STUDIA INFORMATICA 2003
Volume 24 Number 2A (53)

Stefan WĘGRZYN, Ryszard WINIARCZYK
Instytut Informatyki Teoretycznej i Stosowanej PAN

REPRODUKCJA I MULTIPLIKACJA JAKO INFORMATYCZNE TECHNOLOGIE WYTWARZANIA*

Streszczenie. Rozróżnia się dwa przypadki wzrostu liczby obiektów w danym środowisku. Pierwszy to wzrost tej liczby oparty na zjawisku reprodukcji obiektów, a drugi to wzrost tej liczby oparty na zjawisku multiplikacji obiektów. Przeprowadza się analizę i przedstawia modele dla tych dwóch przypadków. Oba te przypadki można brać pod uwagę przygotowując podstawy molekularnych procesów wytwarzania.

Słowa kluczowe: samoreplikacja, informatyczne technologie wytwarzania.

REPRODUCTION AND MULTIPLICATION AS INFORMATICS MANUFACTURING TECHNOLOGIES

Summary. There exist two cases of objects development at a given environment. The first one is based on objects reproduction phenomena and the second is based on multiplication phenomena. In the article the analysis and models of these cases are presented. The both cases may be considered as the basis of molecular manufacturing processes.

Keywords: self-replication, informatics manufacturing technologies.

1. Wprowadzenie

Dla funkcjonowania systemów informatyki zdolnych do realizacji procesów wytwarzania bardzo istotnymi są procesy reprodukcji i multiplikacji.

* Projekt KBN 7 T11C 017 21

Można je tak scharakteryzować:

Niech będzie dany zbiór określonych elementów, który nazwiemy środowiskiem. Załóżmy, że dzięki doprowadzeniu energii z zewnątrz elementy środowiska znajdują się w ruchu i że występuje ich swobodne poruszanie się.

Obiektem będziemy nazywali zamkniętą część środowiska wyodrębnioną w taki sposób, że wymiana elementów między środowiskiem a obiektem jest możliwa tylko przy spełnieniu określonych warunków.

Dzięki temu obiekt pobiera ze środowiska i gromadzi elementy potrzebne do jego rozwoju i rozwija się.

Jeśli w wyniku rozwoju obiektu następuje wytworzenie i wprowadzenie przez niego do środowiska elementu inicjującego początek niezależnego rozwoju w tym środowisku kolejnego następnego obiektu według tego samego programu, który był podstawą rozwoju pierwszego obiektu, to nazywamy to zjawiskiem reprodukcji.

Jeżeli natomiast w wyniku rozwoju obiektu następuje jego podział na obiekty zdolne do dalszego niezależnego ich rozwoju i podziałów, to nazywamy to zjawiskiem multiplikacji.

Można zauważyć, że zgodnie z tymi określeniami w przypadku reprodukcji wszystkie elementy nowego obiektu, obiektu potomnego, są pobierane bezpośrednio ze środowiska, a obiekt matka po wygenerowaniu elementu inicjującego proces reprodukcji pozostaje nadal w środowisku, inicjując powstawanie dalszych kolejnych obiektów potomnych.

W przypadku natomiast multiplikacji obiekty potomne powstają z obiektu „matki", która po rozpadzie na obiekty potomne przestaje istnieć pozostawiając po sobie tylko obiekty potomne.

1.1. Przyjęte modele systemów

Przyjmiemy zgodnie z [1], że system rozwijający opiera się na zbiorze elementów środowiska n różnych kategorii $a_1,\ldots a_i,\ldots,a_n$, na któryoh mogą być wykonywane odpowiadające im operacje $A_1\ldots,A_i,\ldots A_n$.

Przyjmujemy następujące możliwe operacje:

a) transformacja T

$T(a_i)=a_j$ lub $a_i \to a_j$

b) bifurkacja $B(a_i)=a_j a_k$ lub $a_i \to a_j a_k$

c) bifurkacja ze zmianą kierunku zaznaczoną nawiasem $C(a_i)=a_j(a_k)$ lub $a_i \to a_j(a_k)$

d) liniowa generacja $L(a_i)=a_j a_i$ lub $a_i \to a_j a_i$

e) generacja rotacyjna $R(a_j)=a_j(a_i)$ lub $a_i \to a_j(a_i)$

f) stagnacja $S(a_i) = a_i$ lub $a_i \to a_i$

$$SG = A_1 A_2 \ldots A_m = \underset{i=1}{\overset{m}{SEQ}} A_i$$

Produkt można w tym ujęciu określić słowem genetycznym SG rozpiętym nad skończonym zbiorem Z różnych elementów, co zapiszemy w następujący sposób:

$$\frac{SG}{Z} = \frac{\underset{i=1}{\overset{m}{SEQ}} A_i}{a_1, a_2, \ldots a_m}$$

1.2. System przemijający

Ograniczenie liczby operacji, które można zrealizować nad danym elementem można zaznaczyć przez umieszczenie liczby ograniczającej przed symbolem operacji, np.:

$$mL(a_i) \text{ lub } mR(a_i) = \begin{cases} a_j a_i \text{ lub } a_j(a_i) & n_{a_i} < m \\ a_j & n_{a_i} = m+1 \end{cases} \tag{1}$$

$$mS(a_i) = \begin{cases} a_i & n_{a_i} < m \\ - & n_{a_i} = m+1 \end{cases} \tag{2}$$

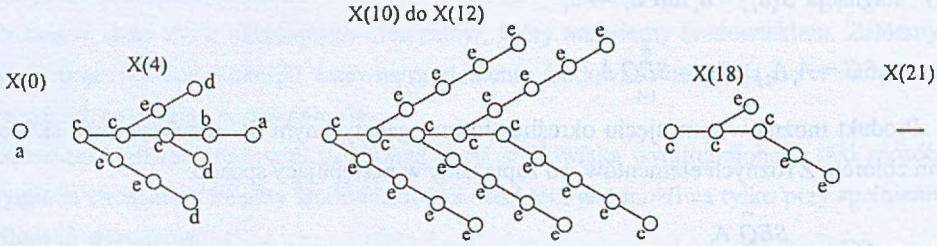
Wprowadzenie ograniczeń na liczbę operacji prowadzi do systemów przemijających. Takim systemem przemijającym jest np. system X określony relacją (3).

$$X = \frac{4LC14S3L9S}{a, b, c, d, e} \tag{3}$$

System (3) powstaje w wyniku realizacji następujących operacji:

$$\begin{array}{l} 4L \;\; a \to 4 \text{ razy } ba, \text{ następnie b} \\ | \\ C \;\; b \to c(d) \\ | \\ 14S \;\; c \to 14 \text{ razy c, następnie} — \\ | \\ 3L \;\; d \to 3 \text{ okresy } e\, d, \text{ następnie } e \\ | \\ 9S \;\; e \to 9 \text{ okresów } e, \text{ następnie} — \end{array} \tag{4}$$

Rozwój i przemijanie systemu opisanego relacjami (4) są przedstawione na rys. 1.



Rys.1. Przebieg procesu rozwoju i przemijania systemu X określonego relacja (3)
Fig.1. Development process running and system X ceasing defined by relation (3)

1.3. Reprodukcja pojedyncza

Odradzanie się systemów przemijających na drodze reprodukcji staje się możliwe po wprowadzeniu w słowie genetycznym SG sprzężenia zwrotnego, operacji opóźnienia oznaczonej symbolem [Dm], gdzie m jest liczbą okresów opóźnienia. Np.

$$[D25]Ta_i = \begin{cases} a_i, n_{ai} < 25 \\ a_j, n_{ai} = 25 \end{cases}$$

Gdyby to zastosować do rozpatrywanego poprzednio systemu (3), to otrzymamy

$$Y = \frac{\boxed{BX[D25]T}}{\alpha, a, b, c, d, \beta} = \frac{\boxed{B4LC14S4L9S[D25]T}}{\alpha, a, b, c, d, \beta} \tag{5}$$

co prowadzi do następujących operacji

B $\alpha \rightarrow a\beta$

4L $a \rightarrow$ 4 okresy ba, następnie b

C $b \rightarrow c(d)$

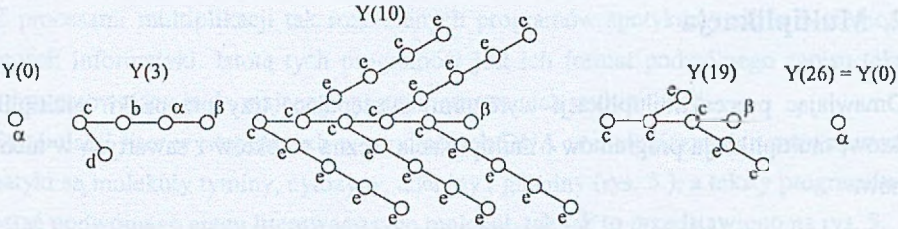
14S $c \rightarrow$ 14 okresy c, następnie — (6)

3L $d \rightarrow$ 3 okresy ed, następnie e

9S $e \rightarrow$ 9 okresy e, następnie —

[D25]T $\beta \rightarrow \alpha$ po 25 okresach

Ilustracja procesu odradzania się systemu (6) jest przedstawiona na rys. 2.



Rys. 2. Ilustracja pojedynczej reprodukcji systemu (5)
Fig. 2. Illustration of single system reproduction (5)

1.4. Reprodukcja multikrotna

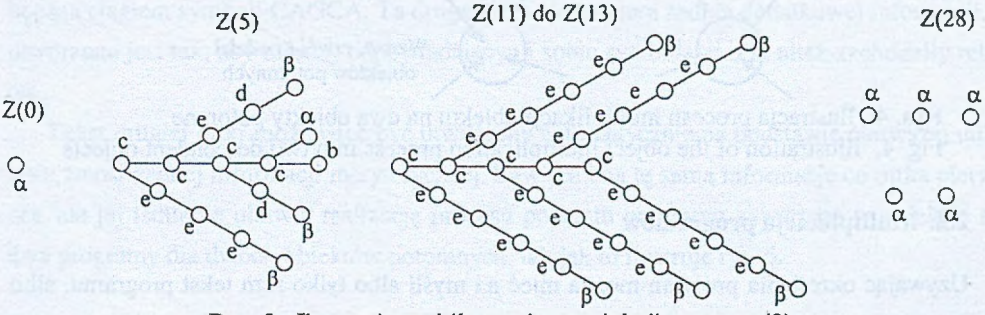
Reprodukcję multikrotną przedstawimy na przykładzie systemu Z, określonego w następujący sposób:

$$Z = \frac{\boxed{4LC14SB4L9S[D25]T}}{a,b,c,\alpha,d,e,\beta} \tag{7}$$

Wynikające z relacji (7) operacje elementarne są następujące:

$$\begin{array}{ll} 4L & a \rightarrow 4 \text{ okresy } ba, \text{ następnie } b \\ C & b \rightarrow c(\alpha) \\ 14S & c \rightarrow 14 \text{ okresów } c, \text{ następnie } — \\ B & \alpha \rightarrow d\beta \\ 3L & d \rightarrow 3 \text{ okresy } ed, \text{ następnie} \\ 9S & e \rightarrow 9 \text{ okresów } e, \text{ następnie } — \\ [D25]T & \beta \rightarrow \alpha \text{ po 25 okresach} \end{array} \tag{8}$$

Ilustracja odradzającego się systemu (8) jest przedstawiona na rys. 3.



Rys. 3. Ilustracja multikrotnej reprodukcji systemu (8)
Fig. 3. Illustration of multiply system reproduction (8)

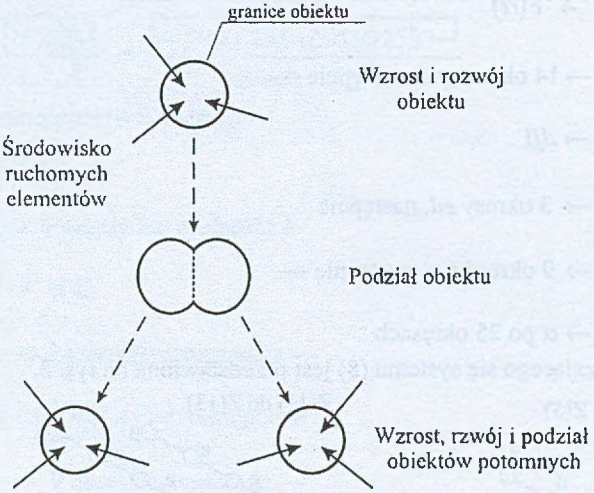
2. Multiplikacja

Omawiając proces multiplikacji wyróżnimy następujące trzy przypadki: multiplikacja obiektów, multiplikacja programów i multiplikacja łączna obiektów i zawartych w nich programów.

2.1. Multiplikacja obiektów

Niech w środowisku swobodnych ruchomych elementów znajduje się obiekt taki, że dzięki otaczającej go powierzchni wymiana elementów między nim a środowiskiem jest możliwa tylko przy spełnieniu określonych warunków. Dzięki temu obiekt gromadzi w sobie potrzebne do jego wzrostu elementy i rozwija się do momentu, w którym staje się możliwy i następuje jego podział na obiekty potomne. Mogą one, podobnie jak obiekt pierwotny, rosnąć, rozwijać się, a po dojściu do określonej wielkości podzielić się na kolejne obiekty potomne. Taki proces będziemy nazywać procesem multiplikacji.

Przebieg procesu multiplikacji dla przypadku podziału obiektu na dwa obiekty potomne ilustruje rysunek 4.



Rys. 4. Ilustracja procesu multiplikacji obiektu na dwa obiekty potomne
Fig. 4. Illustration of the object multiplication process into two descendent objects

2.2. Multiplikacja programów

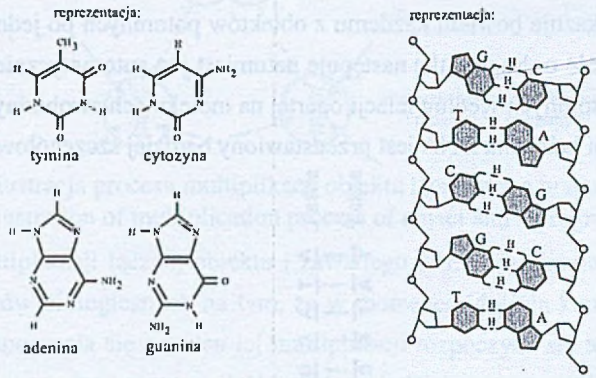
Używając określenia program można mieć na myśli albo tylko sam tekst programu, albo tekst programu utrwalony w określonej materialnej formie. W tym opracowaniu używając określenia program będziemy mieć na myśli tekst programu utrwalony w materialnej formie.

Z procesami multiplikacji tak rozumianych programów spotykamy się w biologicznych systemach informatyki. Istotą tych programów jest ich format podwójnego zapisu tekstu z góry przygotowujący je do mającego nastąpić procesu ich multiplikacji.

Symbolami liter występujących w programach DNA w biologicznych nanosystemach informatyki są molekuły tyminy, cytozyny, adeniny i guaniny (rys. 5), a teksty programów mają postać podwójnego ciągu literowego tych molekuł, tak jak to przedstawiono na rys. 5.

a) Symbole liter **b) Format tekstów**

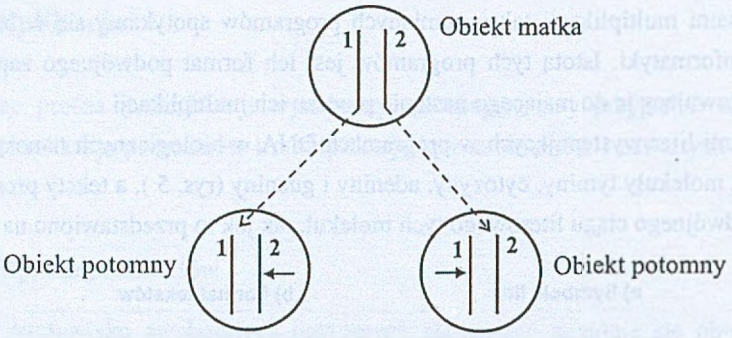
< nukleotyd > ::= A | T | G | C < tekst > ::=<tekst><nukleotyd>|<nukleotyd>



Rys. 5. Sposób kodowania informatycznych programów w biologicznych systemach informatyki
a) symbole liter biologicznych języka programowania, b) format tekstów programów
Fig. 5. Method of informatics programs coding in biological systems of informatics
a) symbols of biological letters in programming language, b) formats of programs text

Podwójny zapis tekstu programu polega na tym, że obok pierwszej nitki programu będącego np. w przypadku przedstawionym na rys. 5 ciągiem symboli GTCGT biegnie druga nitka będąca ciągiem symboli CAGCA. Ta druga nitka nie zawiera żadnej dodatkowej informacji, a utworzona jest tak, aby między odpowiadającymi sobie symbolami obu nitek zachodziły relacje.

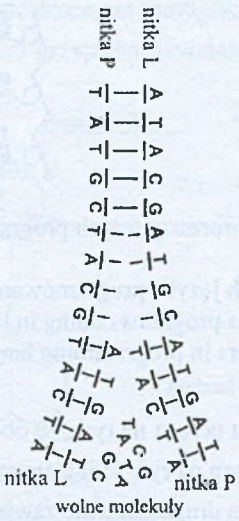
Tekst drugiej nitki może więc być utworzony automatycznie na podstawie pierwszej nitki i nie wnosi żadnej informacji merytorycznej. Zawiera ona tę samą informację co nitka pierwsza, ale jej istnienie ułatwia realizację procesu podziału programu zawartego w obiekcie na dwa programy dla dwóch obiektów potomnych, tak jak to ilustruje rys. 6.



Rys. 6. Podwójny zapis programu w obiekcie znacznie ułatwia jego multiplikację
Fig. 6. Double notation of program contained in object facilitates its multiplication

Obiekt matka przekazuje bowiem każdemu z obiektów potomnych po jednej nitce swego programu, a uzupełnienie o drugą nitkę następuje natomiast już automatycznie, oddzielnie w każdym z obiektów potomnych według relacji opartej na molekułach swobodnych.

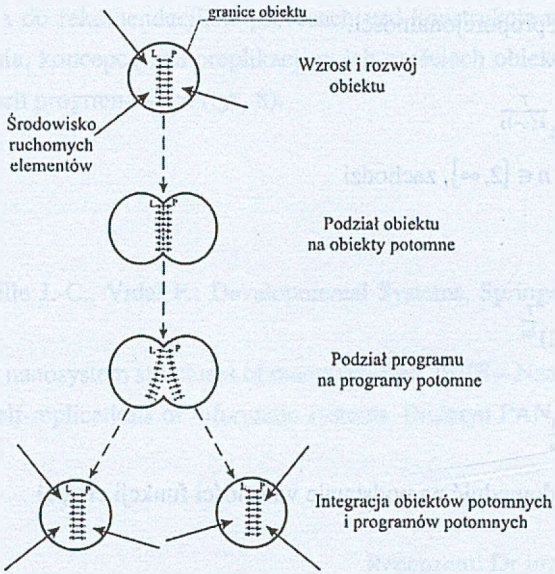
Proces multiplikacji łańcucha DNA jest przedstawiony bardziej szczegółowo na rys. 7.



Rys. 7. Mechanizm multiplikacji DNA
Fig. 7. Mechanism of DNA multiplication

2.3. Multiplikacja łączna obiektów z zawartymi w nich programami

Ilustracja procesu multiplikacji łącznej obiektu i zawartego w nim programu jest przedstawiona na rys. 8.



Rys. 8. Ilustracja procesu multiplikacji obiektu i zawartego w nim programu
Fig. 8. Illustration of multiplication process of object and a program contained in it

Proces multiplikacji łącznej obiektu i zawartego w nim programu polega np. w komórkach organizmów biologicznych na tym, że w momencie dojścia komórki do odpowiedniej wielkości i rozpoczęcia się procesu jej multiplikacji rozpoczyna się też proces multiplikacji znajdującego się w nim programu DNA, który rozdziela się na z góry przygotowane dwie oddzielne nitki, na rys. 7 oznaczono je literami L i P i kieruje je oddzielnie, każdą do jednego z dwóch tworzonych nowych obiektów, gdzie zostają uzupełnione o brakujące części, a więc w jednej L o P a w drugiej P o L.

2.4. Dobór okresów multiplikacji

Jeżeli założyć, że czas potrzebny do osiągnięcia takiego wzrostu obiektu, przy którym możliwa jest jego multiplikacja, jest proporcjonalny do liczby obiektów potomnych, które powstają w momencie multiplikacji obiektu matki, to można wykazać, że najszybszy wzrost całkowitej liczby l obiektów utworzonych w dowolnym przedziale czasu T następuje wówczas, gdy n = 2 to znaczy, obiekty dzielą się na dwie części.

Oznaczmy:

T_p - czas potrzebny do osiągnięcia przez obiekt warunków samoreplikacji,

n - liczba obiektów powstających w momencie multiplikacji,

$l(n)$ - liczba obiektów po czasie T.

Zakładamy:

$$T_p = k(n-1) \tag{9}$$

gdzie k współczynnik proporcjonalności.

Mamy więc

$$l(n) = n^{\frac{T}{T_p}} = n^{\frac{T}{k(n-1)}} \tag{10}$$

Udowodnimy, że dla $n \in [2, \infty]$, zachodzi

$$l(n) > l(n+1) \tag{11}$$

czyli, że

$$n^{\frac{T}{k(n-1)}} > (n+1)^{\frac{T}{kn}} \tag{12}$$

a po uproszczeniu

$$n^{\frac{1}{n-1}} > (n+1)^{\frac{1}{n}} \tag{13}$$

Relację (14) można udowodnić na podstawie własności funkcji ciągłej

$$y(x) = x^{f(x)} \tag{14}$$

gdzie $f(x) = \frac{1}{x-1}$, dla której zachodzi

$$\frac{dy}{dx} = \frac{1}{x-1} x^{\frac{1}{x-1}} \left[\frac{1}{x} - \frac{\ln x}{x-1} \right] < 0 \text{ dla } x \in [2, \infty] \tag{15}$$

Funkcja ciągła $x^{\frac{1}{x-1}}$ a zatem i funkcja dyskretna $n^{\frac{1}{n-1}}$ są więc w przedziale $n \in [2, \infty]$ funkcjami malejącymi, co udowadnia relację (13).

Przy założeniu relacji (9) najszybszy wzrost całkowitej liczby obiektów występuje dla ich dychotomicznych podziałów, to znaczy dla $n = 2$, co np. zachodzi w przypadku samoreplikacji komórek organizmów biologicznych.

3. Konkluzja

Jednym z aktualnych kierunków rozwoju systemów informatyki są prace nad takimi systemami informatyki, które można by wykorzystać do przemysłowej realizacji procesów wytwarzania różnych produktów, a nawet obiektów.

Bardzo istotna cechą tych produktów i obiektów będzie własność samopowielania się, a więc samoreplikacji czy samomultiplikacji.

Dlatego celem niniejszego opracowania była analiza procesów samoreplikacji i samomultiplikacji, aby wskazać zachodzące między nimi różnice i możliwości w informatycznych systemach wytwarzania [2].

Analiza ta skłania do rekomendacji, w procesach nad konstrukcją nanoinformatycznych systemów wytwarzania, koncepcji samoreplikacji w ich częściach obiektowych i samomultiplikacji w ich częściach programowych (rys. 8).

LITERATURA

1. Węgrzyn S., Gille J.-C., Vidal P.: Developmental Systems, Springer-Verlag, New York 1990.
2. Nowak S.:, The nanosystem structures of manufacturing, IEEE – Nano, 2002
3. Węgrzyn S.:, Self-replications of informatic systems, Biuletyn PAN, Vol. 49(2001) no.1, s. 161-165.

Recenzent: Dr inż. Przemysław Szmal

Wpłynęło do Redakcji 24 marca 2003 r.

Abstract

There exist two cases of objects development at a given environment. The first one is based on objects reproduction phenomena (Fig. 2, Fig. 3) and the second is based on multiplication phenomena (Fig. 7.). In the article the analysis and models of these cases are presented. In the first case the mother object can initiate several times consecutive processes of its reproduction, in the second case the mother object cease to exist after its multiplication during which it is divided into e.g. two independent descendents. These descendents continue the process of growth and division. The both cases may be considered as the basis of molecular manufacturing processes (Fig. 8).

Adresy

Stefan WĘGRZYN: Instytut Informatyki Teoretycznej i Stosowanej PAN, ul. Bałtycka 5, 44-100 Gliwice, Polska, wegrzyn@iitis.polsl.gliwice.pl.

Ryszard WINIARCZYK: Instytut Informatyki Teoretycznej i Stosowanej PAN, ul. Bałtycka 5, 44-100 Gliwice, Polska, ryswin@iitis.gliwice.pl.