		dalsza analiza wstrzymana do następnego słowa kluczowego
2		początek dalszej analizy po poprzednim błędzie
3		nie jest używany
4	1, 2, 3	oczekiwany był znak przecinka
5	1, 2, 3	oczekiwano jednego ze znaków "/", "[" lub "]"
6	1, 2, 3	oczekiwano kwadratowego nawiasu zamykającego
7	1, 2, 3	oczekiwano liczby lub kwadratowego nawiasu otwierającego
8	1, 2, 3	prawdopodobnie pominięto kwadratowy nawias otwierający
9		nie jest używany
10		nie jest używany
11		nie jest używany
12	68	parametr bez listy klientów musi być ostatni w zestawie
13	3	definicja AS nie jest dozwolona dla parametru z listą klientów
14	9	analizowany parametr nie może zawierać listy rodzajów klientów
15	9	parametry obiektu typu SOURCE nie mogą zawierać list rodzajów klientów

Tab. 5.5: Komunikaty uzupełniające

numer wiadomości	numer błędu któremu wiadomość towarzyszy	komunikat
16÷22		nie są używane
23	8	możliwy błąd w definicji klasy klientów
24	3	przesłanie probabilistyczne jako parametr musi być ujęte w nawiasy
25	1, 2, 3	oczekiwano nazwy regulaminu szeregowania klientów
26	3	AS nie może być definicją parametru występującego w zestawie.
27	1, 2, 3	niekompletna liczba parametrów rozkładu losowego

Tab. 5.6: Komunikaty uzupełniające (dokonczenie)

5.5 Komunikaty FATAL o wadliwym działaniu programu CONV

1. Komunikaty typu FATAL nie powinny występować przy sprawnie działającym programie CONV. Jeżeli wystąpią, oznacza to wykrycie błędu w programie CONV, o którym należy poinformować producenta.
2. Błąd typu FATAL powoduje natychmiastowe zatrzymanie programu CONV z komunikatem: CONVERSION HALTED.

5.6 Inne komunikaty

1. Nieprawidłowy identyfikator zbioru z opisem sieci (nie ma SOUR)
SOURCE FILE IDENTIFIER INVALID
2. Zbiór z opisem jest pusty
SOURCE FILE EMPTY
3. Przerwanie dalszej konwersji
CONVERSION HALTED
4. Zbiór z opisem się skończył; nie napotkano symbolu *END*
SOURCE FILE EXHAUSTED *END* SYMBOL NOT FOUND
5. Koniec analizy syntaktycznej

END OF SYNTACTICAL ANALYSIS — { OK
 WARNING
 ERRORS }

6. Koniec analizy semantycznej

END OF SEMANTICAL ANALYSIS — { OK,
WARNING,
ERRORS }

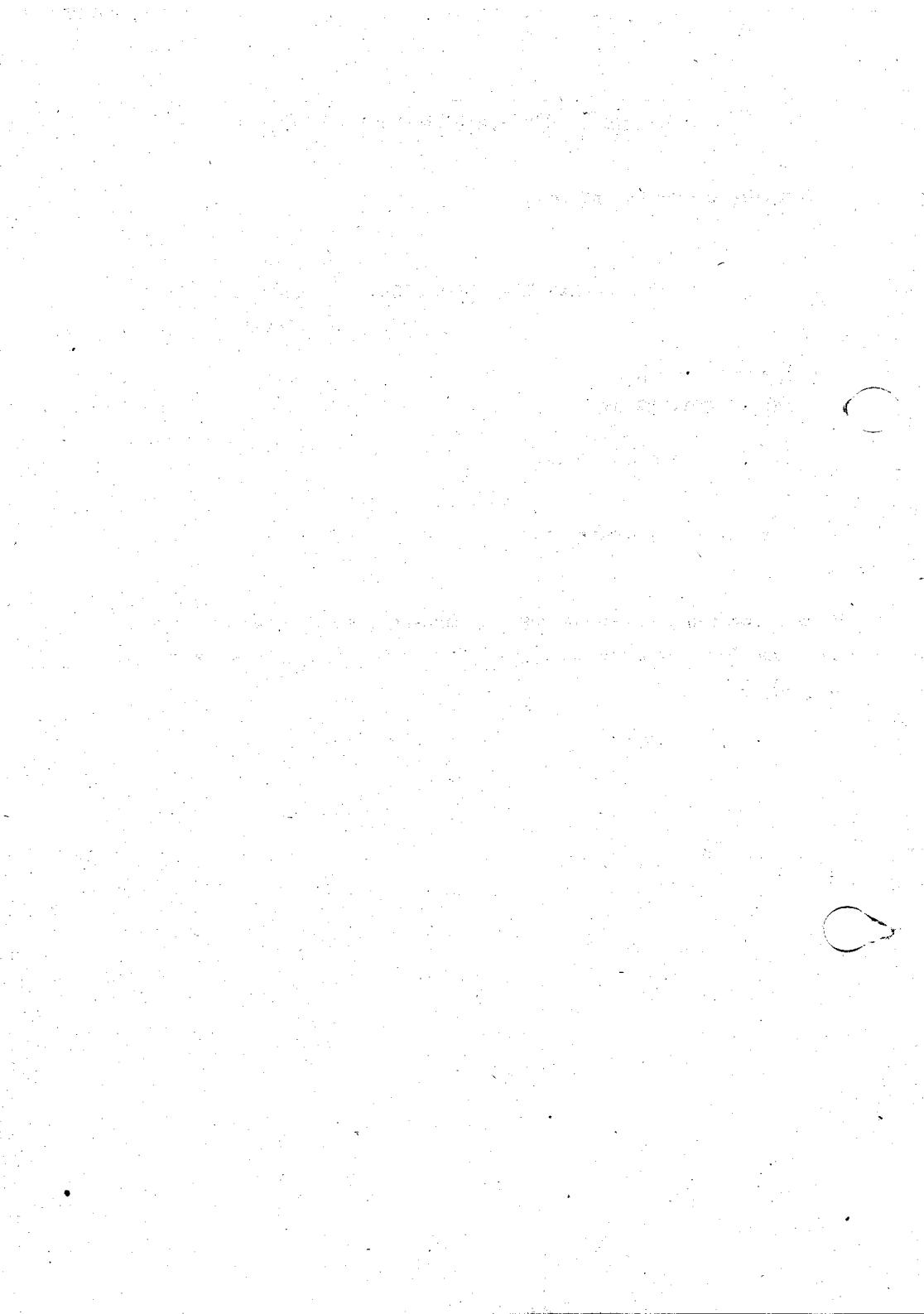
7. Koniec konwersji

END OF CONVERSION

8. Ilość straconych komunikatów

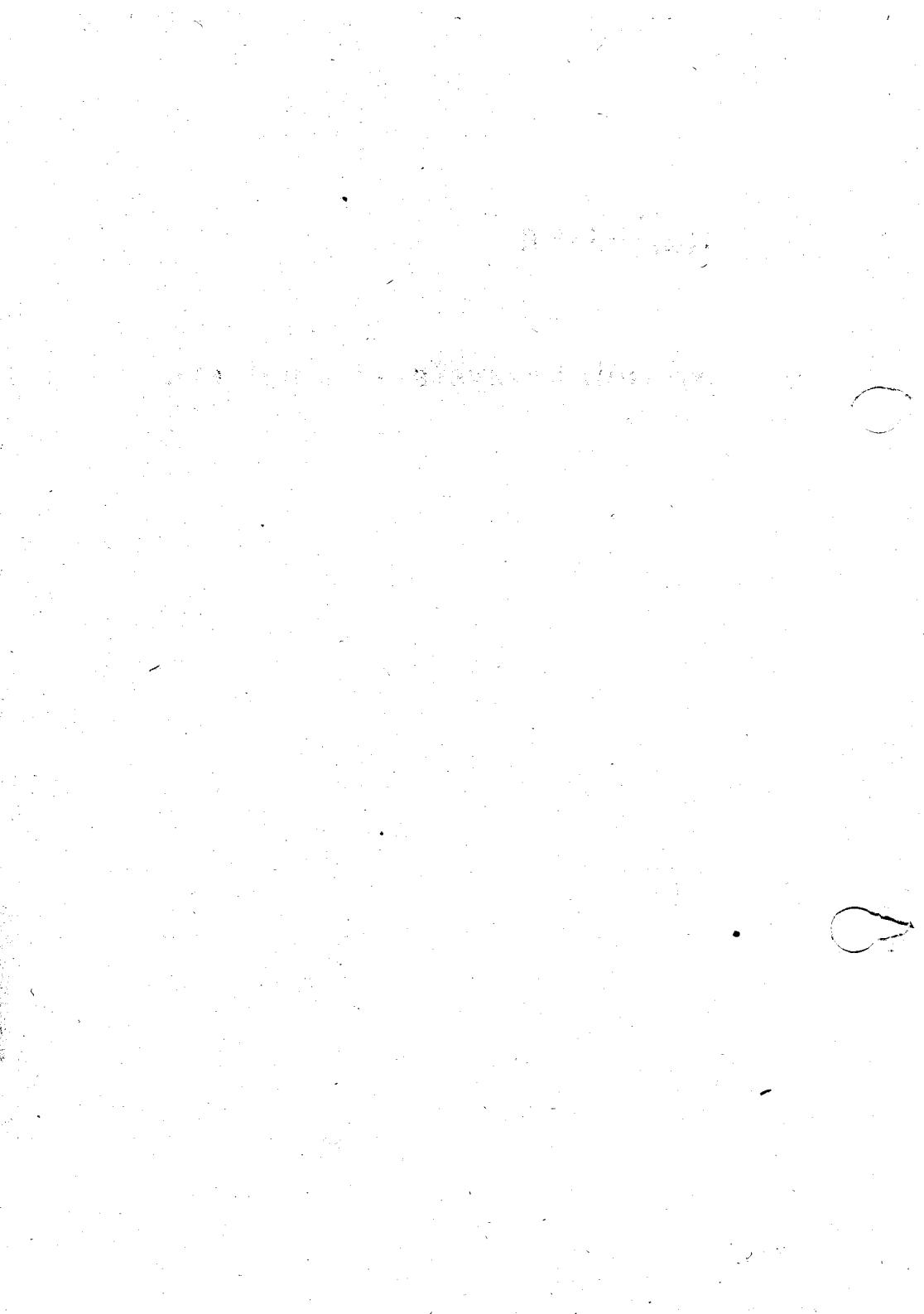
liczba REPORTS LOST

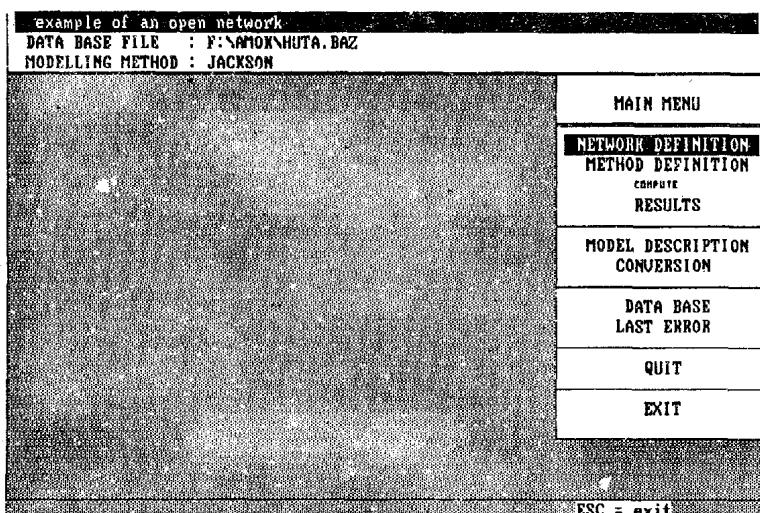
Po wyprowadzeniu któregośkolwiek z komunikatów 1÷4, konwersja zostaje zatrzymana. Nie zostaje również wygenerowany plik z danymi wejściowymi (baza danych).



Rozdział 6

Sposób korzystania z pakietu



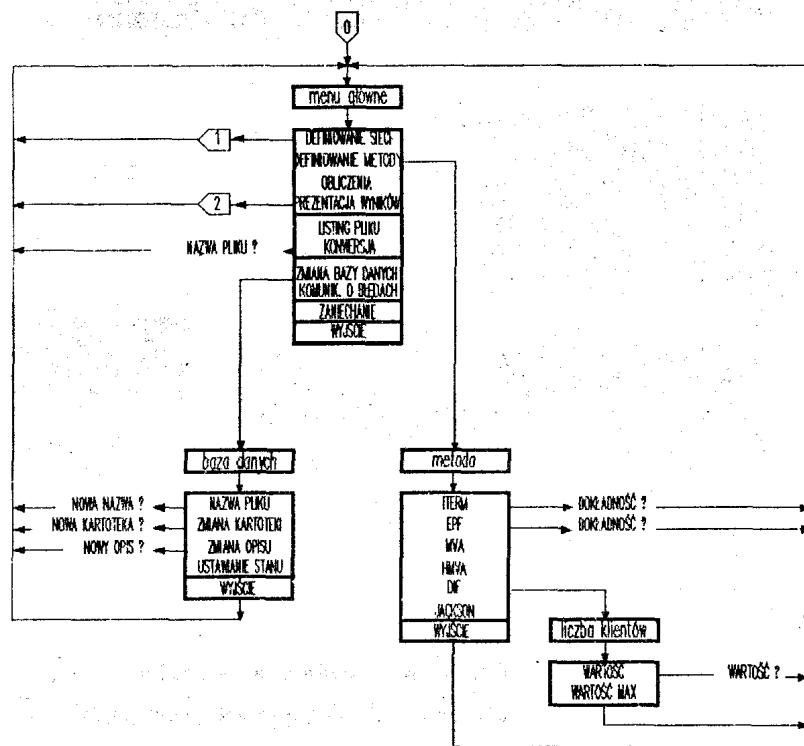


Rys. 6.1: Kopia ekranu z widocznym na nim głównym menu

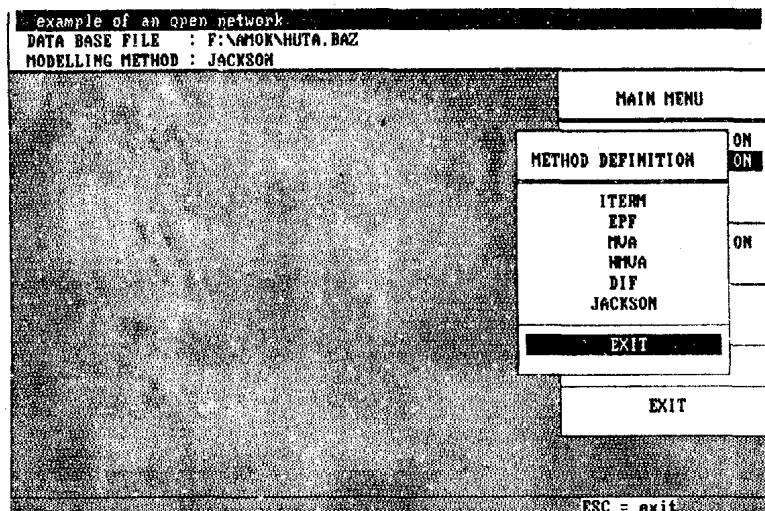
6.1 Menu główne

W pakiecie AMOK przyjęto dwa sposoby definiowania modelu. Jeden — wstawowy — wymaga opisu modelu w języku AMOKu przedstawionym w rozdziale 4, a następnie konwersji tego modelu przy pomocy konwertora. Drugi ze sposobów — interaktywny — pozwala na definiowanie nowego modelu lub dokonywanie zmian w modelu już istniejącym. Dokonuje się tego poprzez system rozwijanych menu. Po uruchomieniu pakietu (zlecienniem IA) ukazuje się na ekranie menu główne (patrz rys. 6.1), którego strukturę przedstawia rys. 6.2. Menu to umożliwia wybór jednej z następujących akcji:

- definiowanie sieci (NETWORK DEFINITION) — przygotowanie nowego lub zmiana już istniejącego modelu kolejkowego;



Rys. 6.2: Struktura menu głównego oraz powiązanych z nim menu zmiany pliku z bazą i menu definiowania metody



Rys. 6.3: Kopia ekranu z menu definiowania metody

- definiowanie metody (METHOD DEFINITION) — wybór metody, według której model będzie analizowany. Niektóre metody wymagają dodatkowych parametrów, o które system pyta użytkownika za pomocą pomocniczych okienek (rys. 6.3);
- obliczenia (COMPUTE) — uruchomienie wybranego modułu obliczeniowego;
- prezentacja wyników (RESULTS) — wybór i prezentacja w odpowiadającej użytkownikowi formie i zakresie wyświetlania otrzymanych wyników modelowania.

```

przykładowa baza danych
DATA BASE FILE : F:\AMOK\BASEFILE.BAZ
MODELLING METHOD : MVA

LISTING OF : basefile

SOUR przykładowa baza danych
*DESCRIPTION*
/STATION/ NAME = DISTTERM ! terminale zdalne !
           SERVICE(:DISTANT) = ERL(5,750)
           TRANSIT(:DISTANT) = TRANSM
           SCHEDULE = IS
/STATION/ NAME = LOCTERM ! terminale lokalne !
           SERVICE(:LOCAL) = ERL(3,500)
           SCHEDULE = IS
           TRANSIT(:LOCAL) = CPU
/STATION/ NAME = TRANSM ! linia transmisyjna !
           SERVICE = EXP(100)
           TRANSIT = [0..5] CPU,
                      [0..5] DISTTERM
/STATION/ NAME = CPU ! jednostka centralna !
           SERVICE(:DISTANT) = COX(1,6,0..5,6)
           SERVICE(:LOCAL) = COX(1,4,0..2,5)

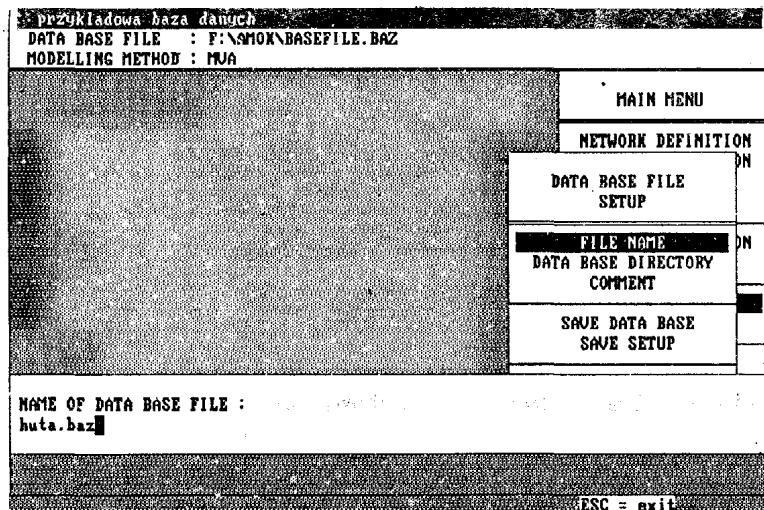
↓ = Down | PgDn = Page Down | ESC = exit

```

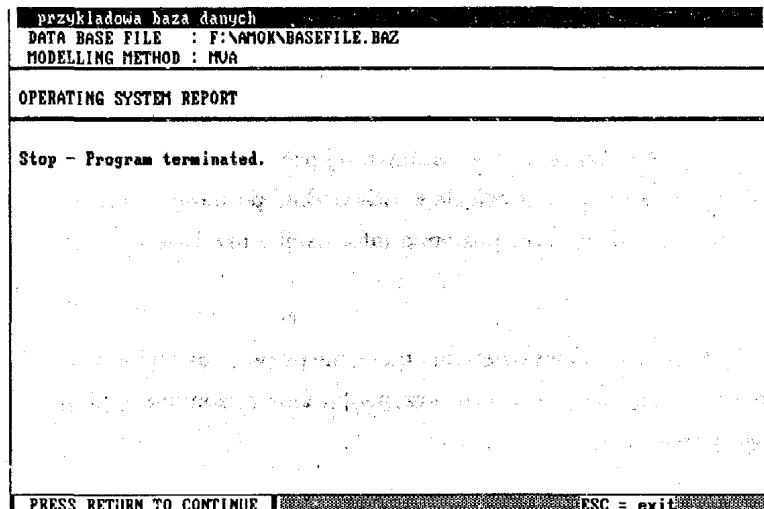
Rys. 6.4: Kopia ekranu podczas wyświetlania pliku z opisem modelu

Ponadto dodatkowe pozycje menu głównego pozwalały na podejmowanie innych akcji jak:

- wyświetlanie zawartości pliku tekstowego (MODEL DESCRIPTION) — w szczególności opisu modelu w języku AMOKu (rys. 6.4);
- konwersja pliku z opisem modelu (CONVERSION) — uruchomienie konwertora przekształcającego opis modelu w języku AMOKu na bazę danych;
- zmiana bazy danych (DATA BASE) — wybór innego pliku z bazą danych. Jest to równoznaczne ze zmianą modelu (rys. 6.5);
- wyświetlenie ostatniego komunikatu modułu obliczeniowego lub konwertora (LAST ERROR) — rys. 6.6.



Rys. 6.5: Kopia ekranu podczas zmiany pliku z bazą danych



Rys. 6.6: Kopia ekranu podczas wyświetlania ostatniego komunikatu

Oprócz tego AMOK oferuje dwa sposoby zakończenia pracy:

- z możliwością zrezygnowania z dokonanych zmian (QUIT) — w tym przypadku jest wymagane potwierdzenie użytkownika;
- normalne (END) — połączone ze schowaniem aktualnej postaci bazy danych w pliku dyskowym.

Uruchamiane poprzez menu główne podstawowe akcje AMOKu (definiowanie sieci, obliczenia oraz prezentacja wyników) realizowane są w zasadzie niezależnie od siebie. Jednakże ze względu na ich logiczne powiązanie, umożliwiono użytkownikowi bezpośrednie przechodzenie między nimi, z pominięciem menu głównego. W tym celu menu każdej akcji podstawowej ma dodatkowe pozycje, z nazwami pozostałych akcji podstawowych. Jest to jedyne odstępstwo od przyjętej w pakiecie zasady, zgodnie z którą wyjście z każdego menu niższego poziomu powoduje powrót do menu wywołującego.

W trakcie pracy użytkownik może zaniechać kontynuacji podjętych już działań. W tych menu, w których znajduje się pozycja WYJŚCIE (EXIT), naciśnięcie klawisza ESC jest równoznaczne z wybraniem tej pozycji. W menu, w których pozycja EXIT nie występuje, naciśnięcie klawisza ESC powoduje anulowanie dotychczas wprowadzonych na tym poziomie informacji i przejście do menu wyższego poziomu.

Zdarza się, że procedura wyświetlania menu nie pozwala na wybór jednej lub kilku jego pozycji, zaznaczonych w tym przypadku zmniejszonymi napisami. Istnieją dwie tego przyczyny:

1. Podjęcie działania wskazanego pozycją menu jest niemożliwe ze względu na aktualny stan modelowania lub stan samego modelu kolejkowego (np. nie

można wykonać obliczeń ani tym bardziej wyświetlić wyników przy nieistniejącym modelu kolejkowym lub nie można zmienić nazwy nieistniejącego obiektu).

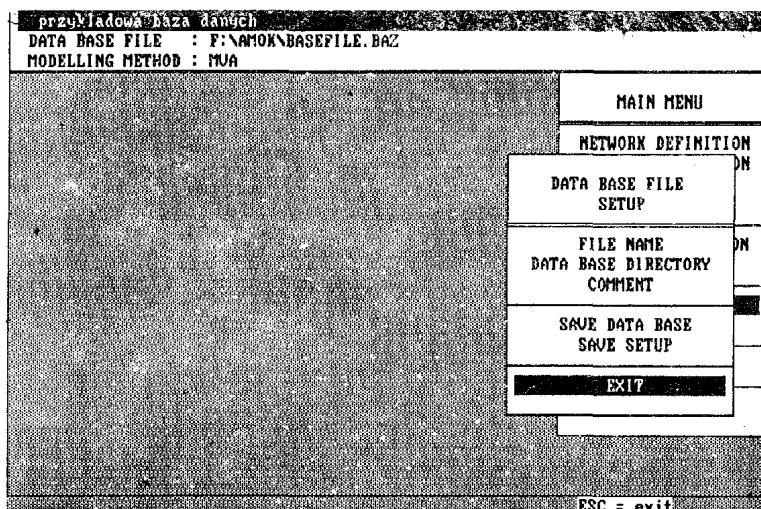
2. Podczas tworzenia modelu kolejkowego podjęcie działania wskazanego pozycją menu jest zakazane, jeśli prowadziłoby do stworzenia nieprawidłowego modelu. Postać modelu kolejkowego musi odpowiadać określonym regułom, wynikającym z jego logiki lub ograniczeń stosowanych metod analizy. Część tych reguł może być kontrolowana już podczas tworzenia modelu, inne dopiero w trakcie pracy modułu obliczeniowego.

6.2 Menu DATA BASE

Po wyborze pozycji **DATA BASE** menu głównego na ekranie konsoli rozwija się menu zmiany bazy danych (patrz rys. 6.7, struktura — rys. 6.2).

Akcje, które umożliwia powyższe menu są następujące:

- zmiana nazwy pliku z bazą danych (**FILE NAME**) — powoduje zmianę pliku z bazą danych, zmiany w poprzedniej bazie danych będą zapamiętane lub nie — zależnie od decyzji użytkownika;
- zmiana kartoteki (**DATA BASE DIRECTORY**) — przejście do nowej kartoteki z bazami danych;
- zmiana komentarza bazy danych (**COMMENT**) — tekst użytkownika będzie zapamiętyany w bazie danych (do 76 znaków);
- zapamiętanie aktualnego stanu parametrów standardowych pakietu (**SAVE SETUP**).



Rys. 6.7: Kopia ekranu zmiany bazy danych

6.3 Definiowanie sieci kolejek

Definiowanie sieci kolejek jest realizowane przez moduł DIALOG. Kompletny opis modelu kolejkowego wymaga podania następujących elementów:

- liczba i rodzaje obiektów kolejkowych (źródeł, stanowisk, podsieci);
- liczba klas klientów krążących pomiędzy obiektami oraz liczba klientów w ramach każdej klasy (jeżeli jest ona stała, tj. gdy drogi klientów należących do tej klasy są opisane zamkniętym łańcuchem);
- parametry każdego z obiektów, określające sposób zachowania się w nich klientów w funkcji typu obiektu oraz klasy klientów.

Strukturę menu definiowania sieci, klas i klientów przedstawia rys. 6.8, a menu definiowania parametrów — rys. 6.9.

6.3.1 Menu definiowania obiektów kolejkowych

Menu główne “definiowania obiektów kolejkowych” (patrz rys. 6.10) zawiera pozycje pozwalające całkowicie określić pierwszy z wymienionych trzech elementów opisu modelu kolejkowego. Są to pozycje:

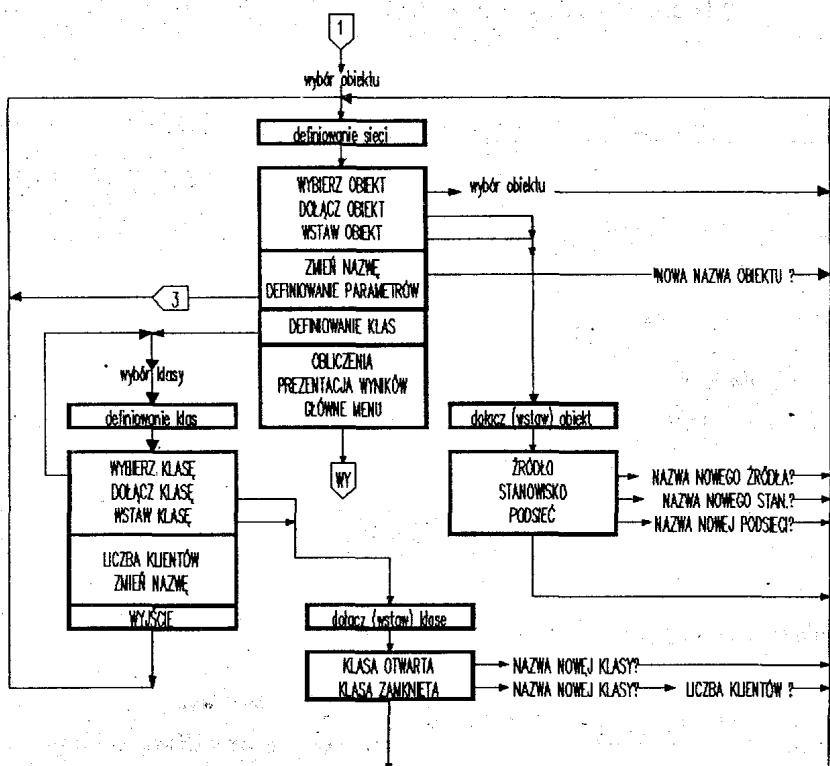
- dołącz obiekt — umożliwia dołączenie do opisu modelu kolejnych obiektów;
- wstaw obiekt — umożliwia wstawienie obiektu przed wskazany istniejący już obiekt kolejkowy.

Wybranie powyższych pozycji menu powoduje przejście do menu określającego typ obiektu (źródło, stanowisko, podsieć); następnie trzeba podać nazwę nowego obiektu (rys. 6.11). Jeżeli użytkownik nie zaniecha swego działania, to nowy obiekt o zadanym typie i nazwie zostanie włączony do sieci kolejek.

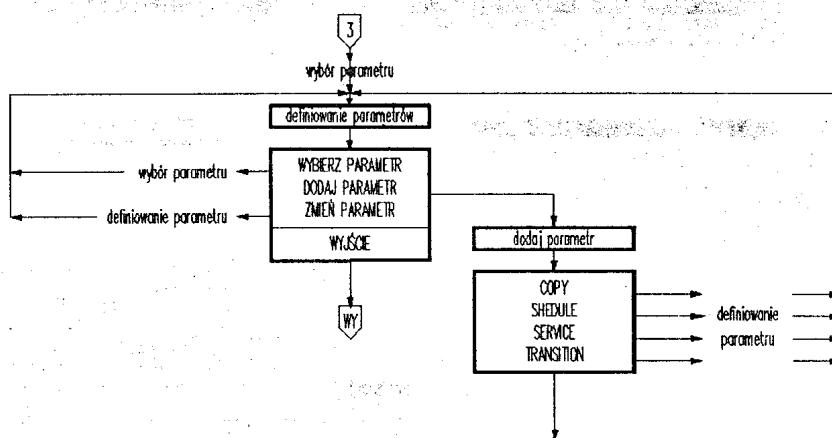
Druga z wymienionych powyżej pozycji menu wymaga wskazania obiektu, przed który chcemy wstawić nowy obiekt. Czynność tę umożliwia wybranie pozycji menu “wybierz obiekt”, która przenosi znacznik na zdefiniowane dotychczas obiekty kolejkowe, pozwalając na wskazanie jednego z nich.

Pomimo powrotu do menu głównego znacznik pozostaje na wybranym przez nas obiekcie. Dzięki temu możliwe staje się wybrane wspomnianej pozycji “wstaw obiekt” oraz dwóch innych:

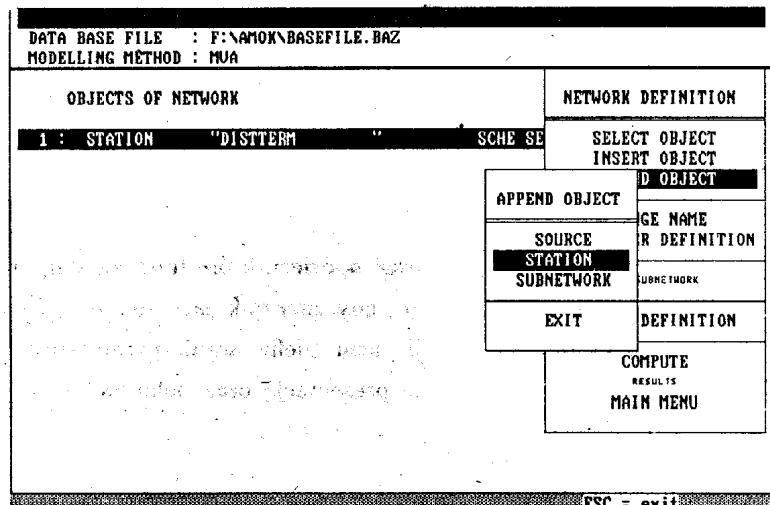
- zmień nazwę — umożliwiającej zmianę istniejącej nazwy obiektu na inną;



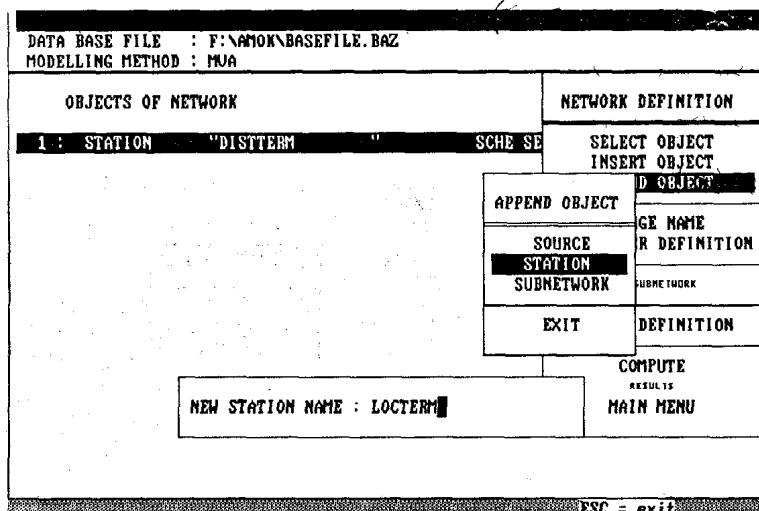
Rys. 6.8: Struktura menu definiowania sieci, klas i obiektów



Rys. 6.9: Struktura menu definiowania parametrów



Rys. 6.10: Kopia ekranu z menu definiowania obiektów kolejkowych



Rys. 6.11: Kopia ekranu podczas definiowania typu i nazwy obiektu

- definiowanie parametrów — umożliwiającej przejście do działania związanego z określeniem drugiego elementu opisu — definicji parametrów wskazanego obiektu.

W ramach sieci kolejek można definiować podsieci, które funkcjonalnie mają te same właściwości co sieć główna. Stąd, gdy znacznik zatrzymamy na obiekcie będącym podsiecią, w miejsce pozycji menu “definiowanie parametrów” ukaże się wejście do podsieci, umożliwiające prezentację oraz definiowanie obiektów zawartych w podsieci.

W ramach podsieci można tworzyć następne podsieci, przy czym głębokość tego zagnieżdżenia została ograniczona do 9. (Sieć główna ma poziom zagnieżdżenia zerowy). Tworzenie podsieci wymagało stworzenia kolejnych pozycji menu:

- wyjście z podsieci — pozwala na opuszczenie podsieci i prezentację, oraz definiowanie obiektów opuszczanej podsieci;

Inna pozycja menu

- definiowanie klas

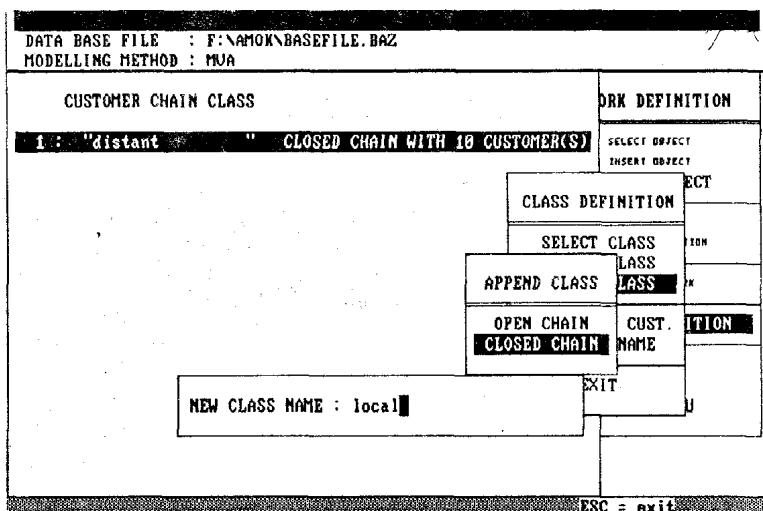
pozwala na realizację trzeciego ze wspomnianych elementów opisu modelu kolejkowego, t.j. umożliwia zdefiniowanie klas klientów krążących pomiędzy obiektami.

6.3.2 Definiowanie klas klientów

Definiowanie klas klientów jest bardzo podobne do definiowania obiektów sieci kolejek. Menu definiowania klas obejmuje kilka pozycji identycznych, odnoszących się tym razem do klasy. Są to:

- wybierz klasę — ustawia znacznik na jednej z dotychczas zdefiniowanych klas;
- wstaw klasę — wstawia nową klasę przed klasą wskazaną znacznikiem;
- dołącz klasę — umożliwia wprowadzenie pierwszej klasy bądź dołączenie nowej klasy do klas istniejących;
- zmień nazwę — pozwala zmienić istniejącą nazwę klasy na inną.

Podczas wstawiania lub dołączania klasy za pomocą innego menu definiujemy typ klasy, a następnie jej nazwę (patrz rys. 6.12). Możliwe są dwa typy klas —



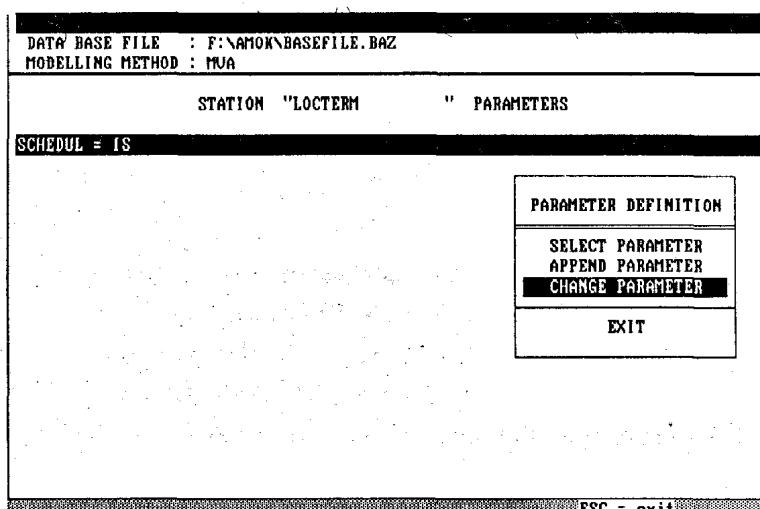
Rys. 6.12: Kopia ekranu podczas definiowania typu i nazwy klasy

otwarte lub zamknięte. W przypadku klasy zamkniętej liczba klientów jest stała i trzeba ją podać. Konieczność zapewnienia możliwości zmiany tej liczby wymaga dodatkowej pozycji menu:

- zmiana liczby klientów

pozwalająca zdefiniować nową liczbę klientów w klasie zamkniętej.

Dla uproszczenia modelowania zakłada się, że jeżeli w modelu kolejkowym krażą klienci tylko jednej klasy, to nie jest konieczne jawne definiowanie tej klasy. Klasa ta jest definiowana automatycznie w AMOKu jako tzw. "klasa bez nazwy". Typ klasy (łańcuch otwarty lub zamknięty) zależy od tego, czy w modelu kolejkowym zdefiniowano źródła. W przypadku zamkniętego łańcucha klientów zapewniona jest możliwość definiowania liczby klientów w łańcuchu. W przypadku, gdy klasa o niezdefiniowanej nazwie została już utworzona, to utworzenie następnych klas



Rys. 6.13: Kopia ekranu z menu definiowania parametrów

jest możliwe po jawnym nadaniu nazwy tej pierwszej klasie. Czynimy to wybierając polecenie "zmień nazwę".

6.3.3 Definiowanie parametrów

Menu definiowania parametrów (rys. 6.13) obiektu umożliwia następujące akcje:

- wybierz parametr — umożliwia wskazanie (poprzez usytuowanie znacznika) jednego z dotychczas zdefiniowanych parametrów obiektu celem np. jego przeddefiniowania,

- dodaj parametr — umożliwia dodanie parametru, który jeszcze nie został zdefiniowany w obiekcie (uwaga : niektóre parametry są opcjonalne i nie wymagają definiowania),
- zmień parametr — umożliwia zmianę definicji istniejącego parametru obiektu.

Zakłada się, że obiekt sieci kolejek będący stanowiskiem obsługi może być zdefiniowany czterema parametrami: COPY, SCHEDULE, SERVICE, TRANSITION; zaś źródło — trzema (COPY, SERVICE, TRANSITION). Ponadto w przypadku stanowiska obsługi parametry SERVICE i TRANSITION mogą być różne dla różnych klas klientów przechodzących przez to stanowisko — stąd może ich być po kilka w opisie. Wybranie pozycji "dodaj parametr" prowadzi do wyświetlenia powyższych czterech typów parametrów, z których możliwy jest wybór tylko tych, które są dozwolone w obiekcie i jeszcze nie zdefiniowane. W przypadku parametrów SERVICE (wzgl. TRANSITION) stanowiska obsługi, dołączenie nowych parametrów jest możliwe tylko wtedy, gdy istniejące już w opisie stanowiska parametry SERVICE (wzgl. TRANSITION) są zależne od klas klientów, co zaznacza się podczas ich definiowania.

Po wybraniu typu parametru przechodzimy do procedury jego definiowania. Sposób definiowania oraz menu są identyczne jak w przypadku wybrania pozycji "zmień parametr" z tym, że w pierwszym przypadku odnoszą się do nowoutworzonego parametru, a w drugim — do już istniejącego.

6.3.4 Wykonywanie obliczeń

Po skompletowaniu opisu sieci i określeniu metody można przystąpić do wykonania obliczeń. Dokonuje się tego poprzez wybranie z głównego menu pakietu

pozycji OBLICZENIA (COMPUTE). Następuje wówczas załadowanie i uruchomienie odpowiedniego modułu obliczeniowego. Po wykonaniu obliczeń sterowanie ponownie wraca do głównego modułu AMOKu.

	globalnie	klasa 1	klasa 2	klasa n
obiekt 1					
obiekt 2	/				
obiekt m					



– wyniki wyświetlane w momencie rozpoczęcia prezentacji

Rys. 6.14: Wielowymiarowa tablica wyników

6.4 Prezentacja wyników

W wyniku analizy modelu kolejkowego otrzymuje się zbiór wyników zaprezentowany schematycznie w tablicy na rys. 6.14. Każdy z elementów tej tablicy stanowi zbiór parametrów przedstawiony na rys. 6.15. Uzupełnieniem tego zbioru może być rozkład liczby klientów na stanowisku lub w kolejce. Moduł prezentacji wyników umożliwia wyświetlenie na ekranie wyników wybranego wiersza względnie wybranej kolumny tablicy. Wyświetlane wyniki mogą dotyczyć całego zbioru bądź wskazanego parametru. W ostatnim wypadku możliwy jest również wybór formy ich prezentacji (liczby, histogram, wykres, graf).

Po wybraniu pozycji RESULTS w menu głównym użytkownik otrzymuje na ekranie zebrane podstawowe wyniki obliczeń dla wszystkich obiektów i globalnie,

rodzaj elementu	rodzaj parametru		
stanowisko	czas reakcji	średnia l. wykonywanych operacji	przepustowość
kolejka			
kanały obsługi			

Rys. 6.15: Zbiór wyników dla jednego obiektu i jednej klasy klientów

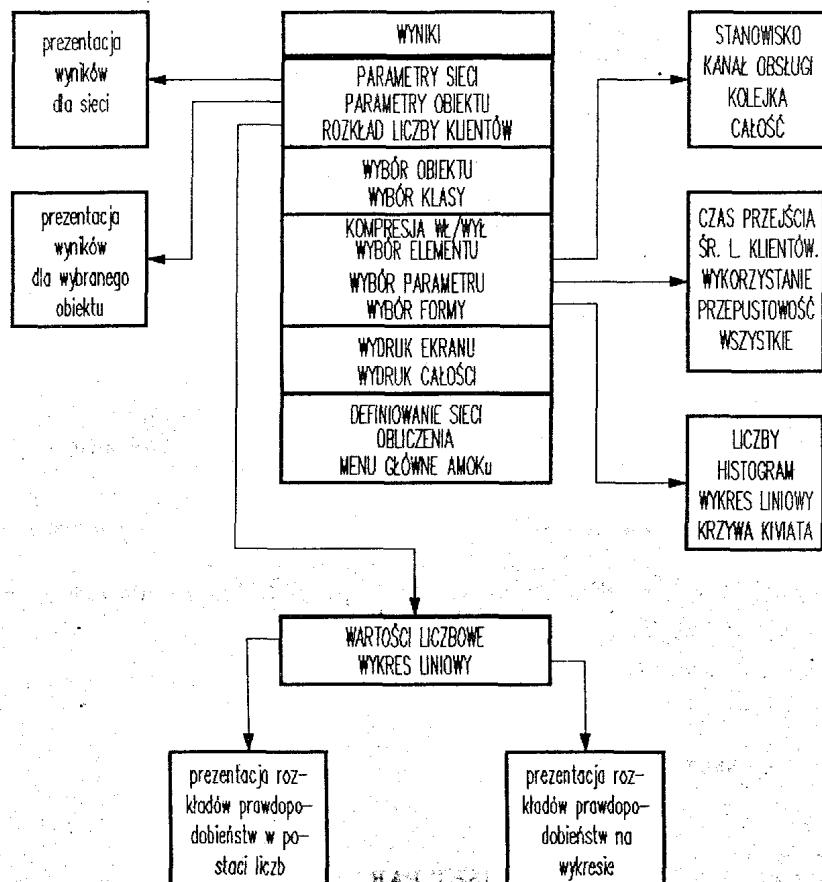
niezależnie od klas (patrz rys. 6.16). Po naciśnięciu klawisza ENTER na ekranie ukazuje się menu prezentacji wyników, którego struktura została przedstawiona na rys. 6.17.

Menu to umożliwia wybór prezentacji następujących wyników:

- parametry całej sieci (NETWORK PARAMETERS) globalnie lub dla wybranej klasy (kolumna tablicy z rys. 6.14), w formie takiej samej jak po wejściu do modułu prezentacji (rys. 6.16);
- parametry wybranego obiektu dla wszystkich klas klientów (OBJECT PARAMETERS) — wiersz tablicy z rys. 6.14;
- rozkład liczby klientów (STATE PROBABILITIES) dla tych metod, dla których jest on obliczany.

przykładowa baza danych					
		PASSAGE TIME	NB.OF CUSTOM.	UTILISATION	THROUGHPUT
NETWORK					
1. STAT. DISTTERM	7.5000E+02	1.2034E+00		1.6046E-03	
2. STAT. LOCTERM	5.0000E+02	4.1959E+00		8.3917E-03	
3. STAT. TRANSM	1.4143E+02	4.5387E-01	3.2092E-01	3.2092E-03	
	QUEUE	4.1430E+01	1.3296E-01		3.2092E-03
	SERVER	1.0000E+02	3.2092E-01	3.2092E-01	3.2092E-03
4. STAT. CPU	1.1545E+01	7.8000E-01	4.4005E-01	6.7632E-02	
5. STAT. DISK1	5.6861E+02	7.5790E+00	9.3303E-01	1.3329E-02	
	QUEUE	4.9861E+02	6.6460E+00		1.3329E-02
	SERVER	7.0000E+01	9.3303E-01	9.3303E-01	1.3329E-02
6. STAT. DISK2	5.6861E+02	7.5790E+00	9.3303E-01	1.3329E-02	
	QUEUE	4.9861E+02	6.6460E+00		1.3329E-02
	SERVER	7.0000E+01	9.3303E-01	9.3303E-01	1.3329E-02
7. STAT. DISK3	1.0051E+02	1.6046E+00	6.2202E-01	8.8860E-03	
	QUEUE	1.1051E+02	9.8199E-01		8.8860E-03
	SERVER	7.0000E+01	6.2202E-01	6.2202E-01	8.8860E-03
MORE					

Rys. 6.16: Kopia ekranu z podstawowymi wynikami obliczeń



Rys. 6.17: Menu prezentacji wyników oraz powiązanych z nim menu wyboru

Następne dwie pozycje menu pozwalają użytkownikowi ustalić:

- FOR OBJECT — (rys. 6.18) obiekt, dla którego mają być prezentowane wyniki po wyborze OBJECT PARAMETERS. Obiekt można również wybrać podczas prezentacji parametrów sieci w NETWORK PARAMETERS;
- FOR CLASS — (rys. 6.19) klasę, dla której mają być prezentowane wynik po wyborze NETWORK PARAMETERS. Klasę można również wybrać podczas prezentacji parametrów obiektu w OBJECT PARAMETERS.

Pozostała część menu prezentacji wyników zawiera funkcje pomocnicze:

- przełącznik kompresji (COMPRESS ON/OFF) — w trybie zagęszczonym usuwane są puste linie dotyczące klas, które na danym stanowisku nie występują lub stanowiska, przez które klienci danej klasy nie przechodzą;
- funkcja wyboru elementu stanowiska (SET ELEMENT), dla którego mają być prezentowane wyniki (patrz rys. 6.20):
 - stanowisko w całości (STATION),
 - kolejka (QUEUE),
 - kanał obsługi (SERVER),
 - wszystkie (ALL);
- funkcja wyboru parametru (SET PARAMETER), który ma być prezentowany (patrz rys. 6.21):
 - czas przejścia (PASSAGE TIME) — rys. 6.22,
 - liczba klientów (NO. OF CUSTOMERS) — rys. 6.23,
 - wykorzystanie (UTILISATION) — rys. 6.24,
 - przepustowość (THROUGHPUT) — rys. 6.25,
 - wszystkie parametry (ALL).

przykładowa baza danych				
DATA BASE FILE : F:\AMON\BASEFILE.BAZ				
MODELLING METHOD : MVA				
NETWORK		PASSAGE TIME	NB.OF CUSTOM. UT	
		RESULTS		
		NETWORK PARAMETERS OBJECT PARAMETERS STATE PROBABILITIES		
1. STAT. DISTTERM		7.5000E+02	1.2034E+00	
2. STAT. LOCTERM		5.0000E+02	4.1959E+00	
3. STAT. TRANSM		1.4143E+02	4.	
QUEUE	SERVER	4.1430E+01		
		1.0000E+02		
4. STAT. CPU		1.1545E+01	7.	
5. STAT. DISK1		5.6861E+02	7.	
QUEUE	SERVER	4.9861E+02		
		7.0000E+01		
6. STAT. DISK2		5.6861E+02	7.	
QUEUE	SERVER	4.9861E+02		
		7.0000E+01		
7. STAT. DISK3		1.0051E+02	1.	
QUEUE	SERVER	1.1051E+02		
		7.0000E+01		
M O R E		FOR OBJECT OF MAIN NETWORK		
		STATION "DISTTERM" "		
		STATION "LOCTERM" "		
		STATION "TRANSM" "		
		STATION "CPU" "		
		STATION "DISK1" "		
		STATION "DISK2" "		
		STATION "DISK3" "		
		STATION "DISK4" "		
		EXIT		

Rys. 6.18: Przykład ekranu z rozwiniętą listą obiektów

przykładowa baza danych			
DATA BASE FILE : F:\AMON\BASEFILE.BAZ			
MODELLING METHOD : MVA			
STAT.: CPU		PASSAGE TIME	NB.OF CUSTOM. UT
		RESULTS	
STATION GLOBAL		1.1545E+01	7.0000E-01
QUEUE GLOBAL		NETWORK PARAMETERS	
SERVER GLOBAL		OBJECT PARAMETERS	
		PROBABILITIES	
1. CLSD DISTANT		1.5047E+01	
2. CLSD LOCAL		8.9130E+00	
		FOR CLASS	
		CLOSED "DISTANT" "	
		CLOSED "LOCAL" "	
		GLOBAL	
		R OBJECT	
		OR CLASS	
		SS OFF (ON)	
		T ELEMENT	
		PARAMETER	
		PRESENTATION FORM	
		PRINT SCREEN	
		PRINT ALL	
		NETWORK DEFINITION	
		COMPUTE	
		MAIN MENU	

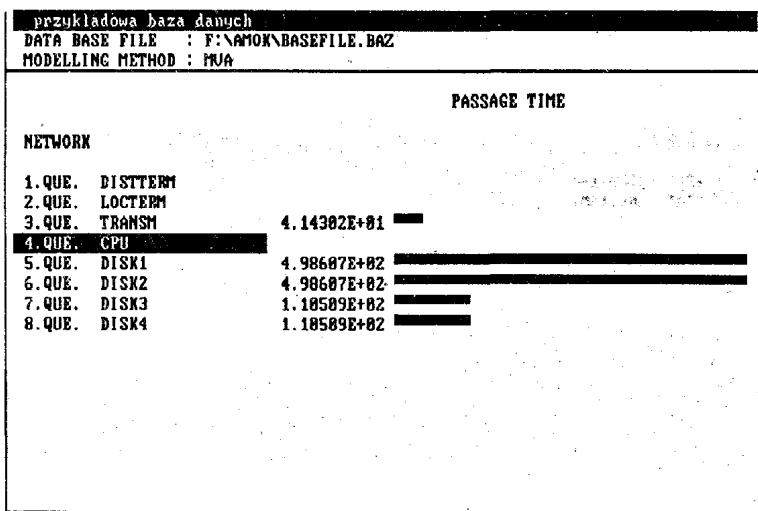
Rys. 6.19: Przykład ekranu z rozwiniętą listą klas

przykładowa baza danych			
DATA BASE FILE : F:\AMOK\BASEFILE.BAZ			
MODELLING METHOD : MUA			
	PASSAGE TIME	NB.OF CUSTOM.	RESULTS
NETWORK			
1. STAT. DISTTERM	7.5000E+02	1.2034E+00	NETWORK PARAMETERS
2. STAT. LOCTERM	5.0000E+02	4.1959E+00	OBJECT PARAMETERS
3. STAT. TRANSM	1.4143E+02	4.5387E+00	STATE PROBABILITIES
QUEUE	4.1430E+01	1.329	FOR OBJECT
SERVER	1.0000E+02	3.289	FOR CLASS
4. STAT. CPU	1.1545E+01	7.0000E+00	COMPRESS OFF (ON)
5. STAT. DISK1	5.6061E+02	7.5790E+00	SET ELEMENT
QUEUE	4.9861E+02	6.6468E+00	SET PARAMETER
SERVER	7.0000E+01	9.3303E-01	PRESENTATION FORM
6. STAT. DISK2	5.6061E+02	7.5790E+00	PRINT SCREEN
QUEUE	4.9861E+02	6.6468E+00	PRINT ALL
SERVER	7.0000E+01	9.3303E-01	
7. STAT. DISK3	1.0051E+02	1.6040E+00	NETWORK DEFINITION
QUEUE	1.1051E+02	9.8199E-01	COMPUTE
SERVER	7.0000E+01	6.2202E-01	MAIN MENU
more			

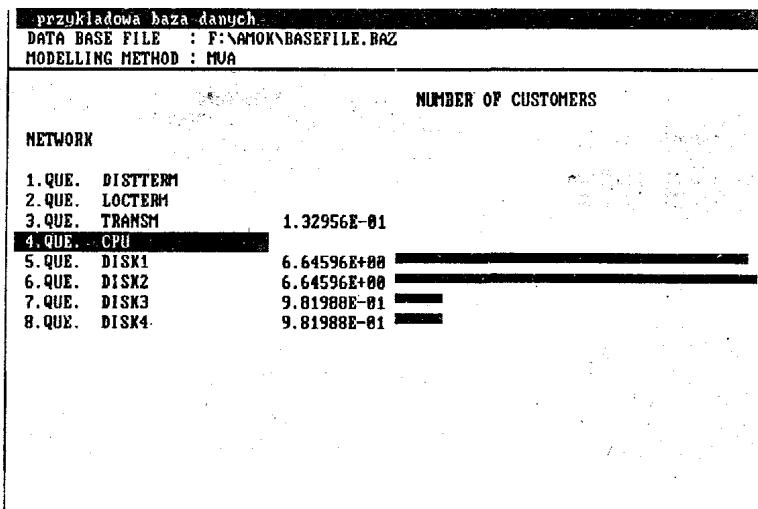
Rys. 6.20: Kopia ekranu z wyborem elementu obiektu

przykładowa baza danych			
DATA BASE FILE : F:\AMOK\BASEFILE.BAZ			
MODELLING METHOD : MUA			
	PASSAGE TIME	NB.OF CUSTOM.	RESULTS
NETWORK			
1. QUE. DISTTERM			NETWORK PARAMETERS
2. QUE. LOCTERM			OBJECT PARAMETERS
3. QUE. TRANSM			STATE PROBABILITIES
4. QUE. CPU	4.14302E+01	1.3295	FOR OBJECT
5. QUE. DISK1	4.98607E+02	6.6459	FOR CLASS
6. QUE. DISK2	4.98607E+02	6.6459	OFF (ON)
7. QUE. DISK3	1.10509E+02	9.8199	ELEMENT
8. QUE. DISK4	1.10509E+02	9.81998E-01	PRESENTATION FORM
			SET PARAMETER
			PRINT SCREEN
			PRINT ALL
			NETWORK DEFINITION
			COMPUTE
			MAIN MENU

Rys. 6.21: Kopia ekranu z wyborem parametru obiektu



Rys. 6.22: Kopia ekranu z wynikami obliczeń czasów przejścia.



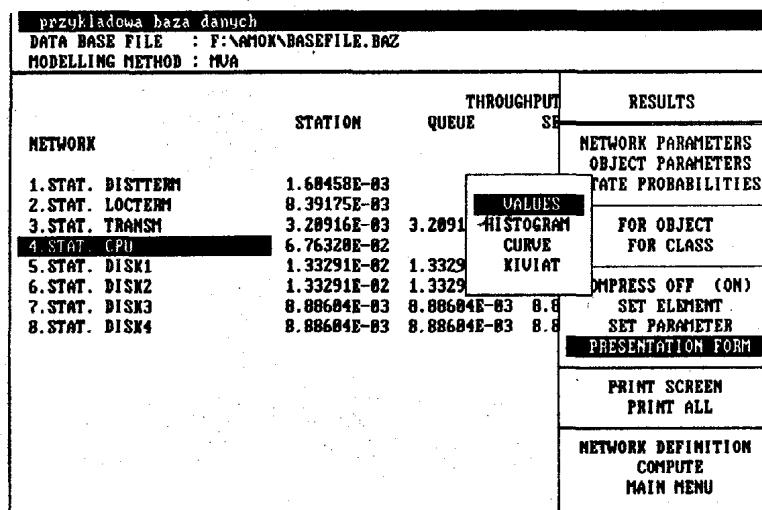
Rys. 6.23: Kopia ekranu z wynikami obliczeń śr. liczby klientów w kolejce

przykładowa baza danych			
	STATION	QUEUE	UTILISATION SERVER
NETWORK			
1. STAT. DISTTERM			
2. STAT. LOCTERM			
3. STAT. TRANSM	3.28916E-01		3.28916E-01
4. STAT. CPU	4.48853E-01		4.48853E-01
5. STAT. DISK1	9.33834E-01		9.33834E-01
6. STAT. DISK2	9.33834E-01		9.33834E-01
7. STAT. DISK3	6.22823E-01		6.22823E-01
8. STAT. DISK4	6.22823E-01		6.22823E-01

Rys. 6.24: Kopia ekranu z wynikami obliczeń wykorzystania

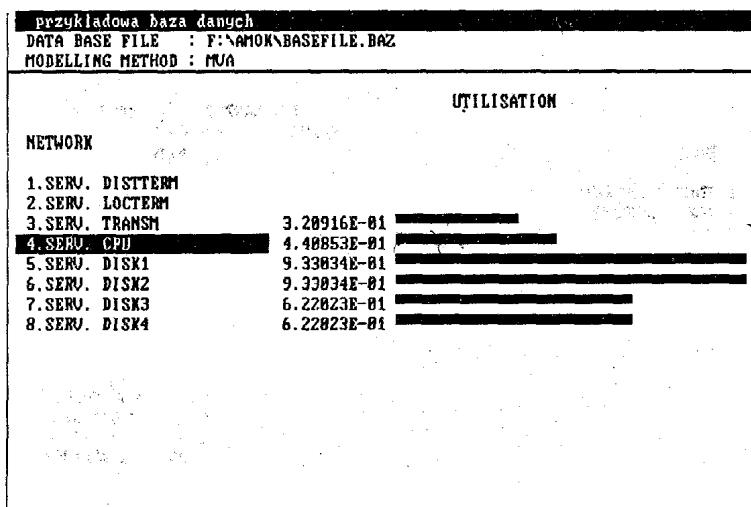
przykładowa baza danych			
	STATION	QUEUE	THROUGHPUT SERVER
NETWORK			
1. STAT. DISTTERM	1.68458E-03		1.68458E-03
2. STAT. LOCTERM	8.39175E-03		8.39175E-03
3. STAT. TRANSM	3.28916E-03	3.28916E-03	3.28916E-03
4. STAT. CPU	6.76328E-02		6.76328E-02
5. STAT. DISK1	1.33291E-02	1.33291E-02	1.33291E-02
6. STAT. DISK2	1.33291E-02	1.33291E-02	1.33291E-02
7. STAT. DISK3	8.88684E-03	8.88684E-03	8.88684E-03
8. STAT. DISK4	8.88684E-03	8.88684E-03	8.88684E-03

Rys. 6.25: Kopia ekranu z wynikami obliczeń przepustowości

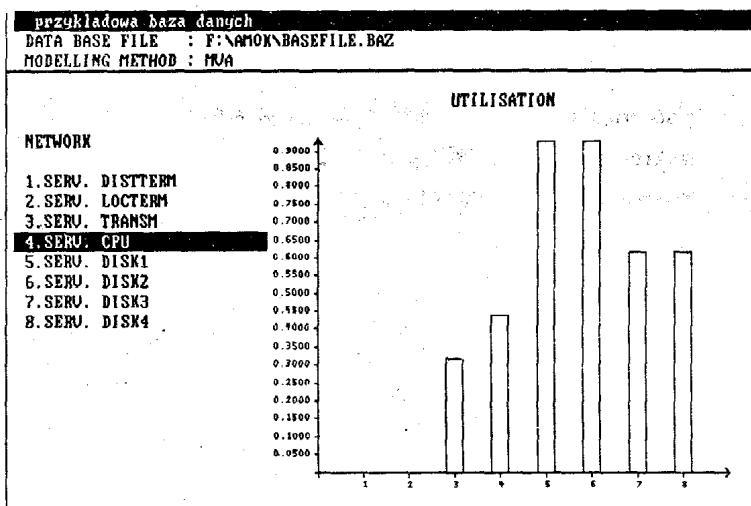


Rys. 6.26: Kopia ekranu z wyborem formy prezentacji

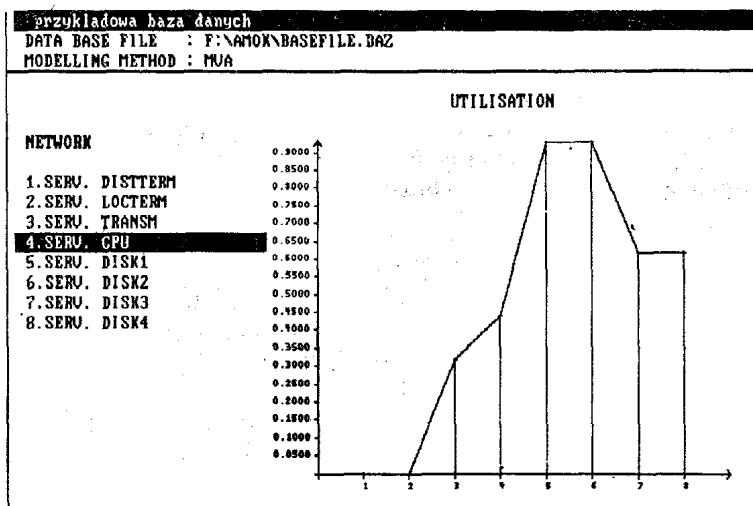
- funkcja wyboru formy prezentacji (PRESENTATION FORM), za pomocą której można wybrać jeden z następujących wariantów (rys. 6.26):
 - wartości liczbowe (VALUES) — rys. 6.27,
 - histogram (HISTOGRAM) — rys. 6.28,
 - wykres liniowy (CURVE) — rys. 6.29,
 - krzywa Kiviata (KIVIAT) — rys. 6.30;



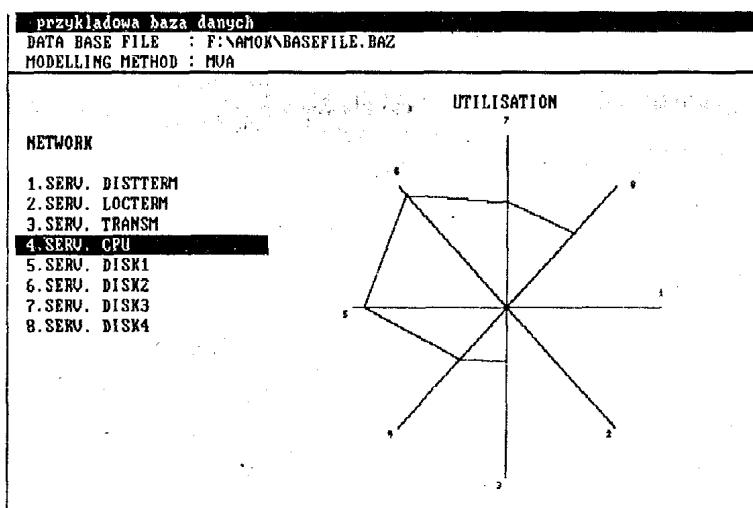
Rys. 6.27: Kopia ekranu z wynikami obliczeń w formie tabeli liczb



Rys. 6.28: Kopia ekranu z wynikami obliczeń w formie histogramu



Rys. 6.29: Kopia ekranu z wynikami obliczeń w formie wykresu



Rys. 6.30: Kopia ekranu z wynikami obliczeń w formie krzywej Kiviata

Użytkownik może również otrzymać kopię wyników na papierze (na drukarce zgodnej z IBM Graphics Printer):

- w formie kopii aktualnego ekranu (PRINT SCREEN) — patrz rys. 6.31;
- w formie wydruku wyników obliczeń wszystkich parametrów dla wszystkich klas i obiektów (rys. 6.32).

Ostatnia część menu prezentacji wyników pozwala na przejście do pozostałych głównych modułów pakietu:

- modułu definiowania sieci (NETWORK DEFINITION),
- modułu obliczeń (COMPUTE),
- głównego menu pakietu (MAIN MENU).

W każdym momencie pracy modułu prezentacji wyników użytkownik może się wyciągnąć z podjętej akcji przez naciśnięcie klawisza ESC.

CLASS: GLOBAL		PASSAGE TIME	NB. OF CUSTOM.	UTILISATION	THROUGHPUT
NETWORK					
1.STAT.CPU	2.6398E-01	1.6321E+00	6.8140E-01	6.1825E+00	
2.STAT.DISK1	2.0791E-01	8.5466E-01	4.9328E-01	4.1107E+00	
QUEUE	8.7914E-02	3.6138E-01			4.1107E+00
SERVER	1.2000E-01	4.9328E-01	4.9328E-01	4.1107E+00	
3.STAT.DISK2	2.2930E-01	3.0901E-01	2.4257E-01	1.3476E+00	
QUEUE	4.9305E-02	6.6443E-02			1.3476E+00
SERVER	1.8000E-01	2.4257E-01	2.4257E-01	1.3476E+00	

Rys. 6.31: Przykład wydruku ekranu z wynikami obliczeń

PASSAGE TIME NB.OF CUSTOM. UTILISATION THROUGHPUT				
STAT.: TERMINALE				
STATION GLOBAL	2.5136E+01	1.8204E+01		7.2422E-01
QUEUE GLOBAL				
SERVER GLOBAL				
STAT.: CPU				
STATION GLOBAL	2.6398E-01	1.6321E+00	6.8140E-01	6.1825E+00
QUEUE GLOBAL				
SERVER GLOBAL				
1.OPEN DBASE1	2.4282E-01	1.0784E+00	4.4413E-01	4.4413E+00
2.OPEN DBASE2	1.8022E-01	2.8321E-01	1.1000E-01	1.5714E+00
3.OPEN ADMIN	1.5937E+00	2.7043E-01	1.2727E-01	1.6969E-01
STAT.: DISK1				
STATION GLOBAL	2.0791E-01	8.5466E-01	4.9328E-01	4.1107E+00
QUEUE GLOBAL	8.7914E-02	3.6138E-01		4.1107E+00
SERVER GLOBAL	1.2000E-01	4.9328E-01	4.9328E-01	4.1107E+00
1.OPEN DBASE1	2.0765E-01	8.3832E-01	4.8446E-01	4.0372E+00
QUEUE	8.7651E-02	3.5386E-01		4.0372E+00
SERVER	1.2000E-01	4.8446E-01	4.8446E-01	4.0372E+00
2.OPEN ADMIN	2.2238E-01	1.6339E-02	8.8169E-03	7.3475E-02
QUEUE	1.0238E-01	7.5222E-03		7.3475E-02
SERVER	1.2000E-01	8.8169E-03	8.8169E-03	7.3475E-02
STAT.: DISK2				
STATION GLOBAL	2.2930E-01	3.0901E-01	2.4257E-01	1.3476E+00
QUEUE GLOBAL	4.9305E-02	6.6443E-02		1.3476E+00
SERVER GLOBAL	1.8000E-01	2.4257E-01	2.4257E-01	1.3476E+00
1.OPEN DBASE2	2.2917E-01	2.8810E-01	2.2629E-01	1.2572E+00
QUEUE	4.9169E-02	6.1813E-02		1.2572E+00
SERVER	1.8000E-01	2.2629E-01	2.2629E-01	1.2572E+00
2.OPEN ADMIN	2.3120E-01	2.0910E-02	1.6280E-02	9.0443E-02
QUEUE	5.1199E-02	4.6306E-03		9.0443E-02
SERVER	1.8000E-01	1.6280E-02	1.6280E-02	9.0443E-02
END AT # 0				

Rys. 6.32: Przykład wydruku wszystkich wyników obliczeń

Rozdział 7

Przykłady wykorzystania pakietu



7.1 Model systemu transakcyjnego

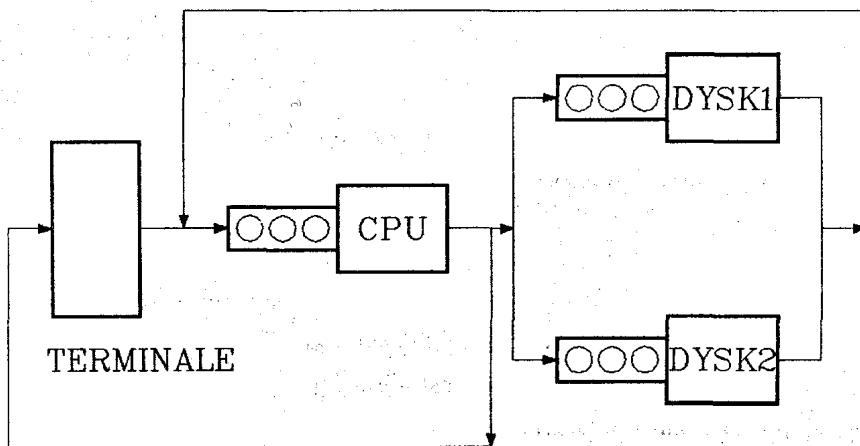
Rozpatrzmy przykład modelu wielodostępnego systemu transakcyjnego. System składa się z jednostki centralnej, dwu stacji dysków i zbioru terminali.

Przetwarzane są trzy typy transakcji. Transakcje pierwszych dwóch typów są generowane przez użytkowników: użytkownicy klasy pierwszej przechowują dane w pierwszej stacji dysków, natomiast użytkownicy klasy drugiej przechowują dane w drugiej stacji dysków. Trzeci typ transakcji składa się z jednej transakcji wykonyjącej prace administracyjne i statystyczne w odniesieniu do baz danych na obu dyskach. Transakcje pierwszego typu wymagają dostępu do danych na dysku 1 średnio 10 razy w ciągu jedego wykonania, transakcje drugiego typu odwołują się do danych na dysku 2 średnio 4-krotnie w trakcie jednego przetwarzania. Transakcja administratora systemu odwołuje się średnio 13 razy do pierwszej stacji dysków i 16 razy do drugiej stacji dysków.

Model systemu jest przedstawiony na rysunku 7.1. Terminale są reprezentowane przez jedno stanowisko opóźniające (stanowisko o nieskończonej liczbie kanałów obsługi), jednostka centralna (CPU) to stanowisko obsługujące w trybie podziału czasu, stacje dysków są modelowane przez jednokanałowe stanowiska FCFS (obsługiwany jest tylko jeden klient, a kolejność obsługi jest zgodna z kolejnością nadchodzenia do stanowiska) o nazwach DYSK1 i DYSK2.

Liczba klientów krążących w sieci jest równa liczbie aktywnych transakcji, a ta z kolei jest równa liczbie aktywnych terminali. W modelu są rozpatrywane trzy klasy klientów odpowiadające trzem typom transakcji.

Prawdopodobieństwa przejść są obliczane na podstawie średniej, względnej liczby wizyt na stanowiskach, np. dla pierwszej klasy klientów prawdopodobieństwo przejść pomiędzy CPU i dyskiem 1 jest równe $10/11 \approx 0.909$, pomiędzy CPU



Rys. 7.1: Analizowany model systemu transakcyjnego

i terminalami $1/11 \approx 0.091$.

Zakładamy, że czas namysłu w terminalu może być aproksymowany przez rozkład wykładniczy dla pierwszego typu klientów i przez rozkład Erlanga odpowiednio 3-go i 5-go rzędu dla drugiego i trzeciego typu transakcji. Odpowiednio czasy namysłu wynoszą: 20 sek, 30 sek, 120 sek.

Czasy obsługi w CPU (czasy nieprzerwanego przetwarzania transakcji) mają rozkład wykładniczy (średnia = 0.2 sek) dla pierwszego typu transakcji, Coxa (2 fazy, średnia = 0.14 sek) dla drugiego typu i Erlanga (2-go stopnia, średnia = 1.5 sek) dla transakcji administratora systemu. W systemie jest aktywnych 20 terminali. Połowa z nich generuje transakcje pierwszego, a połowa drugiego typu ($N_1 = N_2 = 10$). Czasy obsługi w dyskach są niezależne od klasy transakcji i mają rozkład wykładniczy o średniej 0.12 sek w dysku 1 i 0.18 sek w dysku 2.

Problemy do rozwiązania:

- Jakie są parametry wydajnościowe systemu dla podanych powyżej danych ?
 - Jaki jest wpływ transakcji administratora na wydajność systemu ?
 - Założymy, że:
 - przypadek a: liczba aktywnych terminali generujących transakcje pierwszego typu zostanie zmniejszona ($N_1 = 5, N_2 = 10$),
 - przypadek b: liczba aktywnych terminali pozostanie bez zmian, ($N_1 = 10, N_2 = 10$), ale jednostka centralna CPU zostanie zastąpiona przez szybszą — wszystkie czasy zostaną dwukrotnie skrócone.
- Jaki wpływ będą miały te zmiany na czas reakcji systemu ?

Rozwiązanie:

Opis badanego modelu w języku AMOKu jest przedstawiony na rysunku 7.2.

Wyniki (patrz rys. 7.3) pokazują, że system jest przeladowany, a wąskim gardlem jest jednostka centralna CPU. Współczynnik wykorzystania CPU wynosi $\varrho_{CPU} = 0.9662$, co oznacza, że jednostka centralna jest zajęta przez 96.62% czasu. Trzy typy transakcji wykorzystują, odpowiednio, 64.02%, 19.84% i 12.75% czasu CPU.

Dyski są znacznie mniej obciążone: $\varrho_{dysk1} = 0.3536, \varrho_{dysk2} = 0.2123$.

Czas reakcji systemu jest długi: $R_1 = 14.35$ sek dla transakcji pierwszego typu i $R_2 = 5.37$ sek dla drugiego typu. Przepustowość systemu tj. liczba transakcji obsłużonych w ciągu sekundy (równa przepustowości terminali) wynosi $\lambda_1 = 0.29$, $\lambda_2 = 0.28$ dla pierwszego i drugiego typu transakcji.

SOUR MODEL SYSTEMU TRANSAKCYJNEGO

DESCRIPTION

```

/STATION/ NAME = TERMINALE
          SCHEDULE = IS
          SERVICE(:DBASE1) = EXP[20]
          ! TRANSAKCJE PIERWSZEGO TYPU !
          SERVICE(:DBASE2) = ERL[3,30]
          ! TRANSAKCJE DRUGIEGO TYPU !
          SERVICE(:ADMIN) = ERL[5,120]
          ! TRANSAKCJA ADMINISTRACYJNA !
          TRANSIT = CPU

/STATION/ NAME = CPU
          SCHEDULE = PS
          SERVICE(:DBASE1) = EXP[0.2]
          SERVICE(:DBASE2) = COX[1,0.1,0.5,0.08]
          SERVICE(:ADMIN) = ERL[2,1.5]
          TRANSIT(:DBASE1) = [0.909] DISK1,[0.091] TERMINALE
          TRANSIT(:DBASE2) = [0.8] DISK2,[0.2] TERMINALE
          TRANSIT(:ADMIN) = [0.433] DISK1,
                            [0.533] DISK2,
                            [0.034] TERMINALE

/STATION/ NAME = DISK1
          SERVICE = EXP[0.12]
          TRANSIT = CPU

/STATION/ NAME = DISK2
          SERVICE = EXP[0.18]
          TRANSIT = AS(DISK1)

*END*

```

Rys. 7.2: Opis modelu transakcyjnego

Po wyeliminowaniu transakcji administratora systemu (przeliczamy model z dwoma tylko typami transakcji) wykorzystanie CPU wyniesie $\varrho_{CPU} = 0.9078$, a czasy reakcji $R_1 = 11.25$ sek i $R_2 = 4.21$ sek (patrz rys. 7.4).

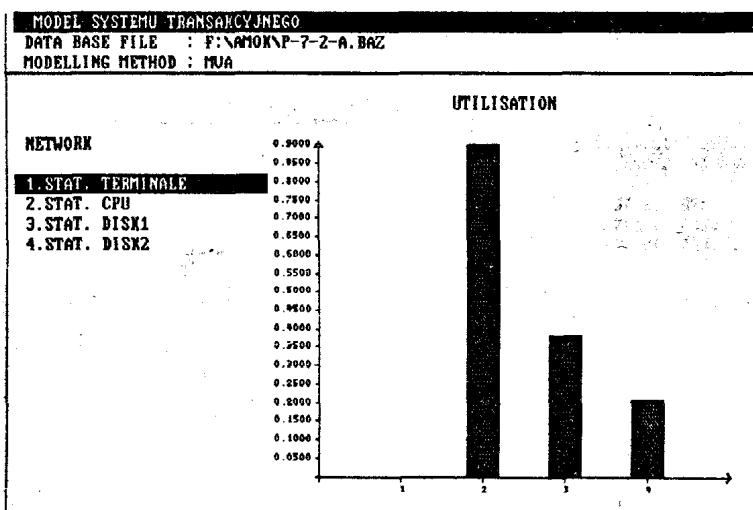
Ograniczenie liczby transakcji pierwszego typu ($N_1 = 5$) nieco poprawi sytuację: $\varrho_{CPU} = 0.8021$ ($80.21\% = 39.82\% + 21.16\% + 19.22\%$), czas reakcji spadnie do $R_1 = 7.61$ sek, $R_2 = 3.07$ sek, ale rzeczywistą poprawę da przyspieszenie CPU. W tym przypadku $\varrho_{CPU} = 0.6814$, $R_1 = 4.75$ sek, $R_2 = 1.82$ sek. Przepustowość wzrośnie do $\lambda_1 = 0.40$ i $\lambda_2 = 0.31$ transakcji/sek (patrz rys. 7.5).

MODEL SYSTEMU TRANSAKCYJNEGO					
DATA BASE FILE : F:\AMOK\NP-7-2.BAZ					
MODELLING METHOD : MUA					
PASSAGE TIME NB.OF CUSTOM. UTILISATION THROUGHPUT					
NETWORK					
1. STAT. TERMINALE	2.5408E+01	1.4680E+01			5.7776E-01
2. STAT. CPU	1.1764E+00	5.5330E+00	9.6624E-01	4.7039E+00	
3. STAT. DISK1	1.7794E-01	5.2431E-01	3.5359E-01	2.9466E+00	
QUEUE	5.7939E-02	1.7072E-01		2.9466E+00	
SERVER	1.2080E-01	3.5359E-01	3.5359E-01	2.9466E+00	
4. STAT. DISK2	2.2213E-01	2.6202E-01	2.1232E-01	1.1796E+00	
QUEUE	4.2127E-02	4.9693E-02		1.1796E+00	
SERVER	1.8000E-01	2.1232E-01	2.1232E-01	1.1796E+00	

MODEL SYSTEMU TRANSAKCYJNEGO					
DATA BASE FILE : F:\AMOK\NP-7-2.BAZ					
MODELLING METHOD : MUA					
UTILISATION					
NETWORK					
1. STAT. TERMINALE					
2. STAT. CPU	9.66243E-01	██████████			
3. STAT. DISK1	3.53591E-01	██████			
4. STAT. DISK2	2.12325E-01	████			

Rys. 7.3: Wyniki obliczeń modelu systemu transakcyjnego

MODEL SYSTEMU TRANSAKCYJNEGO				
DATA BASE FILE : F:\AMON\NP-7-2-A.BAZ				
MODELLING METHOD : MVA				
	PASSAGE TIME	NB.OF CUSTOM.	UTILISATION	THROUGHPUT
NETWORK				
1. STAT. TERMINALE	2.4774E+01	1.5168E+01		6.1226E-01
2. STAT. CPU	8.8874E-01	3.9857E+00	9.0702E-01	4.9776E+00
3. STAT. DISK1	1.8384E-01	5.0750E-01	3.8353E-01	3.1961E+00
	QUEUE	6.3843E-02	2.8485E-01	3.1961E+00
	SERVER	1.2080E-01	3.8353E-01	3.1961E+00
4. STAT. DISK2	2.2109E-01	2.5850E-01	2.1046E-01	1.1692E+00
	QUEUE	4.1093E-02	4.8846E-02	1.1692E+00
	SERVER	1.8000E-01	2.1846E-01	2.1846E-01



Rys. 7.4: Wyniki obliczeń modelu systemu transakcyjnego bez transakcji administratora

MODEL SYSTEMU TRANSAKCYJNEGO					
DATA BASE FILE : F:\AMON\P-7-2.BAZ					
MODELLING METHOD : MVA					
PASSAGE TIME NB.OF CUSTOM. UTILISATION THROUGHPUT					
NETWORK					
1. STAT. TERMINALE	2.7898E+01	1.3219E+01	4.8796E-01		
2. STAT. CPU	6.1188E-01	2.2190E+00	8.0286E-01	3.6313E+00	
3. STAT. DISK1	1.4627E-01	2.7204E-01	2.2384E-01	1.8654E+00	
QUEUE	2.6267E-02	4.8997E-02		1.8654E+00	
SERVER	1.2888E-01	2.2384E-01	2.2384E-01	1.8654E+00	
4. STAT. DISK2	2.2633E-01	2.8923E-01	2.3883E-01	1.2779E+00	
QUEUE	4.6327E-02	5.9283E-02		1.2779E+00	
SERVER	1.8888E-01	2.3883E-01	2.3883E-01	1.2779E+00	

MODEL SYSTEMU TRANSAKCYJNEGO					
DATA BASE FILE : F:\AMON\P-7-2.BAZ					
MODELLING METHOD : MVA					
STAT.: CPU					
STATION	GLOBAL	PASSAGE TIME	NB.OF CUSTOM.	UTILISATION	THROUGHPUT
QUEUE	GLOBAL	2.6398E-01	1.6321E+00	6.8148E-01	6.1825E+00
SERVER	GLOBAL				
1. CLSD	DBASE1	2.4202E-01	1.8784E+00	4.4413E-01	4.4413E+00
2. CLSD	DBASE2	1.8822E-01	2.8621E-01	1.1868E-01	1.5714E+00
3. CLSD	ADMIN	1.5937E+00	2.7843E-01	1.2727E-01	1.6969E-01

Rys. 7.5: Wyniki obliczeń modelu systemu transakcyjnego z ograniczoną liczbą transakcji

7.2 Czas czekania na remont maszyn

W zakładzie przemysłowym pracuje 20 maszyn, ulegających w sposób losowy awariom. Rozkład czasu między awariami można przybliżyć rozkładem wykładniczym o średniej 150 dni. Awarie są dwójakiego rodzaju. Awaria pierwszego typu występuje z prawdopodobieństwem 0.3 i jej naprawa w warsztacie nr 1 trwa średnio 10 dni. Awarie drugiego typu występują z prawdopodobieństwem 0.7, a ich naprawa w warsztacie nr 2 wymaga średnio 5 dni pracy. Oba rozkłady można przybliżyć rozkładem wykładniczym. Jakie jest prawdopodobieństwo, że uszkodzona maszyna będzie czekała na naprawę i ile wynosi średni czas czekania?

Zapis postawionego zadania w języku AMOKu przedstawia rys. 7.6. Wyniki obliczeń są przedstawione na następnym rysunku. Jak można z nich odczytać, prawdopodobieństwo, że warsztat nr 1 pracuje, wynosi 0.374, a prawdopodobieństwo, że drugi warsztat pracuje ma wartość 0.436, a więc prawdopodobieństwo, że maszyna będzie czekała na naprawę jest równe $0.3 \times 0.374 + 0.7 \times 0.436 = 0.4174$.

Średni czas czekania na naprawę w pierwszym warsztacie wynosi 5.3 dni, a w drugim warsztacie 3.4 dni.

SOUR CZAS CZEKANIA NA REMONT

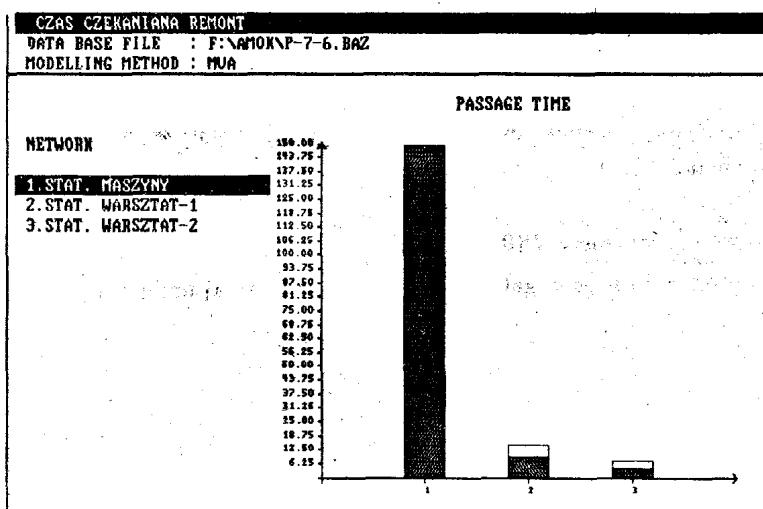
DESCRIPTION

```
/STATION/ NAME = MASZINY
           SCHEDULE = IS
           SERVICE = EXP[150]
           TRANSIT = [0.3] WARSZTAT-1, [0.7] WARSZTAT-2
/STATION/ NAME = WARSZTAT-1
           SERVICE = EXP[10]
           TRANSIT = MASZINY
/STATION/ NAME = WARSZTAT-2
           SERVICE = EXP[5]
           TRANSIT = MASZINY
```

END

Rys. 7.6: Opis modelu zakładu remontu maszyn

CZAS CZEKANIA REMONT					
DATA BASE FILE : F:\AMOK\NP-7-6.BAZ					
MODELLING METHOD : MVA					
PASSAGE TIME NB.OF CUSTOM. UTILISATION THROUGHTPUT					
NETWORK					
1. STAT. MASZYNY	1.5800E+02	1.8698E+01		1.2466E-01	
2. STAT. WARSZTAT-1	1.5307E+01	5.7241E-01	3.7397E-01	3.7397E-02	
QUEUE	5.3065E+00	1.9845E-01		3.7397E-02	
SERVER	1.0000E+01	3.7397E-01	3.7397E-01	3.7397E-02	
3. STAT. WARSZTAT-2	8.3579E+00	7.2930E-01	4.3629E-01	8.7259E-02	
QUEUE	3.3579E+00	2.9300E-01		8.7259E-02	
SERVER	5.0000E+00	4.3629E-01	4.3629E-01	8.7259E-02	



Rys. 7.7: Wyniki obliczeń modelu zakładu remontowego

7.3 Liczba łóżek w gabinecie zabiegowym po-gotowania ratunkowego

Jak świadczą zebrane dane statystyczne, do gabinetu zabiegowego w stacji po-gotowania ratunkowego kierowani są średnio trzej pacjenci w ciągu doby, a czas ich pobytu w gabinecie wynosi średnio 6 godzin. Odstępy między zgłoszeniami pacjentów mają w przybliżeniu rozkład wykładniczy, a czas ich pobytu w ambu-latorium można aproksymować rozkładem Erlanga drugiego rzędu.

Należy określić niezbędną liczbę łóżek w gabinecie zabiegowym tak, by prawdo-podobieństwo iż wszystkie łóżka będą zajęte i następnego pacjenta będzie trzeba przewieźć do szpitala w sąsiedniej miejscowości, było mniejsze od 0.1 %.

Rys. 7.8 przedstawia zapis problemu w języku AMOKu. Stanowisko PACJENCI jest tu źródłem zgłoszeń, a stanowisko LÓZKA - badanym obiektem. Zakładając, że wszystkie zgłoszenia mogą być przyjęte (stanowisko LÓZKA jest typu INFI-NITE SERVER) obliczamy prawdopodobieństwo sytuacji, że w stanowisku jest 0,1,...,10 klientów.

Suma prawdopodobieństw $P(0) + P(1) + P(2) + P(3) + P(4)$ (patrz rys. 7.9) jest większa od 0.999, a więc w gabinecie niezbędne są co najmniej 4 łóżka.

SOUR LICZBA LOZEK W GABINECIE ZABIEGOWYM***DESCRIPTION***

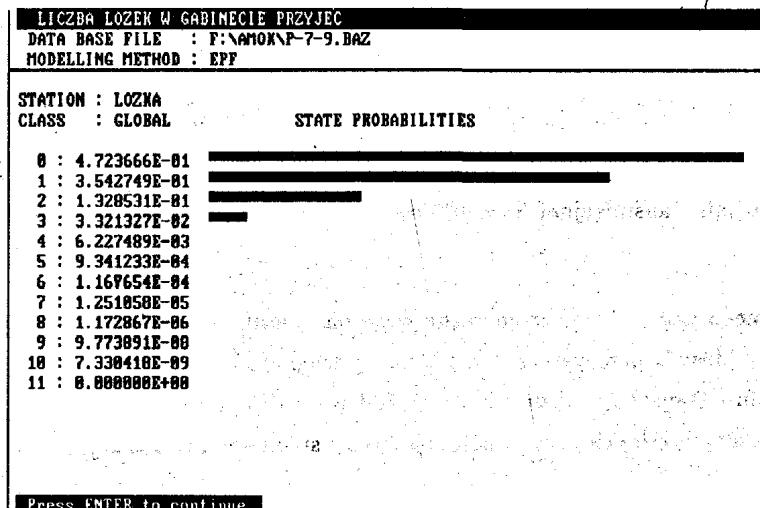
```

/STATION/ NAME = PACJENCI
          SERVICE = EXP[8]
          TRANSIT = LOZKA
/STATION/ NAME = LOZKA
          SCHEDULE = IS
          SERVICE = ERL[2,6]
          TRANSIT = PACJENCI

```

END

Rys. 7.8: Opis modelu gabinetu zabiegowego



Rys. 7.9: Wyniki obliczeń modelu gabinetu zabiegowego

7.4 Model systemu komputerowego zarządzającego bazą danych

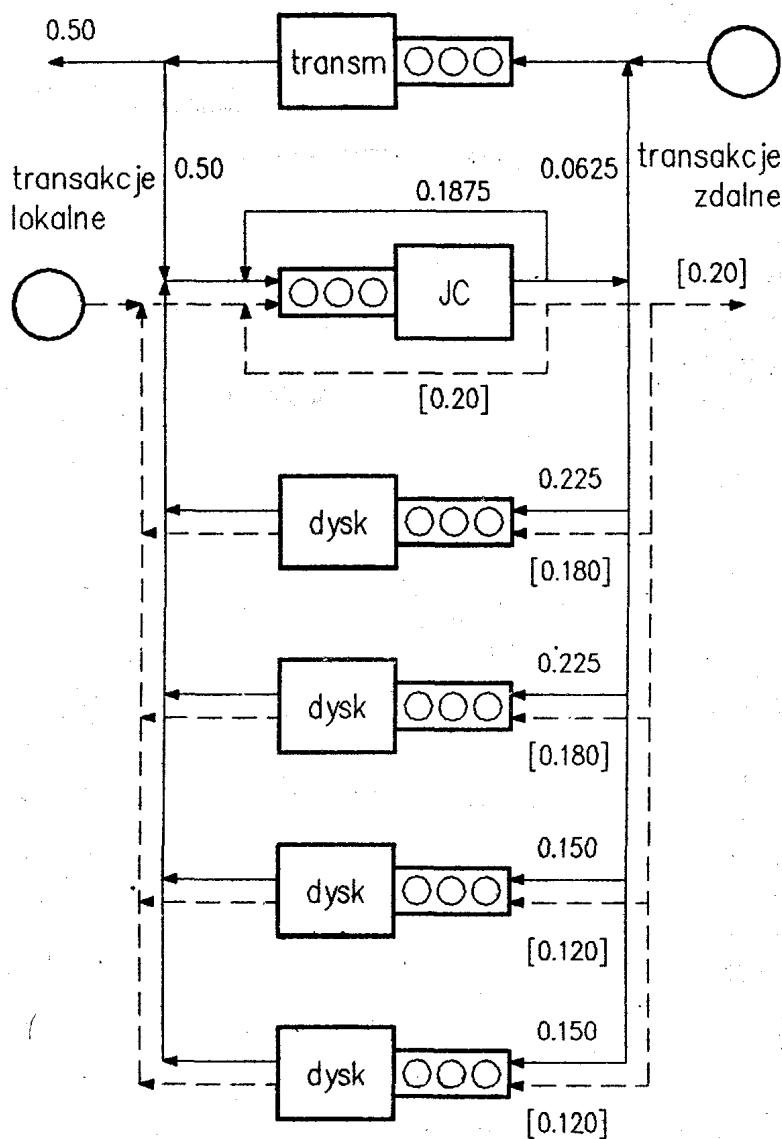
Rysunek 7.10 przedstawia model sieci otwartej sporządzony dla rzeczywistego systemu obsługującego duży zakład przemysłowy - prowadzącego ewidencję gospodarki materiałowej, energetycznej, zamówień, ich realizacji, jak również sprawy osobowe załogi. System wykonuje zadania zdalne, napiływające linią transmisyjną (pierwsza klasa klientów) i zadania lokalne, generowane z końcówek interakcyjnych przy systemie (druga klasa klientów). Transakcje obu typów różnią się częstotliwością korzystania z dysków, co ma swój wyraz w wartościach prawdopodobieństw przejścia między stanowiskami.

Zakładamy, że czasy obsługi transakcji można przybliżyć rozkładami wykładniczymi, których wartość średnia wynosi:

- dla dysków $S_1 = S_3 = S_4 = 70 \text{ ms}$;
- dla jednostki centralnej $S_5 = 6 \text{ ms}$;
- dla linii transmisyjnej $S_6 = 100 \text{ ms}$.

Drogi obiegu zadań w systemie zaznaczono na rysunku linią ciągłą dla klientów pierwszej klasy i przerywaną dla klientów drugiej klasy. Średni czas między nadejściem transakcji lokalnych $\lambda_2 = 500 \text{ ms}$. Należy zbadać pracę systemu dla powyższych danych przy zmieniającym się strumieniu transakcji zdalnych.

Zapis modelu w języku pakietu przedstawiono na rys. 7.11 i 7.12. Wykres na rys. 7.13 przedstawia średnie czasy reakcji systemu i poszczególnych stanowisk dla transakcji zdalnych, w funkcji intensywności nadchodzenia tych transakcji.



Rys. 7.10: Model systemu komputerowego

Widać, że najdłuższe czasy reakcji występują dla dysków 1 i 2 — to one są przyczyną stosunkowo długiego czasu reakcji systemu dla transakcji zdalnych. Czas reakcji systemu rośnie bardzo szybko, ze wzrostem strumienia zdalnych transakcji — podobnie jak czasy reakcji dysków 1 i 2. Wzrost czasu reakcji dysków 3 i 4 oraz linii transmisyjnej jest znacznie łagodniejszy. Dla czasów między nadejściemi transakcji zdalnych krótszych niż 300 ms (intensywności większych od 3.33 [1/s]), system traci stabilność (brak stanu stacjonarnego) za sprawą przeciążenia dysków 1 i 2 (kolejki na nich rosną teoretycznie do nieskończoności). Zwiększąc natomiast odstępy czasu między transakcjami zdalnymi, obserwujemy istotne skrócenie czasu reakcji dysków 1 i 2; przy czasach przekraczających 700 ms najbardziej obciążonym stanowiskiem w sieci staje się linia transmisyjna.

SOUR przyklad sieci otwartej

DESCRIPTION

```
/SOURCE/ NAME = DISTTERM ! terminale zdalne !
        SERVICE = EXP[500]
        TRANSIT = TRANSM:DISTANT

/SOURCE/ NAME = LOCTERM    ! terminale lokalne !
        SERVICE = EXP[500]
        TRANSIT = CPU:LOCAL

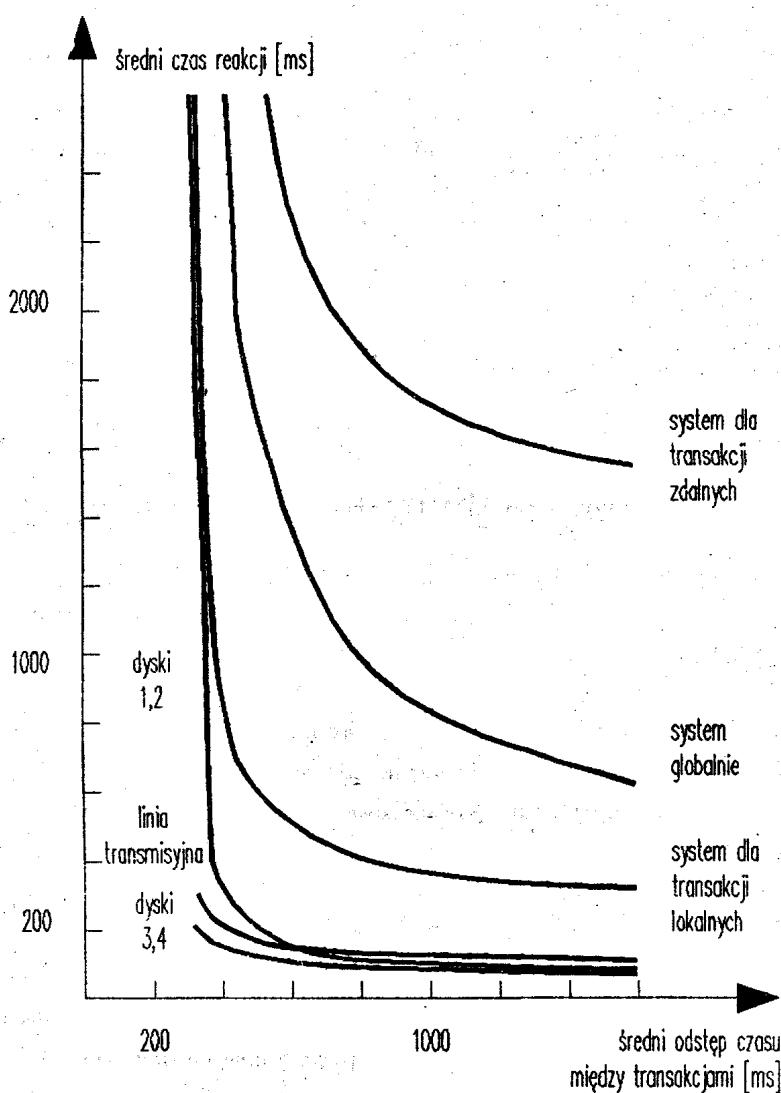
/STATION/ NAME = TRANSM    ! linia transmisyjna !
          SERVICE = EXP[100]
          TRANSIT = [0.5] CPU,
                      [0.5] OUT

/STATION/ NAME = CPU      ! jednostka centralna !
          SERVICE(:DISTANT) = COX[1,6,0.5,6]
          SERVICE(:LOCAL) = COX[1,4,0.2,5]
          SCHEDULE = PS
          TRANSIT(:DISTANT) = [0.1875] CPU,[0.0625] TRANSM,
                                [0.225] DISK1,[0.225] DISK2,
                                [0.15] DISK3,[0.15] DISK4
          TRANSIT(:LOCAL) = [0.2] CPU,[0.18] DISK1,
                            [0.18] DISK2,[0.12] DISK3,
                            [0.12] DISK4,[0.2] OUT
```

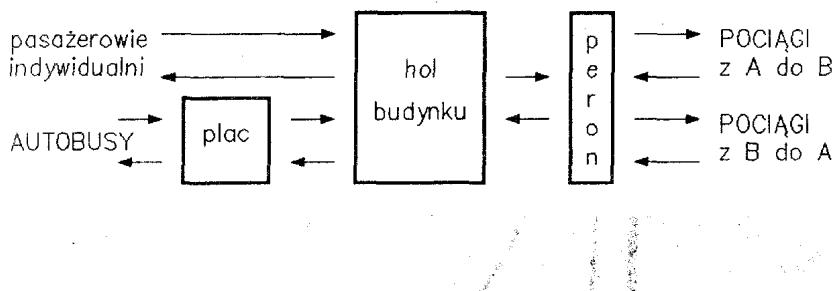
Rys. 7.11: Zapis modelu systemu komputerowego w języku AMOKu

```
/STATION/ NAME = DISK1
          SERVICE = EXP[70]
          TRANSIT = CPU
/STATION/ NAME = DISK2
          COPY = DISK1
/STATION/ NAME = DISK3
          COPY = DISK1
/STATION/ NAME = DISK4
          COPY = DISK1
*END*
```

Rys. 7.12: Zapis modelu systemu komputerowego w języku AMOKu
(dokończenie)



Rys. 7.13: Średnie czasy reakcji systemu i poszczególnych stanowisk dla transakcji zdalnych w funkcji intensywności nadejść tych transakcji



Rys. 7.14: Ruch pasażerów na przystanku przesiadkowym

7.5 Analiza przepustowości przystanku przesiadkowego podmiejskiej linii kolejowej

Wprowadzenie

Przedmiotem analizy jest ruch pieszych i środków komunikacji na przystanku przesiadkowym kolejowo-autobusowym, jednym z węzłów kolejowego ruchu podmiejskiego. Do przystanku przesiadkowego, usytuowanego w obrębie osiedla mieszkaniowego i zakładów pracy, przybywają ludzie pieszo lub autobusami. Stąd — po ewentualnym zakupie biletu w automacie (w przypadku braku 1 letu miesięcznego) — pasażerowie udają się pociągiem podmiejskim w jednym z dwóch kierunków. Z kolei inni pasażerowie podróżują w przeciwnym kierunku, tzn. wysiadają z przyjeżdżających pociągów i kierują się w dalszą drogę pieszo lub autobusami. Opisany ruch pasażerów na przystanku przesiadkowym ilustruje rysunek 7.14.

Analiza ma na celu takie dobranie liczby automatów biletowych oraz częstotliwości kursowania autobusów, które w godzinach szczytu komunikacyjnego zapewni maksymalną przepustowość przystanku, przy jednoczesnym opty-

malnym wykorzystaniu automatów i zapełnieniu autobusów. Wyniki analizy ruchu na pojedynczym przystanku przesiadkowym wskazywać mogą również na potrzebę wprowadzenia poprawki w częstotliwości kursowania pociągów zapewniającej, podobnie jak w przypadku autobusów, maksymalną do osiągnięcia przepustowość przystanku. Właściwą częstotliwość kursowania pociągów, zapewniającą z drugiej strony optymalne ich zapełnienie, uzyskuje się w wyniku analizy wszystkich przystanków przesiadkowych podmiejskiej linii kolejowej.

Dane wejściowe

Wielkości przewidywane bądź zebrane podczas bezpośrednich obserwacji i pomiarów, charakteryzujące ruch pasażerów oraz środków komunikacji na przystanku przesiadkowym i stanowiące dane wejściowe do analizy są następujące:

- λ_a — średnia liczba pasażerów przybywających do przystanku autobusami w ciągu jednej minuty;
- λ_i — średnia liczba pasażerów przybywających do przystanku pieszo w ciągu jednej minuty;
- λ_A — średnia liczba pasażerów wysiadających z pociągów jadących w kierunku A w ciągu jednej minuty;
- λ_B — średnia liczba pasażerów wysiadających z pociągów jadących w kierunku B w ciągu na jednej minuty;
- p_a — procent pasażerów, którzy po wyjściu z pociągu udają się do autobusów;
- t_a — średni czas dojścia od stanowiska autobusów do budynku przystanku przesiadkowego;
- t_h — średni czas przejścia przez hol budynku;
- t_s — średni czas przejścia przez tunel i schody łączące budynek z peronami;

- μ_s — przepustowość schodów tzn. maksymalna liczba pasażerów, którzy opuszczają schody w ciągu jednej minuty;
- t_p — średni czas przejścia przez peron w drodze do pociągu lub z pociągu do schodów;
- t_b — średni czas zakupu biletu w automacie biletowym;
- b — liczba automatów biletowych;
- μ_b — przepustowość automatów biletowych: $\mu_b = b/t_b$;
- p_b — procent pasażerów korzystających z automatów biletowych;
- ν_a — częstotliwość kursowania autobusów wyrażona liczbą autobusów na minutę;
- T_a — średni czas oczekiwania na odjazd autobusu: $T_a = (1/\nu_a)/2$;
- z_a — liczba miejsc w autobusie: $z_a = 60$;
- μ_a — średnia liczba pasażerów, którzy mogą odjechać autobusami w ciągu jednej minuty: $\mu_a = \nu_a * z_a$;
- ν_A — częstotliwość kursowania pociągów w kierunku A wyrażona liczbą pociągów na minutę;
- ν_B — częstotliwość kursowania pociągów w kierunku B wyrażona liczbą pociągów na minutę;
- T_A — średni czas oczekiwania na odjazd pociągu w kierunku A:
$$T_A = (1/\nu_A)/2$$
- T_B — średni czas oczekiwania na odjazd pociągu w kierunku B:
$$T_B = (1/\nu_B)/2$$
- z_A — średnia liczba wolnych miejsc w pociągu w kierunku A;

- z_B — średnia liczba wolnych miejsc w pociągu w kierunku B;
- μ_A — średnia liczba pasażerów, którzy mogą odjechać pociągami w kierunku A w ciągu jednej minuty: $\mu_A = \lambda_A + \nu_A * z_A$;
- μ_B — średnia liczba pasażerów, którzy mogą odjechać pociągami w kierunku B w ciągu jednej minuty: $\mu_B = \lambda_B + \nu_B * z_B$;
- p_A — procent pasażerów odjeżdżających w kierunku A.

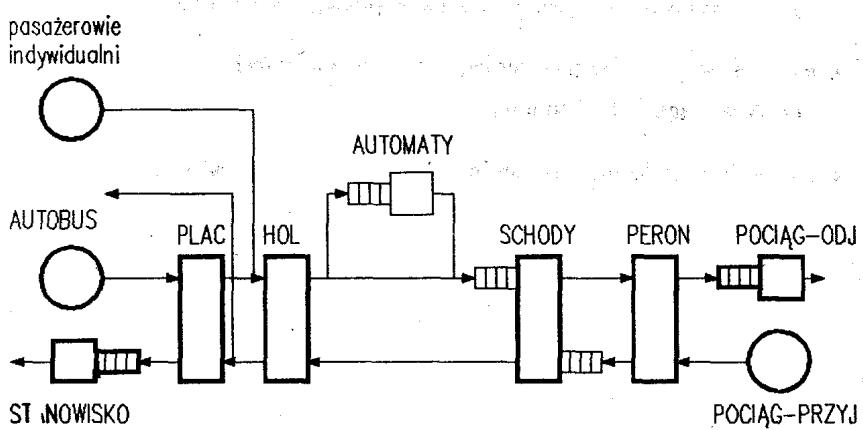
Parametry charakteryzujące przepustowość przystanku osiągane w wyniku jego analizy to:

- T — średni czas jaki tracą pasażerowie od momentu przybycia do przystanku przesiadkowego do momentu jego opuszczenia oraz elementy składowe tego czasu, w tym głównie czas tracony na czekaniu w kolejach i na odjazd;
- n_i — średnia liczba osób przebywająca na przystanku przesiadkowym w różnych jego miejscach.

Należy dążyć do minimalizacji tych parametrów, równoznacznej ze zwiększeniem przepustowości przystanku przesiadkowego w zakresie ograniczonym możliwościami przewozowymi środków komunikacji.

Model kolejkowy

Analiza problemu zostanie przeprowadzona w kategoriach teorii obsługi masowej w oparciu o zbudowany do tego celu model kolejkowy procesu, którego schemat przedstawia rysunek 7.15. Model ten stanowi zespół powiązanych ze sobą stanowisk obsługi reprezentujących poszczególne miejsca przystanku przesiadkowego. Pomiędzy stanowiskami obsługi przechodzą tzw. klienci odpowiadający w rzeczywistości poruszającym się pasażerom.

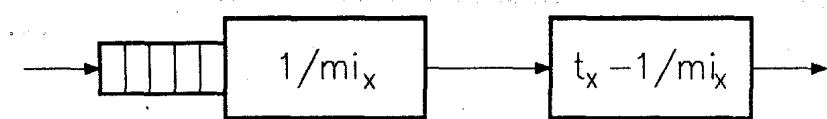


Rys. 7.15: Model kolejkowy ruchu pasażerów na przystanku przesiadkowym

Przyjmuje się dla uproszczenia, że stanowiska obsługi oznaczone na modelu jako PLAC, HOL i PERON są typu IS tzn., że są w stanie pomieścić dowolną liczbę klientów. W rzeczywistości oznacza to, że nie przewiduje się stłoczenia pasażerów w tych miejscach. Oznacza to również, że średni czas potrzebny na przejście tych stanowisk jest w zasadzie stały i niezależny od liczby klientów. Założenie to upraszczające obliczenia jest o tyle słusze, że ewentualne wydłużenie tego czasu jest w kalkulowane w czas przejścia schodami lub oczekiwania na środki komunikacji, a ponadto, jego prawdziwość można zweryfikować analizując otrzymane wynik

Stanowiska obsługi: STANOWISKO, AUTOMATY, SCHODY, POCIĄG-ODJ, opisane są dwoma parametrami:

- t_x — średnim czasem obsługi na stanowisku;
- μ_x — przepustowośćią stanowiska tzn. maksymalną średnią liczbą klientów, którzy mogą opuścić stanowisko w ciągu jednej minuty.



Rys. 7.16: Tandem dwóch stanowisk o średnim czasie obsługi t_x i przepustowości μ_x

Drugi parametr określa jednocześnie średni minimalny odstęp czasu pomiędzy kolejnymi klientami opuszczającymi stanowisko. W analizowanym modelu czas ten jest zawsze mniejszy od czasu obsługi na stanowisku.

Stanowiska obsługi o podanych wyżej właściwościach zostały w analizowanym modelu zastąpione tandemami dwóch podstawowych w sensie teorii obsługi maseowej typów stanowisk: pierwszym — z jednym kanałem obsługi, z naturalnym regulaminem szeregowania klientów i średnim czasie obsługi $\tau_{1x} = 1/\mu_x$ — oraz drugim — typu IS ze średnim czasem obsługi $\tau_{2x} = t_x - \tau_{1x}$ (patrz rys. 7.16).

Klient przybywający do pierwszego stanowiska napotyka jedynego jego kanał obsługi wolny albo zajęty przez obsługiwanyego klienta, który wcześniej zjawił się na stanowisku. W pierwszym przypadku zajmuje ten kanał obsługi i rozpoczyna obsługę, w drugim — ustawia się w kolejce i oczekuje na zwolnienie dla niego kanału. Taki regulamin obsługi powoduje, że na wyjściu stanowiska pojawiają się klienci w średnich odstępach czasu nie mniejszych niż $1/\mu_x$. Po opuszczeniu pierwszego stanowiska, klienci udają się do drugiego stanowiska tandemu — typu IS. Nieograniczona liczba kanałów obsługi tego typu stanowiska pozwala każdemu nadchodzącemu klientowi zająć kolejny wolny kanał i natychmiast rozpocząć obsługę, która trwać będzie średnio $t_x - 1/\mu_x$. Stąd sumaryczny średni czas, jaki potrzebuje klient na przejście tandemu stanowisk wynosi t_x , natomiast przepustowość tego

tandemu ograniczona przepustowością pierwszego stanowiska — μ_x . Wydłużenie czasu przejścia może być spowodowane koniecznością czekania klienta w kolejce pierwszego stanowiska tandemu odpowiadającym w rzeczywistości czekaniu w kolejce do kaszy czy automatu, spowolnieniu wynikającym z zatłoczenia na schodach i tunelu łączącym perony z holiem budynku względnie koniecznością czekania na kolejny środek transportu w sytuacji, gdy aktualny jest już zapelniony do granic.

Rozwiążanie modelu

Analizowany model kolejkowy przystanku przesiadkowego został zapisany w języku AMOKu. Postać zapisu tego modelu przedstawiona została na rys. 7.17, 7.18, 7.19.

SOUR MODEL KOLEJKOWY PRZYSTANKU PRZESIADKOWEGO***DESCRIPTION*****! ----- ZRODŁA Klientow -----****/SOURCE/ NAME = AUTOBUS****SERVICE = EXP [0.04] ! 1/LAMBDA_MALE-A****TRANSIT = PLAC : DO****/SOURCE/ NAME = INDYWIDUALNI****SERVICE = EXP [0.05] ! 1/LAMBDA_I****TRANSIT = HOL : DO****/SOURCE/ NAME = POCIAG-PRZYJ****SERVICE = EXP [0.033] ! 1/LAMBDA_A +****! 1/LAMBDA_B****TRANSIT = PERON : Z****! ----- MIEJSCA PRZYSTANKU PRZESIADKOWEGO -----****/STATION/ NAME = STANOWISKO****SERVICE = EXP [0.033] ! 60 MIEJSC****! CO 2 MINUTY****TRANSIT = STANOWISKO--****/STATION/ NAME = STANOWISKO--****SCHEDULE = IS****SERVICE = EXP [0.967] ! AUTOBUS****! CO 2 MIN****TRANSIT = OUT**

Rys. 7.17: Opis modelu przystanku kolejowego w języku AMOKu

```
STATION/ NAME = PLAC
    SCHEDULE = IS
    SERVICE = EXP [ 0.8 ]           ! MALE-T_MALE-A
    TRANSIT ( DO ) = HOL
    TRANSIT ( Z ) = STANOWISKO

/STATION/ NAME = HOL
    SCHEDULE = IS
    SERVICE = EXP [ 0.4 ]           ! MALE-T_MALE-H
    TRANSIT ( DO ) = [ 0.8 ] SCHODY ,
                                    [ 0.2 ] AUTOMATY ! MALE-P_MALE-B
    TRANSIT ( Z ) = [ 0.65 ] PLAC ,   ! MALE-P_MALE-A
                                    [ 0.35 ] OUT

/STATION/ NAME = AUTOMATY
    SERVICE = EXP [ 0.067 ]          ! 3 AUTOM.
                                         ! PO 12 SEK.JEDEN
    TRANSIT = AUTOMATY--
```

/STATION/ NAME = AUTOMATY--

```
    SCHEDULE = IS
    SERVICE = EXP [ 0.133 ]          ! 12 SEKUND ZAKUP BILETU
    TRANSIT = SCHODY
```

Rys. 7.18: Opis modelu przystanku kolejowego w języku AMOKu (kontynuacja)

```
/STATION/ NAME = SCHODY
    SERVICE = EXP [ 0.01 ] ! 100 NA MINUTE
    TRANSIT = SCHODY--
/STATION/ NAME = SCHODY--
    SCHEDULE = IS
    SERVICE = EXP [ 0.6 ] ! OK 40 SEKUND SCHODAMI
    TRANSIT ( DO ) = PERON
    TRANSIT ( Z ) = HOL
/STATION/ NAME = PERON
    SCHEDULE = IS
    SERVICE = EXP [ 0.1 ]
    TRANSIT ( DO ) = POCIAG-ODJ
    TRANSIT ( Z ) = SCHODY
/STATION/ NAME = POCIAG-ODJ
    SERVICE = EXP [ 0.02 ] ! 250 MIEJSC CO 5 MINUT
    TRANSIT = POCIAG-ODJ--
/STATION/ NAME = POCIAG-ODJ--
    SCHEDULE = IS
    SERVICE = EXP [ 2.48 ] ! POCIAG CO 5 MINUT
    TRANSIT = OUT
*END*
```

Rys. 7.19: Opis modelu przystanku kolejowego w języku AMOKu (dokończenie)

	liczba automatów				
	1	2	3	4	5
średni czas zakupu biletów w automacie (w minutach)	liczba przybywających pasażerów	1.1	0.3	0.24	0.17
średnia liczba oczekujących do automatów pasażerów	większa od przepustowości automatów	9.9	2.7	2.16	2.00

Tab. 7.1: Zależność czasu czekania w kolejce do automatu i liczby oczekujących pasażerów od liczby automatów

Spośród wielu metod analizy modeli kolejkowych dostępnych w pakiecie AMOK została wybrana metoda Jacksona, odpowiednia dla przedstawionego problemu. Ograniczenia metody wymagały zastosowania rozkładów wykładniczych do zdefiniowania parametrów określających odstępy czasowe między zdarzeniami.

Wyniki

Model został przeliczony dla wielu wariantów wartości następujących parametrów:

- liczby automatów;
- częstotliwości kursowania autobusów i pociągów.

Ważniejsze otrzymane wyniki w funkcji powyższych parametrów zostały przedstawione w tabelach 7.1, 7.2 i 7.3.

	częstotliwość kursowania autobusów co				
	1min	2min	3min	4min	5min
średni czas oczekiwania (w minutach)	0.47	1.08	liczba przybywających do stanowiska pasażerów większa od możliwości		
średnia liczba pasażerów	20.32	47.00	odwiezienia ich autobusami		

Tab. 7.2: Zależność czasu oczekiwania na autobus i liczby oczekujących pasażerów od częstotliwości kursowania autobusów

	średni czas oczekiwania	średnia liczba pasażerów	współczynnik zajętości
pociąg	5.36 min	120.6	90%
autobus	1.08 min	23.5	71%
schody	0.65 min	50.6	82%
automaty	0.3 min	2.7	60%

Tab. 7.3: Średnie czasy oczekiwania, średnie liczby pasażerów, stopnie zajętości (autobus co 2 minuty, 3 automaty)

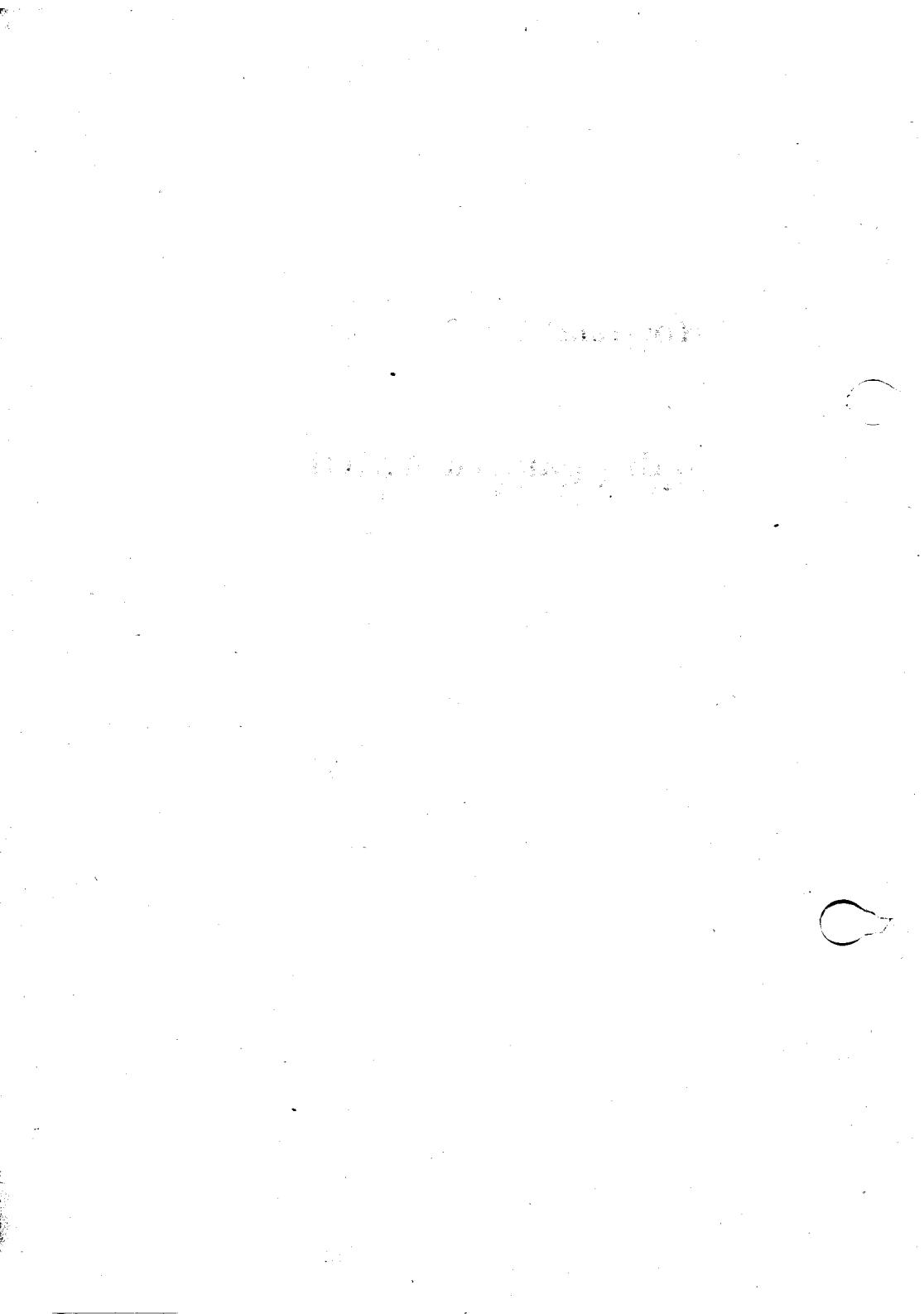
Wnioski

Z analizy wynika, że optymalna liczba automatów biletowych wynosi 3. Powyżej tej liczby nie widać istotnych zmian w zmniejszeniu czasu oczekiwania pasażerów w kolejach po zakup biletu. Częstotliwość kursowania autobusów powinna być ustalona na poziomie — co 2 minuty jeden autobus. Zapewnia to jednocześnie ich średnie zapełnienie w 70 %. Dalsze zmiany ustalanych w wyniku analizy parametrów nie wpływają znacznie na poprawę przepustowości, gdyż wąskim gardłem stają się schody. Analiza modelu wskazuje również na potrzebę zwiększenia częstotliwości kursowania pociągów.

Osiągnięte metodą analitycznymi ocen optymalne parametry przepustowości przystanku przesiadkowego powinny być następnie zweryfikowane na drodze symulacji, która pozwoli przybliżyć model do rzeczywistości i ocenić dokładność metody analitycznej. Dzięki dotychczasowej analitycznej analizie modelu konieczność przeprowadzenia symulacji ogranicza się tylko do jednego konkretnego, otrzymanego w wyniku tej analizy przypadku. Zastosowanie pakietu AMOK daje tu istotne zmniejszenie czasu niezbędnego do analizy problemu i tym samym obniżkę kosztów przeprowadzanych badań.

Rozdział 8

Jądro pakietu AMOK



8.1 Przegląd metod analizy

Powtórzmy raz jeszcze, że modelowanie w AMOKu polega na obliczeniu wskaźników efektywności pracy charakteryzujących stan ustalony sieci stanowisk obsługi o dowolnej topologii.

Należy obliczyć mierniki efektywności pracy systemu: *średnia długość kolejki* przed każdym stanowiskiem, *średni czas czekania w kolejce, przepustowość stanowiska, współczynnik wykorzystania stanowiska, średni czas reakcji całego systemu* a także, jeśli to możliwe, *rozkład prawdopodobieństwa* liczby zadań obecnych w każdym stanowisku.

Nie ma uniwersalnej metody rozwiązywania postawionego powyżej problemu. W zależności od przyjętych założeń co do rozkładów czasu obsługi i regulaminów szeregowania klientów w kolejkach, zadanie może być rozwiązane w sposób analityczny (dokładny lub przybliżony) — albo też jego analityczne rozwiązanie nie jest możliwe i trzeba stosować *symulację cyfrową*.

Sieci stanowisk obsługi możemy ogólnie podzielić na dekomponowalne i niedekomponowalne. W uproszczeniu, sieć dekomponowalna składa się ze stanowisk spełniających jeden z warunków:

1. rozkład czasu obsługi wykładniczy, identyczny dla wszystkich klas; regulamin kolejki naturalny (FIFO),
2. rozkład czasu obsługi wykładniczy, hiperwykładniczy, Erlanga lub Coxa; regulamin szeregowania klientów stosowy z wywłaszczeniem (LIFO-PR) lub stanowisko obsługi bez kolejki (IS lub PS) (dokładne definicje rozkładów i regulaminów kolejki znajdują się w rozdziale 3.2.).

Ponadto, w przypadku otwartej sieci dekomponowalnej, wejściowy strumień klientów (ze źródła) musi mieć rozkład wykładniczy. Pozostałe sieci — to sieci niedekomponowalne. Większość opracowanych metod analizy sieci daje dla sieci dekomponowalnych rozwiązanie dokładne, zaś dla niedekomponowalnych — przybliżone.

Moduły obliczeniowe AMOKu odpowiadają metodom analitycznym. Modułu symulacyjnego wersja 2.xx AMOKu nie posiada. Metody analityczne są metodami dokładnymi lub przybliżonymi. Metody dokładne są zawarte w modułach JACKSON i MVA; są one związane z najogólniejszym modelem analitycznym, zwanym od nazwisk autorów modelem BCMP [1]. Model ten zakłada wiele klas klientów, sieci mieszane (otwarte dla pewnych klas klientów, zamknięte dla pozostałych) oraz stanowiska obsługi z regulaminem FIFO, PS, IS lub LIFO-PR. W stanowiskach typu FIFO czas obsługi ma rozkład wykładniczy, jednakowy dla wszystkich klas klientów; w pozostałych stanowiskach dopuszczalny jest rozkład typu Coxa, różny dla każdej klasy. Największym problemem numerycznym modelu BCMP jest obliczanie stałej normalizacyjnej, która służy do skalowania prawdopodobieństw stanów: liczba stanów szybko wzrasta wraz ze wzrostem liczby stanowisk i liczby klientów w systemie i np. liczba stanów zamkniętej sieci 10 stanowisk, w której krąży 25 klientów przekracza 52 mln. Dlatego AMOK analizuje dwa modele o zawiązonych nieco założeniach: moduł JACKSON rozwiązuje otwartą sieć BCMP z czasami obsługi niezależnymi od liczby klientów w kolejce — w tym przypadku stała normalizacyjna wyraża się prostym wzorem; moduł MVA rozwiązuje zamkniętą sieć BCMP przy użyciu algorytmu wartości średnich [3] — który, przez zastosowanie procedury rekursywnej rozwiązującej kolejno sieć z $1, 2, \dots, N$ klientami, unika obliczania stałej normalizacyjnej.

Moduł HMVA zawiera przybliżoną wersję algorytmu MVA i jest wykorzystywany dla dużych i zawierających wielu klientów zamkniętych sieci BCMP [4,1]. Pozostałe moduły są realizacją metod przybliżonych.

Zasady metod stosowanych w poszczególnych modułach omówimy poniżej. Typy sieci analizowanych przez te moduły opisuje tabela zamieszczona w Rozdziale 2.

Wprowadźmy następujące oznaczenia, stosowane w opisie poszczególnych metod obliczeniowych:

Dla sieci:

- M — liczba stanowisk w sieci,
- N — liczba klientów wszystkich klas w sieci,
- $N = (N^1, \dots, N^K)$ — wektor populacji klientów,
- K — liczba klas klientów,
- λ — liczba klientów przychodzących do sieci w jednostce czasu.

Dla stanowiska m w sieci:

- $A_m(x), a_m(x)$ — dystrybuanta i funkcja gęstości rozkładu odstępu czasu między kolejnymi zgłoszeniami; wartość średnia tego rozkładu to $E[a_m] = 1/\lambda_m$, gdzie λ_m jest średnią liczbą klientów nadchodzących do stanowiska m w jednostce czasu; σ_{am}^2 jest wariancją, a $C_{am}^2 = \sigma_{am}^2 \lambda_m^2$ współczynnikiem zmienności rozkładu A_m ,
- $B_m(x), b_m(x)$ — dystrybuanta i funkcja gęstości rozkładu czasu obsługi; wartość średnia jest równa $E[b_m] = s_i = 1/\mu_m$, gdzie μ_m jest współczynnikiem intensywności obsługi — średnią liczbą klientów, jaką mogłoby obsłużyć stanowisko w jednostce czasu, gdyby pracowało bez przerwy; σ_{bm}^2 jest wariancją, a $C_{bm}^2 = \sigma_{bm}^2 \mu_m^2$ współczynnikiem zmienności rozkładu B_m ,

- n_m — liczba klientów na stanowisku m ,
- λ_m — przepustowość — średnia liczba klientów obsłużonych w jednostce czasu,
- e_m — średnia liczba wizyt klienta na stanowisku m , $e_m = \lambda_m / \lambda$,
- ϱ_m — współczynnik wykorzystania stanowiska — stosunek czasu, w którym stanowisko było zajęte do czasu obserwacji,
- r_{mj} — prawdopodobieństwo, że klient po zakończeniu obsługi w stanowisku m przejdzie do kolejki stanowiska j ; r_{m0} jest prawdopodobieństwem, że bezpośrednio po zakończeniu obsługi w stanowisku m klient opuści sieć; r_{0m} jest prawdopodobieństwem, że klient rozpocznie pobyt w sieci od stanowiska m ; w sieci zamkniętej $r_{0m} = r_{m0} = 0$; $R = [r_{mj}]$ nazywamy macierzą tranzycji.

W przypadku wielu klas klientów parametr odnoszących się do klasy k posiada dodatkowy indeks górny, np.:

- N^k — liczba klientów klasy k ,
- n_m^k — liczba klientów klasy k na stanowisku m ,
- s_m^k — średni czas obsługi klienta k w stanowisku m ,
- λ_m^k — przepustowość klienta klasy k na stanowisku m ,
- t_m^k — średni czas reakcji klienta klasy k na stanowisku m ,

Wartości e_i i λ_i otrzymujemy z równań ruchu klientów. Dla sieci otwartej:

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^M \lambda_j r_{ji}, \quad i = 1, \dots, M \quad (8.1)$$

w przypadku jednej klasy klientów, oraz

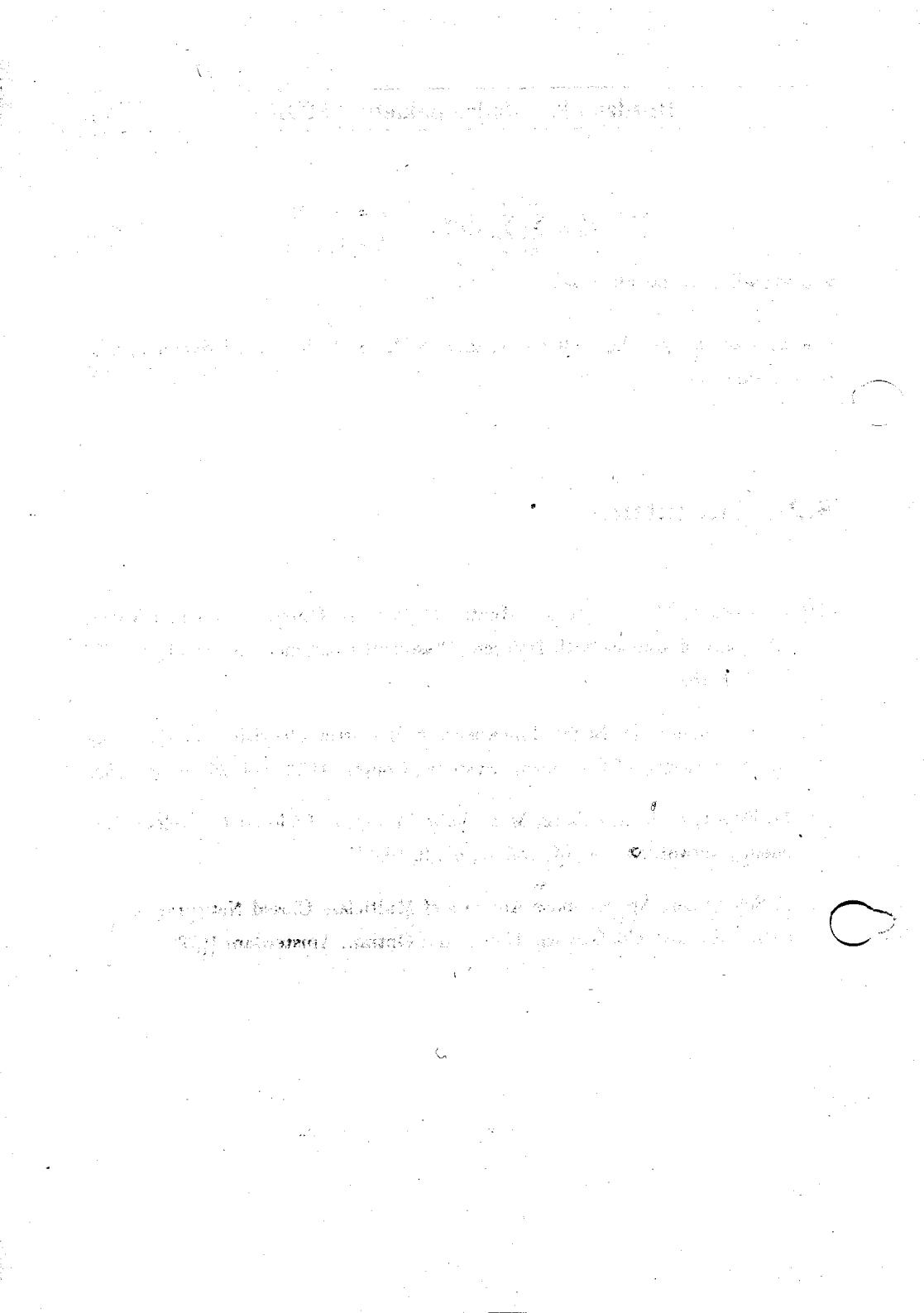
$$\lambda_i^k = \lambda_{0i}^k + \sum_{j=1}^M \sum_{l=1}^K \lambda_j^l r_{ji}^{lk}, \quad i = 1, \dots, M \\ k = 1, \dots, K \quad (8.2)$$

w przypadku K klas klientów.

Dla sieci zamkniętej $\lambda_{0i} = 0$ i równania (8.1), (8.2) nie mają jednoznacznego rozwiązania.

8.2 Literatura

- [1] F. Baskett, M. Chandy, R. Muntz, J. Palacios, Open, Closed and Mixed Networks of Queues with Different Classes of Customers, J. ACM, vol. 22, no. 2, 1975.
- [2] K. M. Chandy, D. Neuse, Linearizer: A Heuristic Algorithm for Queueing Network Models of Computing Systems, Comm. ACM, vol. 25, no. 2, 1982.
- [3] M. Reiser, S. S. Lavenberg, Mean Value Analysis of Closed Multichain Queueing Networks, J. ACM, vol. 27, no. 2, 1980.
- [4] P. Schweitzer, Approximate Analysis of Multiclass Closed Networks of Queues, Proc. Int. Conf. Stoch. Contr. and Optim., Amsterdam 1979.



Rozdział 9

Program JACKSON

86-198

86-198



9.1 Opis metody

Moduł rozwiązuje sieć otwartą typu BCMP [1]. Algorytm został opracowany zgodnie z [3]. Dla sieci otwartej równania ruchu (8.2) mają jednoznaczne rozwiązanie. Calkowita przepustowość stanowiska jest sumą przepustowości dla klas:

$$\lambda_m = \sum_{r=1}^R \lambda_m^r, \quad m = 1, 2, \dots, M,$$

a przepustowość całej sieci λ to

$$\lambda = \sum_{m=1}^M \lambda_{0m}$$

Rozkład prawdopodobieństw stanów dla stanowisk FIFO, PS i LIFO-PR można zapisać jako:

$$p_m(n_m) = g_m^{-1} n_m! \prod_{r=1}^R \frac{\varrho_m^r k_m^r}{k_m^r!}, \quad \varrho_m^r = \frac{\lambda_m^r}{\mu_m^r},$$

a dla stanowiska IS jako:

$$p_m(n_m) = g_m^{-1} \prod_{r=1}^R \frac{\varrho_m^r k_m^r}{k_m^r!}, \quad \varrho_m^r = \frac{\lambda_m^r}{\mu_m^r},$$

gdzie:

n_m — liczba klientów na stanowisku m ,

k_m^r — liczba klientów klasy r na stanowisku m ,

g_m — stała normalizacyjna stanowiska m .

Korzystając z warunku normalizacyjnego:

$$\sum_{n_m \geq 0} p_m(n_m) = 1, \quad m = 1, 2, \dots, M$$

możemy obliczyć stałe normalizacyjne.

Dla stanowisk posiadających kolejki typu FIFO, PS i LIFO-PR otrzymamy:

$$g_m = \sum_{n_m \geq 0} n_m! \prod_{r=1}^R \frac{\varrho_m^r k_m^r}{k_m^r!},$$

co można doprowadzić do postaci:

$$g_m = \sum_{i=0}^{\infty} \varrho_m^i = \frac{1}{1 - \varrho_m}, \quad \varrho_m = \sum_{r=1}^R \varrho_m^r.$$

Stałą normalizacyjną dla stanowiska IS można zapisać jako:

$$g_m = \sum_{n_m \geq 0} \prod_{r=1}^R \frac{\varrho_m^r k_m^r}{k_m^r!} = \sum_{i=0}^{\infty} \frac{\varrho_m^i}{i!} = e^{\varrho_m}, \quad \varrho_m = \sum_{r=1}^R \varrho_m^r,$$

a stała normalizacyjna dla rozpatrywanego typu sieci ma postać:

$$G = \prod_{m=1}^M g_m.$$

Na podstawie $p_m(n_m)$ oblicza się pozostałe interesujące nas parametry sieci.

9.2 Uwagi o programie

Opis sieci rozwijywanej programem JACKSON może zawierać co najwyżej:

- 30 źródeł (SOURCE),
- 30 stanowisk obsługi (STATION),
- 8 klas klientów.

Opis źródła musi spełniać następujące wymagania:

- rozkład odstępu czasu między generowanymi klientami musi być wykładniczy (EXP),
- w opisie parametru TRANSITION nie może wystąpić parametr OUT w sposób jawni ani opcjonalny.

Opis stanowiska obsługi musi zaś spełniać wymaganie:

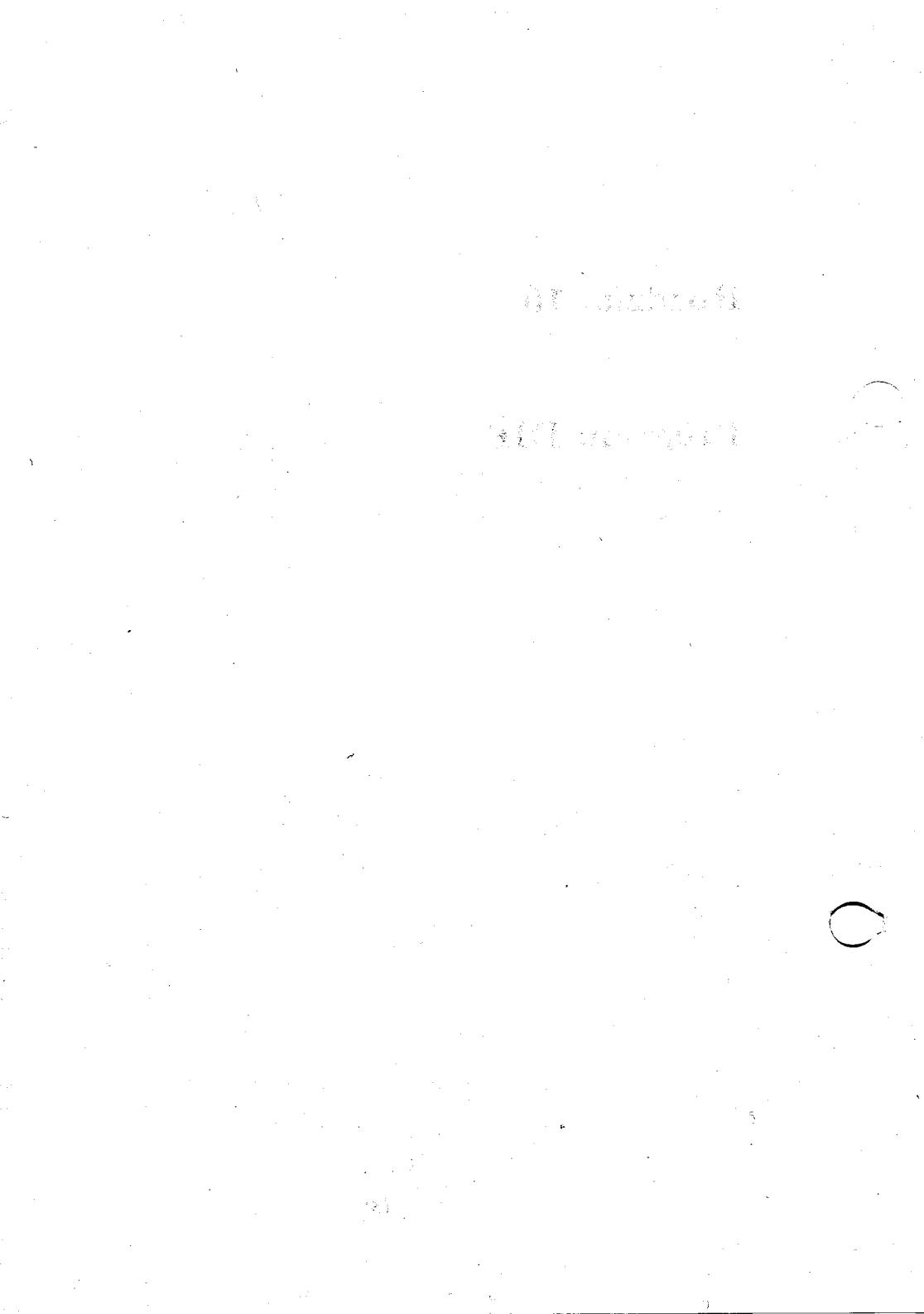
- rozkłady czasów obsługi muszą być wykładnicze (EXP) lub Coxa (COX).

9.3 Literatura

- [1] F. Baskett, M. Chandy, R. Muntz, J. Palacios, Open, Closed and Mixed Networks of Queues with Different Classes of Customers, J. ACM, vol. 22, no. 2, 1975.
- [2] J.R. Jackson, Jobshop-Like Queueing Systems, Management Science, vol. 10, pp.131-142, 1963.
- [3] D. Merle, Contribution a l'étude d'un analyseur de modèles à réseaux de files d'attente, algorithme de calculs et applications, Thèse de docteur ingénieur, Institut National Polytechnique de Lorraine, 1978.

Rozdział 10

Program DIF



10.1 Opis metody

Program DIF realizuje metodę aproksymacji dyfuzyjnej, pozwalającą analizować otwarte sieci stanowisk z dowolnym typem strumienia wejściowego, dowolnym rozkładem czasu obsługi i z naturalnym (FIFO) regulaminem szeregowania klientów.

Wymagana informacja o rozkładach odstępu czasu między generowanymi klientami i czasów obsługi ogranicza się do dwóch pierwszych momentów tych rozkładów, tzn. wyniki aproksymacji dyfuzyjnej są takie same dla różnych, lecz posiadających tę samą wartość średnią i wariancję rozkładów.

Metoda aproksymacji dyfuzyjnej w wykorzystywanej w pakiecie postaci pozwala analizować otwarte sieci stanowisk typu G/G/1 — stanowisk z dowolnym typem strumienia wejściowego, dowolnym rozkładem czasu obsługi i z naturalnym (FIFO) regulaminem szeregowania klientów. Dopuszcza się dowolną liczbę klas klientów; klienci poszczególnych klas różnią się czasem obsługi i drogą w sieci, lecz czekają na usługę we wspólnej kolejce. Klienci mogą zmieniać klasę w momencie zmiany stanowiska.

Podstawą tej przybliżonej metody jest zastąpienie dyskretnego procesu $N(t)$, odpowiadającego liczbie klientów na stanowisku obsługi, ciągłym procesem dyfuzji $X(t)$ o podobnych właściwościach. Wymagana informacja o rozkładach $A_i(x)$, $B_i(x)$ ogranicza się do dwóch pierwszych momentów tych rozkładów, tzn. wyniki aproksymacji dyfuzyjnej są takie same dla różnych, lecz posiadających tę samą wartość średnią i wariancję rozkładów $A_i(x)$, $B_i(x)$ oraz $A'_i(x)$, $B'_i(x)$.

Dyskusję błędów metody, rosnących wraz z $C_{a_i}^2$, $C_{b_i}^2$, można znaleźć w [1], [5].

10.1.1 Aproksymacja dyfuzyjna pojedynczego stanowiska obsługi

Klienci napływają do stanowiska obsługi w niezależnych od siebie odstępach czasu a_1, a_2, \dots, a_k , których rozkład $A(t)$ ma wartość średnią $1/\lambda$ i wariancję σ_a^2 . Oznamczymy

$$T_k = \sum_{i=1}^k a_i.$$

Zgodnie z centralnym twierdzeniem granicznym rozkład zmiennej

$$\frac{T_k - k/\lambda}{\sigma_a \sqrt{k}}$$

zbliża się do rozkładu normalnego wraz ze wzrostem k :

$$\lim_{k \rightarrow \infty} P\left[\frac{T_k - k/\lambda}{\sigma_a \sqrt{k}} \leq x\right] = \Phi(x), \quad (10.1)$$

gdzie

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\xi^2/2} d\xi,$$

a co za tym idzie, liczba klientów nadeszłych w odpowiednio długim czasie $[0; t]$ (tak, by liczba k była dostatecznie duża) ma w przybliżeniu rozkład normalny o wartości średniej λt i wariancji $\sigma_a^2 \lambda^3 t$.

Podobnie liczba klientów obsłużonych w dostatecznie długim czasie $[0; t]$ ma w przybliżeniu rozkład normalny o wartości średniej μt i wariancji $\sigma_b^2 \mu^3 t$ — o ile stanowisko pracuje bez przerwy. Wynika stąd, że zmiana liczby klientów w czasie $[0; t]$, $N(t) - N(0)$, ma w przybliżeniu rozkład normalny o wartości średniej $(\lambda - \mu)t$ i wariancji $(\sigma_a^2 \lambda^3 + \sigma_b^2 \mu^3)t$.

Aproksymacja dyfuzyjna dla stanowiska G/G/1 [4] zastępuje proces $N(t)$, którego zmiany w dostatecznie długim czasie mają rozkład normalny, ciągłym procesem

$X(t)$, którego nieskończonie małe zmiany

$$dX(t) = X(t + dt) - X(t)$$

mają rozkład normalny o średniej βdt i wariancji αdt , przy czym

$$\beta = \lambda - \mu,$$

$$\alpha = \sigma_a^2 \lambda^3 + \sigma_b^2 \mu^3 = C_a^2 \lambda + C_b^2 \mu. \quad (10.2)$$

$$(10.3)$$

Jest to proces dyfuzji z prądem, którego warunkowa funkcja gęstości prawdopodobieństwa

$$f(x, t; x_0) = P[x \leq X(t) < x + dx \mid X(0) = x_0]$$

spełnia równanie

$$\frac{\partial f(x, t; x_0)}{\partial t} = \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x}. \quad (10.4)$$

Proces $N(t)$ nie przyjmuje wartości ujemnych, dlatego przyjmuje się, że proces $X(t)$ po osiągnięciu wartości $x = 0$ zachowuje tę wartość przez czas, który jest wielkością losową o rozkładzie wykładniczym z parametrem λ , bezpośrednio po czym proces dyfuzji rozpoczyna się w punkcie $x = 1$. Czas, przez który $X(t) = 0$ odpowiada okresowi bezczynnemu, w którym w stanie nie ma eventów, a przeskok do wartości $X(t) = 1$ odpowiada nadaniem do stanowiska pierwszego po okresie bezczynnym klienta. Przy tak określonym warunku brzegowym równanie dyfuzji ma postać [2]:

$$\begin{aligned} \frac{\partial f(x, t; x_0)}{\partial t} &= \frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x} + \lambda_0 p_0(t) \delta(x - 1), \\ \frac{dp_0}{dt} &= \lim_{x \rightarrow 0} [\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0)] - \lambda_0 p_0(t) \delta(x - 1), \end{aligned} \quad (10.5)$$

gdzie $p_0(t) = P[X(t) = 0]$.

W stanie stacjonarnym, gdy

$$\lim_{t \rightarrow \infty} p_0(t) = p_0, \quad \lim_{t \rightarrow \infty} f(x, t; x_0) = f(x),$$

rozwiązaniem równań (10.5) jest funkcja

$$f(x) = \begin{cases} \frac{\lambda p_0}{-\beta} (1 - e^{zx}), & z \neq \frac{2\beta}{\alpha}, \\ \frac{\lambda p_0}{-\beta} (e^{-z} - 1) e^{zx}, & z = \frac{2\beta}{\alpha}. \end{cases} \quad (10.6)$$

Po uwzględnieniu warunku normalizacyjnego otrzymujemy $p_0 = 1 - \varrho$. Wartości $f(x)$, $x = 1, 2, \dots$ rozwiązania (10.6) są przybliżonymi wartościami $p(n)$, $n = 1, 2, \dots$, dla stanowiska G/G/1.

Jeżeli strumień wejściowy λ zawiera K klas klientów i $\lambda = \sum_{k=1}^K \lambda^k$, to w parametrach α, β uwzględnia się [3]

$$C_a^2 = \sum_{k=1}^K \frac{\lambda^k}{\lambda} C_a^{k2}, \quad \frac{1}{\mu} = \sum_{k=1}^K \frac{\lambda^k}{\lambda} \frac{1}{\mu^k},$$

$$C_b^2 = \mu^2 \sum_{k=1}^K \frac{\lambda^k}{\lambda} \frac{1}{\mu^{k2}} (C_b^{k2} + 1) - 1.$$

$f(n)$ określa rozkład $p(n)$ liczby klientów wszystkich klas obecnych w stanowisku, a prawdopodobieństwo, że w kolejce znajduje się ν klientów klasy k wynosi

$$p^k(\nu) = \sum_{n=\nu}^{\infty} [p(n) \sum_{k=1}^K \binom{n}{\nu} \left(\frac{\lambda^k}{\lambda} \right)^\nu \left(1 - \frac{\lambda^k}{\lambda} \right)^{n-\nu}], \quad k = 1, \dots, K. \quad (10.7)$$

10.1.2 Aproksymacja sieci stanowisk

W analizie sieci oblicza się strumienie wejściowe do poszczególnych stanowisk, po czym każde z tych stanowisk jest uważane za niezależne i opisane przy pomocy modelu pojedynczego stanowiska.

Wartość parametru λ_i strumienia wejściowego do stanowiska i w sieci określają równania ruchu (8.1) i (8.2). Trzeba jeszcze określić drugi moment rozkładów odstępów czasu między klientami pojawiającymi się w danym stanowisku.

Wykorzystuje się tu [3]:

- zależność pomiędzy strumieniem wejściowym i wyjściowym każdego stanowiska (ścisła dla M/G/1, przybliżona dla G/G/1)

$$d_i(t) = \varrho_i b_i(t) + (1 - \varrho_i) a_i(t) * b_i(t),$$

gdzie $*$ jest operacją splotu,

skąd

$$C_{di}^2 = C_{ai}^2(1 - \varrho_i) + \varrho_i^2 C_{bi}^2 - \varrho_i(1 - \varrho_i) \quad (10.8)$$

dla jednej klasy klientów, oraz

$$C_{di}^{k,2} = \lambda_i \sum_{k=1}^K \frac{\lambda_i^k}{(\mu_i^k)^2} [(C_{bi}^k)^2 + 1] + 2\varrho_i(1 - \varrho_i) + (C_{ai}^2 + 1)(1 - \varrho_i) - 1 \quad (10.9)$$

w przypadku wielu klas klientów;

- założenie, iż wariancja ogólnej liczby klientów, którzy nadeszli w jednostce czasu do stanowiska j z innych stanowisk i z zewnątrz, jest sumą wariancji poszczególnych składowych tego strumienia. Otrzymuje się

$$C_{aj}^2 = \frac{1}{\lambda_j} \sum_{i=1}^M r_{ij} \lambda_i [(C_{di}^2 - 1)r_{ij} + 1] + \frac{C_{0j}^2 \lambda_{0j}}{\lambda_j}; \quad (10.10)$$

parametry λ_{0j} , C_{0j}^2 odnoszą się do strumienia zewnętrznego.

W przypadku K klas klientów uwzględniamy

$$C_{aj}^2 = \frac{1}{\lambda_j} \sum_{l=1}^K \sum_{k=1}^K r_{ij}^{kl} \lambda_i [(C_{di}^{k^2} - 1) r_{ij}^{kl} + 1] + \sum_{k=1}^K \frac{C_{0j}^{k^2} \lambda_{0j}^k}{\lambda_j}. \quad (10.11)$$

Równania (10.8), (10.9) lub (10.10), (10.11) tworzą liniowy układ, pozwalający obliczyć C_{ai}^2 , co wraz z obliczonymi z 8.1 lub 8.2 λ_i pozwala określić β_i , α_i dla każdego stanowiska.

10.2 Uwagi o programie

Opis sieci rozwiązywanej programem DIF może zawierać co najwyżej:

- 30 źródeł (SOURCE),
- 30 stanowisk obsługi (STATION),
- 4 klasy klientów.

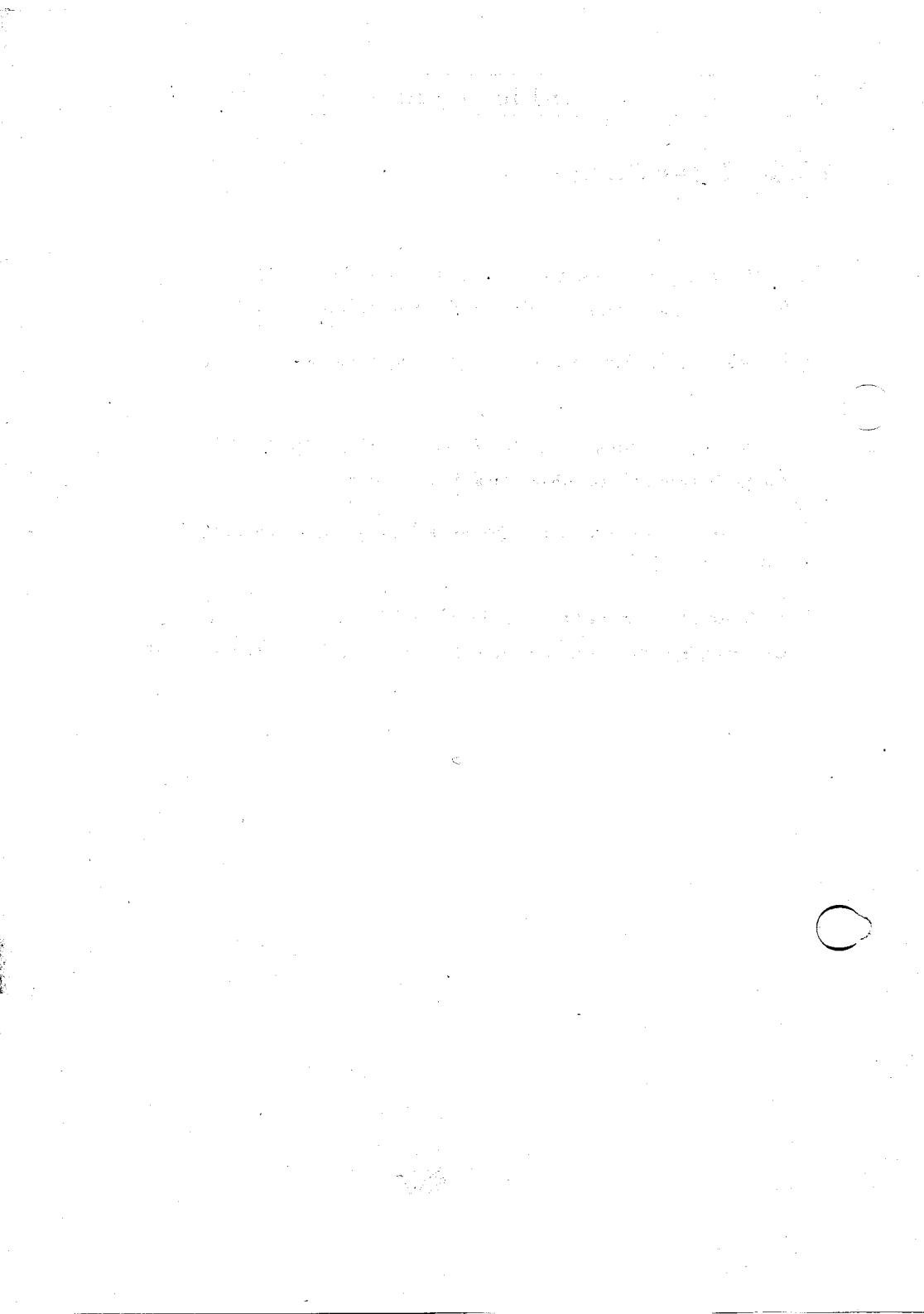
Rozkłady odstępów czasu między generowanymi przez źródła klientami i rozkłady czasów obsługi w stanowiskach mogą być zdefiniowane jako ogólne (G), w przypadku innych rozkładów program sam oblicza ich współczynnik zmienności.

Przy definiowaniu źródeł w opisie parametru TRANSITION nie może wystąpić parametr OUT w sposób jawnny ani opcjonalny.

Regulamin szeregowania klientów musi być naturalny (FIFO);

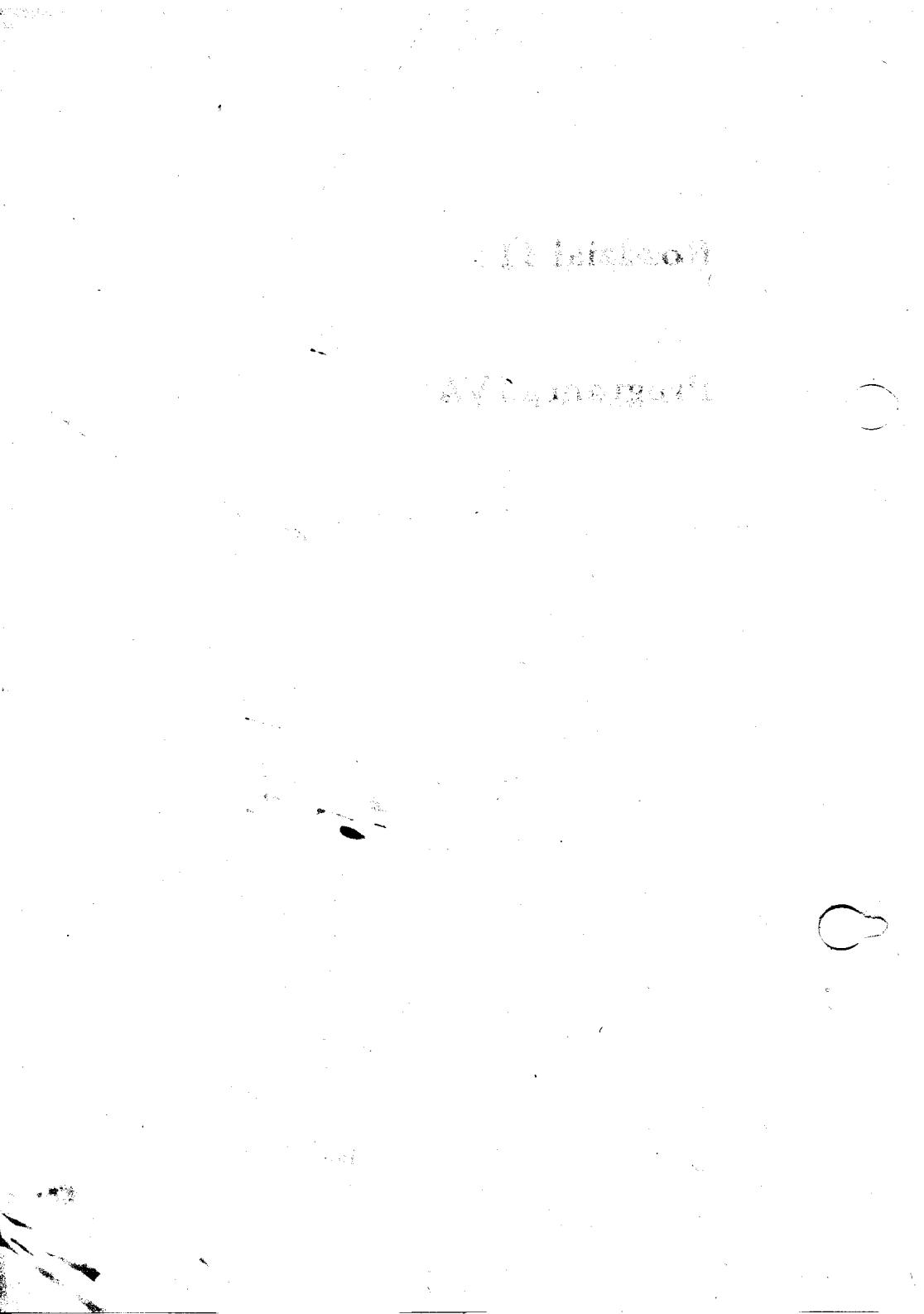
10.3 Literatura

- [1] G. Balbo, Approximate Solutions of Queueing Network Models of Computer Systems, Ph.D. Thesis, Purdue Univ., West Lafayette, 1979.
- [2] E. Gelenbe, On Approximate Computer Systems Models, J. ACM, vol. 22, no. 2, 1975.
- [3] E. Gelenbe, G. Pujolle, The Behaviour of a Single Queue in a General Queueing Network, Acta Informatica, Fasc. 7, 1976.
- [4] G. F. Newell, Applications of Queueing Theory, Chapman and Hall, London 1971.
- [5] M. Reiser, H. Kobayashi, Accuracy of the Diffusion Approximation for Some Queueing Systems, IBM J. of Res. Develop., vol. 18, pp.110–124, 1974.



Rozdział 11

Program MVA



11.1 Opis metody

W programie MVA wykorzystywana jest metoda wartości średnich, pozwalająca na dokładne rozwiązywanie zamkniętych sieci dekomponowalnych. Jeżeli w sieci znajdują się stanowiska z regulaminem naturalnym i różnymi od wykładniczych rozkładami czasów obsługi, a więc sieć nie jest dekomponowalna, wyniki obliczeń są przybliżone.

Dopuszczalny jest podział klientów na klasy. Podejście do rozwiązywania sieci kolejkowej metodą wartości średnich pozwala na otrzymanie "miar pracy" stanowisk wprost, bez obliczania stałej normalizacyjnej, stanowiącego istotny problem numeryczny w innych metodach.

Obliczane są tylko wartości średnie (przede wszystkim średnia długość kolejki, średni czas reakcji i przepustowość) natomiast nie oblicza się rozkładów długości kolejek w stanowiskach.

Podstawą metody jest wykorzystanie zależności pomiędzy średnim czasem reakcji a średnią długością kolejki w sieci o liczbie klientów pomniejszonej o 1. Ta zasadnicza zależność, zastosowana do każdego stanowiska i do każdej klasy, przy "czbie klientów wzrastającej co 1 od 0 do założonej, pozwala na rekurencyjne rozwiązanie modelu.

Metoda wartości średnich rozwiązuje zamkniętą sieć BCMP, której klienci nie zmieniają swojej przynależności do klas (każda sieć, której klienci zmieniają klasę, przechodząc z jednego stanowiska na inne, można zastąpić siecią o stałej przynależności do klas — por. [2]) bez obliczania stałej normalizacyjnej. Metoda opiera się na zależnościach pomiędzy średnim czasem oczekiwania a średnią długością kolejki w sieci o liczbie klientów pomniejszonej o 1. Złożoność obliczeniowa metody jest zbliżona do algorytmu splotowego, a zajętość pamięci nieco

większa. Nie ma natomiast problemów z przekraczaniem zakresu liczb zmiennooprzecinkowych. Metoda wartości średnich jest znaczco szybsza w przypadku sieci o stanowiskach wielokanalowych. Jest także, dzięki swej prostocie, intuicyjnie zrozumiała.

Wprowadźmy następujące oznaczenia:

- $S(k)$ — zbiór stanowisk należących do łańcucha k , tj. odwiedzanych przez klientów klasy k ,
- $K(m)$ — zbiór klas klientów odwiedzających stanowisko m ,
- $\underline{1}^k$ — wektor bazy ortonormalnej (standardowej) w \mathbb{R}^K ,
- $\underline{0}$ — K -wymiarowy wektor zerowy.

Podstawą metody jest następujące twierdzenie — por. [3] [1]:

Średni czas oczekiwania klientów klasy k na stanowisku m jest powiązany ze średnią długością kolejki na tym stanowisku w sieci z jednym klientem mniej, następującą zależnością:

$$\bigwedge_{\sum N^k > 0} t_m^k(N) = \begin{cases} s_m^k \{1 + E[n_m(N - \underline{1}^k)]\} & \text{dla stanowiska typu} \\ & \text{FCFS, PS lub LCFS-} \\ & \text{PR,} \\ s_m^k & \text{dla stanowiska typu IS,} \\ t_m^k(N) = 0 & \text{dla } N^k = 0. \end{cases} \quad (11.1)$$

Zależność tę wiążemy z twierdzeniem Little'a zastosowanym do każdego łańcucha

i do każdego stanowiska oddzielnie. Z równań Little'a dla łańcucha k mamy:

$$\lambda_l^k = \frac{N^k}{\sum_{m \in S(k)} e_m^k t_m^k} . \quad (11.2)$$

Suma w mianowniku jest średnim czasem, jaki upływa pomiędzy kolejnymi wizytami klienta w stanowisku l . Równanie Little'a dla stanowiska m :

$$E(n_m^k) = \lambda_m^k t_m^k , \quad E(n_m) = \sum_{k \in K(m)} E(n_m^k) . \quad (11.3)$$

Równania (11.2) – (11.3) pozwalają na iteracyjne obliczanie takich wielkości opisujących sieć, jak : średnia długość kolejki, średni czas reakcji i przepustowość, jeśli zastosujemy je dla $m = 1, 2, \dots, M$ oraz $k = 1, 2, \dots, K$ i rozpoczęmy od wartości początkowej :

$$E[n_m^k(0)] = 0 .$$

11.2 Uwagi o programie

Opis sieci rozwiązywanej programem MVA może zawierać co najwyżej:

- 30 stanowisk obsługi (STATION),
- 10 klas klientów.

Nie ma ograniczeń na liczbę klientów.

11.3 Literatura

- [1] S. S. Lavenberg, M. Reiser, Stationary State Probabilities at Arrival Instants for Closed Queueing Networks with Multiple Types of Customers, *J. Appl. Probab.*, vol. 17, no. 4, 1980.
- [2] M. Reiser, H. Kobayashi, Queueing Network Models with Multiple Closed Chains: Theory and Computational Algorithms, *IBM J. Res. Develop.*, vol. 19, 1975.
- [3] K. C. Sevcik, I. Mitrani, The Distribution of Queueing Network States at Input and Output Instants, w: M.Arato et al (edts.), *Performance of Computer Systems*, North Holland, 1979.

Rozdział 12

Program HMVA

8000000000

8000000000



12.1 Opis metody

Algorytm wykorzystywany w dokładnej analizie wartości średnich wymaga przeprowadzenia obliczeń dla całego ciągu populacji klientów w systemie: od systemu z jednym klientem aż do systemu z interesującą nas liczbą klientów. Dla dużej populacji klientów zgrupowanych w wiele klas obliczenia są czasochłonne — np. rozwiązanie sieci zawierającej 10 stanowisk i 5 klas klientów, po 10 klientów w każdej klasie wymaga od algorytmu MVA ponad 8 mln. operacji arytmetycznych, natomiast jeżeli klienci nie są rozróżnialni, tzn. jeżeli w sieci jest 50 klientów tej samej klasy, algorytm MVA wymaga ok. 500 operacji arytmetycznych. Dlatego też dla dużych populacji klientów została zaproponowana metoda przybliżona, bazująca na zależnościach MVA ([4], [1]). Metoda ta będzie oznaczana jako HMVA. Wyniki dla populacji $\underline{N} - 1^r$ są tu obliczane na podstawie wyników dla populacji \underline{N} , następnie są stosowane zależności metody MVA dla obliczenia nowych wyników dla populacji \underline{N} . Iteracja $\underline{N} \rightarrow \underline{N} - 1^r \rightarrow \underline{N}$ jest powtarzana aż do osiągnięcia wymaganej dokładności. Stosowane są następujące przybliżone zależności:

$$F_m^k(\underline{N}) = E[n_m^k(\underline{N})/N^k], \quad (12.1)$$

$$D_m^{kj}(\underline{N}) = F_m^k(\underline{N} - 1^j) - F_m^k(\underline{N}), \quad (12.2)$$

$$E[n_m^k(\underline{N} - 1^j)] = (\underline{N} - 1^j)^k [F_m^k(\underline{N}) + D_m^{kj}(\underline{N})]. \quad (12.3)$$

Znaczenie powyższych wielkości jest następujące:

$F_m^k(N)$ — część klientów klasy k , która znajduje się na stanowisku m ,

$D_m^{kj}(N)$ — zmiana $F_m^k(N)$ spowodowana zmniejszeniem populacji o jednego klienta klasy j .

Metoda HMVA wykorzystuje fakt, że mając dane wielkości $E[n_m^k(\underline{N})]/N^k$ oraz $D_m^{kj}(N)$, można obliczyć $E[n_m^k(\underline{N} - 1^j)]$. Założeniem metody jest:

$$D_m^{kj}(N) = D_m^{kj}(\underline{N} - 1^j),$$

czyli F jest liniowo zależne od liczby klientów w sieci.

12.2 Uwagi o programie

Opis sieci rozwiązywanej programem HMVA może zawierać co najwyżej:

- 30 stanowisk obsługi (STATION),
- 10 klas klientów.

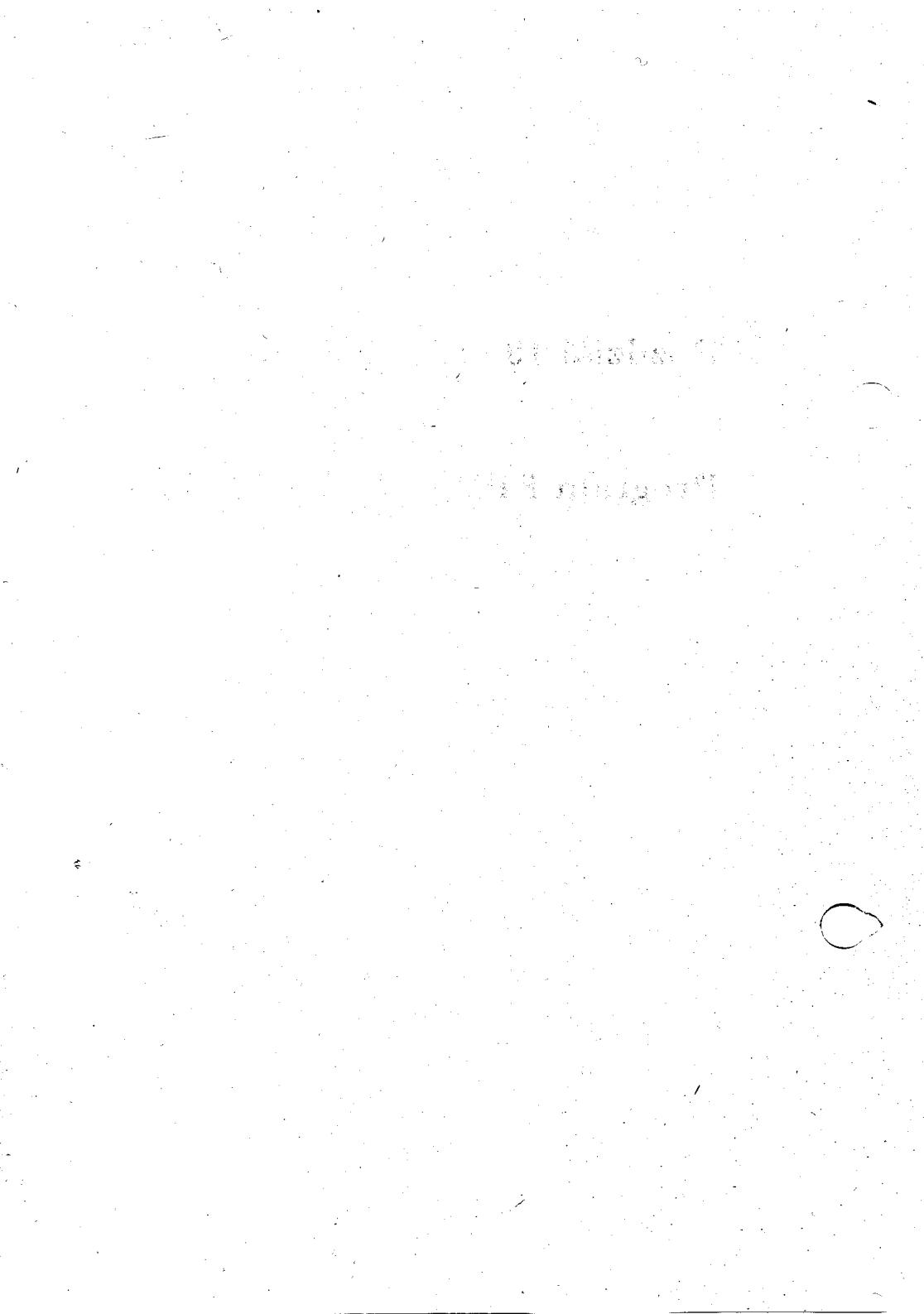
Nie ma ograniczeń na liczbę klientów.

12.3 Literatura

- [1] K. M. Chandy, D. Neuse, Linearizer: A Heuristic Algorithm for Queueing Network Models of Computing Systems, Comm. ACM, vol. 25, no. 2, 1982.

Rozdział 13

Program EPF



13.1 Opis metody

Program EPF realizuje metodę rozszerzonej formy iloczynowej, służącą do otrzymywania przybliżonego rozwiązania zamkniętych sieci niedekomponowalnych z jedną klasą klientów [1]. Jednocześnie rozwiązuje ona zamknięte sieci dekomponowalne dając wyniki dokładne. Zaletami metody są prostota koncepcji i wyniki bliskie dokładnym przy małych wariancjach rozkładów czasów obsługi na stanowiskach.

Metoda EPF nawiązuje do sposobu rozwiązywania zamkniętych dekomponowalnych sieci stanowisk obsługi.

Rozwiązanie zamkniętych sieci stanowisk obsługi z wykładowiczym rozkładem czasu obsługi na wszystkich stanowiskach oraz z naturalnym regulaminem obsługi ma postać iloczynową:

$$p(n_1, n_2, \dots, n_M) = \frac{1}{G(N)} p_1(n_1) p_2(n_2) \dots p_M(n_M)$$

gdzie:

- n_i — liczba klientów na i -tym stanowisku obsługi,
- M — liczba stanowisk obsługi,
- G — stała normalizacyjna,
- $p(n_i)$ — prawdopodobieństwo, że na i -tym stanowisku znajduje się n_i klientów,
- $p(n_1, n_2, \dots, n_M)$ — prawdopodobieństwo stanu ustalonego sieci, w którym na i -tym stanowisku obsługi znajduje się n klientów, $i = 1, 2, \dots, M$,

a $N = \sum_{i=1}^M n_i$.

Metoda EPF zakłada, że w przypadku, gdy rozkłady czasów obsługi nie są wykładnicze, lecz dowolne, rozwiążanie ma w przybliżeniu formę iloczynową, a $p_i(n_i)$ mają postać jak w odosobnionych stanowiskach typu M/G/1/N. Ich postać określa się na podstawie $p'_i(n_i)$ dla stanowiska M/G/1:

$$p_i(n_i) = \frac{p'_i(n_i)/(1-\varrho)}{1 + \frac{\varrho}{(1-\varrho)} \sum_{\nu=0}^{N-1} p'_i(\nu)}, \quad n_i = 0, \dots, N-1,$$

$$p_i(N) = 1 - \sum_{n_i=0}^{N-1} p_i(n_i).$$

$p'(n)$ obliczamy z zależności Pollaczka-Chinczyna

$$Q(z) = \bar{b}(\lambda - \lambda z) \frac{(1-\varrho)(1-z)}{\bar{b}(\lambda - \lambda z) - z}, \quad (13.1)$$

gdzie $Q(z) = \sum_{k=0}^{\infty} p'(k)z^k$ jest funkcją tworzącą $p'(n)$, a $\bar{b}(s)$ jest transformatą Laplace'a funkcji gęstości rozkładu czasu obsługi.

Konieczne obliczenia wykonano dla występujących w języku AMOKu typów rozkładów.

W sieci zamkniętej, wartości względnej przepustowości i -tej gałęzi sieci e_i mogą być obliczone tylko z dokładnością do stałego czynnika, który oznaczać będziemy przez X_0 . Tak więc, równie dobrze możemy napisać:

$$\varrho_i = X_0 e_i s_i, \quad X_0 > 0.$$

W przypadku sieci separowalnych nie stwarza to problemu, gdyż rozwiążanie w formie iloczynowej jest niezależne od wyboru X_0 . Nie jest tak jednak, gdy stanowiska mają rozkład dowolny. W rezultacie, w metodzie EPF rozwiążanie

sieci w stanie ustalonym jest funkcja X_0 . Właściwe X_0 musi prowadzić do spełnienia warunku przepustowości, tj. równania (8.1). Wiedząc, że względne przepustowości spełniają ten sam warunek, możemy napisać:

$$\frac{\lambda_1}{e_1} = \frac{\lambda_2}{e_2} = \dots = \frac{\lambda_M}{e_M} = h = \text{const.}$$

Załóżmy teraz, że podstawiając jakąś wartość X_0 do wzorów na $p_i(n_i)$ oraz stosując formę iloczynową rozwiązania, obliczymy absolutne przepustowości. Niech

$$h_i = \frac{\lambda_i}{e_i}, \quad i = 1, 2, \dots, M.$$

Aby spełniony został warunek przepustowości, X_0 musi być takie, że

$$\bigwedge_{1 \leq i, j \leq M} h_i = h_j.$$

Realizację powyższego warunku można aproksymować poszukiwaniem X_0 minimalizującego sumę najmniejszych kwadratów o postaci:

$$L(X_0) = \sum_{i=1}^M \left(\frac{h_i}{E(h)} - 1 \right)^2 X_i,$$

gdzie: $X_i = e_i s_i$, $E(h)$ — wartość oczekiwana h .

Ze względów praktycznych jako $E(h)$ przyjmuje się h_s , gdzie s jest numerem najbardziej obciążonego stanowiska obsługi w sieci. Tak więc, poszukiwanie właściwego X_0 polega na znalezieniu X_0 minimalizującego funkcję:

$$L(X_0) = \sum_{i=1}^M \left(\frac{h_i}{h_s} - 1 \right)^2 \varrho_i.$$

W dokonanej implementacji, minimum $L(X_0)$ znajduje się przeszukując zalożony przedział zmienności $L(X_0)$ ze stałym krokiem, z kolejnym zawężaniem przedziału przeszukiwania wokół wyznaczonego argumentu minimum.

Po znalezieniu właściwego X_0 obliczamy dla kolejnych stanowisk przepustowość λ_i , średnią liczbę klientów w systemie $E(n_i)$, średnie obciążenie stanowiska ϱ_i .

13.2 Uwagi o programie

Opis sieci rozwiązywanej programem EPF może zawierać co najwyżej:

- 30 stanowisk obsługi.

W sieci może krążyć maksimum 50 klientów.

13.3 Literatura

- [1] A. W. Shum, Queueing Models for Computer Systems with General Service Time Distribution, Garland, New York 1980.

Rozdział 14

Program ITERM

1870-1871

1870-1871



14.1 Opis metody

Program ITERM wykorzystuje metodę iteracyjną stanowiska równoważnego, służącą do otrzymywania rozwiązania zamkniętej sieci stanowisk obsługi z jedną klasą klientów. Moduł rozwiązuje zamkniętą sieć typu BCMP, która może być uzupełniona o stanowiska z regulaminem naturalnym i rozkładem czasu obsługi typu Cox'a. Dla sieci dekomponowalnych rozwiązanie jest dokładne, dla pozostałych wyniki są przybliżone.

Istotą metody jest zredukowanie sieci do dwu stanowisk — wyróżnionego stanowiska i oraz stanowiska zastępczego, reprezentującego resztę sieci. Czas obsługi na stanowisku zastępczym posiada rozkład wykładniczy o średniej zależnej od liczby klientów w stanowisku. Parametry stanowiska zastępczego są obliczane na podstawie przepustowości sieci mającej zwarte stanowisko i (czas obsługi równy zero). Rozwiązanie sieci dwustanowiskowej złożonej z wykładniczego stanowiska zastępczego i dowolnego stanowiska i jest znane i dostarcza $p_i(n_i)$, $i = 1, \dots, M$ (prawdopodobieństwa obsadzenia stanowiska klientami), co z kolei pozwala na obliczenie szukanych parametrów sieci (średnie długości kolejek, obciążenie stanowisk itd.).

Rozwiązanie otrzymuje się z następującego układu równań :

$$\left. \begin{aligned} p(n) &= \frac{\prod_{j=1}^M e_j^{n_j}}{G(N)} \\ \nu_j(i) &= \frac{e_j G_j(N-i)}{G_j(n-i+1)} \cdot \frac{Q_j(i-1)}{Q_j(i)} \end{aligned} \right\} \begin{array}{l} i = 0, \dots, N, \\ j = 1, \dots, M, \end{array} \quad (14.1)$$

gdzie:

$$\underline{eR} = \underline{e} = (e_1, \dots, e_M), \quad (14.2)$$

$$A_j(n_j) = \begin{cases} 1 & n_j = 0, \\ \prod_{j=1}^{n_j} \nu_j(i) & n_j > 0, \end{cases} \quad (14.3)$$

$$G(N) = \sum_{\underline{n}} \prod_{j=1}^M \frac{e_j^{n_j}}{A_j(n_j)}, \quad (14.4)$$

$$G_x(u) = \sum_{\substack{\underline{n}: \sum n_i = u, n_x = 0}} \prod_{j=1}^M \frac{e_j^{n_j}}{A_j(n_j)}, \quad (14.5)$$

$Q_j(\bullet)$ jest prawdopodobieństwem asymptotycznym stanu j-tego stanowiska, rozpatrywanego niezależnie od sieci, o wykładniczym strumieniu wejściowym, zależnym od liczby klientów na stanowisku (czas obsługi jest niezmienny) i intensywności strumienia

$$\lambda_j(i) = \frac{e_j G_j(N - i - 1)}{G_j(N - i)}. \quad (14.6)$$

Objaśnienia:

- (14.1) — $p(\underline{n})$ to prawdopodobieństwa asymptotyczne stanu w sieci identycznej, co badana strukturze i stanowiskach wykładniczych o intensywności $\nu_j(n_j)$ (jest to sieć iloczynowa o znanym rozwiązaniu [3] lub [4]),
- $\nu_j(i)$ to intensywność obsługi j-tego stanowiska, odzwierciedlenie warunku $\nu(i)Q(i) = \lambda(i-1)Q(i-1)$,
- (14.2) — równanie ruchu klientów w sieci,
- (14.4) — $G(N)$ to stała normalizacyjna modelowanej sieci,

(14.5) — $G_x(u)$ to stała normalizacyjna w sieci zwartej na stanowisku x ,

(14.6) — $\lambda_j(i)$ to przepustowość gałęzi, w której zwarto stanowisko j .

Rozwiązywanie iteracyjne powyższego układu równań ma postać (brak dowodu zbieżności):

$$\nu_j^{(0)}(i) = \begin{cases} 0 & i = 0 \\ \mu_j & i > 0 \end{cases} \quad (\text{intensywność obsługi w sieci oryginalnej}) \quad (14.7)$$

$$\nu_j^{(l)}(i) = \frac{e_j G^{(l-1)}(N-i)}{G_j^{(l-1)}(N-i+1)} \cdot \frac{Q_j^{(l-1)}(i-1)}{Q_j^{(l-1)}(i)} \quad (14.8)$$

$$i = 0, \dots, N, \quad j = 1, \dots, M$$

Warunkiem końca iteracji jest:

$$\left| \frac{N - \sum_{j=1}^M \sum_{i=0}^N i Q_j(i)}{N} \right| < \varepsilon, \quad (14.9)$$

$$\left| \frac{X_i - \frac{1}{M} \sum_{j=1}^M X_j}{\frac{1}{M} \sum_{j=1}^M x_j} \right| < \varepsilon, \quad i = 1, \dots, M, \quad (14.10)$$

gdzie:

$$X_j = \frac{1 - Q_j(0)}{e_j \mu_j^{-1}}. \quad (14.11)$$

Objaśnienia:

(14.9) — warunek liczby klientów w sieci,

(14.10) — z twierdzenia Chang-Lavenberga o wykorzystaniu stanowisk [2].

Stale normalizacyjne są liczone przy wykorzystaniu metody Buzena [1]

$$G(n) = g(n, M), \quad (14.12)$$

$$g(n, m) = \sum_{k=0}^n \frac{(e_m)^k}{A_m(k)} g(n-k, m-1), \quad (14.13)$$

przy czym

$$g(n, 1) = (e_1)^n / A_1(n), \quad n = 0, \dots, N, \quad (14.14)$$

$$g(0, m) = 1, \quad m = 1, \dots, M, \quad (14.15)$$

Prawdopodobieństwa asymptotyczne stanowisk izolowanych otrzymuje się:

- dla stanowisk z kolejką z regułaminem szeregowania naturalnym — według Marie'go [5],
- dla stanowisk z regułaminem szeregowania typu PS, LCFS-PR, IS — według [6].

Przyjmuje się podaną na rysunku 4.1 postać rozkładu Coxa. Dla regulaminu FCFS, $p_1 = 1$. We wzorze (14.7) μ ma postać:

$$\frac{1}{\mu} = \sum_{i=1}^F \left(\prod_{j=1}^i p_j \right) (1 - p_{i+1}) EXP[s_i] . \quad (14.16)$$

14.2 Uwagi o programie

Maksymalna liczba klientów i stanowisk obsługi są zależne od złożoności modelu (obszary robocze są przydzielane dynamicznie). Sposób deklarowania liczby klientów podano w rozdziale opisującym sposób korzystania z programów pakietu.

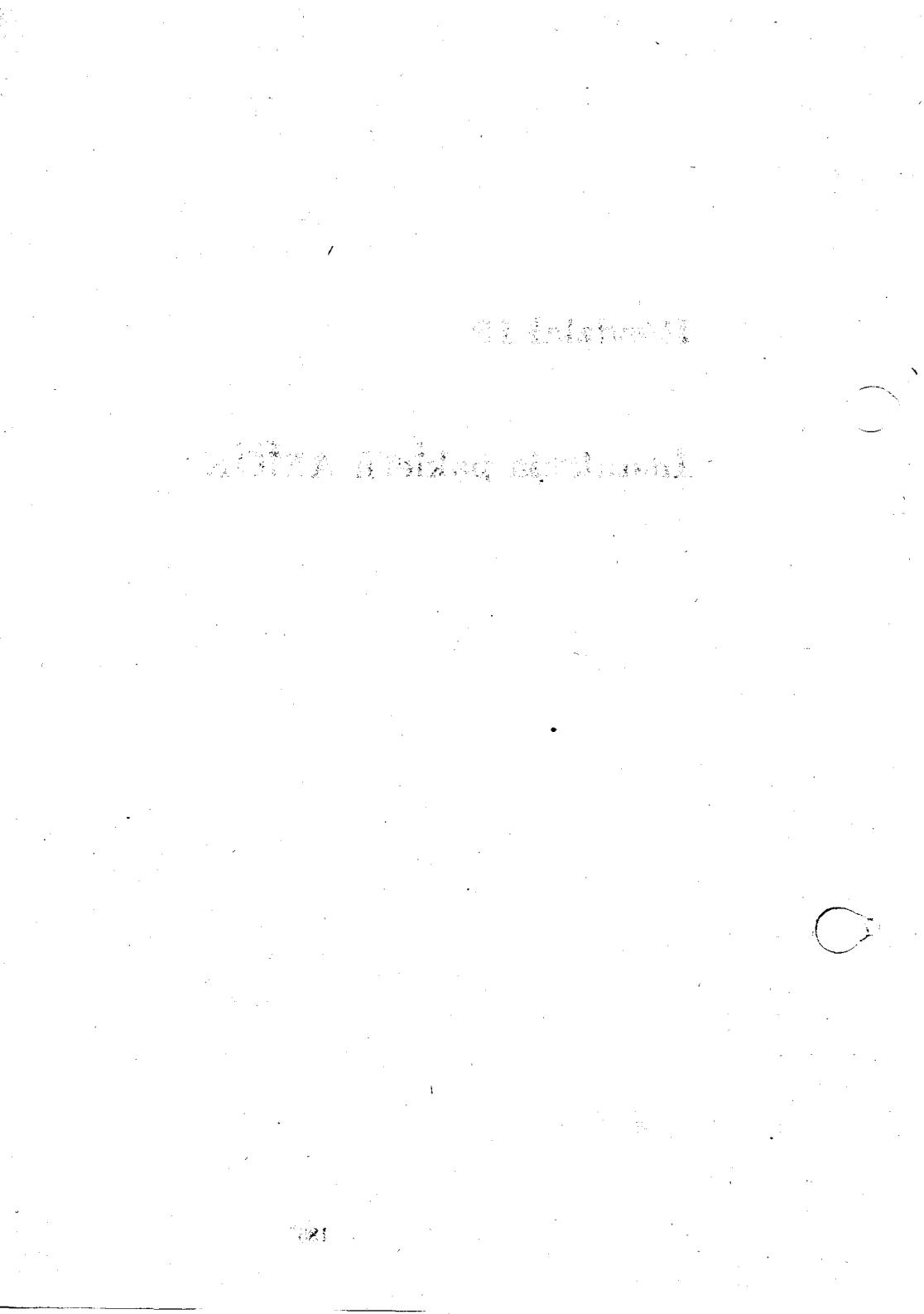
14.3 Literatura

- [1] J. P. Buzen, Computational Algorithms for Closed Queueing Networks with Exponential Servers, Comm. ACM, vol. 16, no. 9, 1973.
- [2] A. Chang, S. S. Lavenberg, Work Rates in Closed Queueing Networks with General Independent Servers, Oper. Res., vol. 22, no. 4, 1972
- [3] W. J. Gordon, G. F. Newell, Closed Queueing Systems with Exponential Servers, Operations Research, vol. 15, no. 2, pp.252- 267, 1967.
- [4] S.S.Lavenberg i in., Computer Performance Modeling Handbook, Academic Press, 1983.

- [5] R. Marie, An Approximate Analytical Method for General Queueing Networks, IEEE Trans. on Software Eng., vol. SE-5, no. 5, 1979.
- [6] M. Reiser, C. H. Sauer, Queueing Network Models: Methods of Solution and their Program Implementation, w: K.M.Chandy, R.T.Yeh (edts.), Current Trends in Programming Methodology, vol. 3, Prentice Hall, 1978.

Rozdział 15

Instalacja pakietu AMOK



Pakiet programowy AMOK stanowi zestaw programów i plików pomocniczych powiązanych ze sobą zależnościami funkcjonalnymi, do przechowania którego na dysku twardym wymagany jest obszar o pojemności ok. 3 MB. Do uruchomienia pakietu, a dokładniej — jego programów obliczeniowych — niezbędne jest ok. 580 KB wolnej pamięci operacyjnej, a więc może okazać się celowe usunięcie z pamięci komputera programów rezydentnych.

Pakiet AMOK istnieje w różnych wersjach uwzględniających parametry mikrokomputera (np. rodzaj karty graficznej), na którym ma on pracować. Tworzenie odpowiedniego wariantu pakietu przebiega automatycznie, po uruchomieniu makroinstrukcji instalacyjnej INSTALL.BAT z odpowiednimi argumentami:

[d :]INSTALL d1 d2

gdzie:

- d — nazwa dysku z makroinstrukcją INSTALL,
- d1 — nazwa kieszeni, w której znajduje się dyskietka instalacyjna Nr 1,
- d2 — nazwa dysku twardego, na którym będzie dokonywane inicjowanie AMOKa (przy czym nie jest konieczne, aby był to dysk bieżący).

Użycie INSTALL bez parametrów podaje wzór poprawnego wywołania.

INSTALL można startować bezpośrednio z dyskietki (założone jest wtedy, aby $d=d1$, o ile d występuje):

Przykłady

zainstalowanie AMOKa na dysku D z dyskietki w kieszeni A.

C:\>A:INSTALL A D

albo

A:\>INSTALL A D

albo

D:\>A:INSTALL A D

Pakiet można też instalować po uprzednim przekopiowaniu INSTALL.BAT na dowolny dysk twardy:

Przykład

zainstalowanie AMOKa na dysku F z dyskietki w kieszeni B.

D:\>COPY B:INSTALL.BAT

D:\>INSTALL B F

Podczas sesji instalacyjnej makroinstrukcja żąda zmian dyskietek i — w przypadku, gdy mikrokomputer wyposażony jest w koprocesor arytmetyczny — pyta, czy użytkownik chce go wykorzystywać:

Do you want to use your Math Coprocessor? (y/n)

Kopiowanie plików przykładowych z dyskietki Nr 5 jest opcjonalne i zależy od odpowiedzi użytkownika na pytanie:

Do you want to copy example files now? (y/n)

Pliki te można wprowadzić na dysk twardy również niezależnie od instalacji poprzez użycie zleceń DOSa czy też innego narzędziowego programu kopiującego. Makroinstrukcja instalacyjna wprowadza wybrane pliki pakietu AMOK oraz ewentualnie pliki z przykładami — do kartoteki \AMOK, która w razie potrzeby jest tworzona automatycznie i staje się kartoteką bieżącą po zakończeniu pracy INSTALL. Wówczas można już bezpośrednio uruchomić pakiet AMOK:

Przykład

D:\AMOK> IA

Należy zwrócić uwagę, że w ciągu jednej sesji instalacji nie wykorzystuje się wszystkich pięciu dyskietek, o wyborze dyskietek i kolejności ich użycia decyduje makroinstrukcja INSTALL.

Zawartość dyskietek:

- Nr 1 — INSTALL.BAT, inne pliki instalacyjne i program konwersji CONV;
- Nr 2 — EGZEKUTOR dla karty monochromatycznej (typu Hercules) oraz dla karty kolorowej;
- Nr 3 — plik metod obliczeniowych bez użycia koprocesora arytmetycznego;
- Nr 4 — plik metod obliczeniowych z użyciem koprocesora arytmetycznego;
- Nr 5 — przykładowe opisy sieci w języku AMOKu i pliki z bazami danych (po konwersji).

UWAGI:

1. Próba przeniesienia plików zainstalowanego AMOKu na dysk twardy innego mikrokomputera na drodze prostego kopiowania plików pakietu może zakończyć się niepowodzeniem. Jedyną pewną metodą uzyskania kompletnego i poprawnego zestawu plików AMOKu jest użycie jego makroinstrukcji instalującej INSTALL.
2. Wywołanie IA ustawia zmienną systemową BREAK na ON.

Na dyskietce Nr 1 znajduje się program FXARC.COM firmy PKWARE Inc. Nie zostały w nim wprowadzone żadne zmiany i dostarczany jest nieodpłatnie zgodnie z licencją firmy PKWARE Inc.

Wyświetlenie licencji: [d:]fxarc -l

Errata do wersji 2.11

Rozdział 2

Podstawowe pojęcia modeli kolejkowych

Moduł	Liczba źródeł	Liczba stanowisk	Liczba klas	Liczba klientów
JACKSON	30	30	8	—
DIF	30	30	5	—
MVA	—	30	10	∞
HMVA	—	30	10	∞
EPF	—	30	1	30
ITERM	—	10	1	50

Tab. 2.2: Maksymalne rozmiary sieci analizowanych przez wersję 2.11 AMOKu

Rozdział 4

Język opisu sieci kolejek

4.- Postać zapisu w języku AMOKu

Opis sieci kolejek rozpoczyna się słowem ***DESCRIPTION*** i kończy słowem ***END***. Słowo ***END*** jest ostatnim analizowanym elementem opisu; ewentualny, występujący po nim zapis jest nieistotny dla analizy.

Pomiędzy słowami ***DESCRIPTION*** i ***END*** występują opisy poszczególnych obiektów kolejkowych oraz podsieci.

Postać zapisu:

SOUR *tekst użytkownika*

DESCRIPTION

opis obiektów kolejkowych i podsieci

END

4.- Podsieci

1. Podsieć jest fragmentem sieci, w której zgrupowano opisy kilku obiektów sieci.
2. Wprowadzenie podsieci w opisie sieci kolejek jest opcjonalne.
3. Podsieci mogą być zagnieźdzane, tzn. podsieć może zawierać w sobie oprócz opisu obiektów — opisy innych podsieci. Maksymalna głębokość zagnieźdzania podsieci wynosi 9.
4. Opis każdej podsieci rozpoczyna identyfikator początku podsieci składający się ze standardowej nazwy **SUBNETWORK**, poprzedzonej i zakończonej znakiem “/”.

Koniec opisu podsieci (zamknięcie podsieci) wyznacza identyfikator końca podsieci składający się ze standardowej nazwy **SUBEND**, poprzedzonej i zakończonej znakiem “/”.

Koniec opisu podsieci wyznacza również koniec sieci i koniec opisu ***END***.

Uwaga: Występowanie identyfikatora końca podsieci nie poprzedzonego odpowiadającym mu identyfikatorem początku podsieci jest niedozwolone.

5. Każdej podsieci można przypisać nazwę przez użycie parametru **NAME** umieszczonego za identyfikatorem początku podsieci. Parametr ten jest tutaj opcjonalny.
6. Za identyfikatorem końca podsieci można umieścić nazwę zamkanej podsieci zdefiniowaną uprzednio w parametrze **NAME**. W tym przypadku zamknięciu ulegają podsieci zawarte wewnątrz zamkanej podsieci. Zamiast jednej nazwy można podać oddzielone przecinkami nazwy kilku (niekoniecznie wszystkich) zamkanych podsieci. Nazwy te powinny występować w kolejności od nazwy podsieci najbardziej zagnieżdżonej do nazwy podsieci najmniej zagnieżdżonej.

Użytkowa nazwa podsieci, umieszczona za identyfikatorem końca podsieci, jest opcjonalna. Gdy nie zostanie podana, zamkana jest ostatnio wprowadzona do opisu, najbardziej zagnieżdżona podsieć.

Przykład:

DESCRIPTION

/SUBNETWORK/

podsieć bez nazwy

/SUBNETWORK/ NAME=CPU

podsieć o nazwie CPU

/SUBNETWORK/

podsieć bez nazwy

/SUBEND/

/SUBNETWORK/ NAME=S-1

podsieć o nazwie S-1

/SUBEND/ CPU

END

Rozdział 13

Program EPF

13.2 Uwagi o programie

Opis sieci rozwiązywanej programem EPF może zawierać co najwyżej:

- 30 stanowisk obsługi.

W sieci może krażyć maksimum 30 klientów.