

22 maja 2017

Damian Filipowicz  
Maciej Marciniak  
Dawid Wiktorski  
I12-Z1

# Sprawozdanie z laboratorium sztucznej inteligencji Informatyka, sem. VI


## **Spis treści:**

<b>Opis zadania</b>	3
<b>Założenia realizacyjne</b>	4
1. Założenia dodatkowe	4
2. Metody, strategie oraz algorytmy wykorzystywane do rozwiązania zadania.	4
3. Języki programowania, narzędzia informatyczne i środowiska używane do implementacji systemu.	4
<b>Podział prac</b>	5
<b>Opis implementacji</b>	5
<b>Użytkowanie i testowanie programu</b>	6
<b>Tekst programu</b>	7
Program prologowy	7
Program graficzny Javy	9

# I. Opis zadania

Celem zadania semestralnego jest rozwiązanie łamigłówki S562JK (znajdującej się na stronie <http://www.logic-puzzles.org/pdf/S562JK.pdf>) za pomocą sztucznej inteligencji. Program tworzony będzie w języku programowania SWI-Prolog z wykorzystaniem paradygmatu CLP (tzn. z więzami).

Łamigłówka S562JK polega na uzupełnieniu macierzy 4x5 na podstawie pisemnych wskazówek. Tabela składa się z czterech kolumn zawierających daty ślubu, imiona, nazwy drinków oraz tytułu piosenki. Wiersze są powiązaniem tych 4 zmiennych ze sobą, tworząc opis pięciu osób. Widok łamigłówki znajduje się na Obrazie 1.

Wedding Years		Chandler	Holden	Jane	Skylar	Tiffany	cherry cola	cola	diet cola	grape soda	root beer	All By Myself	Happy Times	Moon River	One More Time	Summer Blues
	1925															
	1941															
	1947															
	1972															
	1976															
Songs	All By Myself															
	Happy Times															
	Moon River															
	One More Time															
	Summer Blues															
Soft Drinks	cherry cola															
	cola															
	diet cola															
	grape soda															
	root beer															

**Logic Puzzles**  
Presented by Puzzle Baron

Puzzle ID: S562JK  
For hints, solutions and more puzzles, go to [www.Printable-Puzzles.com](http://www.Printable-Puzzles.com)!

Wedding Years	First Names	Soft Drinks	Songs
1925			
1941			
1947			
1972			
1976			

- The 5 people were the one who drinks cola, Chandler, the person married in 1947, the singer known for Moon River, and the one who drinks root beer.
- The one who drinks cherry cola doesn't sing Moon River or One More Time.
- Either the person married in 1972 or the person married in 1976 loves to drink root beer.
- The one who drinks root beer got married before the singer known for All By Myself.
- Holden got married before the one who drinks cola.
- The singer known for Moon River got married before Skylar.
- Of Chandler and Holden, one loves to drink grape soda and the other got married in 1972.
- The singer known for All By Myself is Jane.
- The person married in 1941 is not Tiffany.
- The singer known for Summer Blues loves to drink grape soda.

Obraz 1. Łamigłówka S562JK

Na stronie twórcy łamigłówki można znaleźć rozwiązanie zagadki. Uzupełniona tabela znajduje się na Obrazie 2.

1925	Tiffany	Diet cola	Moon River
1941	Chandler	Grape soda	Summer Blues
1947	Skylar	Cherry cola	Happy Times
1972	Holden	Root beer	One More Time
1976	Jane	Cola	All By Myself

Obraz 2. Rozwiązanie łamigłówki S562JK

## II. Założenia realizacyjne

### 1. Założenia dodatkowe

Dane wejściowe nie są przedstawione w postaci tekstu, ale liczb, których wartości zależne są od położenia w poszczególnych tabelach.

### 2. Metody, strategie oraz algorytmy wykorzystywane do rozwiązania zadania.

Podstawową metodyką realizacji projektu było użycie biblioteki clpfd, czyli programowania z więzami. Dodatkowo użyto funkcji nth1 w celu powiązania ze sobą ograniczeń pomiędzy tabelami.

### 3. Języki programowania, narzędzia informatyczne i środowiska używane do implementacji systemu.

Problem łamigłówki S562JK został rozwiązany w języku programowania logicznego Prolog oraz z użyciem interfejsu SWI-Prolog w wersji 7. Wykorzystano bibliotekę clpfd (ang. Constraint Logic Programming over Finite Domains), która umożliwia programowanie z więzami. Interfejs graficzny programu zrealizowany został w języku Java w środowisku Eclipse wraz z wgranym dodatkiem Designer Swing.

### III. Podział prac

Podział prac przedstawiony poniżej przedstawia przydział zadań do osób wnoszących największy wkład w dany fragment projektu. Każdy członek zespołu uczestniczył w każdym etapie realizacji.

Autor	Podzadanie
Damian Filipowicz	Sformułowanie powiązań pomiędzy tabelami
Dawid Wiktorski	Utworzenie ograniczeń na podstawie wskazówek łamigłówki
Maciej Marciniak	Przygotowanie programu graficznego

### IV. Opis implementacji

#### Struktury wykorzystane w programie:

W programie posiadamy 6 list (A, B, C, D, E, F) z których każda reprezentuje odpowiednią tabelę z łamigłówki. W każdej z list, pozycja (indeks) na niej odpowiada wartości z kolumny, a wartość w tym miejscu odpowiada pozycji z wiersza. Zilustrowany podział tablic łamigłówki przedstawiono na obrazie 3.

		First Names					Soft Drinks					Songs				
		Chandler	Holden	Jane	Skylar	Tiffany	cherry cola	cola	diet cola	grape soda	root beer	All By Myself	Happy Times	Moon River	One More Time	Summer Blues
Wedding Years	1925															
	1941															
	1947		A						B					C		
	1972															
	1976															
Songs	All By Myself															
	Happy Times															
	Moon River		D						E							
	One More Time															
	Summer Blues															
Soft Drinks	cherry cola															
	cola															
	diet cola															
	grape soda															
	root beer															
			F													

Obraz 3.Podział tablic łamigłówki

*Funkcje, procedury lub predykaty zdefiniowane w programie;*

- funkcja `s562jk` służąca do wywołania programu z parametrami `A1...A5`, `B1...B5` oraz `C1... C5` odpowiadającym wynikom łamigłówki,
- `nth1`- funkcja służąca w przypadku tego programu do wyciągania wartości z podanego indeksu lub zapisywania do elementu o podanym indeksie wartości.

## V. Użytkowanie i testowanie programu

Uruchomienie programu napisanego w Javie przedstawi taki widok:



Wedding Year	First Names	Soft Drinks	Songs
1925	Tiffany	Diet cola	Moon River
1941	Chandler	Grape soda	Summer Blues
1947	Skylar	Cherry cola	Happy Times
1972	Holden	Root beer	One More Time
1976	Jane	Cola	All By Myself

Jest to rozwiązanie łamigłówki zgodne z tym podanym przez producentów gry. Testowanie programu prologowego odbywało się poprzez wywołanie predykatu `s562jk` i porównywaniu wyników uzyskanych z treścią zagadki co pozwalało na wychwycenie więzów, których programowi jeszcze brakowało.

Do uruchomienia programu niezbędne jest posiadanie programu SWI-Prolog w wersji minimalnej 7 oraz zainstalowanej Javy 1.8u45 64 bitowej. Dodatkowo należy utworzyć zmienne środowiskowe:

- `SWI_HOME_DIR` z odniesieniem do folderu z instalacją prologa,
- `LIB` z odniesieniem do folderu z bibliotekami Javy oraz biblioteką `jpl.jar` Prologa,
- dodać do zmiennej `PATH` lokalizację folderu bin plików instalacyjnych SWI-Prolog.

Program uruchamia się z poziomu konsoli systemowej poleceniem `java -jar S562JK.jar` lub poprzez dwuklik na ikonę programu.

Zaimplementowany program nie posiada danych wejściowych, stąd wywołanie programu daje tylko jeden wynik dedykowany pod konkretną łamigłówkę.

## VI. Tekst programu

### Program prologowy

Program w języku Prolog znajduje się w Listingu 1.

```
:- use_module(library(clpfd)).

s562jk(
    A1,A2,A3,A4,A5,
    B1,B2,B3,B4,B5,
    C1,C2,C3,C4,C5):-

    X1 = [A1,A2,A3,A4,A5],
    X2 = [B1,B2,B3,B4,B5],
    X3 = [C1,C2,C3,C4,C5],
    X4 = [D1,D2,D3,D4,D5],
    X5 = [E1,E2,E3,E4,E5],
    X6 = [F1,F2,F3,F4,F5],

    /* Ograniczenia podstawowe */
    X1 ins 1..5,
    all_different(X1),
    X2 ins 1..5,
    all_different(X2),
    X3 ins 1..5,
    all_different(X3),
    X4 ins 1..5,
    all_different(X4),
    X5 ins 1..5,
    all_different(X5),
    X6 ins 1..5,
    all_different(X6),

    /******zdanie 1 *****/
    A1 in 1..2 ∨ 4..5,
    D1 in 1..2 ∨ 4..5,
    F1 in 1 ∨ 3..4,
    B2 in 1..2 ∨ 4..5,
    C3 in 1..2 ∨ 4..5,
    E2 in 1..2 ∨ 4..5,
    E5 in 1..2 ∨ 4..5,
    B5 in 1..2 ∨ 4..5,

    /******zdanie 2 *****/
    E1 in 1..3 ∨ 5,

    /******zdanie 3 *****/
    B5 in 4 ∨ 5,

    /******zdanie 4 *****/
    E5 #< 5,
    B5 #< 5,
    C1 #> B5,

    /******zdanie 5 *****/
    A2 #< B2,
    F2 #\= 2,
    B2 #\= 1,

    /******zdanie 6 *****/
    A4 #> 1,
    D4 in 1..2 ∨ 4..5,
    C3 #< A4,

    /******zdanie 7 *****/
    A1 #= 4 #∨ A2 #= 4,
    A1 #= 4 #<==> F2 #= 4,
    A2 #= 4 #<==> F1 #= 4,

    /******zdanie 8 *****/
    D3 #= 1,
    D1 #\= 1,
    D2 #\= 1,
    D4 #\= 1,
    D5 #\= 1,

    /******zdanie 9 *****/
    A5 #\= 2,

    /******zdanie 10 *****/
    E4 #= 5,
    E1 #\= 5,
    E2 #\= 5,
    E3 #\= 5,
    E5 #\= 5,
    B4 #= C5,
```

```

/* Powiązania pomiędzy tabelami */
nth1(D1,[C1,C2,C3,C4,C5],A1),
nth1(D2,[C1,C2,C3,C4,C5],A2),
nth1(D3,[C1,C2,C3,C4,C5],A3),
nth1(D4,[C1,C2,C3,C4,C5],A4),
nth1(D5,[C1,C2,C3,C4,C5],A5),

/*****/
nth1(E1,[B1,B2,B3,B4,B5],C1),
nth1(E2,[B1,B2,B3,B4,B5],C2),
nth1(E3,[B1,B2,B3,B4,B5],C3),
nth1(E4,[B1,B2,B3,B4,B5],C4),
nth1(E5,[B1,B2,B3,B4,B5],C5),

/*****/
nth1(F1,[B1,B2,B3,B4,B5],A1),
nth1(F2,[B1,B2,B3,B4,B5],A2),
nth1(F3,[B1,B2,B3,B4,B5],A3),
nth1(F4,[B1,B2,B3,B4,B5],A4),
nth1(F5,[B1,B2,B3,B4,B5],A5),

/*****/
nth1(D1buf,[A1,A2,A3,A4,A5],C1),
nth1(D1buf,[D1,D2,D3,D4,D5],1),

nth1(D2buf,[A1,A2,A3,A4,A5],C2),
nth1(D2buf,[D1,D2,D3,D4,D5],2),

nth1(D3buf,[A1,A2,A3,A4,A5],C3),
nth1(D3buf,[D1,D2,D3,D4,D5],3),

nth1(D4buf,[A1,A2,A3,A4,A5],C4),
nth1(D4buf,[D1,D2,D3,D4,D5],4),

nth1(D5buf,[A1,A2,A3,A4,A5],C5),
nth1(D5buf,[D1,D2,D3,D4,D5],5),

/*****/
nth1(F1buf,[A1,A2,A3,A4,A5],B1),
nth1(F1buf,[F1,F2,F3,F4,F5],1),

nth1(F2buf,[A1,A2,A3,A4,A5],B2),
nth1(F2buf,[F1,F2,F3,F4,F5],2),

nth1(F3buf,[A1,A2,A3,A4,A5],B3),
nth1(F3buf,[F1,F2,F3,F4,F5],3),

nth1(F4buf,[A1,A2,A3,A4,A5],B4),
nth1(F4buf,[F1,F2,F3,F4,F5],4),

nth1(F5buf,[A1,A2,A3,A4,A5],B5),
nth1(F5buf,[F1,F2,F3,F4,F5],5),

/*****/
nth1(E1buf,[C1,C2,C3,C4,C5],B1),
nth1(E1buf,[E1,E2,E3,E4,E5],1),

nth1(E2buf,[C1,C2,A3,A4,A5],B2),
nth1(E2buf,[E1,E2,E3,E4,E5],2),

nth1(E3buf,[C1,C2,C3,C4,C5],B3),

nth1(E3buf,[E1,E2,E3,E4,E5],3),

nth1(E4buf,[C1,C2,C3,C4,C5],B4),
nth1(E4buf,[E1,E2,E3,E4,E5],4),

nth1(E5buf,[C1,C2,C3,C4,C5],B5),
nth1(E5buf,[E1,E2,E3,E4,E5],5),
!.

```

Listing 1. Program w języku Prolog



## Program graficzny Javy

Kod aplikacji napisanej w Java znajduje się w Listingu 2.

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JTable;
import java.awt.BorderLayout;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.SwingConstants;
import org.jpl7.Query;
import java.awt.Font;
import java.awt.Color;
import javax.swing.table.DefaultTableModel;
import java.awt.Toolkit;

public class S562JK {
    private JFrame frame;
    private String[] columnNames = {"Wedding Year", "First Names", "Soft Drinks", "Songs"};
    private JTable table;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    S562JK window = new S562JK();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public S562JK() {
        initialize();
    }
    private void initialize() {
        frame = new JFrame();
        frame.setTitle("\u0141amig\u0142\u00F3wka S562JK");
        frame.setResizable(false);
        frame.setBounds(100, 100, 673, 160);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        String t1 = "consult('Lamiglowka S562JK.pl')";
        Query q1 = new Query(t1);
        q1.hasSolution();
        String t2 = "s562jk(A1,A2,A3,A4,A5, B1,B2,B3,B4,B5, C1,C2,C3,C4,C5)";
        Query q2 = new Query(t2);
    }
}
```

```

String[][] wynik = new String[5][4];
wynik[0][0] = "1925";
wynik[1][0] = "1941";
wynik[2][0] = "1947";
wynik[3][0] = "1972";
wynik[4][0] = "1976";

wynik[q2.oneSolution().get("A1").intValue()-1][1] = "Chandler";
wynik[q2.oneSolution().get("A2").intValue()-1][1] = "Holden";
wynik[q2.oneSolution().get("A3").intValue()-1][1] = "Jane";
wynik[q2.oneSolution().get("A4").intValue()-1][1] = "Skylar";
wynik[q2.oneSolution().get("A5").intValue()-1][1] = "Tiffany";

wynik[q2.oneSolution().get("B1").intValue()-1][2] = "Cherry cola";
wynik[q2.oneSolution().get("B2").intValue()-1][2] = "Cola";
wynik[q2.oneSolution().get("B3").intValue()-1][2] = "Diet cola";
wynik[q2.oneSolution().get("B4").intValue()-1][2] = "Grape soda";
wynik[q2.oneSolution().get("B5").intValue()-1][2] = "Root beer";

wynik[q2.oneSolution().get("C1").intValue()-1][3] = "All By Myself";
wynik[q2.oneSolution().get("C2").intValue()-1][3] = "Happy Times";
wynik[q2.oneSolution().get("C3").intValue()-1][3] = "Moon River";
wynik[q2.oneSolution().get("C4").intValue()-1][3] = "One More Time";
wynik[q2.oneSolution().get("C5").intValue()-1][3] = "Summer Blues";

table = new JTable(wynik, columnNames);
table.setModel(new DefaultTableModel(
    wynik,
    new String[] { "Wedding Year", "First Names", "Soft Drinks", "Songs" }
) {
    Class[] columnTypes = new Class[] {
        String.class, Object.class, Object.class, String.class
    };
    public Class getColumnClass(int columnIndex) {
        return columnTypes[columnIndex];
    }
});
table.setBackground(Color.WHITE);
table.setFillsViewportHeight(true);
table.setFont(new Font("Tahoma", Font.PLAIN, 17));
JScrollPane scrollPane = new JScrollPane(table);
frame.getContentPane().add(scrollPane, BorderLayout.CENTER);

JLabel lblNewLabel = new JLabel("S562JK");
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC, 20));
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
frame.getContentPane().add(lblNewLabel, BorderLayout.NORTH);
}
}

```

Listing 2. Program w języku Java