Instytut Sterowania i Systemów Informatycznych
Wydział Informatyki, Elektrotechniki i Automatyki
Uniwersytet Zielonogórski

Podstawy Systemów Dyskretnych – Laboratorium[1]
Informatyka[2] – stopień pierwszy – tytuł inżyniera – Semestr II rok I
Zbiór zadań

1 Maksymalnie miękkie wprowadzenie do programowanie w systemie Maxima

Zadanie przedstawione w tej części obejmują trzy pierwsze laboratoria.

1. Wykonać następujące obliczenia w środowisku Maxima:
   (a) `1+2*3-5/4;`,
   (b) `1/2;`, `1/2 + 1/3;`,
   (c) `sqrt(5);`, `sqrt(3);`, `sqrt(1/2);`,
   (d) `sqrt(5.0);`, `sqrt(3.0);`, `sqrt(1.0/2.0);`,
   (e) `ln(4)`, `exp(4);`,
   (f) `ln(4.0)`, `exp(4.0);`,
   (g) `5!;`, `10!;`, `15!;`, `20!;`, `100!;`, `200!;`,
   (h) `float(sqrt(7));`,
   (i) `(1/2)^4; 0.5^4; float( (1/2)^4 );`,
   (j) `alpha, beta, gamma, Alpha, Beta, Gamma;`,
   (k) `float(%pi);`,
   (l) `alpha: 2*%pi/3;`,
   (m) `sin(alpha), cos(alpha), tan(alpha), cot(alpha);`,
   (n) `ln(alpha);, exp(alpha);, abs(-7.2);`,
   (o) `float(ln(alpha)), float(exp(alpha));`,
   (p) `a[3]; b[1]; a_2; c_72;`

   W każdym przypadku sprawdzić poprawność wyrażenia, i odnaleźć poprawną funkcję realizującą np. obliczenia dotyczące logarytmów, funkcji wykładniczych bądź trygonometrycznych.

2. Wyznaczyć przybliżenia przynajmniej z dokładnością 0.0001 następujących liczb (zmienna fpprec określa dokładność, należy wykorzystać też bfloat – konwersja do typu bigfloat):
   (a) $\sqrt{2}$, $\sqrt{3}$,

[1] by M.S. luty, marzec, kwiecień, maj roku pańskiego 2005 oraz luty, marzec 2006, luty 2007, marzec 2010, luty 2019, luty 2021, opracowane min. na podstawie książki Matematyka Dyskretna, Kenneth A.Ross, Charles R.B.Wright, Wydawnictwa Naukowe PWN, wydanie trzecie, Warszawa 2000

[2] Zbiór zadań dotyczy kierunku informatyku w trybie stacjonarnym (studia dzienne) oraz w trybie niestacjonarnym (studia zaoczne).

(b) $\sqrt[3]{4}$, $\sqrt[4]{5}$.

3. Obliczyć wartości wyrażeń:

   (a) $\frac{3}{2} + \frac{1}{3}$ oraz $1 + \frac{\pi}{2}$

   (b) $\sin(1)$, $\sin(\pi)$, $\sin(x)$,

   (c) $\int \mathrm{tg}(t)dt$,

   (d) $\sqrt{32}$.

4. Wyznaczyć wartości następujących wyrażeń, oraz wyjaśnić rolę symboli `:`, `%`, `$`, `'` (dwukropek, procent oraz znak dolara i apostrof):

   (a) `x:2; y:5;`, `(x+3*y)/(2*x-7*y);`,

   (b) `float(%);`,

   (c) `a:2.5;`,

   (d) `print("hello!");`

   (e) `print("hello!")$;`

   (f) `'sqrt(2) = float(sqrt(2));`

   (g) `aa: 1024; 'aa^2 = aa^2;`

   A także odpowiedzieć na pytanie jak usunąć wszystkie zmienne lub jedną wskazaną zmienną, podać sposób wyświetlania wszystkich zmiennych.

5. Usunąć niewymierność z mianownika:

   (a) $\frac{12}{5\sqrt{2}}$, $\frac{6\sqrt{2}-4}{8\sqrt{2}}$

   (b) $\frac{2-\sqrt{5}}{\sqrt{5}-1}$, $\frac{3\sqrt{3}-4\sqrt{2}}{2\sqrt{3}-3\sqrt{2}}$

   (c) $\frac{4\sqrt[3]{2}-5}{3\sqrt[3]{2}}$, $\frac{\sqrt{6}-\sqrt{18}}{\sqrt{2}}$

   (d) $\frac{2-\sqrt{2}}{\sqrt{2}+1}$, $\frac{\sqrt{2}-3}{3-\sqrt{2}}$

   (e) $\frac{-\sqrt{27}+3\sqrt{3}}{7-4\sqrt{3}}$, $\frac{8-3\sqrt{20}}{5-2\sqrt{5}}$

6. Czy liczba $\frac{1}{\sqrt[3]{4}-\sqrt[3]{3}}$ jest równa:

   (a) $\sqrt[3]{4} + \sqrt[3]{3}$,

   (b) $\sqrt[3]{12} + \sqrt[3]{9} + 2\sqrt[3]{2}$,

   (c) $\sqrt[3]{4} + \sqrt[3]{3} + 2\sqrt[3]{12}$.

7. Liczba $(23 + \sqrt{97})^{17} + (23 - \sqrt{97})^{17}$ jest:

   (a) całkowita parzysta,

   (b) całkowita nieparzysta,

   (c) niewymierna.

8. Podać wyrażenia w środowisku Maxima, które utworzą poniższe wyrażenie:

   (a) $a + \frac{b}{c} + d$, $\frac{a+b}{c+d}$,

   (b) $\frac{1}{3}\frac{\frac{a}{b+c}-d^2}{e}$,

(c) $\left(1+3\frac{P}{h^2}\right)\left(1-(\frac{a}{l})^{0.6}\right)$,

(d) $\frac{c\cdot a_1\cdot a_2\sqrt{\frac{g(h_1-h_2)}{s}}}{3+\sqrt{a_1^2-a_2^2}}$,

(e) $2\frac{\pi\varepsilon}{\arccos\left(a^2+b^2-\frac{1}{2}\frac{d^2}{ab}\right)}$

9. Przykłady definicji funkcji i podstawień w wyrażeniach:

   (a) `f(x) := sin(x)/x;`,

   (b) `f(beta), f(1.57);`, `f(a+b);`

   (c) pochodna funkcji: `diff(f(x),x);`, druga pochodna `diff(diff(f(x),x), x);`,

   (d) `g(x) := tan(x);`,

   (e) całka funkcji `integrate(g(s),s);`, całka określona `integrate(g(s),s,%pi/4, %pi/3);`, `float(%);`.

   (f) przykład z podstawieniem dla wyrażenia `h : sin(x)/x;`,

   (g) `subst(y,x,h);`, `float(subst(1.57,x,h));`,

   (h) `diff(h,x);`.

10. Podstawowe wykresy (sprawdzić które polecenia tworzące wykres mają odmiany „wx" tj. np. plot2d i wxplot2d):

    (a) `contour_plot(x^2 + y^2, [x,-4,4],[y,-4,4]);`,

    (b) `f(x) := 0.5*x^3 + 2*x^2 - 3*x;`,

    (c) `plot2d(f(x),[x,-6,3]);`, `wxplot2d(f(x),[x,-6,3]);`,

    (d) `r:2;`, `plot2d([parametric, r*sin(t), r*cos(t), [t, -8*%pi, 8*%pi]]);`,

    (e) `plot3d(log(x^2 * y^2),[x,-2,2],[y,-2,2],[z,-8, 4],`
    `[palette,false],[color, magenta]);`,

    (f) `mandelbrot([iterations,30], [x, -2, 1], [y, -1.2, 1.2], [grid, 400, 400]);`,

    (g) zaprezentować drugi z fraktali tj. zbiór Julia (z pomocą funkcji julia).

11. Narysować wykresy funkcji (zwrócić uwagę na dziedziny poszczególnych funkcji):

    (a) $f(x)=x$, $f(x)=x^2$, $f(x)=x^3$, $f(x)=x^4$,

    (b) $f(x)=\frac{1}{x}$,

    (c) $f(x)=|x|$, $f(x)=|x|-3$, $f(x)=|x|+3$, $f(x)=|x-3|$, $f(x)=|x+3|$, $f(x)=|x-3|+3$, $f(x)=|x+3|-3$,

    (d) $f(x)=\sin x$, $f(x)=\cos x$, $f(x)=\operatorname{tg} x$, $f(x)=\operatorname{ctgh} x$, $f(x)=\frac{1}{\operatorname{tg} x}$,

    (e) $f(x)=\log_2 x$, $f(x)=\log_{\frac{1}{2}} x$, $f(x)=2^x$, $f(x)=\sqrt{x}$.

12. Narysować wykresy funkcji trygonometrycznych:

    (a) $y=2\sin x$, $y=-\frac{1}{2}\cos x$, $y=\sin\frac{1}{2}x$, $y=|\cos x|$

    (b) $y=\sin|x|$, $y=2\cos\frac{x}{2}+1$, $y=1-|\cos x|$, $y=\operatorname{tg}\left(\frac{-x}{2}\right)$

    (c) $y=\cos\left(2x-\frac{\pi}{6}\right)$, $y=\sin\left(2x+\frac{\pi}{3}\right)$

13. Podstawowe przekształcenia algebraiczne i metoda solve, find_root:

(a) `factor(x^2-b^2);`, `expand(%);`,

(b) `solve(a*x+b,x);`, `solve(x^2-2*x+3,x);`,

(c) `solve(a*x^2 + 3*b*y=5,x);`,

(d) rozwiązać równanie $x = cos(x)$,

(e) rozwiązać równanie $x = cos(x)$) numerycznie tj. skorzystać z funkcji find_root.

14. Definicja funkcji bez argumentów:

```
myprocname():=block(
    print("Hello, I am function without arguments!!!")
)$
```

uruchomienie następuje po podaniu nazwy `myprocname()$`, a przejście do nowej linii bez uruchomienia polecenia to użycie tylko klawisza enter:

```
myprocname();
```

Definicja funkcji z argumentami:

```
cylinder(r,h):=block([b,c,v,s],
    b:0,  c:0, v: 0, s: 0,
    b: %pi * r^2,
    ... inne usunięte fragmenty
    v: h * 2,
    return([v,b])
);
```

Ostatnie wyrażenie return zwraca jedną wartość tj. listę, ale dostęp do elementów jest możliwy po indeksach, a odczytujemy je w następujący sposób:

```
a: cylinder(r, h);
a[1];
a[2];
```

Wyjaśnić rolę wyrażenia `[b,c,v,s]`

15. Zdefiniować funkcje do obliczania pól, objętości następujących figur oraz brył:

(a) pola kwadratu, prostokąta, koła, deltoidu, trapzeu,

(b) objętość kuli, prostopadłościanu, ostrosłupa.

16. Opracować funkcje do rozwiązywania równania kwadratowego i zastosować ją aby rozwiązać dwa równania:

(a) $3x^2 - 5x + 2 = 0$,

(b) $x^2 - 3x + 2 = 0$.

17. Opracować podobne funkcje dla równań trzeciego oraz czwartego stopnia.

18. Wykonać następujące przykłady, i wyjaśnić rolę słowa `sum`:

(a) `sum(k,k,1,10);`, `sum(k^2,k,1,10);`,

(b) `sum(a[k],k,1,10);`,

(c) obliczyć sumę elementów $3 \cdot k^2 + \frac{k}{2}$ dla $k = 4..15$,

(d) `t: makelist (x^2, x, 1, 10, 1);`, `create_list (x^i, i, [1, 3, 7]);`,

(e) `cons(a,[b,c,d]);`,

(f) `apply(max, t);`, `rest(t,4);`, `rest(t, -4);`,

(g) `first(t);`, `last(t);`, `rest(rest(t,6-1), 8-10);`,

a także wyjaśnić rolę i znaczenie funkcji make_list, create_list, cons, wyjaśnić też działanie przykładu: `rest(rest(t,6-1), 8-10);`.

19. Konstrukcja warunkowa dostępna w środowisku Maxima, określana jest też jako operator specjalny:

```
if cond_1 then expr_1 else expr_0
```

Można stosować też elif.

```
if cond_1 then expr_1 elseif cond_2 then expr_2 elseif ... else expr_0
```

Przykład użycia:

```
a : 2$
b : 5$
if a < b then
  print("a is less than b")
else
  print("a is not less than b")$
```

Należy zwrócić uwagę kiedy nie stosujemy znaku średnika oraz na zastosowanie znaku dolara.

Zaimplementować funkcję o postaci:

$$f(x) = \begin{cases} -(x-1)^2 + 2 & \text{dla } x < 0 \\ 2x + 1 & \text{dla } 0 \leqslant x \leqslant 2 \\ 5e^{(-(x-2)^2)} & \text{dla } x > 2 \end{cases}$$

Narysować wykres funkcji f(x).

20. Podstawowe formy pętli for w środowisku Maxima:

```
for variable: initial_value step increment thru limit do body
for variable: initial_value step increment while condition do body
for variable: initial_value step increment unless condition do body
for variable in list do body
```

Przykład użycia konstrukcji for:

```
for k:1 thru 100 do (
    if primep(k) then
        print(k)
)$
```

For odliczany do tyłu:

```
for k:100 thru 1 step -1 do (
    if primep(k) then
        print(k)
)$
```

Wyjaśnić jak działają poniższe pętle for:

(a) `for a:-3 thru 35 step 7 do display(a)$`,

(b) `s: 0$ for i: 1 while i <= 10 do s: s+i$ s;`,

(c) `x:1$ thru 10 do x:x+x$ x;`,

(d) `for count: 2 next 3*count thru 20 do display (count)$`,

(e)
```
poly: 0$
for i: 1 thru 5 do
    for j: i step -1 thru 1 do
        poly: poly + i*x^j$
poly;
```

21. Zaimplementować w postaci funkcji Maxima iteracyjną metodę obliczania pierwiastka kwadratowego z liczby rzeczywistej:

$$x_i = \frac{1}{2}\left(x_{i-1} + \frac{s}{x_{i-1}}\right), \text{ dla } i = 1, 2, 3, \ldots \tag{1}$$

Funkcja powinna przyjmować trzy argumenty, liczbą z której obliczamy pierwiastek, wartość początkową, oraz maksymalną liczbę iteracji.

22. Opracować funkcję obliczającą n-tą liczbę Fibonacciego wg wzoru:

$$F_1 = 0, F_2 = 1, F_k = F_{k-1} + F_{k-2}, \text{ dla } k = 3, 4, \ldots, n \tag{2}$$

Opracowana funkcja powinna współpracować z następującym przykładem:

```
n: 10;
F: Fib2(10)
F[6], F[7], F[8];
for i: 1 while i <= n do print(F[i]);
```

jeśli Fib2 jest nazwą opracowanej funkcji.

23. Opracować funkcję generującą elementy następującego ciągu:

$$\sum_{k=1}^{n}(2k-1)^2 = 1^2 + 3^2 + 5^2 + 7^2 + \ldots + (2n-1)^2 \tag{3}$$

Opracować wersję funkcję stosując pętlę for.

24. Dla szeregu Taylora przybliżającego funkcję sinus:

$$\sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \ldots \tag{4}$$

zapis w notacji z sumą o n składnikach:

$$\sin x = \sum_{k=1}^{n} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \tag{5}$$

Podać procedurę obliczającą przybliżenie funkcji sin, o dwóch argumentach: x – argument oraz n – liczba składników.

Procedura powinna pozwolić obliczyć i narysować wykres szeregu Taylora oraz funkcji sin w celu porównania przybliżenia wartości sin szeregiem Taylora :

- `taylorsine(t, 5);`,
- `taylorsine(%pi/3, 5);`,
- `taylorsine(float(%pi/3), 5);`,
- porównać wykres otrzymany za pomocą `taylorsine` oraz wbudowanej funkcji `sin`.

25. Podać kod dla środowiska Maxima, który wyznaczy punkt stały następującej iteracji:

$$x_{k+1} = \sin(x_k) \text{ dla } k = 0, 1, 2, \ldots, \tag{6}$$

dla zadanej dokładności δ.

2 Rachunek zdań

1. Zbudować tabelę prawdy dla zdań (wykorzystać możliwości przeciążania operatorów oraz tworzeniach nowych w środowisku Maxima):
   (a) $\neg(p \wedge q)$
   (b) $\neg((p \leftrightarrow q) \vee (p \rightarrow r)) \rightarrow (\neg q \wedge p)$
   (c) $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$
   (d) $(p \wedge q) \leftrightarrow \neg(\neg p \vee \neg q)$
2. Opracować funkcję, lub odpowiednie fragmenty kodu, sprawdzające czy dane wyrażenie logiczne jest tautologią, powinna zostać także wyznaczona tabela wartości logicznych.
3. Zakładamy, że p, q, r przyjmują wartości logiczne. Sprawdzić spełnialność poniższych formuł:
   (a) $(p \wedge q) \rightarrow r)$
   (b) $(p \rightarrow r) \rightarrow q$
   (c) $\neg p \leftrightarrow (q \vee r)$
   (d) $\neg(p \leftrightarrow (q \vee r))$
   (e) $\neg(p \vee q) \wedge r$
4. Sprawdzić, czy podane poniżej zdania są tautologiami (wykorzystując opracowaną funkcję z zadania 2):
   (a) $(p \wedge q) \vee \neg(p \rightarrow q)$
   (b) $(\neg p \vee \neg q) \rightarrow (p \rightarrow \neg q)$
   (c) $((p \rightarrow q) \wedge \neg p) \rightarrow \neg q$
   (d) $(p \rightarrow r) \wedge (q \rightarrow s) \wedge (\neg p \vee \neg s) \rightarrow (\neg p \vee \neg q)$
   (e) $((p \rightarrow q) \vee r) \wedge (\neg p \rightarrow r)$

(f) $(p \to q) \wedge (\neg p \to r) \to (r \to \neg q)$
(g) $((\neg p \to q) \to r) \to \neg(p \to q)$
(h) $(((p \to q) \to r) \to \neg p) \to \neg q$
(i) $((p \to q) \to p) \to q$
(j) $p \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q)$
(k) $q \vee r \to (p \vee q \to q \vee r)$

3 Algebra zbiorów

1. Wykorzystując m. in. pakiet draw w środowisku Maxima opracować zestaw procedur do tworzenia diagramów Venna dla trzech zbiorów tradycyjnie oznaczanych jako A, B, C. Opracowane procedury powinny oferować dostęp do operacji sumy oraz iloczynu, pozostałe operacje można odtworzyć za pomocą wspomnianych dwóch operacji i wyrażeń logiki boolowskiej.

2. Pokazać za pomocą diagramów Venna następujące zależności:

   (a) $(A \setminus B) \cup C$
   (b) $(A \cap B) \cup C$
   (c) $((A \cap B) \cup (A \cap C)) \cap (B \cap C)^c$,
   (d) $(A \cup B) \cap A^c \subseteq B$
   (e) $A \cap (B \oplus C) = (A \cap B) \oplus (A \cap C)$
   (f) $A \oplus B \subseteq (A \oplus C) \cup (B \oplus C)$

4 Relacje, grafy i macierze

1. Wykorzystać pojęcie macierzy do rozwiązania następującego układu równań liniowych:

$$\begin{bmatrix} 3 & 5 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

   Przykłady tworzenia macierzy:

   - `T: matrix( [1,2,0], [0,3,4], [5,0,6] );`, w tym przypadku macierz jest budowana poprzez podawanie całych wierszy,
   - zmiana elementu w macierzy `T[2][2]: 8;`
   - tworzenie macierzy element po elemencie: `T: zeromatrix(3,3);`, a następnie: `T[1][1]: 4; T[1][2]: 2; T[2][2]: 1; T[2][3]: 2; T[3][1]: 8; T[3][3]: 9;`

2. Wykorzystać funkcję genmatrix oraz definicję lambda wyrażenia: `f: lambda ([i, j], random (10));` do utworzenie macierzy:

   - `A: genmatrix(f, 3, 3);`,
   - `B: genmatrix(f, 3, 3);`.

   Wytłumaczyć, dlaczego następujące wyrażenie języka Maxima:

```
f[i,j]:=random(10);
```

tworzy tylko za pierwszym wykorzystaniem poprawnie macierz, a następne próby generują macierz o tych samych elementach.

3. Utworzyć pomocniczą funkcję `createRandomMatrix` do tworzenia macierzy z wylosowanym elementach za pomocą funkcji random. Dla dwóch macierzy o losowo wybranych elementach (utworzone macierzy można wyświetlić z użyciem ich nazw tj. `A; B;`, upewnić się iż macierze nie zawierają zbyt dużo zer):

   ```
   A: createRandomMatrix(4, 3, 0, 5):
   B: createRandomMatrix(3, 5, 0, 5):
   ```

   Wykonać następujące operacje:

   - mnożenie,
   - transpozycję,
   - odczytanie fragmentu macierzy np. `D := A[ 1..3, 1..2 ];` (zapis za pomocą pseudokodu),
   - wyznaczyć odwrotność macierzy,
   - utworzyć macierz jednostkową.

4. Mamy następujące macierze:

$$A = \begin{bmatrix} -1 & 0 & 2 \\ 1 & 3 & -2 \\ 4 & 2 & 3 \end{bmatrix}, B = \begin{bmatrix} 6 & 8 & 5 \\ 4 & -2 & 7 \\ 3 & 1 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 3 \\ 2 & -4 \\ 5 & -2 \end{bmatrix}$$

   Wyznaczyć macierze wykorzystując środowisko Maxima, o ile będzie to możliwe:

   (a) A^T, C^T,
   (b) $A + B$, $A + C$,
   (c) $(A + B)^T$,
   (d) $A^T + B^T$,
   (e) $B + B^T$, $C + C^T$,
   (f) $(A + A) + B$,

   gdzie A^T oznacza transpozycję macierzy.

5. Mamy następujące macierze:

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 0 & 2 \end{bmatrix} B = \begin{bmatrix} 2 & 1 \\ -1 & 0 \\ -2 & 3 \end{bmatrix} C = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

   Wyznaczyć następujące macierze, o ile one istnieją:

   (a) AB, BA, ABA
   (b) $A + B^T$, $3A^T - 2B$
   (c) $(AB)^2$
   (d) AC, BC, C^2
   (e) C^TC, CC^T

(f) $73C$

6. Niech macierze A, B, C będą określone w następujący sposób:

$$A = \begin{bmatrix} -1 & 4 \\ 2 & 5 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$$

   (a) $(AB)C$, $A(BC)$
   (b) $B(AC)$, $(BA)C$
   (c) AB, BA
   (d) AC, CA
   (e) A^2

7. Tworzenie grafów, Maxima korzysta z „pakietu graphs", można go dołączyć do istniejącego dokumentu:

```
load("graphs");
```

   Przykłady tworzenie grafów:

   - graf nieskierowany:

```
g: create_graph([1,2,3,4,5,6,7,8,9],
                    [[1, 2], [1, 3], [2, 3], [6, 7], [7, 8]]);
print_graph(g);
draw_graph(g);
```

   - graf skierowany:

```
h: create_graph([1,2,3,4,5], [[1, 2], [2, 3], [2, 5], [3, 4]],
                             'directed=true);
print_graph(h);
draw_graph(h, edge_width=2, show_id=true);
```

     Można również zaprezentować macierz incydencji:

```
adjacency_matrix(h);
```

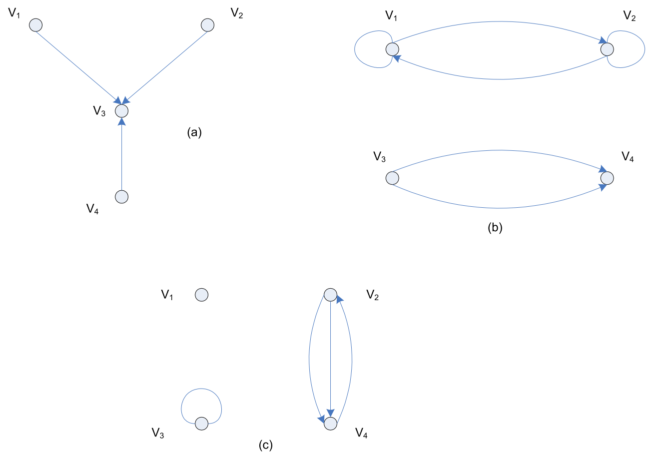
   - utworzenie grafu na podstawie macierzy incydencji:

```
A: matrix( [0, 1, 0, 0],
           [1, 0, 1, 0],
           [0, 1, 0, 1],
           [0, 0, 1, 0]);
g: from_adjacency_matrix(A);
draw_graph(g);
vertices(g);
edges(g);
```

8. Wyznacz macierze grafów skierowanych przedstawianych na poniższym Rysunku 1:

9. Dla każdej z poniższych macierzy incydencji narysować za pomocą Maximy i pakietu graphs odpowiedni graf skierowany:

$$\begin{pmatrix} 0 & 0 & 2 & 1 \\ 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Rysunek 1: Grafy do zadania

10. Dla każdej z poniższych macierzy incydencji narysować za pomocą Maximy i pakietu graphs odpowiedni graf nieskierowany:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Narysować też graf skierowany na podstawie drugiej macierzy.

11. Opracować procedury sprawdzające, podstawowe własności relacji opisanej grafem skierowanym tj. czy relacja jest zwrotna, przeciw-zwrotna, symetryczna, przeciw-symetryczna, antysymetryczna, przechodnia.

12. Opracować procedurę aktualizacji macierzy A zgodnie z regułami gry w życie:

$$G = \begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

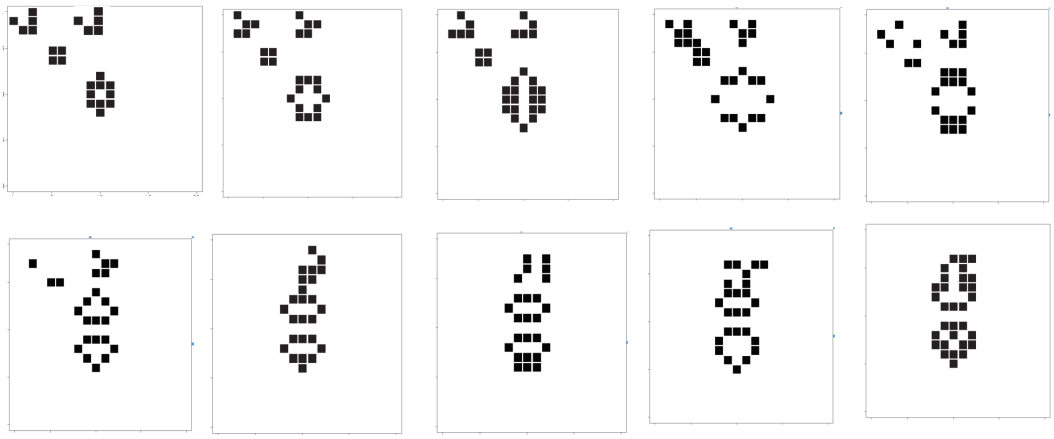
Obowiązują cztery reguły gry w życie:

- żywa komórka (wartość jeden) z mniej niż dwoma żywymi sąsiadami, umiera, tj. otrzymuje zero,
- żywa komórka (wartość jeden) z dwoma lub trzema żywymi sąsiadami przeżywa, tj. jej stan się nie zmienia,
- żywa komórka (wartość jeden) z więcej niż trzema żywymi sąsiadami umiera, tj. otrzymuje zero,
- martwa komórka posiadająca trzech żywych sąsiadów ożywa.

Macierz można zaprezentować w następujący sposób:

```
gdim: matrix_size(g);
draw(
    gr2d(colorbox=false,
        palette=gray,
        image(g,1,1,gdim[1],gdim[2]))
);
```

Pierwsze iteracje macierzy A powinny dać następujące postacie macierzy opisującej aktualny stan gry w życie:



5 Indukcja i rekurencja

1. Opracować procedurą która wspomaga dowody indukcyjne w pierwszym etapie sprawdzania poprawności indukcji dla $n = 1, 2, 3, \ldots$, inaczej mówiąc dla pierwszych elementów mających spełniać badaną formułę. Przykładowe wywołanie może przedstawiać się w następujący sposób:

   ```
   checkFirstStepInd( lefteqn, righteqn, n, 1, 5)
   ```

   Argumenty `lefteqn`, `righteqn` to procedury opisujące odpowiednio lewą i prawą stronę badanej formuły dowodzonej za pomocą indukcji matematycznej.

2. Wykorzystując elementy zdefiniowane w zadaniu 5.1 oraz zakładając, że $n \in \mathbb{N}$, udowodnić indukcyjnie:

   (a) $\sum_{i=1}^{n} i = \dfrac{n(n+1)}{2}$,

(b) $\sum_{i=0}^{n} r^i = \frac{r^{n+1}-1}{r-1}$,

(c) $\sum_{i=1}^{n} i^2 = 1+4+9+\ldots+n^2 = \frac{n(n+1)(2n+1)}{6}$ dla $n \in \mathbb{P}$,

(d) $1^3+2^3+3^3+\ldots+n^3 = \sum_{k=1}^{n} k^3 = \frac{1}{4}n^2(n+1)^2$ dla $n \in \mathbb{P}$,

(e) $4+10+16+\ldots+(6n-2) = n(3n+1)$ dla wszystkich $n \in \mathbb{P}$,

(f) $\frac{1}{1\cdot 5}+\frac{1}{5\cdot 9}+\frac{1}{9\cdot 13}+\ldots+\frac{1}{(4n-3)(4n+1)} = \frac{n}{4n+1}$ dla $n \in \mathbb{P}$,

(g) $3+11+\ldots+(8n-5) = 4n^2-n$ dla $n \in \mathbb{P}$.

3. Opracować procedurę, która rozwiązuje rekurencje o postaci (nowy element powstaje z pomocą dwóch poprzednich elementów):

$$s_n = as_{n-1} + bs_{n-2} \text{ gdy } n \geqslant 2,$$

gdzie $a, b \in \mathbb{Z}$ oraz s_0, s_1 także są elementami ze zbioru liczb naturalnych.

4. Podać wzór jawny na s_n dla następujących ciągów, wykorzystując funkcję opracowaną w poprzednim zadaniu:

(a) $s_0 = 2,\ s_1 = -1$ oraz $s_n = s_{n-1} + 6s_{n-2}$ gdy $n \geqslant 2$,

(b) $s_0 = 2$ oraz $s_n = 5s_{n-1}$ gdy $n \geqslant 1$,

(c) $s_0 = 1,\ s_1 = 8$ oraz $s_n = 4s_{n-1} - 4s_{n-2}$ gdy $n \geqslant 2$,

(d) $s_0 = 1,\ s_1 = 4$ oraz $s_n = s_{n-2}$ gdy $n \geqslant 2$,

(e) $s_0 = 1,\ s_1 = -3$ oraz $s_n = -2s_{n-1} + 3s_{n-2}$ gdy $n \geqslant 2$.

6 Kombinatoryka

1. Zaimplementować funkcję implementującą symbol Newtona w dwóch odmianach:

(a) w bezpośredniej wykorzystującej definicję symbolu Newtona,

(b) poprawionej uwzględniającej wartości jakie przyjmują obliczenia pośrednie.

7 Arytmetyka modularna oraz arytmetyka dużych liczb całkowitych

1. Dokonać implementacji funkcji NWD (największy wspólny dzielnik, wersja podstawowa i rozszerzona przydatna do rozwiązywania kongruencji – RNWD), algorytm dzielenia w środowisku Maxima (pseudo kod znajduje się na liście ćwiczeniowej) oraz operację mod – dzielenia modulo z resztą. Zaimplementować także procedurę MLES (modular linear equation solver) do rozwiązywania kongruencji. Pseudo kod MLES:

```
MLES(a, b, n) ->
  (d, x', y') = RNWD(a, n)
  if d|b then
      x0 = x'(b/d) mod n
      for i=0 to d-1
       pokaz (x0 + i(n/d)) mod n
  else
      pokaz "brak rozwiązań"
```

2. Stosując opracowane procedury rozwiązać następujące kongruencje:

   (a) $10x \equiv 1 \pmod{37}$,

   (b) $5x \equiv 1 \pmod{26}$,

   (c) $17x \equiv 1 \pmod{26}$,

   (d) $8x \equiv 6 \pmod{15}$,

   (e) $643x \equiv 1 \pmod{2000}$.

3. Stosując opracowane procedury rozwiązać układy kongruencji:

   (a) $\begin{cases} x \equiv 1 \pmod{13} \\ x \equiv 4 \pmod{15} \end{cases}$,

   (b) $\begin{cases} x \equiv 0 \pmod{13} \\ x \equiv 65 \pmod{99} \end{cases}$,

   (c) $\begin{cases} x \equiv 8 \pmod{13} \\ x \equiv 65 \pmod{99} \end{cases}$.

4. Dokonać implementacji algorytmu szybkiego potęgowania modularnego. Postać w pseudokodzie jest następująca:

```
bits = postac_binarna(b);
m := liczba_bitow(bits);
a := a mod n;
result := 1;
x := a;
for i := 0 to m do
begin
    if bits[i] = 1 then
    begin
        result := result * x;
        result := result mod n;
    end
    x = x * x;
    x = x mod n;
end
```

5. Obliczyć następujące potęgi modularne wykorzystując szybki algorytm potęgowania modularnego:

   (a) $12^{56} \bmod 7$

   (b) $3^{1853} \bmod 13$

   (c) $4^{4000} \bmod 12$

   (d) $8^{234} \bmod 18$

   (e) $15^{90} \bmod 11$

   (f) $3^{10000000} \bmod 5$

   (g) $3^{333333} \bmod 33$

6. (Zadanie dodatkowe) Wykonać, wykorzystując funkcje implementujące rozszerzony algorytm Euklidesa oraz potęgowanie modularne, implementację algorytmu RSA. Opracowane funkcje mają pozwalać na szyfrowanie i deszyfrowanie tekstu np.: jeśli para (e, n) to klucz publiczny, a (d, n) reprezentuje klucz prywatny, to przykłady wykorzystania funkcji prezentują się następująco:

```
plainText: "ciąg znaków";
encmsg: encryptionRSA(plainText, e, n);
decmsg: decryptionRSA(encmsg, d, n);
```