Kolokwium nr 1 z programowania obiektowego
29 kwietnia 2OO9

Zespół badawczy zajmujący się dziedziną sztucznego życia prowadzi eksperymenty ze sztucznymi organizmami wyposażonymi w kod genetyczny, zapisany w postaci ciągu małych liter alfabetu angielskiego. Podczas rozmnażania organizm przekazuje potomkowi swój kod genetyczny, który może być zmodyfikowany wskutek działania mutacji.

Warianty mutacji:

(a) *mutacja jednostajna*: dla każdej pozycji w kodzie decydujemy niezależnie, z prawdopodobieństwem równym odwrotności długości kodu, czy ma zostać zmodyfikowana,
(b) *mutacja jednopunktowa*: wybieramy losowo pozycję w kodzie, która ma zostać zmodyfikowana,
(c) *mutacja brzegowa*: modyfikujemy krańcowe pozycje kodu.

Badacze rozważają użycie w przyszłości jeszcze innych wariantów mutacji.

W obecnej fazie eksperymentów *modyfikacja* polega na zastąpieniu litery zajmującej wybraną pozycję w kodzie przez słowo wygenerowane losowo przy użyciu zadanej *gramatyki bezkontekstowej*.
Generator słów używany przez eksperymentatorów realizuje wyprowadzenie lewostronne, zastępując po kolei każdy napotkany symbol nieterminalny produkcją wybraną losowo z zestawu produkcji przypisanych temu symbolowi. Długość wyprowadzenia (tj. liczba zastosowanych podstawień tekstowych) dla generowanego słowa jest ograniczona do 100; przekroczenie tego limitu jest traktowane jako niepowodzenie, skutkujące mutacją letalną (śmiertelną).

Gramatyka, wyprowadzenie lewostronne i opis gramatyki.

Gramatyka bezkontekstowa to (formalnie) uporządkowana czwórka <T, N, P, S>, gdzie T - niepusty zbiór symboli terminalnych (krótko: *terminali*), N - niepusty zbiór symboli nieterminalnych (*nieterminali*), P - zbiór *produkcji* oraz S - symbol początkowy (*aksjomat*). Zbiory T i N nie mają wspólnych elementów i łącznie tworzą *alfabet* gramatyki A. Każdy symbol nieterminalny X ma przyporządkowany (niepusty) zbiór produkcji postaci X -> z, gdzie z jest pewnym słowem nad alfabetem A (dopuszcza się *słowo puste*, nie zawierające żadnych symboli); słowo z nazywamy wówczas prawą stroną produkcji. Aksjomat jest wybranym symbolem nieterminalnym.

Przykład gramatyki:
T = { a, b }
N = { S, P, Q ,R }
P = { S -> P, S -> Q, S -> R, P -> a, P -> aP, P -> aPb, Q -> b, Q -> Qb,
Q -> aQb, R -> <puste> }
S = S

Wyprowadzenie lewostronne to ciąg słów, rozpoczynający się od aksjomatu i kończący się słowem złożonym z samych terminali taki, że kolejne słowo powstaje z poprzedniego przez zastąpienie skrajnie lewego wystąpienia nieterminala przez prawą stronę pewnej przyporządkowanej mu produkcji.
Przyjmujemy, że terminale są zakodowane w postaci małych liter alfabetu angielskiego (a-z), a nieterminale - za pomocą dużych liter tegoż alfabetu (A-Z).

Przykład wyprowadzenia lewostronnego długości 4 (nie liczymy elementu, od którego zaczynamy):
S => P => aP => aaPb => aaab

Opis gramatyki podany jest wg następującej konwencji:
(i) napis (String) reprezentujący symbole terminalne
(ii) napis reprezentujący symbole nieterminalne (pierwsza litera napisu określa aksjomat)
(iii) tablica tablic zawierających prawe strony produkcji zakodowane w postaci napisów; każdy wiersz tablicy zawiera zestaw prawych stron dla danego nieterminala (w kolejności określonej w (ii)).

Opis gramatyki zdefiniowanej powyżej:
(i) "ab"

(ii) `"SPQR"`
(iii) `{{"P","Q","R"},{"a","aP","aPb"},{"b", "Qb", "aQb"},{""}}`

Polecenia:

(1) Zaprojektuj i opisz klasy w Javie potrzebne do implementacji mutacji kodu genetycznego sztucznych organizmów.
(2) Zaimplementuj metodę `main` w klasie `Test` symulującą tworzenie zmutowanych kodów genetycznych przy użyciu każdego z opisanych wariantów mutacji dla wybranej gramatyki.
(3) Zaimplementuj pozostałe metody i konstruktory użyte w rozwiązaniu.

==========

Przydatne metody w klasie `Character`:

```
static boolean isUpperCase(char ch)
static boolean isLowerCase(char ch)
```

Przydatne metody w klasie `String`:

```
int length()
char[] toCharArray()
char charAt(int index)
```

==========

Powodzenia!