THE DOMINO GAME AND THE EULER CYCLE

[1]SILVESTRE ASCENCIÓN GARCÍA SÁNCHEZ, [2]CARLOS AQUINO RUIZ,
[3]CELEDONIO ENRIQUE AGUILAR MEZA

[1] Prof., Escuela Superior de Ingeniería Mecánica y Eléctrica, del IPN

[2]Prof., Escuela Superior de Ingeniería Mecánica y Eléctrica, del IPN

[3] Prof., Escuela Superior de Ingeniería Mecánica y Eléctrica, del IPN

E-mail: [1]silvestregarcia@hotmail.com, [2]caquino@ipn.mx, [3]cele_ag@ipn.mx

ABSTRACT
Leonhard Euler's work in 1736 on the problem of the bridges of Königsberg is considered the first result of graph theory. By using the domino game it could be illustrated how using all the tiles to create a game that begins and ends with the same piece, as to form an Euler cycle. It begins by defining the Euler cycle, then the domino game is represented as a graph and shown under which terms a domino game meets the Euler condition. Then, using the logic programming language Prolog a method is constructed that finds all domino games forming an Euler cycle only if they meet the Euler condition.

Keywords: *graph, path, cycle, degree of a vertex*

1. INTRODUCTION

The origin of the Eulerian cycle theory was posed and solved by the same Leonhard Euler in 1736. Such problem is called Seven Bridges of Königsberg City (East Prussia in the eighteenth century and now Kaliningrad, Russian province) giving rise to the theory of graphs.

The problem is stated as follows: two islands of the Pregel River in Königsberg unite with each other and the mainland by seven bridges, in that order of ideas the question was posed: Is it possible to take a walk starting from any of the four parts of the mainland, crossing each bridge only once and returning to the starting point? (See figure 1)

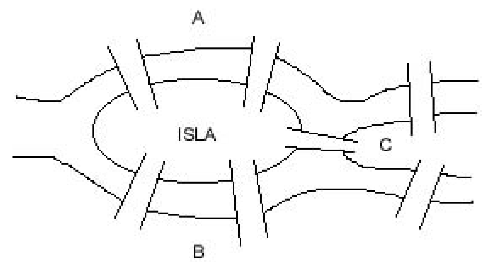

Figure 1. The Seven Bridges of Königsberg

Euler proved that it was not possible since the number of lines affecting each point are not even or pairs (necessary condition to enter and exit of each point, and to return to the starting point, in different ways at all times).

During last three decades, the assessment of potential of the sustainable eco-friendly alternative sources and refinement in technology has taken place to a stage so that economical and reliable power can be produced. Different renewable sources are available at different geographical locations close to loads, therefore, the latest trend is to have distributed or dispersed power system. Examples of such systems are wind-diesel, wind-diesel-micro-hydro-system with or without multiplicity of generation to meet the load demand. These systems are known as hybrid power systems. To have automatic reactive load voltage control SVC device have been considered. The multi-layer feed-forward ANN toolbox of MATLAB 6.5 with the error back-propagation training method is employed.

2. METHODOLOGY

Euler Cycle

An Euler cycle is one that passes through all the edges E of an undirected graph G = (V, E) only once, where V is the set of vertices and E the set of edges. Euler showed that the existence of such a cycle, the graph must be connected (without isolated vertices) and with all its grade vertices (number of incident edges at vertices) pair.

The Domino Game

The game consists of 28 domino tiles, each divided into two equal parts and with the different combinations of the integers from 0 to 6 plus the tiles that repeat the number (see Figure 3).

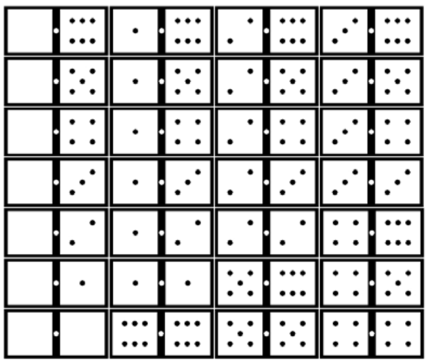

Figure 3. The 28 domino game pieces or tiles

So, if you could create a domino game with the integers from 0 to n, the number of available tokens or tiles would be given by the combinations of n +1 numbers taken two by two, plus the tiles n +1 that repeat the number; this is:

$$\#\text{ de fichas} = C(n+1,2) + n + 1 = \frac{(n+1)!}{2!\,(n-1)!} + n + 1 = \frac{(n+1)(n+2)}{2} \qquad (1)$$

The domino game can be modeled using a non-directed graph, in which the set of vertices V is given by the integers from 0 to n and where the tiles represent the set of edges. In Figure 4 shows the graph for n = 3. The edges that start and end at the same vertex (also called links represent the tiles that repeat number.

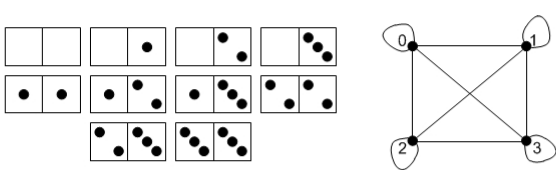

Figure 4. Domino game with n = 3 (left) and its representation by graph (right).

Given an integer n, the degree of each vertex is given by n +2, since each vertex has n edges and tie or loop each edge adds one degree while loop adds two. So n must be an even number in order to exist an Euler Cycle. In the jargon of domino an Euler cycle represents a game that is closed using all the tiles. Figure 5 shows an Euler cycle for n = 2.

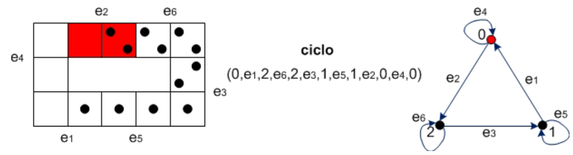

Fig. 5. Euler cycle for n = 2. On the left is shown the arrangement of tabs, center the sequence of vertices and edges and on the right the corresponding cycle in the graph.

The process path

Then it was designed a procedure called path in the logic programming language Prolog. This procedure is a modified version of the procedure with the same name found in Bratko, I. Let G be a graph and A, Z two vertices in G. The relationship was designed:

Path (A, Z, G, P, L) (2)

Where path is a path with no repeated edges from A to Z of length L (number of edges). The method used in path is explained as follows:

P is a path with no repeated edges from A to Z of length L in G:

(1) If A = Z, then P = [A] with L = 0 or,

(2) Find a path with no repeated edges P1 of Y to Z of length L1 and a path from A to Y that has no edges of P1.

The path procedure requires the definition of other path1 procedure call (also represents a modified version of the procedure that is in path1 Bratko):

path1 (A, P1, L1, G, P, L) (3)

In relation path1, A is a vertex, G is a graph, P1 is a path in G of length L1, and P is a path in G does not repeat edges from A to the beginning of P1 and continues along length P1 to Z L. The relationship between the two procedures is shown in Figure 6.

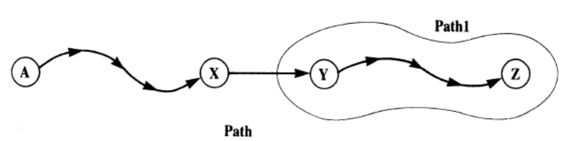

Figure 6. The relationship path 1: path is a path from A to Z, the last part of path overlaps path 1.

The method used in path1 is described below:
P1 is a path that does not repeat Y Z edges in G of length L1:
(1) If Y = A, then P = P1 and L = L1 otherwise,
(2) If X is adjacent to Y, G1 is the graph that results from deleting the edge {X, Y} of G then P must meet the relationship path1 (A, [X | P1], L2, G1, S, L) where L2 = L1 + 1.

The Euler Procedure
The method finds all paths path between two points A, C on G that do not repeat edges, even those in which A = Z (cycles). To find an Euler cycle just find paths that begin and end in the same node, and whose length equals the number of edges of the graph. The relationship is then defined as:
Euler (A, G, P)
The method used in Euler is stated as follows:
P is an Euler cycle in G that begins and ends in A if:

(1) P is a path of length L in G which begins and ends in A and L equals the number of edges in G.

Other procedures used and adapted for the case of non-simple graphs (with loops) are listed below.
adjacent (X, Y, W, G) (5)
X is adjacent to Y by G = (V, E):
(1) If the directed edge (X, Y) is in E with X ≠ Y or,
(2) If the directed edge (Y, X) is in E.

Erase (X,Y,W,G,G1) (6)
G1 = (V, E1) is to delete the undirected edge {X, Y} of G = (V, E):
(1) If E1 is deleting the directed edge (X, Y) with X ≠ Y of E or
(2) If E1 is deleting the directed edge (Y, X) of E.

The complete program is shown below:

```
length([],0).

length([_|T],L) :-
length(T,L1),
L = L1 + 1.
member(X,[X|_]).
member(X,[_|T]) :-
member(X,T).
del(X,[X|T],T).

del(X,[Y|T1],[Y|T2]) :-
del(X,T1,T2).
adjacent(X, Y, g(_,E)) :-
member(e(X,Y),E),
X<>Y; % Cláusula 1
member(e(Y,X),E). % Cláusula 2
erase(X,Y,g(V,E),g(V,E1)) :-
del(e(X,Y),E,E1),
X<>Y; % Cláusula 1
del(e(Y,X),E,E1). % Cláusula 2
path1(A,[A|T1],L,_,[A|T1],L). % Cláusula 1
path1(A,[Y|T1],L1,G,P,L) :- % Cláusula 2
adjacent(X,Y,G),
L2=L1+1,
erase(X,Y,G,G1),
path1(A,[X,Y|T1],L2,G1,P,L).
path(A,Z,G,P,L) :-
path1(A,[Z],0,G,P,L).
euler(A, g(V,E),P) :-
path(A, A, g(V,E),P,L),
length(E,L).
```

Note that when the methods adjacent and seek erase or delete a loop respectively (that is edges of the form (X, X)) both come only in clause 2, thereby avoiding double counting a loop. Other procedures that were used in the program are member and length, which count the number of elements in a set and indicate whether or not an element belongs to a set.

3. RESULTS

Then as shown in Prolog in its procedural form, encounters the Euler cycles beginning at the vertex 0 of the graph in Fig. 5 and then compares with the execution of the program. The order set of directed edges that was used was E = [e1, e2, e3, e4, e5, e6] = [(0, 0), (1, 1), (2, 2), (0, 1), (0.2), (1.2)]

In Fig. 7 it shows the procedural Euler method, note that the first time you enter the clause 2 of the procedure path1 variable Y = 0, while the adjacent method first searches directed edges of the form (X, 0) and finds none as clause 1 has the condition X ≠ 0, then searches directed edges of the form (0, X) and in lexicographic order e1, e4 and e5. After the procedure path1 it continues until it finds closed edges that will include all the edges. The order in which they are found is listed from top to bottom in the search tree.

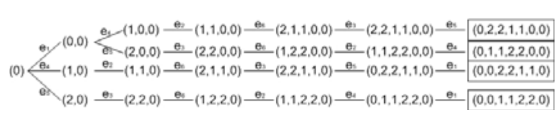

Figure 7. Procedural form the Euler method, the order of the cycles which begin and end at 0 shown from top to bottom.

The result after the execution of the program is shown in Figure 8. It is noted that it corresponds to what was developed in Fig. 7.

```
[Inactive C:\Users\SILVER~1\AppData\Local\Temp\goal$000.exe]
P=[0,2,2,1,1,0,0]
P=[0,1,1,2,2,0,0]
P=[0,0,2,2,1,1,0]
P=[0,0,1,1,2,2,0]
4 Solutions
```

Figure 8. Results of the Euler query execution (a, g ([0, 1, 2], [e (0, 0), and (1, 1) and (2, 2), e (0, 1), e (0, 2), and (1, 2)]), P). It shows the different results of P in order.

4. CONCLUSIONS

The Domino Game can be represented by a graph, and the items that are closed after using all tiles represent an Euler cycle. In this paper we show that it is a necessary condition in order to have an Euler Cycle that the biggest number that appears on the tiles must be even.

Then, by using Logic Programming Language Prolog constructs a method is constructed that allows to find all games of domino that may form an Euler Cycle for any set of tiles that comply with the Euler condition. Finally, we show how the method (procedural form) obtains the Euler Cycles in particular for the case n = 2 and it is verified by implementing the program in Prolog.

REFRENCES:

Deransart, P., AbdelAli, E. & Laurent C. (1991). Prolog: *The standard reference manual.* Springer-Verlag Berlin Heildelberg 1996.

Iranzo, P.J. & María, A.F. (2007). *Programación Lógica Teoría y Práctica*. Pearson Educación S.A.

[Johnsonbaugh, R. (2005). Matemáticas Discretas Sexta Edición. *Trayectorias y ciclos* (pp. 329-336). Pearson Educación de México, S.A. de C.V.

Armenta, R.A. (2010). Matemáticas Discretas *Permutaciones y combinaciones* (pp. 306-310). Alfaomega Grupo Editor, S.A. de C.V., México.

Bratko, I. (1986). Prolog programming for artificial intelligence. *Graphs* (pp. 224-232) Addison – Wesley Publishing Company

AUTHOR PROFILES:

Ing. Aquino Carlos Ruiz was born in Mexico on April 2, 1973 is in Communications and Electronics Engineering from the National Polytechnic Institute IPN, a research professor and areas of development are applied computing, data networking and security.

Ing. Celedonio Enrique Aguilar Meza was born in Mexico on march 3, 1959 and holds a degree in communications and electronics from the National Polytechnic Institute IPN, is currently research professor and development areas are the processing and handling, coding and compression, safety and efficient transmission.

Master´s Degree Silvestre Ascención García Sánchez. Born in Mexico DF on Jul 11, 1975, holds a degree in Communications and Electronics from the National Polytechnic Institute in and earned his Master of Science in Engineering with specialization in Microelectronics Digital Signal Processing by the National Polytechnic Institute in 2005.