

J. WINKOWSKI (Warszawa)

O symulacji algorytmicznej

1. Wprowadzenie. Nie ma dotychczas ostatecznie ukształtowanej teorii symulacji. Będziemy jednak starali się przedstawić temat w sposób możliwie ścisły, używając pojęć matematycznych. Najważniejsze z nich to pojęcia algorytmu i obliczenia. *Algorytmem* nazywamy parę $M = (S; f)$ złożoną ze zbioru S (zbiór *stanów*) i operacji jednoargumentowej (na ogół częściowej) f w S (*funkcja przejścia*). Algorytm stanowi przepis przekształcania krok po kroku pewnych danych (stany) według stałej reguły (*funkcja przejścia*). Na przykład znany algorytm dodawania liczb w zapisie dziesiętnym jest przepisem przekształcania odpowiednich napisów aż do uzyskania napisu końcowego, który jest reprezentacją szukanego wyniku. *Obliczeniem* albo *procesem obliczeniowym* algorytmu $M = (S; f)$ nazywamy każdy ciąg s_0, s_1, \dots stanów ze zbioru S , w którym $s_{n+1} = f(s_n)$, i który jest nieskończony lub kończy się stanem s_k , dla którego $f(s_k)$ nie jest określone. Takie obliczenie jest w pełni wyznaczone przez algorytm M i przez stan s_0 i oznaczamy je przez $\text{Obl}(M, s_0)$. Stan s_0 nazywamy *początkowym*, a ostatni stan s_k (o ile istnieje) – *końcowym*. Stan początkowy s_0 reprezentuje dane początkowe, stany s_1, s_2, \dots – *wyniki pośrednie*, a ewentualny stan końcowy s_k – *wynik końcowy*. Tak np. proces obliczeniowy algorytmu dodawania jest ciągiem kolejno otrzymywanych napisów. Pierwszy z napisów reprezentuje dodawane liczby.

Ogólnie metoda symulacji polega na badaniu zjawisk poprzez prowadzenie eksperymentów z innymi, podobnymi zjawiskami. W przypadku symulacji algorytmicznej rolę takich podobnych zjawisk spełniają obliczenia (procesy obliczeniowe) odpowiednich algorytmów. Obliczenia, mające z natury charakter dyskretny, nadają się szczególnie dobrze do symulacji zjawisk dyskretnych, tzn. takich, których stan zmienia się jedynie w pewnych chwilach $t_1 \leq t_2 \leq \dots$. Natomiast zjawiska o charakterze ciągłym trzeba na ogół aproksymować zjawiskami dyskretnymi. Tutaj będziemy mówili tylko o zjawiskach dyskretnych.

Symulacja zjawiska jest wykonywana na podstawie jego opisu. Opis obejmuje najprostsze fakty charakteryzujące mechanizm zjawiska, ale nie obejmuje wielu własności, wynikających wprawdzie z opisu, ale trudnych do przewidzenia. Symulacja pozwala sprawdzić czy zjawisko ma pewne własności, obliczyć różne charakterystyki itd. Dzięki użyciu komputera można to wykonać w stosunkowo krótkim czasie.

W przypadku zjawisk losowych symulacja daje materiał statystyczny do szacowania interesujących charakterystyk metodami statystyki matematycznej. Jest to o tyle istotne, że obliczenie takich charakterystyk bezpośrednio na podstawie opisu zjawiska jest bardzo żmudne, a często wręcz niewykonalne.

Niekiedy przez symulację zjawisk fizycznych można uzyskać rozwiązanie problemów czysto matematycznych. Przykładem może być problem wyznaczania długości najkrótszej drogi od wierzchołka A do wierzchołka B w grafie skierowanym, którego łukom są

przypisane pewne długości. Dla rozwiązania wystarczy symulować rozchodzenie się (ze stałą prędkością) wzdłuż łuków grafu pewnego zaburzenia spowodowanego chwilowym impulsem w wierzchołku A . Szukana długość jest iloczynem prędkości rozchodzenia się zaburzenia i czasu potrzebnego na pierwsze dotarcie zaburzenia od A do B (por. [3]).

Symulacja może dostarczyć informacji o charakterze poszukiwanych rozwiązań niektórych problemów matematycznych, wyobrażeń o procesach fizycznych, chemicznych, biologicznych, ekonomicznych itp.

Wszystko to czyni z metody symulacji wygodne narzędzie badawcze, które może być wykorzystywane do weryfikowania różnych teorii zjawisk, prognozowania, badania działania projektowanych urządzeń czy organizacji i tym podobnych celów.

2. Pojęcie opisu i modelu zjawiska. W praktyce dysponujemy nieformalnym opisem zjawiska. Ten opis, albo inaczej *teorię* zjawiska, staramy się wyrazić za pomocą pojęć matematycznych, tzn. skonstruować *model* matematyczny.

Ze zjawiskiem wiążemy pewien zbiór *stanów* S i *prawa*, według których osiąga ono stany z S , poczynając od pewnego początkowego $s_0 \in S$.

W przypadku deterministycznym staramy się znaleźć takie funkcje τ, ϕ , by działanie praw sprowadziło się do tego, że zjawisko osiągnąwszy w chwili t stan $s \in S$ pozostaje w nim do chwili

$$(1) \quad \bar{t} = \tau(t, s),$$

następnie przechodzi do stanu

$$(2) \quad \bar{s} = \phi(t, s)$$

itd. Możliwość znalezienia funkcji τ, ϕ zależy od sposobu charakteryzacji stanów zjawiska, tzn. od wyboru zbioru S . Funkcje τ, ϕ określone są (być może częściowo) w $T \times S$, gdzie T jest zbiorem chwil (liczby rzeczywiste), i przyjmują odpowiednio wartości z T i S . Funkcja τ reprezentuje upływ czasu i w związku z tym powinna spełniać warunek

$$(3) \quad t \leq \tau(t, s).$$

Zbiór S i funkcje τ, ϕ można oczywiście wybierać na wiele sposobów.

PRZYKŁAD 1. Niech zjawisko polega na obsłudze zgłaszających się klientów przez pewną stację obsługi. Początkowo stacja nie pracuje. W chwili $t = 0$ zgłasza się klient i stacja rozpoczyna jego obsługiwanie. Następni klienci są obsługiwani, jeśli zastają stację wolną, bądź czekają w kolejce na zwolnienie stacji. Klienci zgłaszają się w stałych odstępach czasu o długości d , a czas obsłużenia jednego klienta wynosi e .

Charakteryzujemy stan zjawiska przez liczbę q klientów w kolejce, chwilę α zgłoszenia się ostatniego klienta i chwilę β ostatniego rozpoczęcia obsługi. Ze zjawiskiem wiążemy zbiór S wszystkich takich stanów, tzn. trójek (q, α, β) z całkowitym $q \geq 0$ i z rzeczywistymi α, β . Przez T oznaczamy zbiór nieujemnych liczb rzeczywistych. Prawa, według których przebiega zjawisko, wyrażamy przez funkcje τ, ϕ określone w

$$(4) \quad X_0 = \{(t, q, \alpha, \beta) \in T \times S \mid \alpha \leq t \leq \alpha + d, \beta \leq t\}$$

formułami

$$(5) \quad \tau(t, q, \alpha, \beta) = \begin{cases} \min\{\alpha + d, \beta + e\} & \text{dla } t < \beta + e, \\ t & \text{dla } \beta + e \leq t \text{ i } q > 0, \\ \alpha + d & \text{dla } \beta + e \leq t \text{ i } q = 0, \end{cases}$$

$$\phi(t, q, \alpha, \beta) = \begin{cases} (q+1, \alpha+d, \beta) & \text{dla } t < \beta + e \text{ i } \alpha + d \leq \beta + e, \\ & \text{lub } \beta + e \leq t \text{ i } \alpha + d = t, \\ & \text{lub } \beta + e \leq t \text{ i } q = 0, \\ (q-1, \alpha, t) & \text{dla } \beta + e \leq t < \alpha + d \text{ i } q > 0, \\ (q, \alpha, \beta) & \text{dla } t < \beta + e < \alpha + d. \end{cases}$$

Jeśli bowiem w chwili t stacja pracuje ($t < \beta + e$), to najbliższą chwilą zmiany stanu jest chwila pierwszego zgłoszenia się klienta (gdy $\alpha + d \leq \beta + e$) lub chwila zakończenia wykonywanej obsługi (gdy $\beta + e < \alpha + d$). Jeśli stacja jest wolna ($\beta + e \leq t$), to albo czeka ją klienci ($q > 0$) i w chwili t stacja zaczyna pracować, albo nie ma klientów ($q = 0$) aż do chwili pierwszego zgłoszenia się klienta. Natomiast osiągnięcie jednego ze stanów $(q + 1, \alpha + d, \beta)$, $(q - 1, \alpha, t)$, (q, α, β) oznacza odpowiednio zgłoszenie się klienta, rozpoczęcie obsługi i zakończenie obsługi. *

W przypadku losowym uzależniamy dodatkowo działanie praw od pewnych czynników losowych tak, by

$$(6) \quad \bar{t} = \tau(t, s, \omega), \quad \bar{s} = \phi(t, s, \omega),$$

gdzie ω reprezentuje czynniki losowe i pochodzi z pewnego zbioru Ω z σ -ciałem podzbiorów \mathcal{F} oraz z miarą prawdopodobieństwa P określoną na \mathcal{F} (a więc jest elementem przestrzeni prawdopodobieństwa (Ω, \mathcal{F}, P)). Wtedy można przedefiniować zbiór stanów i prawa przyjmując

$$(7) \quad S' = S \times \Omega, \quad \phi'(t, s, \omega) = (\phi(t, s, \omega), \omega),$$

i sprowadzić tym samym losowość zjawiska do losowości wyboru stanu początkowego $s'_0 = (s_0, \omega)$. Wówczas stan początkowy staje się zmienną losową σ określoną na przestrzeni prawdopodobieństwa (Ω, \mathcal{F}, P) formułą

$$(8) \quad \sigma(\omega) = (s_0, \omega),$$

i przyjmującą wartości z $S' = S \times \Omega$.

PRZYKŁAD 2. Niech w zjawisku obsługi z przykładu 1 odstępów między zgłoszeniami kolejnych klientów i kolejne czasy obsługi będą odpowiednio wartościami rzeczywistych zmiennych losowych $D_1, D_2, \dots, E_1, E_2, \dots$ określonych na przestrzeni prawdopodobieństwa (Ω, \mathcal{F}, P) .

Stany zjawiska charakteryzujemy dodatkowo przez liczbę m zgłoszonych klientów i przez liczbę n klientów, których obsługa została rozpoczęta. Ze zjawiskiem wiążemy zbiór S wszystkich takich stanów (tzn. układów (q, α, β, m, n) z całkowitymi $q \geq 0$, $m \geq 0$, $n \geq 0$ i rzeczywistymi α, β), a prawa rządzące przebiegiem zjawiska wyrażamy przez funkcje τ, ϕ określone w

$$(9) \quad X_0 = \left\{ (t, q, \alpha, \beta, m, n, \omega) \in T \times S \times \Omega \mid \alpha \leq t \leq \alpha + D_m(\omega), \beta \leq t \right\}$$

formułami

$$(10) \quad \tau(t, q, \alpha, \beta, m, n, \omega) = \begin{cases} \min\{\alpha + D_m(\omega), \beta + E_n(\omega)\} & \text{dla } t < \beta + E_n(\omega), \\ t & \text{dla } \beta + E_n(\omega) \leq t \text{ i } q > 0, \\ \alpha + D_m(\omega) & \text{dla } \beta + E_n(\omega) \leq t \text{ i } q = 0, \end{cases}$$

$$\phi(t, q, \alpha, \beta, m, n, \omega) = \begin{cases} (q+1, \alpha + D_m(\omega), \beta, m+1, n) & \text{dla } t < \beta + E_n(\omega) \\ & \text{i } \alpha + D_m(\omega) \leq \beta + E_n(\omega), \\ & \text{lub } \beta + E_n(\omega) \leq t \\ & \text{i } \alpha + D_m(\omega) = t, \\ & \text{lub } \beta + E_n(\omega) \leq t \text{ i } q = 0, \\ (q-1, \alpha, t, m, n+1) & \text{dla } \beta + E_n(\omega) \leq t < \alpha + D_m(\omega) \text{ i } q > 0, \\ (q, \alpha, \beta, m, n) & \text{dla } t < \beta + E_n(\omega) < \alpha + D_m(\omega). \end{cases}$$

Natomiast $\phi'(t, q, \alpha, \beta, m, n, \omega)$ różni się od ϕ tym, że po prawej stronie (10) występują odpowiednio $(q+1, \alpha + D_m(\omega), \beta, m+1, n, \omega)$, $(q-1, \alpha, t, m, n+1, \omega)$, $(q, \alpha, \beta, m, n, \omega)$.

Warto zwrócić uwagę, że wzbogacenie opisu stanu o liczby m, n było potrzebne do wskazania właściwych zmiennych losowych (D_m, E_n) w definicji funkcji τ, ϕ .

Gdy zjawisko daje się scharakteryzować jak wyżej, powstaje możliwość obliczania na podstawie stanu początkowego i chwili początkowej kolejnych stanów i chwil ich osiągnięcia. Wystarczy w tym celu wykonywać jednoargumentową operację F określoną (być może częściowo, jeśli funkcja τ lub ϕ jest częściowa) w $X = T \times S$ formułą

$$(11) \quad F(t, s) = (\tau(t, s), \phi(t, s)).$$

Otrzymujemy ciąg

$$(12) \quad x_0, x_1 = F(x_0), x_2 = F(x_1), \dots$$

elementów zbioru X nieskończony lub kończący się elementem $x_k \in X$ z nieokreślonym $F(x_k)$.

Struktura relacyjna

$$(13) \quad M = (X; F)$$

stanowi algorytm abstrakcyjny, który będziemy nazywali *algorytmem symulacji* zjawiska. Stany tego algorytmu reprezentują stany zjawiska i chwile ich osiągnięcia.

Otrzymany ciąg (12) jest obliczeniem $\text{Obl}(M, x_0)$ algorytmu symulacji M ze stanem początkowym x_0 . W przypadku deterministycznym obliczenie $\text{Obl}(M, x_0)$ jest wyznaczone jednoznacznie przez x_0 (gdzie $x_0 = (t_0, s_0)$, t_0 jest chwilą rozpoczęcia, a s_0 — stanem początkowym zjawiska) i to właśnie obliczenie nazywamy *modelem zjawiska*. Natomiast w przypadku losowym mamy do czynienia z tzw. obliczeniem losowym $\text{Obl}(M, \xi)$ (gdzie $\xi = (t_0, \sigma)$, t_0 jest chwilą rozpoczęcia; a σ — losowym stanem początkowym zjawiska) i to obliczenie losowe nazywamy *modelem zjawiska*. Ma ono wiele reali-

zacji postaci $\text{Obl}(M, x_0)$ odpowiadających różnym wartościom $x_0 = (t_0, s_0)$ zmiennej losowej $\xi = (t_0, \sigma)$. Wszystkie te realizacje są obliczeniami tego samego algorytmu M .

Zjawisko może mieć wiele algorytmów symulacji, ponieważ S , τ i ϕ można wybrać na wiele sposobów.

PRZYKŁAD 3. Algorytm $M = (X; F)$ symulacji zjawiska z przykładu 1 otrzymujemy określając w $X_0 \subset X$ operację F według (11); łatwo sprawdzić, że dziedziną operacji F jest zbiór X_0 i dla wszystkich $(t, q, \alpha, \beta) \in X_0$

$$(14) \quad F(t, q, \alpha, \beta) = \begin{cases} (\alpha+d, q+1, \alpha+d, \beta) & \text{dla } t < \beta + e \text{ i } \alpha + d \leq \beta + e, \\ & \text{lub } \beta + e \leq t \text{ i } \alpha + d = t, \\ & \text{lub } \beta + e \leq t \text{ i } q = 0, \\ (t, q-1, \alpha, t) & \text{dla } \beta + e \leq t < \alpha + d \text{ i } q > 0, \\ (\beta + e, q, \alpha, \beta) & \text{dla } t < \beta + e < \alpha + d, \end{cases}$$

oraz $F(t, q, \alpha, \beta) \in X_0$.

Jeśli np. $d = 3$, $e = 5$ i $x_0 = (0, 0, 0, 0)$, to kolejnymi stanami obliczenia $\text{Obl}(M, x_0)$ są

$$(15) \quad \begin{aligned} x_0 &= (0, 0, 0, 0), & x_1 &= (3, 1, 3, 0), & x_2 &= (5, 1, 3, 0), \\ x_3 &= (5, 0, 3, 5), & x_4 &= (6, 1, 6, 5), & x_5 &= (9, 2, 9, 5), \\ x_6 &= (10, 2, 9, 5), & x_7 &= (10, 1, 9, 10) \text{ itd.} \end{aligned}$$

i obliczenie to (nieskończone) jest modelem zjawiska. Obserwując je możemy stwierdzić jak zmienia się wielkość kolejki z upływem czasu, kiedy kolejka osiągnie pewną krytyczną długość itd. Zjawisko jest jednak na tyle proste, że te same wyniki można uzyskać bez symulacji, stosując odpowiednie wzory. *

PRZYKŁAD 4. Algorytm symulacji zjawiska z przykładu 2 otrzymujemy przyjmując $X = T \times S \times \Omega$ i $M = (X; F)$, gdzie dla $(t, q, \alpha, \beta, m, n, \omega)$ ze zbioru X_0 określonego formułą (9)

$$(16) \quad F(t, q, \alpha, \beta, m, n, \omega) = \begin{cases} (\alpha + D_m(\omega), q + 1, \alpha + D_m(\omega), \beta, m + 1, n, \omega) & \text{dla } t < \beta + E_n(\omega) \\ & \text{i } \alpha + D_m(\omega) \leq \beta + E_n(\omega), \\ & \text{lub } \beta + E_n(\omega) \leq t \\ & \text{i } \alpha + D_m(\omega) = t, \\ & \text{lub } \beta + E_n(\omega) \leq t \text{ i } q = 0, \\ (t, q-1, \alpha, t, m, n+1, \omega) & \text{dla } \beta + E_n(\omega) \leq t < \alpha + D_m(\omega) \\ & \text{i } q > 0, \\ (\beta + E_n(\omega), q, \alpha, \beta, m, n, \omega) & \text{dla } t < \beta + E_n(\omega) < \alpha + D_m(\omega). \end{cases}$$

W tym przypadku modelem zjawiska jest proces losowy $\text{Obl}(M, \xi)$, gdzie $\xi = (0, \sigma)$, a σ jest losowym stanem początkowym zjawiska określonym na (Ω, \mathcal{F}, P) formułą

$$(17) \quad \sigma(\omega) = (0, 0, 0, 1, 1, \omega).$$

Przewidzenie wielkości kolejki w zadanej chwili t nie tylko nie jest takie proste jak w przykładzie 3, ale traci w ogóle sens wobec losowości zjawiska. Można co najwyżej mówić o oczekiwanej wielkości kolejki w chwili t . Teoretycznie, znając zmienne losowe $D_1, D_2, \dots, E_1, E_2, \dots$, można próbować ją obliczyć na drodze czysto analitycznej. W praktyce jest to niewykonalne bez specjalnych założeń natury probabilistycznej. Natomiast można szacować tę wielkość metodami statystyki matematycznej, dysponując odpowiednimi danymi z obserwacji zjawiska. Rola symulacji polega na tym, że symulacja może dostarczyć zastępczych danych, równoważnych oryginalnym z punktu widzenia statystyki matematycznej. W tym celu wystarczy losować $\omega \in \Omega$ według miary prawdopodobieństwa P , wykonywać obliczenie $\text{Obl}(M, (0, \sigma(\omega)))$ do osiągnięcia chwili t i rejestrować otrzymaną wielkość kolejki. Opis różnych technik losowania znajdzie Czytelnik w monografiach poświęconych metodom Monte Carlo (np. w [8]). *

3. Konstruowanie algorytmu symulacji. Konstruowanie algorytmów symulacji nastrocza pewne trudności już w prostych przypadkach (por. podane wyżej przykłady). Z drugiej strony, metoda symulacji stosowana jest zwykle do badania skomplikowanych zjawisk (w stosunku do których zawodzą np. inne metody). Dlatego dąży się do rozłożenia zjawiska na możliwie proste składowe. Te składowe odpowiadają zwykle pewnym konkretnym lub fikcyjnym obiektom. Można np. uważać, że w zjawisku, o którym mówiliśmy wyżej, mamy do czynienia z pewnym źródłem klientów i ze stacją obsługi. Można było by brać pod uwagę również klientów.

Staramy się wyspecyfikować zbiór I wszystkich składowych w ten sposób, by każdą zmianę stanu konstruowanego algorytmu symulacji $M = (X; F)$ postaci $\bar{x} = F(x)$ dało się objaśnić akcją pewnej składowej $\mu(x) \in I$. Dokładniej chodzi o to, aby działanie operacji F opisać przez działania pewnych operacji $\bar{f}_{\mu(x)}$ charakterystycznych dla składowych $\mu(x)$ tak, by $F(x) = \bar{f}_{\mu(x)}(x)$. Wtedy algorytm M możemy opisać przez zbiór stanów X , zbiór składowych I , funkcję $\mu: D_\mu \rightarrow I$ z dziedziną $D_\mu \subseteq X$ i przez rodzinę jednoargumentowych (być może częściowych) operacji \bar{f}_i ($i \in I$) w X . Funkcja μ wyznacza składową $\mu(x)$ działającą w stanie $x \in D_\mu$ ($x \notin D_\mu$ oznacza że żadna składowa nie może działać), a \bar{f}_i jest transformacją stanu algorytmu odpowiadająca akcji składowej $i \in I$ (fakt, że \bar{f}_i nie jest określona dla x oznacza, że składowa i nie może działać w stanie x).

PRZYKŁAD 5. Zjawisko z przykładu 1 rozkładamy na dwie składowe A, B zwane odpowiednio źródłem klientów i stacją obsługi. Wtedy $I = \{A, B\}$.

Wszystkie akcje źródła A polegają na dostarczeniu jednego klienta. Każdej takiej akcji odpowiada wykonanie transformacji \bar{f}_A stanu algorytmu symulacji, gdzie \bar{f}_A jest operacją w X określoną dla wszystkich $(t, q, \alpha, \beta) \in X$, dla których $\alpha \leq t \leq \alpha + d$, formułą

$$(17) \quad \bar{f}_A(t, q, \alpha, \beta) = (\alpha + d, q + 1, \alpha + d, \beta).$$

Wielkość $\alpha + d$ reprezentuje tu chwilę osiągnięcia przez zjawisko stanu $(q + 1, \alpha + d, \beta)$.

Natomiast akcje stacji B polegają na rozpoczęciu obsługi lub na zakończeniu trwającej obsługi. Akcja rozpoczęcia obsługi może mieć miejsce tylko wtedy, gdy stacja jest wolna ($\beta + e \leq t$) i są klienci ($q > 0$), a akcja zakończenia trwającej obsługi tylko wtedy, gdy stacja pracuje ($t < \beta + e$). Zatem każdej akcji stacji B odpowiada wykonanie transformacji \bar{f}_B , gdzie dla $(t, q, \alpha, \beta) \in X$ i $\beta \leq t$

$$(18) \quad \bar{f}_B(t, q, \alpha, \beta) = \begin{cases} (t, q-1, \alpha, t) & \text{dla } \beta + e \leq t \text{ i } q > 0, \\ (\beta + e, q, \alpha, \beta) & \text{dla } t < \beta + e, \\ \text{nieokreślona w pozostałych przypadkach.} \end{cases}$$

Wielkość $\beta + e$ w stanie $(\beta + e, q, \alpha, \beta)$ reprezentuje chwilę \bar{t} osiągnięcia przez zjawisko stanu (q, α, β) , w którym stacja B jest wolna ($\beta + e \leq \bar{t}$).

Funkcja $\mu : X_0 \rightarrow \{A, B\}$ wyznaczająca składowe działające w poszczególnych stanach jest określona następująco:

$$(19) \quad \mu(t, q, \alpha, \beta) = \begin{cases} A & \text{dla } t < \beta + e \text{ i } \alpha + d \leq \beta + e, \text{ lub } \beta + e \leq t \text{ i } \alpha + d = t, \\ & \text{lub } \beta + e \leq t \text{ i } q = 0, \\ B & \text{dla } \beta + e \leq t < \alpha + d \text{ i } q > 0 \text{ lub } t < \beta + e < \alpha + d. \end{cases}$$

Łatwo sprawdzić, że funkcja F' określona (dla wszystkich $(t, q, \alpha, \beta) \in X_0$) wzorem

$$F'(t, q, \alpha, \beta) = \bar{f}_{\mu(t, q, \alpha, \beta)}(t, q, \alpha, \beta)$$

jest identyczna z funkcją F określoną wzorem (14). *

PRZYKŁAD 6. W przypadku przykładu 2 można rozumować podobnie. Otrzymujemy

$$(20) \quad \bar{f}_A(t, q, \alpha, \beta, m, n, \omega) = (\alpha + D_m(\omega), q + 1, \alpha + D_m(\omega), \beta, m + 1, n, \omega)$$

przy $\alpha \leq t \leq \alpha + D_m(\omega)$,

$$(21) \quad \bar{f}_B(t, q, \alpha, \beta, m, n, \omega) = \begin{cases} (t, q-1, \alpha, t, m, n+1, \omega) & \text{dla } \beta + E_n(\omega) \leq t \text{ i } q > 0, \\ (\beta + E_n(\omega), q, \alpha, \beta, m, n, \omega) & \text{dla } t < \beta + E_n(\omega), \\ \text{nieokreślona w pozostałych przypadkach} \end{cases}$$

przy $\beta \leq t$,

$$(22) \quad \mu(t, q, \alpha, \beta, m, n, \omega) = \begin{cases} A & \text{dla } t < \beta + E_n(\omega) \text{ i } \alpha + D_m(\omega) \leq \beta + E_n(\omega), \\ & \text{lub } \beta + E_n(\omega) \leq t \text{ i } \alpha + D_m(\omega) = t, \\ & \text{lub } \beta + E_n(\omega) \leq t \text{ i } q = 0, \\ B & \text{dla } \beta + E_n(\omega) \leq t < \alpha + D_m(\omega) \text{ i } q > 0 \\ & \text{lub } t < \beta + E_n(\omega) < \alpha + D_m(\omega), \end{cases}$$

przy wszystkich $(t, q, \alpha, \beta, m, n, \omega)$ ze zbioru X_0 określonego formułą (9). *

Struktury relacyjne $\bar{M}_i = (X; \bar{f}_i)$ odpowiadające składowym $i \in I$ stanowią znów algorytmy. Problem skonstruowania algorytmu symulacji M sprowadza się do skonstruowania algorytmów \bar{M}_i , opisujących w istocie działanie pojedynczych składowych w zjawisku,

oraz funkcji μ , wyznaczającej działające składowe. Nie eliminuje to jednak trudności, ponieważ nie wiadomo z góry, jak wybrać zbiór X stanów algorytmu symulacji. Dlatego łatwiej jest najpierw skonstruować niezależne od siebie algorytmy $M_i = (X_i; f_i)$ symulujące działanie składowych $i \in I$, a następnie utworzyć z nich algorytm symulacji $M = (X; F)$.

Stan $x_i \in X_i$ algorytmu M_i powinien opisywać chwilę osiągnięcia przez zjawisko aktualnego stanu, zwaną dalej *chwilą bieżącą*, i aktualny stan składowej $i \in I$, i powinien odpowiadać jednemu ze stanów $x \in X$ nieznanego jeszcze algorytmu symulacji M . Nie można jednak wymagać tego od wszystkich układów $(x_i)_{i \in I}$ stanów $x_i \in X_i$, ponieważ poszczególne x_i mogą odpowiadać różnym $x \in X$. Tak np. chwile bieżące w x_i, x_j mogą być różne, może brakować naturalnych zależności między stanami składowych itd. Nie mniej jednak, nawet nie znając algorytmu M , możemy zwykle stwierdzić czy układ jako całość jest odpowiednikiem pewnego stanu $x \in X$ (tzn. czy każdy ze stanów x_i odpowiada stanowi x). Trzeba w tym celu odwołać się do wynikającej z opisu zjawiska interpretacji układu jako całości. Jeśli bowiem taka interpretacja jest możliwa, tzn. wszystkie spośród stanów x_i ($i \in I$) odpowiadają stanowi s zjawiska i chwili t jego osiągnięcia, to muszą odpowiadać pewnemu stanowi algorytmu symulacji, albowiem para (t, s) powinna być reprezentowana przez jeden ze stanów tego algorytmu. Wszystkie układy $(x_i)_{i \in I}$ stanów $x_i \in X_i$, które mają interpretację w zjawisku w powyższym znaczeniu, tworzą pewien podzbiór Y produktu $\prod_{i \in I} X_i$.

PRZYKŁAD 7. Weźmy pod uwagę zjawisko z przykładu 1. Stan algorytmu M_A symulacji działania źródła klientów A opisujemy przez liczbę K klientów, którzy mają być obsłużeni (przez nieznaną chwilowo stację obsługi), chwilę a dostarczenia ostatniego klienta i chwilę bieżącą u . Wobec tego stanami algorytmu M_A są trójki (u, K, a) z całkowitym $K \geq 0$ i rzeczywistymi u, a . Natomiast funkcją przejścia jest

$$(23) \quad f_A(u, K, a) = \begin{cases} (a + d, K + 1, a + d) & \text{dla } a \leq u \leq a + d, \\ \text{nieokreślona w pozostałych przypadkach.} \end{cases}$$

Podobnie stan algorytmu M_B symulacji działania stacji B opisujemy przez liczbę Q klientów czekających na obsłużenie (niezależnie od tego skąd pochodzą), chwilę b ostatniego rozpoczęcia obsługi i chwilę bieżącą v . Zbiór X_B stanów tego algorytmu składa się z trójek (v, Q, b) z całkowitym $Q \geq 0$ i z rzeczywistymi v, b , a jego funkcją przejścia jest

$$(24) \quad f_B(v, Q, b) = \begin{cases} (v, Q - 1, v) & \text{dla } b + e \leq v \text{ i } Q > 0, \\ (b + e, Q, b) & \text{dla } b \leq v < b + e, \\ \text{nieokreślona w pozostałych przypadkach.} \end{cases}$$

Wzajemne oddziaływanie źródła A i stacji B na siebie w zjawisku polega na tym, że stacja B obsługuje tych klientów, których dostarcza źródło A , a więc K, Q oznaczają tę samą liczbę. Poza tym u, v oznaczają chwilę wskazywaną przez jeden zegar. Trzeba to brać pod uwagę interpretując stany obu algorytmów w zjawisku. Zatem interpretację w zjawisku mają te układy $((u, K, a), (v, Q, b))$ stanów algorytmów A, B z produktu $X_A \times X_B$, dla których $u = v$ i $K = Q$. Tworzą one zbiór

$$Y = \left\{ ((u, K, a), (v, Q, b)) \in X_A \times X_B \mid u = v \text{ i } K = Q \right\}. *$$

Na to, by z algorytmów M_i symulujących działanie składowych $i \in I$ można było utworzyć algorytm symulacji M , trzeba je skonstruować tak, że

- (i) jeśli $(x_i)_{i \in I} \in Y$ i może nastąpić akcja i -tej składowej, to stan x_i algorytmu M_i determinuje wpływ tej akcji na stany wszystkich M_j ($i \in I$), a otrzymany układ $(\bar{x}_i)_{i \in I}$ należy do Y ,
- (ii) każdy układ $(x_i)_{i \in I} \in Y$ determinuje działającą składową.

Wtedy bowiem jako zbiór stanów algorytmu symulacji można wziąć dowolny zbiór X , który przez pewne przekształcenie h daje się przekształcić na Y , ponieważ wszystkie elementy $x \in X$ takiego zbioru mają interpretację w zjawisku (taką jak $h(x)$) i można w nim określić operacje f_i opisujące akcje składowych, oraz funkcję μ wyznaczającą działające składowe.

Własność (i) oznacza bowiem istnienie funkcji f_{ij} przyporządkowujących stanom x_i, x_j algorytmów M_i, M_j z układów $(x_i)_{i \in I} \in Y$ stany \bar{x}_j algorytmów M_j z układów $(\bar{x}_i)_{i \in I} \in Y$ otrzymywanych z $(x_i)_{i \in I}$ w wyniku akcji i -tej składowej. Te funkcje f_{ij} są określone (na ogół częściowo) w $X_i \times X_j$ i przyjmują wartości z X_j , przy czym dla każdego $i \in I, x_i \in X_i$ wartości $f_{ii}(x_i, x_i), f_i(x_i)$ są równocześnie określone oraz

$$(25) \quad f_{ii}(x_i, x_i) = f_i(x_i),$$

gdy są określone. Otrzymuje się je przez interpretację układów $(x_i)_{i \in I} \in Y$ w zjawisku. Dlatego dowolne operacje \bar{f}_i w X spełniające dla $(x_i)_{i \in I} = h(x)$ warunki

$$(26) \quad h(\bar{f}_i(x)) = (f_{ij}(x_i, x_j))_{j \in I}$$

i określone dla tych $x \in X$, dla których wszystkie $f_{ij}(x_i, x_j)$ ($j \in I$) są określone, opisują akcje składowych.

Z kolei własność (ii) oznacza istnienie funkcji ν przyporządkowującej niektórym układom $y \in Y$ działającą składową $\nu(y) \in I$. Złożenie μ funkcji h, ν jest więc funkcją przyporządkowującą niektórym elementom $x \in X$ działającą składową $\mu(x) \in I$.

PRZYKŁAD 8. Z interpretacji układów stanów algorytmów M_A, M_B w przykładzie 7 wynika, że wpływ akcji każdej ze składowych A, B na stany algorytmów M_A, M_B wyraża się przez funkcje

$$(27) \quad f_{AA}((u, K, a), (\bar{u}, \bar{K}, \bar{a})) = \begin{cases} f_A(u, K, a), & \text{gdy } (u, K, a) = (\bar{u}, \bar{K}, \bar{a}) \text{ i } f_A(u, K, a) \text{ określone,} \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases}$$

$$f_{AB}((u, K, a), (v, Q, b)) = \begin{cases} (\bar{u}, \bar{K}, b), & \text{gdy } u=v \text{ i } K=Q \text{ i } f_A(u, K, a) \text{ określone} \\ & \text{i } f_A(u, K, a) = (\bar{u}, \bar{K}, \bar{a}), \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases}$$

$$f_{BB}((v, Q, b), (\bar{v}, \bar{Q}, \bar{b})) = \begin{cases} f_B(v, Q, b), & \text{gdy } (v, Q, b) = (\bar{v}, \bar{Q}, \bar{b}) \text{ i } f_B(v, Q, b) \text{ określone,} \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases}$$

$$f_{BA}((u, K, a), (v, Q, b)) = \begin{cases} (\bar{v}, \bar{Q}, a), & \text{gdy } u=v \text{ i } K=Q \text{ i } f_B(v, Q, b) \text{ określone} \\ & \text{i } f_B(v, Q, b) = (\bar{v}, \bar{Q}, \bar{b}), \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases}$$

a każdy układ $((u, K, a), (v, Q, b)) \in Y$ ($z u = v, K = Q$) determinuje działającą składową

$$(28) \quad \nu((u, K, a), (v, Q, b)) = \begin{cases} A & \text{dla } u < b + e \text{ i } a + d \leq b + e \text{ lub} \\ & b + e \leq u \text{ i } a + d = u \text{ lub } b + e \leq u \text{ i } K = 0, \\ B & \text{dla } b + e \leq u \leq a + d \text{ i } q > 0 \text{ lub} \\ & u < b + e < a + d. \end{cases}$$

Zatem para algorytmów M_A, M_B ma własności (i), (ii). Jako zbiór stanów algorytmu symulacji można wybrać zbiór X z przykładów 3 i 5, ponieważ przekształcenie h określone wzorem

$$(29) \quad h(t, q, \alpha, \beta) = ((t, q, \alpha), (t, q, \beta))$$

transformuje wzajemnie jednoznacznie X na Y . Natomiast w charakterze operacji opisujących akcje składowych można wziąć funkcje \bar{f}_A, \bar{f}_B z przykładu 5, ponieważ spełniają one warunki (26). Podobnie, składając h i ν otrzymujemy funkcję μ określoną wzorem (19). *

Algorytmy M_i , których rodzina ma własności (i), (ii) i których każdy stan wchodzi w pewien układ $(x_i)_{i \in I} \in Y$, nazywamy *algorytmami symulacji działania składowych zjawiska* lub krótko *algorytmami składowych*. Warto podkreślić, że każdy z nich konstruuje się niezależnie od pozostałych, a oddziaływanie odpowiedniej składowej na pozostałe wyraża się dopiero przez interpretację jego stanów w zjawisku. Trzeba jednak pamiętać, że dla zagwarantowania własności (i), (ii) stan algorytmu każdej składowej powinien zawierać wystarczające informacje o zjawisku. Ze względu na własność (i) powinien on zawierać informację o tym, o d c z e g o z a l e ż y sposób działania składowej, i informację n a k t ó r e w p ł y w a działanie składowej. Własność (ii) może być zagwarantowana metodą, którą opiszemy w następnym punkcie.

PRZYKŁAD 9. Algorytmy M_A, M_B z przykładu 7 tworzą rodzinę algorytmów składowych A, B zjawiska. Własności (i), (ii) stwierdziliśmy w przykładzie 8. Każdy stan $(u, K, a) \in X_A$ wchodzi w każdy układ $((u, K, a), (u, K, b)) \in Y$, a stan $(v, Q, b) \in X_B$ w układ $((v, Q, a), (v, Q, b)) \in Y$. *

Niech algorytmy $M_i = (X_i, f_i)$ stanowią rodzinę algorytmów składowych $i \in I$ i niech Y, X, h, \bar{f}_i, μ będą takie jak wyżej. Wtedy rzutowania $pr_j : \prod_{i \in I} X_i \rightarrow X_j$ przekształcają Y na całe X_i (własność rodziny algorytmów składowych), i tym samym przekształcenia $h_i : X \rightarrow X_i$, powstałe ze złożenia h z pr_i , przekształcają X na X_i .

Przekształcenia h_i wyznaczają w zbiorze X rodzinę równoważności

$$(30) \quad x \equiv_i \bar{x} \text{ wtedy i tylko wtedy, gdy } h_i(x) = h_i(\bar{x}).$$

Z własności (i) wynika, że równoważności \equiv_i ($i \in I$) spełniają następujący warunek:

(iii) dla każdego $J \subseteq I$, jeśli $x \equiv_i \bar{x}$ dla wszystkich $i \in J$, to dla każdego $j \in J$ $f_j(x),$

$\bar{f}_j(\bar{x})$ są równocześnie określone lub nie i, jeśli są określone, to $\bar{f}_j(x) \equiv_i \bar{f}_j(\bar{x})$ dla wszystkich $i \in J$.

Warunek ten oznacza, że dla każdego $J \subseteq I$ równoważność $\equiv_J = \bigcap_{i \in J} \equiv_i$ (taka że $x \equiv_J \bar{x}$

wtedy i tylko wtedy, gdy $x \equiv_i \bar{x}$ dla wszystkich $i \in J$) jest stabilna względem każdej operacji \bar{f}_j ($j \in J$). Jeśli jest on spełniony, to przy naturalnym przekształceniu

$$\pi_J : X \rightarrow X/\equiv_J$$

zbioru X na zbiór X/\equiv_J warstw elementów \equiv_J -równoważnych, przyporządkowującym stanom $x \in X$ warstwy $\pi_J(x)$ stanów \equiv_J -równoważnych z x , każda z operacji \bar{f}_j ($j \in J$) przechodzi na relację

$$\pi_J(\bar{f}_j) = \left\{ (x_J, \bar{x}_J) \in X/\equiv_J \times X/\equiv_J \mid x_J = \pi_J(x) \text{ i } \bar{x}_J = \pi_J(\bar{x}) \text{ i } \bar{x} = \bar{f}_j(x) \right. \\ \left. \text{dla pewnych } x \in X, \bar{x} \in X \right\},$$

która znów jest operacją (w X/\equiv_J). W szczególności dla każdego $i \in I$

$$(31) \quad f'_i = \pi_i(\bar{f}_i)$$

jest operacją w $X'_i = X/\equiv_i$, dzięki czemu struktura relacyjna

$$(32) \quad M'_i = (X'_i; f'_i)$$

stanowi algorytm. Ponieważ $x \equiv_i \bar{x}$ oznacza równość rzutów $pr_i(h(x))$, $pr_i(h(\bar{x}))$, więc istnieje przekształcenie różnowartościowe

$$\varphi_i : X'_i \rightarrow X_i$$

takie, że

$$\varphi_i(\pi_i(x)) = h_i(x),$$

przy czym $\varphi_i(X'_i) = X_i$. Równocześnie z własności (25), (26) operacji \bar{f}_i , f_i wynika, że $\varphi_i(f'_i) = f_i$. Innymi słowy, algorytmy M'_i , M_i są izomorficzne.

Ostatecznie, z konstruowanym algorytmem symulacji $M = (X; F)$ wiąże się pewna struktura relacyjna

$$(33) \quad \Sigma = (X; \{\bar{f}_i\}_{i \in I}, \{\equiv_i\}_{i \in I})$$

z nośnikiem X , jednoargumentowymi operacjami \bar{f}_i określonymi (być może częściowo) w X i z równoważnościami \equiv_i w X , spełniająca warunek (iii) i taka, że algorytmy M'_i określone wzorem (32) są izomorficzne z odpowiednimi algorytmami M_i składowych $i \in I$. Tego rodzaju strukturę nazywamy *systemem* nad rodziną algorytmów M_i ($i \in I$).

Prócz systemu Σ nad $\{M_i\}_{i \in I}$, z algorytmem symulacji wiąże się jeszcze funkcja μ określona w X o wartościach w I , wyznaczająca działające składowe, którą nazywamy *funkcją sterującą* systemu Σ . Para (Σ, μ) , zwana *systemem sterowanym*, wyznacza w pełni algorytm

$$(34) \quad \mathcal{M}(\Sigma, \mu) = (X; F),$$

gdzie $F(x) = \bar{f}_{\mu(x)}(x)$ dla wszystkich $x \in X$, dla których prawa strona jest określona. Ten algorytm, zwany *algorytmem systemu sterowanego* (Σ, μ) jest algorytmem symulacji zjawiska o który chodzi. Tak więc algorytm symulacji zjawiska konstruujemy w postaci algorytmu systemu sterowanego nad rodziną algorytmów składowych.

PRZYKŁAD 10. Dla zjawiska z przykładu 1 tworzymy algorytmy M_A, M_B składowych A, B jak w przykładzie 7. Następnie wybieramy zbiór X , przekształcenie h , i operacje \bar{f}_A, \bar{f}_B jak w przykładzie 8. Wówczas

$$(t, q, \alpha, \beta) \equiv_A (\bar{t}, \bar{q}, \bar{\alpha}, \bar{\beta}) \quad \text{wtedy i tylko wtedy, gdy} \quad t = \bar{t}, q = \bar{q}, \alpha = \bar{\alpha},$$

$$(t, q, \alpha, \beta) \equiv_B (\bar{t}, \bar{q}, \bar{\alpha}, \bar{\beta}) \quad \text{wtedy i tylko wtedy, gdy} \quad t = \bar{t}, q = \bar{q}, \beta = \bar{\beta},$$

i poszukiwanym systemem nad rodziną algorytmów M_A, M_B jest

$$\Sigma = (X; \bar{f}_A, \bar{f}_B, \equiv_A, \equiv_B).$$

Następnie określamy funkcję sterującą μ jak w przykładzie 8. Otrzymany algorytm $\mathcal{M}(\Sigma, \mu)$ jest identyczny z algorytmem z przykładu 5. *

PRZYKŁAD 11. Rozważmy zjawisko różniące się od opisanego w przykładzie 1 tym, że stacji B pomaga obsługiwać klientów stacja C (taka sama jak B i obsługująca klienta w takim samym czasie e). Niech stacja C podejmuje się obsługi tylko wtedy, gdy stacja B jest zajęta. Akcje stacji C i B przebiegają tak samo, więc algorytm M_C symulacji działania stacji C jest identyczny z M_B . Jeśli więc $R \geq 0$ oznacza liczbę klientów czekających na obsłużenie przez którąkolwiek stację, c – chwilę ostatniego rozpoczęcia obsługi przez stację C , i w chwilę bieżącą, to w zbiorze X_C wszystkich trójek (w, R, c) z całkowitym $R \geq 0$ i rzeczywistymi w, c określamy funkcję przejścia f_c wzorem

$$f_c(w, R, c) = \begin{cases} (w, R - 1, w) & \text{dla } c + e \leq w \text{ i } R > 0, \\ (c + e, R, c) & \text{dla } c \leq w < c + e, \\ \text{nieokreślona w pozostałych przypadkach.} \end{cases}$$

W tym przypadku interpretację w zjawisku mają układy $((u, K, a), (v, Q, b), (w, R, c))$, dla których $u = v = w$ i $K = Q = R$, tzn. należące do zbioru

$$Y' = \{((u, K, a), (v, Q, b), (w, R, c)) \in X_A \times X_B \times X_C, u = v = w \text{ i } K = Q = R\}.$$

Niech X' oznacza zbiór wszystkich układów $(t, q, \alpha, \beta, \gamma)$ z całkowitym $q \geq 0$ i rzeczywistymi $t, \alpha, \beta, \gamma, h'$ – przekształcenie X' na Y' określone wzorem

$$h'(t, q, \alpha, \beta, \gamma) = ((t, q, \alpha), (t, q, \beta), (t, q, \gamma)).$$

Przyjmując

$$\begin{aligned} \bar{f}'_A(t, q, \alpha, \beta, \gamma) &= \begin{cases} (\alpha + d, q + 1, \alpha + d, \beta, \gamma) & \text{dla } \alpha \leq t \leq \alpha + d, \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases} \\ \bar{f}'_B(t, q, \alpha, \beta, \gamma) &= \begin{cases} (t, q - 1, \alpha, t, \gamma) & \text{dla } \beta + e \leq t \text{ i } q > 0, \\ (\beta + e, q, \alpha, \beta, \gamma) & \text{dla } t < \beta + e \text{ i } \beta \leq t, \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases} \\ \bar{f}'_C(t, q, \alpha, \beta, \gamma) &= \begin{cases} (t, q - 1, \alpha, \beta, t) & \text{dla } \gamma + e \leq t \text{ i } q > 0, \\ (\gamma + e, q, \alpha, \beta, \gamma) & \text{dla } t < \gamma + e \text{ i } \gamma \leq t; \\ \text{nieokreślona w pozostałych przypadkach,} \end{cases} \end{aligned}$$

oraz

$$(t, q, \alpha, \beta, \gamma) \stackrel{A}{\equiv}' (\bar{t}, \bar{q}, \bar{\alpha}, \bar{\beta}, \bar{\gamma}) \quad \text{wtedy i tylko wtedy, gdy} \quad t = \bar{t}, q = \bar{q}, \alpha = \bar{\alpha},$$

$$(t, q, \alpha, \beta, \gamma) \stackrel{B}{\equiv}' (\bar{t}, \bar{q}, \bar{\alpha}, \bar{\beta}, \bar{\gamma}) \quad \text{wtedy i tylko wtedy, gdy} \quad t = \bar{t}, q = \bar{q}, \beta = \bar{\beta},$$

$$(t, q, \alpha, \beta, \gamma) \stackrel{C}{\equiv}' (\bar{t}, \bar{q}, \bar{\alpha}, \bar{\beta}, \bar{\gamma}) \quad \text{wtedy i tylko wtedy, gdy} \quad t = \bar{t}, q = \bar{q}, \gamma = \bar{\gamma},$$

otrzymujemy system

$$\Sigma' = (X'; \bar{f}'_A, \bar{f}'_B, \bar{f}'_C, \stackrel{A}{\equiv}', \stackrel{B}{\equiv}', \stackrel{C}{\equiv}')$$

nad rodziną algorytmów M_A, M_B, M_C .

Trójka M_A, M_B, M_C stanowi rodzinę algorytmów składowych A, B, C , a funkcję sterującą μ' można określić dla $\alpha \leq t \leq \alpha + d, \beta \leq t, \gamma \leq t$ wzorem

$$\mu'(t, q, \alpha, \beta, \gamma) = \begin{cases} A & \text{dla } t < \beta + e \text{ i } t < \gamma + e \text{ i } \alpha + d \leq \beta + e \text{ i } \alpha + d \leq \gamma + e, \\ & \text{lub } \gamma + e \leq t < \beta + e \text{ i } \alpha + d \leq \beta + e, \\ & \text{lub } \beta + e \leq t < \gamma + e \text{ i } \alpha + d \leq \gamma + e, \\ & \text{lub } \beta + e \leq t \text{ i } \gamma + e \leq t \text{ i } q = 0, \\ B & \text{dla } \beta + e \leq t < \alpha + d \text{ i } q > 0, \\ & \text{lub } t < \beta + e < \alpha + d \text{ i } \beta + e \leq \gamma + e, \\ & \text{lub } \gamma + e \leq t < \beta + e < \alpha + d, \\ C & \text{dla } t < \beta + e \text{ i } \gamma + e \leq t < \alpha + d \text{ i } q > 0, \\ & \text{lub } t < \gamma + e < \alpha + d \text{ i } \gamma + e < \beta + e. \end{cases}$$

Algorytm $\mathfrak{M}(\Sigma', \mu')$ systemu sterowanego (Σ', μ') jest algorytmem symulacji zjawiska. *

4. Symulacja upływu czasu. Teraz wskażemy metodę opisywania stanów algorytmów składowych, zapewniającą możliwość skonstruowania funkcji sterującej. Metoda ta wiąże się ściśle z symulacją upływu czasu w zjawisku.

Algorytm symulacji $M = \mathcal{M}(\Sigma, \mu)$, w którym $\Sigma = (X; \{\bar{f}_i\}_{i \in I}, \{\bar{\equiv}_i\}_{i \in I})$ jest systemem nad rodziną algorytmów $M_i = (X_i; f_i)$ składowych $i \in I$, μ jest funkcją sterującą systemem Σ , oblicza kolejne stany zjawiska i chwile ich osiągania, realizując algorytmy M_i przez wykonanie odpowiednich operacji \bar{f}_i . Stany algorytmów M, M_i zawierają informacje o czasie. Kolejność wykonywania operacji \bar{f}_i odpowiada następstwu akcji składowych w czasie. Funkcja sterująca μ , wyznaczając działające składowe, powoduje pośrednio symulację upływu czasu. Problem skonstruowania takiej funkcji wiąże się więc ściśle z problemem wyboru techniki symulacji upływu czasu.

Na to, żeby było możliwe skonstruowanie funkcji sterującej, układy $(x_i)_{i \in I}$ stanów $x_i \in X_i$ algorytmów M_i powinny determinować działającą składową (własność (ii) z punktu 3). Muszą więc zawierać informację o dotychczasowej historii zjawiska, wyznaczającą w pełni dalszy przebieg zjawiska. Można to zapewnić przez włączenie do opisu stanu każdego algorytmu M_i wspólnej informacji o ewentualnych przyszłych akcjach składowych, dających się przewidzieć na podstawie dotychczasowej historii zjawiska. Wystarczającą informacją jest ciąg

$$(35) \quad L = (i_1, t_1), (i_2, t_2), \dots$$

par (i_k, t_k) reprezentujących przewidywane akcje i wskazujących składową i_k realizującą akcję oraz chwilę zrealizowania t_k . Tego rodzaju ciąg L , uporządkowany tak, że

$$(36) \quad t_1 \leq t_2 \leq \dots$$

nazywamy *planem działania*, a pary, z których się składa — *zdarzeniami*. Składową i_1 w (35) oznaczamy przez SKŁADOWA (L), a chwilę t_1 przez CHWILA (L). O składowej mówimy, że jest *L-bezczynna*, jeśli nie ma zdarzenia w L .

Włączenie planu działania do opisów stanów algorytmów składowych upraszcza te opisy i sprowadza je faktycznie do naturalnych charakteryzacji stanów składowych. Plan działania wchodzi w skład opisu stanu algorytmu symulacji M , a jego pierwsze zdarzenie wyznacza działającą składową i chwilę akcji. Dokładniej, jeśli $L(x)$ oznacza plan działania dla stanu $x \in X$, to taką składową jest

$$(37) \quad \mu(x) = \text{SKŁADOWA}(L(x)),$$

a chwilą akcji jest CHWILA ($L(x)$). Dzięki temu otrzymujemy w naturalny sposób funkcję μ wyznaczającą działającą składową.

Stan (algorytmu symulacji, algorytmów składowych) opisujemy przez informację ϑ o chwili bieżącej, plan działania L , oraz przez pewną informację J . Pożądane jest by te informacje nie były ze sobą związane. Dla wygody przez PLANOWANIE (j, τ) oznaczamy operację wstawienia do planu L w stanie (ϑ, L, J) zdarzenia (j, τ) po wszystkich występujących w L zdarzeniach (i_k, t_k) z $t_k \leq \tau$, a przez UPŁYW CZASU operację zastąpienia w stanie (ϑ, L, J) informacji ϑ informacją CHWILA (L), a następnie usunięcia pierwszego zdarzenia z planu L .

PRZYKŁAD 12. Wróćmy raz jeszcze do przykładu 1. Stan algorytmu M_A źródła A wyrażamy przez chwilę bieżącą ϑ , plan działania L , i liczbę $K \geq 0$ klientów, którzy mają być obsłużeni. Niech

$$\xi(\vartheta, L, K) = (\vartheta, L, K + 1).$$

Funkcję przejścia f_A definiujemy, określając $f_A(\vartheta, L, K)$ jako wynik kolejnego działania operacji ξ , PLANOWANIE (A , CHWILA (L) + d), UPŁYW CZASU, gdy stacja B nie jest L -bezczylna, bądź operacji ξ , PLANOWANIE (B , CHWILA (L)), PLANOWANIE (A , CHWILA (L) + d), UPŁYW CZASU, gdy stacja B jest L -bezczylna. Podobnie stan algorytmu M_B stacji B wyrażamy przez chwilę bieżącą ϑ , plan działania L , i liczbę $Q \geq 0$ klientów czekających na obsłużenie, a funkcję przejścia f_B definiujemy, określając $f_B(\vartheta, L, Q)$ jako wynik działania operacji η , PLANOWANIE (B , CHWILA (L) + e), UPŁYW CZASU, gdy $Q \geq 0$, bądź operacji UPŁYW CZASU, gdy $Q = 0$, gdzie $\eta(\vartheta, L, Q) = (\vartheta, L, Q - 1)$.

Ponieważ w zjawisku $K = Q$, więc stanami algorytmu symulacji mogą być trójki (ϑ, L, q) , w których ϑ oznacza chwilę bieżącą, L plan działania, a $q \geq 0$ oznacza wielkość kolejki. Podobnie jak w przykładzie 8 można łatwo otrzymać operacje \bar{f}_A, \bar{f}_B opisujące akcje składowych. Równoważności \equiv_A, \equiv_B są w tym przypadku identycznościami. *

Stosując te same metody można tworzyć również algorytmy symulacji zjawisk, w których występuje nieskończenie wiele składowych.

PRZYKŁAD 13. Weźmy pod uwagę następujący proces urodzin i zgonów. W każdej chwili $p \geq 2$ będącej liczbą pierwszą rodzi się obiekt $(1, p)$. Obiekt ten żyje p jednostek czasu i ginąc w chwili $2p$ pozostawia po sobie potomka $(2, p)$. Potomek ten żyje znowu p jednostek czasu i ginąc w chwili $3p$ pozostawia po sobie potomka $(3, p)$ itd.

Ten proces odpowiada zachowaniu się systemu składowych (m, p) odpowiadających obiektom (m, p) , i pewnej dodatkowej składowej $(1, 1)$. O wszystkich składowych można zakładać, że występują przez cały czas w systemie, z tym że poza czasami życia odpowiednich obiektów są w wyróżnionym stanie $\#$ i wtedy nie mogą działać. Składowa $(1, 1)$, nie będąca nigdy w stanie $\#$, działa w każdej chwili całkowitej $t \geq 2$, w której nie działają inne składowe, i wtedy wyprowadza składową $(1, t)$ ze stanu $\#$. Każda składowa (m, p) ($p \geq 2$) działa w chwili $(m + 1)p$ wprowadzając składową $(m + 1, p)$ ze stanu $\#$ i przechodząc do stanu $\#$.

Algorytmy składowych definiujemy następująco. Stanami każdego algorytmu $M_{(m,p)}$ są pary (ϑ, L) , gdzie ϑ oznacza chwilę bieżącą, a L plan działania, przy czym L -bezczylność składowej oznacza stan $\#$. Funkcję przejścia $f_{(1,1)}$ algorytmu $M_{(1,1)}$ definiujemy, określając $f_{(1,1)}(\vartheta, L)$ jako wynik działania operacji PLANOWANIE $((1, CHWILA (L), 2 \cdot CHWILA (L))$, PLANOWANIE $((1, 1), CHWILA (L) + 1)$, UPŁYW CZASU. Dla każdego $p \geq 2$ funkcję przejścia $f_{(m,p)}$ algorytmu $M_{(m,p)}$ definiujemy, określając $f_{(m,p)}(\vartheta, L)$ jako wynik działania operacji, PLANOWANIE $((m + 1, p), CHWILA (L) + p)$, ξ , UPŁYW CZASU, gdzie ξ polega na zastąpieniu w L zdarzenia $((1, 1), t)$ zdarzeniem $((1, 1), CHWILA (L) + 1)$. Stanami algorytmu symulacji M mogą być również pary (ϑ, L) i wtedy operacje $\bar{f}_{(m,p)}$ opisujące akcje składowych są identyczne z operacjami $f_{(m,p)}$.

Łatwo zauważyć, że obliczenie $Obl(M, (0, L_0))$ algorytmu symulacji M , w którym $L_0 = ((1, 1), 2)$, daje wszystkie liczby pierwsze. Są to chwile akcji składowej $(1, 1)$. A oto kilka pierwszych stanów tego obliczenia

0, $((1, 1), 2)$
 2, $((1, 1), 3)$, $((1, 2), 4)$
 3, $((1, 2), 4)$, $((1, 1), 4)$, $((1, 3), 6)$
 4, $((1, 1), 5)$, $((1, 3), 6)$, $((2, 2), 6)$
 5, $((1, 3), 6)$, $((2, 2), 6)$, $((1, 1), 6)$, $((1, 5), 10)$
 6, $((2, 2), 6)$, $((1, 1), 7)$, $((2, 3), 9)$, $((1, 5), 10)$
 6, $((1, 1), 7)$, $((3, 2), 8)$, $((2, 3), 9)$, $((1, 5), 10)$ itd.

Przykład ten został zaczerpnięty z [3]. *

5. **Algorytmy symulacji w językach programowania.** Dotychczasowe rozważania miały charakter ogólny. Algorytmy, o których mówiliśmy, były algorytmami abstrakcyjnymi. Nie wiązaliśmy ich ze środkami realizacji obliczeń. Teraz zajmiemy się programowaniem wykonania obliczeń symulujących, tzn. obliczeń algorytmów symulacji przez komputer. Programy wykonania takich obliczeń będziemy nazywali *programami symulacji*. Są to również algorytmy, ale nie pokrywają się z algorytmami symulacji w dotychczasowym znaczeniu.

Program symulacji zawiera:

- część *centralną*, realizującą funkcję przejścia F algorytmu symulacji $M = (X; F)$ na informacjach reprezentujących stany $x \in X$,
- część *przygotowawczą*, która doprowadza algorytm symulacji reprezentowany przez część centralną do odpowiedniego stanu początkowego $x_0 \in X$,
- *cykl*, realizujący obliczenie symulacyjne przez cykliczne wykonywanie części centralnej.

Może on być częścią większego programu (który np. dostarcza mu pewnych informacji). W skład części centralnej wchodzi mniejsze części, które realizują operacje \bar{f}_i opisujące akcje składowych. Jedną z tych części jest wykonywana przy każdym wykonaniu części centralnej.

Jeśli potrafimy rozłożyć zjawisko na małą liczbę składowych, to program symulacji można zbudować stosunkowo prosto. W języku ALGOL 60 można to zrobić następująco.

Numerujemy występujące składowe liczbami $1, \dots, n$. Opisujemy algorytmy $M_i = (X_i; f_i)$ składowych $1, \dots, n$ w formie deklaracji procedur P_i działających na pewnych informacjach globalnych i na swoich parametrach formalnych A_i, B_i, \dots , wywoływanych przez nazwę. Parametry formalne A_i, B_i, \dots , wraz z niektórymi informacjami globalnymi, reprezentują stan algorytmu M_i w instrukcji (cele) procedury P_i . Instrukcja procedury P_i opisuje sposób wykonywania operacji przejścia f_i .

Do informacji globalnych zaliczamy informację o chwili bieżącej postaci

`real ϑ`

i informację o planie działania postaci

`integer array i [1:G]`

`real array t [1:G]`

Każdy plan działania

$$L = (i_1, t_1), \dots, (i_m, t_m),^*$$

gdzie $t_1 \leq \dots \leq t_m$, opisujemy przez początkowe pozycje wektorów i, t przyjmując

$$(38) \quad i[1] = i_1, \dots, \quad i[m] = i_m, \quad t[1] = t_1, \dots, \quad t[m] = t_m, \quad t[m+1] = +\infty.$$

Wtedy planowane zdarzenia są parami $(i[z], t[z])$, gdzie $z = 1, \dots, m$. Dla posługiwania się informacją o chwili bieżącej i o planie działania tworzymy procedury:

SKŁADOWA, funkcyjną, wyznaczającą składową $i[1]$,

CHWILA, funkcyjną, wyznaczającą chwilę $t[1]$,

BEZCZYNNNA, funkcyjną, z parametrem formalnym j wywoływanym przez wartość, wyznaczającą bulowską wartość formuły „w planie działania występuje zdarzenie składowej j ”,

PLANOWANIE, z parametrami formalnymi j, τ wywoływanymi przez wartość, wstawiającą do planu działania zdarzenie (j, τ) po wszystkich zdarzeniach $(i[z], t[z])$ z $t[z] \leq \tau$,

UPEŁYW CZASU, przypisującą zmiennej ϑ wartość $t [1]$ i usuwającą z planu działania pierwsze zdarzenie ($i [1], t [1]$),
 itp. Na przykład procedurę PLANOWANIE trzeba zbudować tak, by wywołana z parametrami aktualnymi j_0, τ_0 w sytuacji (38) wyszukiwała wśród $1, \dots, m$ liczbę z , dla której $t [z] \leq \tau_0 < t [z+1]$, i doprowadzała do sytuacji

$$i[1] = i_1, \dots, i[z] = i_z, i[z+1] = j_0, i[z+2] = i_{z+1}, \dots, i[m+1] = i_m,$$

$$t[1] = t_1, \dots, t[z] = t_z, t[z+1] = \tau_0, t[z+2] = t_{z+1}, \dots, t[m+1] = t_m, t[m+2] = +\infty.$$

PRZYKŁAD 14. Algorytmy źródła A i stacji B z przykładu 12 opisujemy w formie deklaracji

```

procedure źródło ( $K$ ); integer  $K$ ;
  begin  $K := K + 1$ ;
    if BEZCZYNNNA ( $1$ ) then PLANOWANIE ( $2, CHWILA$ );
    PLANOWANIE ( $1, CHWILA + d$ );
    UPEŁYW CZASU
  end;
procedure stacja ( $Q$ ); integer  $Q$ ;
  begin if  $Q=0$  then go to  $L$ ;
     $Q := Q - 1$ ;
    PLANOWANIE ( $2, CHWILA + e$ );
     $L : UPEŁYW CZASU$ 
  end;
  
```

numerując źródło A liczbą 1 , a stację B liczbą 2 .*

Część centralną programu symulacji, która ma reprezentować algorytm symulacji M , tworzymy w formie następującego bloku z deklaracjami procedur P_i :

```

begin procedure  $P_1 (A_1, B_1, \dots)$  ...
  ...
  procedure  $P_n (A_n, B_n, \dots)$  ...
  switch  $SW := L_1, \dots, L_n$ ;
  go to  $SW$  [SKŁADOWA];
 $L_1 : P_1 (A'_1, B'_1, \dots)$ ; go to koniec;
  ...
 $L_n : P_n (A'_n, B'_n, \dots)$ ; go to koniec;
koniec; end;
  
```

Stan algorytmu M jest reprezentowany przez zestaw H informacji globalnych, zawierający informację o chwili bieżącej (zmienna ϑ) i informację o planie działania (wektory i, t). Pewne podzestawy H_i tego zestawu (obejmujące zmienną ϑ i wektory i, t) reprezentują stany poszczególnych algorytmów M_i . Wśród informacji podzestawu H_i są odpowiedniki A'_i, B'_i, \dots parametrów formalnych A_i, B_i, \dots procedury P_i . Operacje \bar{f}_i opisujące akcje składowych $1, \dots, n$ są realizowane przez instrukcje $P_i (A'_i, B'_i, \dots)$. Wybór działającej składowej jest dokonywany przez instrukcję **go to** SW [SKŁADOWA].

PRZYKŁAD 15. W przykładzie 12 część centralna programu symulacji wygląda następująco:

```

begin procedure źródło ( $K$ ) ...
  procedure stacja ( $Q$ ) ...
  switch  $SW := L_1, L_2$ ;
  
```

```

go to SW [SKŁADOWA];
L1: źródło (q); go to koniec;
L2: stacja (q); go to koniec;
koniec: end;

```

Stan algorytmu symulacji jest tu reprezentowany przez zmienną q (wielkość kolejki), informację o chwili bieżącej (zmienna ϑ) i informację o planie działania (wektory i, t).

Część przygotowawcza powinna zawierać instrukcję

```

q := 0;
 $\vartheta$  := 0;
t [1] := +  $\infty$ ;

```

PLANOWANIE (1,0);

a cykl realizujący obliczenie symulacyjne powinien mieć postać

```

for n: 1 step 1 until m do < część centralna >; *

```

Jeśli trzeba brać pod uwagę dużo, a nawet nieskończenie wiele składowych (jak w przykładzie 13), lub jeśli nie wiadomo z góry, jakie obiekty wystąpią, to opisana metoda nie daje się zastosować. Jednak również i wtedy często można efektywnie zbudować program symulacji i wykonać symulację (przynajmniej częściowo). Dla ułatwienia programowania w takich przypadkach budowane są specjalne języki programowania zwane *symulacyjnymi* (np. GPSS, SIMSCRIPT, CSL, SOL, SIMULA). Mają one wiele cech wspólnych. Wymienimy niektóre, biorąc za punkt wyjścia język SIMULA.

W SIMULA wykorzystywane jest podobieństwo zachowania się składowych w zjawisku. Składowe zachowujące się podobnie mają podobne algorytmy symulacji działania (por. przykład 13). Jeśli podobieństwo jest tego rodzaju, że algorytmy można opisać formalnie tak samo, to ten sam opis formalny może stanowić opis algorytmów wielu składowych. Ma on wtedy charakter opisu *klasy* podobnych algorytmów. Opierając się na tym spostrzeżeniu wprowadzono deklaracje klas algorytmów. W SIMULA składową zjawiska reprezentuje się w algorytmie symulacji i w modelu zjawiska nie tyle przez algorytm składowej, co raczej przez ciąg kolejnych stanów algorytmu składowej w modelu zjawiska. Taki ciąg jest nazywany *procesem* i opisuje losy składowej. Język daje możliwości inicjowania procesów, nazywania procesów i ich atrybutów (tzn. informacji charakteryzujących stan), oraz manipulowania procesami i ich atrybutami za pośrednictwem nazw. Przewidziane są środki do posługiwania się planem działania. Operacje takie jak założenie planu działania, wybieranie działającego procesu i niektóre modyfikacje planu działania są implementowane automatycznie i nie wymagają programowania. Pozostałe modyfikacje planu działania programuje się szczególnie prosto dzięki specjalnym instrukcjom do planowania akcji. Możliwe jest tworzenie kolejek procesów itp. Dokładniejsze informacje o SIMULA i o innych językach symulacji znajdzie Czytelnik np. w przeglądowej pracy O. J. Dahla [3].

Bibliografia

- [1] Н. П. Бусленко, *Моделирование сложных систем*, Москва 1968.
- [2] J. N. Buxton (editor), *Simulation programming languages*, Proceedings of the IFIP Working Conference, Oslo, May 1967, North-Holland 1968.
- [3] O. J. Dahl, *Discrete event simulation languages*, Programming languages (ed. F. Genuys), Academic Press, London 1968.
- [4] — i K. Nygaard, SIMULA — *An ALGOL based simulation language*, Comm. of the ACM, September 1966.
- [5] G. Gordon, *System simulation*, Prentice Hall.
- [6] F. F. Martin, *Computer modeling and simulation*, Wiley, New York 1968.
- [7] K. D. Tocher, *The art of simulation*, The English Universities Press Ltd, 1963.
- [8] R. Zieliński, *Metody Monte Carlo*, Warszawa 1970.