27 czerwiec 2017

Damian Filipowicz

Maciej Marciniak

Dawid Wiktorski

I12-Z1

## 

Sprawozdanie z laboratorium sztucznej inteligencji

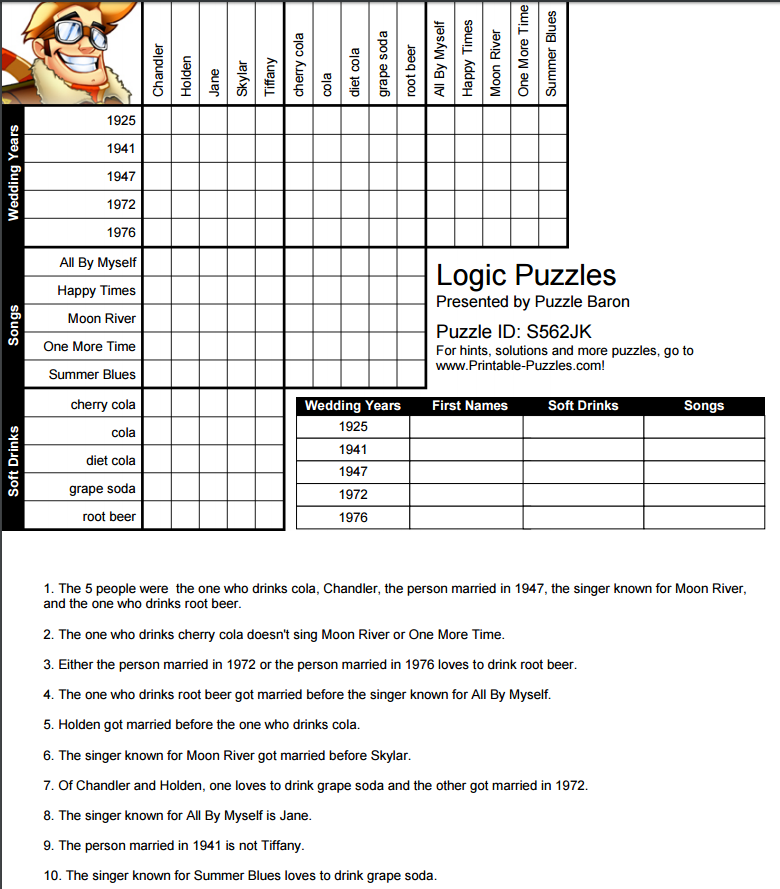
Informatyka, sem. VI

# Opis zadania

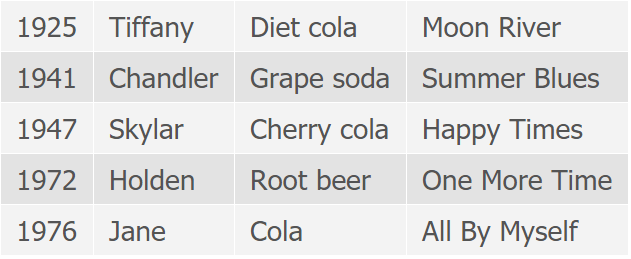
Celem zadania semestralnego jest napisanie programu w predykacie CLP rozwiązujący łamigłówkę S562JK[[1]](#footnote-1) za pomocą sztucznej inteligencji.

Łamigłówka S563JK 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 powiązaniami tych 4 zmiennych ze sobą, tworząc opis pięciu osób. Widok łamigłówki znajduje się na Rys. 1.

Dane wejściowe programu z tabel przeniesione zostały do postaci liczbowej. Parametry są uzależnione od położenia w poszczególnych macierzach zależności pomiędzy kolumnami rozwiązania.

  
Rys 1.Łamigłówka S562JK

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

  
Rys. 2. Rozwiązanie łamigłówki S562JK[[2]](#footnote-2)

# Założenia realizacyjne

## **1. Metody, strategie oraz algorytmy wykorzystywane do rozwiązania zadania.**

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

## **2. 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 7.3.18. 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 Neon.2 Release (4.6.2) wraz z wgranym dodatkiem Designer Swing.

# Podział prac

Podział prac przedstawiony w Tabeli 1 przedstawia przydział zadań do osób wnoszących największy wkład w dany fragment projektu. Każdy członek zespołu uczestniczył w każdym etapie realizacji.

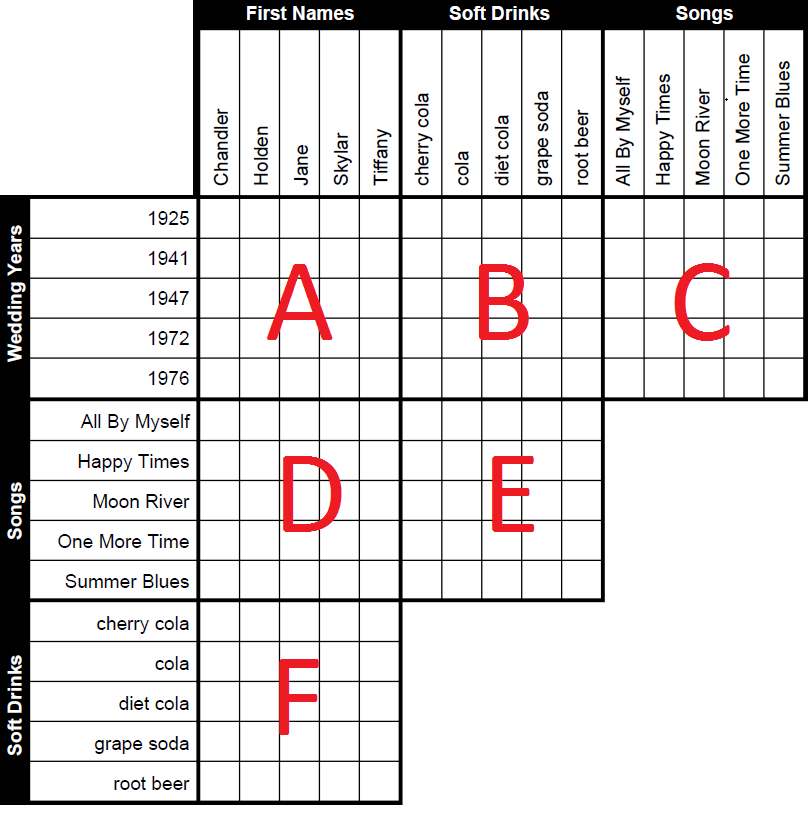
Tabela 1 Podział prac

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

# Opis implementacji

## **Struktury wykorzystane w programie:**

W programie posiadamy 6 list (A, B, C, D, E, F) z których każda reprezentuje odpowiednią tabele 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 łamigłówki na tabele przedstawiono na Rys 3.



Rys 3.Podział łamigłówki na listy

Lista A odzwierciedla macierz zaznaczoną czerwonym znakiem “A” na Rys. 3. Indeks w tej liście oznacza konkretną kolumnę o indeksach numerowanych od 0 do 4, które przyjmują kolejno wartości: Chandler, Holden, Jane, Skylar, Tiffany. Wiersze numerowane są od 1 do 5 i przyjmują kolejno wartości: 1925, 1941, 1947, 1971, 1976. Odpowiednio te wartości umieszczane zostają w konkretnej pozycji listy przypisując datę ślubu do danego imienia osoby.

Lista B odzwierciedla macierz zaznaczoną czerwonym znakiem “B” na Rys. 3. Indeks w tej liście oznacza konkretną kolumnę o indeksach numerowanych od 0 do 4, które przyjmują kolejno wartości: cherry cola, cola, diet cola, grade soda, root beer. Wiersze numerowane są od 1 do 5 i przyjmują kolejno wartości: 1925, 1941, 1947, 1971, 1976. Odpowiednio te wartości umieszczane zostają w konkretnej pozycji listy przypisując datę ślubu do danego napoju.

Lista C odzwierciedla macierz zaznaczoną czerwonym znakiem “C” na Rys. 3. Indeks w tej liście oznacza konkretną kolumnę o indeksach numerowanych od 0 do 4, które przyjmują kolejno wartości: All By Myself, Happy Times, Moon River, One More Time, Summer Blues. Wiersze numerowane są od 1 do 5 i przyjmują kolejno wartości: 1925, 1941, 1947, 1971, 1976. Odpowiednio te wartości umieszczane zostają w konkretnej pozycji listy przypisując datę ślubu do danego utworu muzycznego.

Lista D odzwierciedla macierz zaznaczoną czerwonym znakiem “D” na Rys. 3. Indeks w tej liście oznacza konkretną kolumnę o indeksach numerowanych od 0 do 4, które przyjmują kolejno wartości: Chandler, Holden, Jane, Skylar, Tiffany. Wiersze numerowane są od 1 do 5 i przyjmują kolejno wartości: All By Myself, Happy Times, Moon River, One More Time, Summer Blues. Odpowiednio te wartości umieszczane zostają w konkretnej pozycji listy przypisując osobę do danego utworu muzycznego.

Lista E odzwierciedla macierz zaznaczoną czerwonym znakiem “E” na Rys. 3. Indeks w tej liście oznacza konkretną kolumnę o indeksach numerowanych od 0 do 4, które przyjmują kolejno wartości: cherry cola, cola, diet cola, grape soda, root beer. Wiersze numerowane są od 1 do 5 i przyjmują kolejno wartości: All By Myself, Happy Times, Moon River, One More Times, Summer Blues. Odpowiednio te wartości umieszczane zostają w konkretnej pozycji listy przypisując napój do utworu muzycznego.

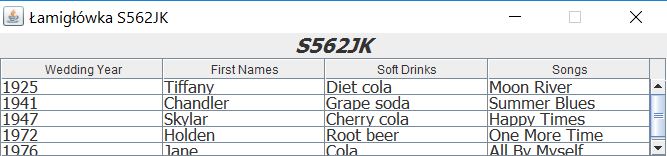
Lista F odzwierciedla macierz zaznaczoną czerwonym znakiem “F” na Rys. 3. Indeks w tej liście oznacza konkretną kolumnę o indeksach numerowanych od 0 do 4, które przyjmują kolejno wartości: Chandler, Holden, Jane, Skylar, Tiffany. Wiersze numerowane są od 1 do 5 i przyjmują kolejno wartości: cherry cola, cola, diet cola, grape soda, root beer. Odpowiednio te wartości umieszczane zostają w konkretnej pozycji listy przypisując napój do danego imienia osoby.

**2. Predykat zdefiniowany w programie.**

● predykat s562jk/15 służący do wywołania programu z parametrami A1…A5, B1...B5 oraz C1… C5 odpowiadającym wynikom łamigłówki oraz listom A, B i C opisanymi w punkcie 1 rozdziału IV. Na początku zdefiniowane są wartości domyślne, następnie każde zdanie wprowadza kolejne ograniczenia do konkretnych zmiennych. Predykat nth1 służy do wyciągania wartości z podanego indeksu lub zapisywania wartości do elementu o podanym indeksie.

# Użytkowanie i testowanie programu

Uruchomienie programu napisanego w Javie przedstawi taki widok:



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 uruchomienie 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 lokację 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.

# 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,  /\* 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), | /\*\*\*\*\*\*\*\*\*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,  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  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

1. Oryginalna treść twórców łamigłówki znajduje się na stronie<http://www.logic-puzzles.org/pdf/S562JK.pdf> odczyt z dnia 5 czerwca 2017, godzina 11.06 [↑](#footnote-ref-1)
2. Rozwiązanie twórców łamigłówki pobrane z strony<http://www.printable-puzzles.com/printable-logic-puzzles.php> odczyt z dnia 5 czerwca 2017, godzina 11.06 [↑](#footnote-ref-2)