INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2013) Vol. 2 (2) 100–112

GAME-TREE STRUCTURES WITH THE COMPLEX COMPLEXITY LEVEL AS A TOOL IN KNOWLEDGE ENGINEERING

ADAM DEPTUŁA

Department of Knowledge Engineering, Opole University of Technology

The paper concerns application of the dependence graphs and game- tree structures with the complex complexity level as a tool in knowledge engineering. Different graph solutions mean connections between input and output data as well as decision variables of the analyzed system. The graph distribution from any vertex in the first stage leads to a tree structure with cycles, and next to a general tree game structure. Game-tree structures from each node have different shape and properties. Algorithmic way to create graphical structures out of a mathematical model describes the optimization method of systematic exploration. Tree structure, with the lowest values of complexity level is the simplest structure.

Keywords: knowledge engineering, dependence graphs, game- tree structures, decision- making, artificial intelligence

1. Introduction

The subject of knowledge engineering and expert systems is considered to be one of the most important areas and directions of contemporary IT development. The main task of the widely understood knowledge engineering is to gather and formalize experts' knowledge into the form of rules used by expert systems and decision support systems. Expert systems, as IT systems, are aimed mainly at solving specialist problems requiring professional expert's opinion based on knowledge so they are connected with gaining and processing it. The main idea of an expert system is the expert's knowledge transfer to the system equipped with the knowledge base. The knowledge representation is a basic term for different kinds of decision processes and drawing conclusions and it is one of major problems which has not been completely solved yet. In the knowledge engineering, knowledge representation is treated as a way of presenting, in the formal language, the whole knowledge range required for intelligent system maintenance [1, 2].

However, their main task is to support the decision making process on the basis of the possessed knowledge. Decision making processes place an emphasis on modern information processing techniques in order to improve the decision making process. Both in the knowledge creation for the enterprise needs and in specifying construction guidelines for the designer, it is necessary to use an appropriate device describing the decision making process. First decisions are made in the planning process and then, they are reinforced are corrected. There is a wide range of tests concerning working out of a methodology supporting decision making processes and controlling in the systems with different complexity scale with the participation of the artificial intelligence [9, 10]. Among tools supporting decision making processes, it is possible to enumerate: decision tables, dendrites, decision logical trees, tree classifiers as well as logical comparisons and graphs.

This article presents dependency graphs concerning signal transmission and game structures as tools supporting decision making processes in the knowledge engineering. The mathematical model of the analysed system (element) is a group of functions which are connected with one another by means of different variables and in this way, they describe connections between figures in the system. As a result of saving and making a decomposition of a graph presenting these functions dependencies, we obtain decomposition groups which structurally describe properties of subsequent sub-systems of a given system. The obtained game structures differ in shape and properties. In order to choose a game structure with the lowest complexity, it is necessary to calculate a complex complexity coefficient for all structures that are obtained.

2. The dependence graph for tree game structures

Directed graph defined as an ordered pair of sets. The first of these contains vertices and the second consists of the edges of the graph. Figure 1 shows an exemplary dependence digraph of the signal flow.

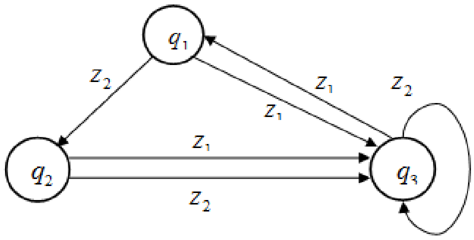

Figure 1. Dependence digraph of the signal flow
Source: Own study based on [3, 11]

Depending directed graph of Figure 1 consists of a set of vertices Q: $Q = \{q_1, q_2, q_3\}$ and edges from the set: $Z = \{z_1, z_2\}$. The graphs distribution from any vertex in the first stage leads to a tree structure with cycles, and next to a general tree game structure. Each structure has a proper analytic notation G_i^+ and G_i^{++}. The algorithm describes the distribution of dependency graph is presented in [5, 11]. Decomposition of the dependence graph of the signal flow from the defined initial vertex G_i^+ in the first stage leads to the expression:

$$G_1^+ = ({}^0 q_1({}^1 z_1q_3({}^2 z_1q_1, z_2q_3)^2, z_2q_2({}^2 z_1q_3, z_2q_3)^2)^1)^0$$

and then to express G_i^{++}:

$$G_1^{++} = ({}^0 q_1({}^1 z_1q_3({}^2 z_1q_1^1, z_2q_3^1)^2, z_2q_2({}^2 z_1q_3({}^3 z_1q_1^2, z_2q_3^1)^3, z_2q_3({}^3 z_1q_1^2, z_2q_3^1)^3)^2)^1)^0$$

where: z_iq_r - the *i*-th edge z_i extending out from the *r*-th vertex q_r.

Considering tree structures we must determine element subordination in the system. Each structure has its proper analytic notation (G_i^+ and G_i^{++}, where *i* is the vertex from which the graph decomposition was started). Each element q_r has always subordinated elements q_i. Both q_r and q_i elements can occur many times in the expression G_i^{++} in the brackets $({}^k ...)^k$ with different values of *k*, i.e. at various stages of the tree structure [5, 6, 11].

Figure 2 shows the tree structure with cycles, while in Figure 3 a tree game structure from the initial vertex q_1.

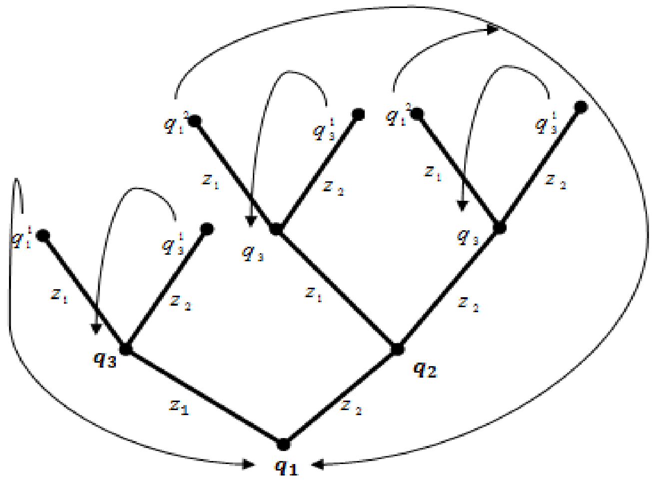

Figure 2. Tree structure with cycles and the initial vertex q_1
Source: Own study based on [5, 11]

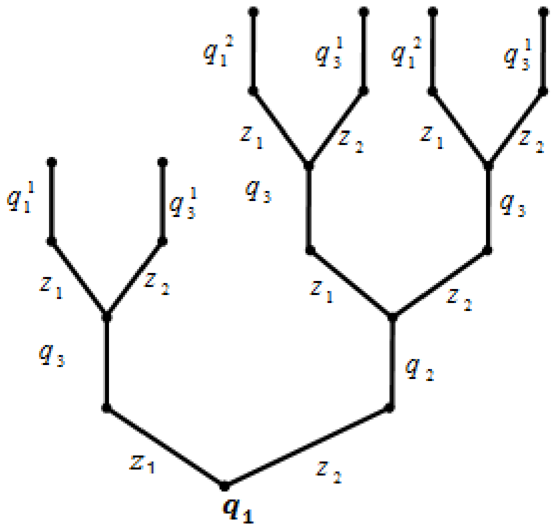

Figure 3. Tree game structure from the initial vertex q_1
Source: Own study based on [5, 11]

3. Complex complexity coefficient for game-tree structures

The level of structure's complexity of is determined by the complex complexity coefficient $L(G_i^{++})$ [3, 4, 8]:

$$L(G_i^{++}) = \sum_{w \in W(L)} \frac{d(w_i)}{h(w_i)+1} \tag{1}$$

where:

$L(G_i^{++})$ - complexity coefficient of structure G_i^{++},

w_i - *i-th* node,

$d(w_i) = \deg(w_i)$ - degree of *i-th* node branching (amount of node branchings),

$h(w_i)$ - distance from the *i-th* node root,

$W(L)$ - set of all nodes.

In order to state the importance of game structures obtained as a result of a dependency graph decomposition from each graph vertex, a complex complexity coefficient is introduced [8] $L^K(G_i^{++})$:

$$L^K(G_i^{++}) = \sum_{w \in W(L)} \frac{d(w_i)}{h(w_i)+1} + \frac{L}{\sum_{l \in L} \frac{1}{h_{l_i}}} \tag{2}$$

where:

$L^K(G_i^{++})$- comoplex complexity coefficient of structure G_i^{++},

L- number of leaves for the *i-th* node branching ($\deg(w_i) \geq 2$),

h_{l_i} - amount (complexity) of the *i-th* leaf.

Then we obtain a better quality evaluation of a given structure by means of taking into consideration decisions which are not branched. The current structure complexity coefficient defined the structure in the macro scale with taking into consideration only the amount of nodes in the structure and the degree of their branching. The complex complexity coefficient of a structure takes into consideration the complexity degree of all leaves going out of each branching node [8]. The Figure 4 shows an example game-tree structure with different coefficients L and L^K.

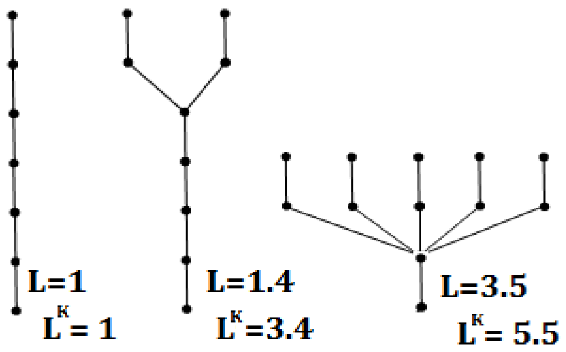

Figure 4. Tree-game structures with different complexity coefficients L and L^K
Source: own study based on [3, 8]

4. Game graphs with different vertex connections as tool in knowledge engineering

An analysis of a mathematical model of a given system can be described by means of game graphs with different exemplary vertex connections $G(1)$, $G(2)$ presented in the Figure 5.

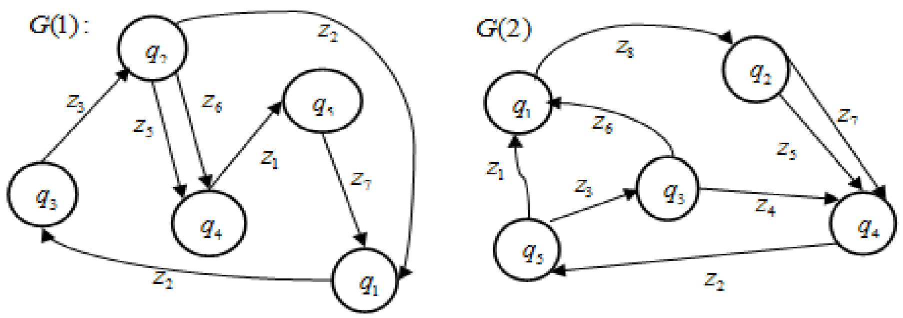

Figure 5. Dependence digraphs *G(1)*, *G(2)* for different connections vertex

For a graph *G(1)* as shown in Figure there is a set *D(1)* tree-game structures:

$$D(1) = \left\{G(1)_{q_1}^{++}, G(1)_{q_2}^{++}, G(1)_{q_3}^{++}, G(1)_{q_4}^{++}, G(1)_{q_5}^{++}\right\}$$

Tree-game structures are shown in figures 6-7.

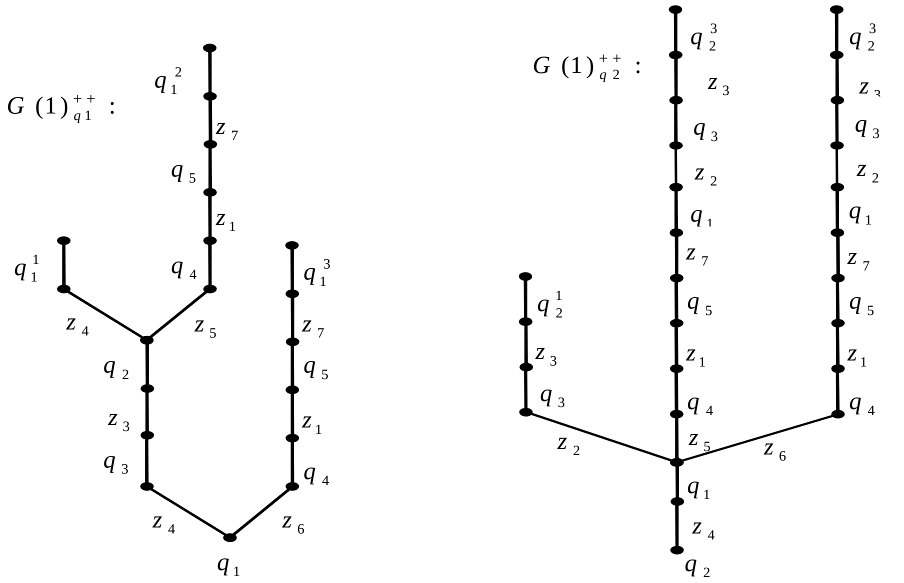

Figure 6. Tree game structures $G(1)^{++}_{q1}$ i $G(1)^{++}_{q2}$

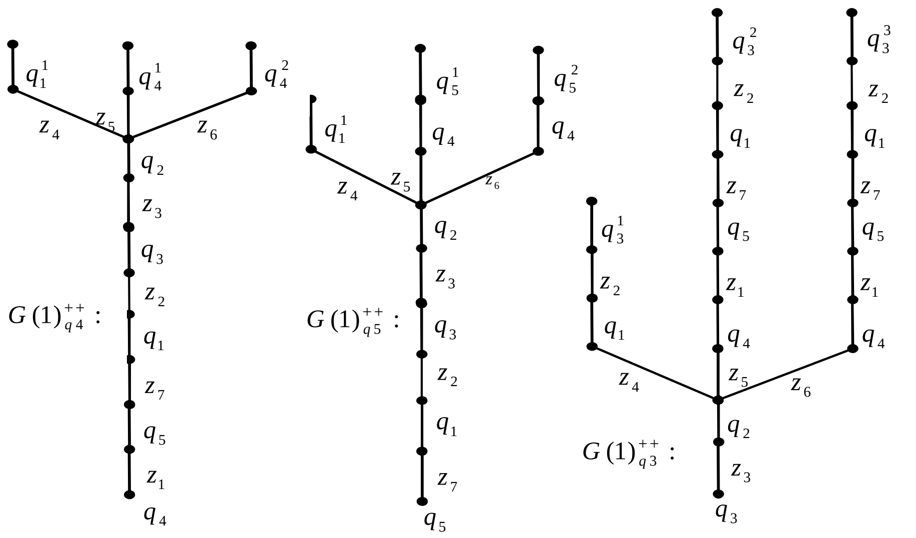

Figure 7. Tree game structures $G(1)^{++}_{q3}$, $G(1)^{++}_{q4}$ i $G(1)^{++}_{q5}$

Analytic expressions $G(1)_i^{+}$ and $G(1)_i^{++}$ for the structures of the figures 6-7:

$$G(1)_{q1}^{+} = ({}^{0} q_1 ({}^{1} z_2 q_3 ({}^{2} z_3 q_2 ({}^{3} z_4 q_1, z_5 q_4 ({}^{4} z_1 q_5 ({}^{5} z_7 q_1)^{5})^{4}, z_6 q_4)^{3})^{2})^{1})^{0},$$

$$G(1)_{q1}^{++} = ({}^{0} q_1 ({}^{1} z_2 q_3 ({}^{2} z_3 q_2 ({}^{3} z_4 q_1^{1}, z_5 q_4 ({}^{4} z_1 q_5 ({}^{5} z_7 q_1^{2})^{5})^{4}, z_6 q_4 ({}^{4} z_1 q_5 ({}^{5} z_7 q_1^{3})^{5})^{4})^{3})^{2})^{1})^{0},$$

$$G(1)_{q2}^{+} = ({}^{0} q_2 ({}^{1} z_4 q_1 ({}^{2} z_2 q_3 ({}^{3} z_3 q_2)^{3})^{2}, z_5 q_4 ({}^{2} z_1 q_5 ({}^{3} z_7 q_1)^{3})^{2}, z_6 q_4)^{1})^{0},$$

$$G(1)_{q2}^{++} = ({}^{0} q_2 ({}^{1} z_4 q_1 ({}^{2} z_2 q_3 ({}^{3} z_3 q_2^{1})^{3})^{2}, z_5 q_4 ({}^{2} z_1 q_5 ({}^{3} z_7 q_1 ({}^{4} z_2 q_3$$
$$({}^{5} z_3 q_2^{2})^{5})^{4})^{3})^{2}, z_6 q_4 ({}^{2} z_1 q_5 ({}^{3} z_7 q_1 ({}^{4} z_2 q_3 ({}^{5} z_3 q_2^{3})^{5})^{4})^{3})^{2})^{1})^{0},$$

$$G(1)_{q3}^{++} = ({}^{0} q_3 ({}^{1} z_3 q_2 ({}^{2} z_4 q_1 ({}^{3} z_2 q_3)^{3}, z_5 q_4 ({}^{3} z_1 q_5 ({}^{4} z_7 q_1)^{4})^{3}, z_6 q_4)^{2})^{1})^{0},$$

$$G(1)_{q3}^{++} = ({}^{0} q_3 ({}^{1} z_3 q_2 ({}^{2} z_4 q_1 ({}^{3} z_2 q_3^{1})^{3}, z_5 q_4 ({}^{3} z_1 q_5 ({}^{4} z_7 q_1 ({}^{5} z_2 q_3^{2})^{5})^{4})^{3}, z_6 q_4 ({}^{3} z_1 q_5 ({}^{4} z_7 q_1$$
$$({}^{5} z_2 q_3^{3})^{5})^{4})^{3})^{2})^{1})^{0},$$

$$G(1)_{q4}^{++} = ({}^{0} q_4 ({}^{1} z_1 q_5 ({}^{2} z_7 q_1 ({}^{3} z_2 q_3 ({}^{4} z_3 q_2 ({}^{5} z_4 q_1, z_5 q_4, z_6 q_4)^{5})^{4})^{3})^{2})^{1})^{0},$$

$$G(1)_{q4}^{++} = ({}^{0} q_4 ({}^{1} z_1 q_5 ({}^{2} z_7 q_1 ({}^{3} z_2 q_3 ({}^{4} z_3 q_2 ({}^{5} z_4 q_1^{1}, z_5 q_4^{1}, z_6 q_4^{2})^{5})^{4})^{3})^{2})^{1})^{0},$$

$$G(1)_{q5}^{++} = ({}^{0} q_5 ({}^{1} z_7 q_1 ({}^{2} z_2 q_3 ({}^{3} z_3 q_2 ({}^{4} z_4 q_1^{1}, z_5 q_4 ({}^{5} z_1 q_5^{1})^{5}, z_6 q_4 ({}^{5} z_1 q_5^{2})^{5})^{4})^{3})^{2})^{1})^{0}.$$

For each of the structures $G(1)_{q1}^{++}$, $G(1)_{q2}^{++}$, $G(1)_{q3}^{++}$, $G(1)_{q4}^{++}$ i $G(1)_{q5}^{++}$ was calculated complex complexity coefficients by the formula (2):

$$L^{K}(G(1)_{q1}^{++}) = \left(\sum_{w \in W(L)} \frac{d(w_i)}{h(w_i)+1} + \frac{L}{\sum_{l \in L} \frac{1}{h_{l_i}}} \right) = \left(\frac{1}{0+1} + \frac{2}{\frac{1}{4}+\frac{1}{6}} \right) + \left(\frac{2}{4+1} + \frac{2}{\frac{1}{2}+\frac{1}{6}} \right) = 10,27$$

$$L^{K}(G(1)_{q2}^{++}) = 8,66, \quad L^{K}(G(1)_{q3}^{++}) = 8,66, \quad L^{K}(G(1)_{q4}^{++}) = 3,33, \quad L^{K}(G(1)_{q5}^{++}) = 4,01$$

For a graph *G(2)* as shown in Figure there is a set *D(2)* tree-game structures:

$$D(2) = \left\{ G(2)_{q_1}^{++}, G(2)_{q_2}^{++}, G(2)_{q_3}^{++}, G(2)_{q_4}^{++}, G(2)_{q_5}^{++} \right\}$$

Tree-game structures are shown in Figures 8-9.

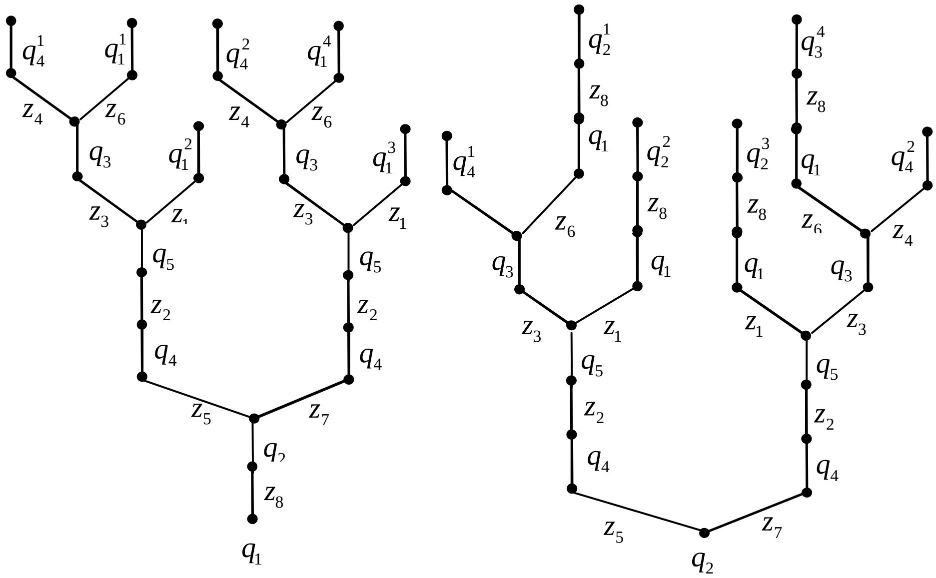

Figure 8. Tree game structures $G(2)_{q1}^{++}$ i $G(2)_{q2}^{++}$

Knowledge engineering is an interdisciplinary branch of solving many different problems as far as exact science and sociology are concerned. Research issues include among others: artificial intelligence, expert systems and in particular, decision support systems. In any of the analysed systems, in the newly recommended work conditions, unknown dependency functions depending on time will change their behaviours. This is why, one rule is in force: graph vertices (the Q set) define functions depending on time, whereas edges (the Z set) define decisions. Game tree-structures from each graph vertex describe the decision making process and the space of the possible to get states of the analysed system.

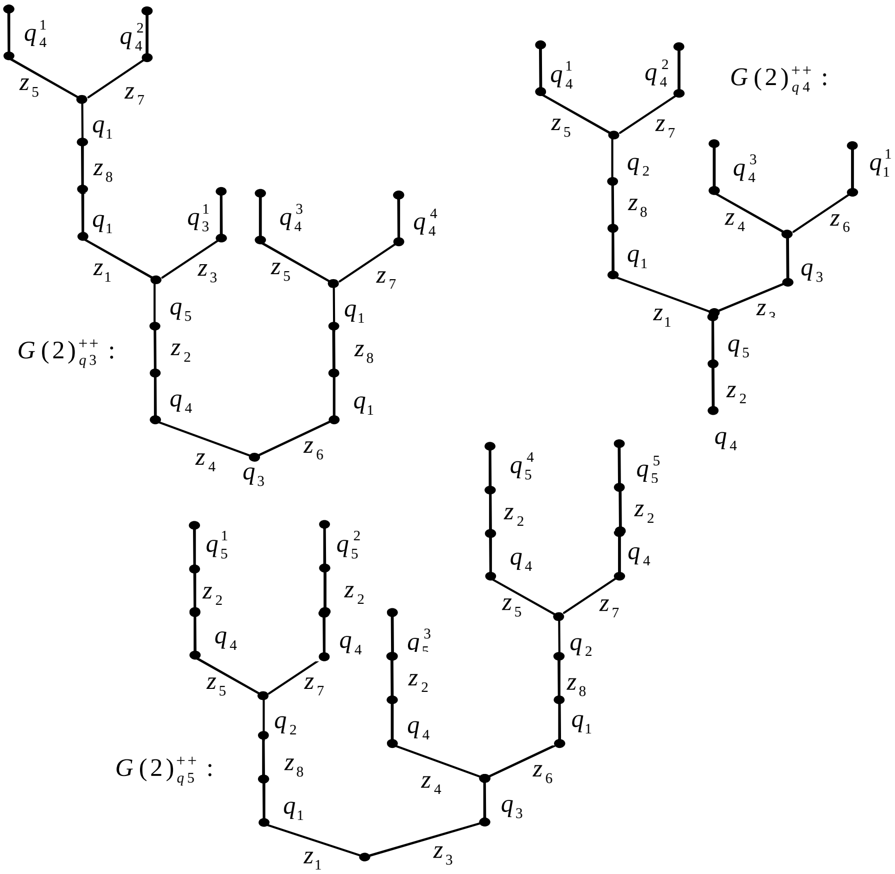

Figure 9. Tree game structures $G(2)^{++}_{q3}$, $G(2)^{++}_{q4}$ i $G(2)^{++}_{q5}$

5. Dependency graph in the description of a mathematical model of a machine set

When we use a dependency graph (Figure 11) in the description of a mathematical model of a machine set (Figure 10), which is described by means of a set of algebraic, differential and integral equations, then the graph vertex is formed by evaluated functions depending on time, whereas decisions are formed by construction and / or exploitation parameters as well as analytic and algebraic transformations [5, 6, 7].

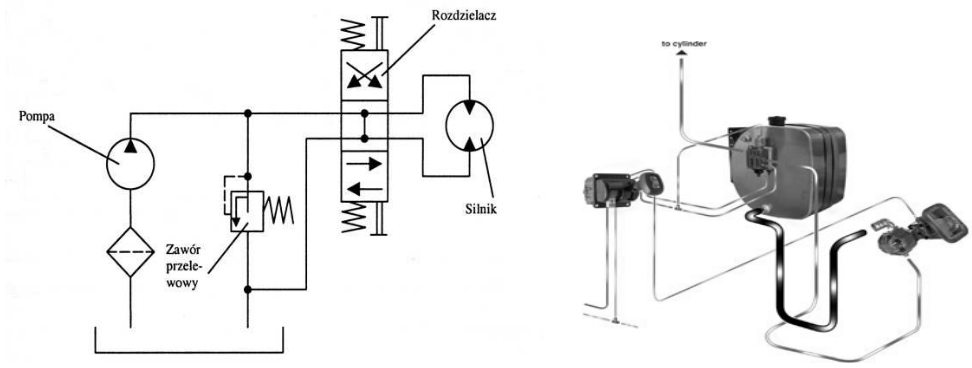

Figure 10. Hydraulic scheme of the system
Source: own study based on [5, 6, 11]

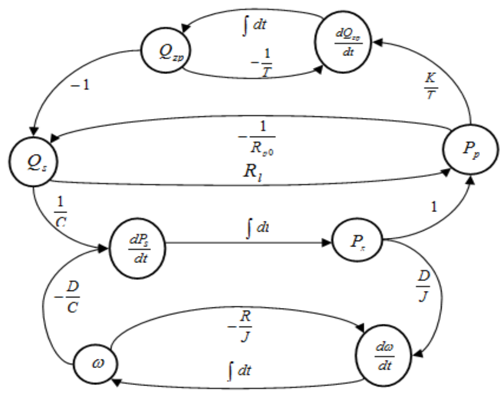

Figure 11. Dependence digraph of the signal flow G(H) for the hydraulic system
Source: own study based on [5, 6, 11]

For a graph for hydraulic system *G(H)* as shown in Figure 11 there is a set *D(H)* tree-game structures:

$$D(H)=\left\{G(H)^{++}_{Qs},G(H)^{++}_{Ps},G(H)^{++}_{\omega},G(H)^{++}_{Pp},G(H)^{++}_{Qzp}\right\}$$

The Figure 12 shows the game-tree structures G^{++}_{ω} and G^{++}_{Qs} from the initial vertex ω and Qs with complex complexity coefficients $L^{K}(G(H)^{++}_{\omega})=17{,}01$ and $L^{K}(G(H)^{++}_{Qs})=18{,}66$.

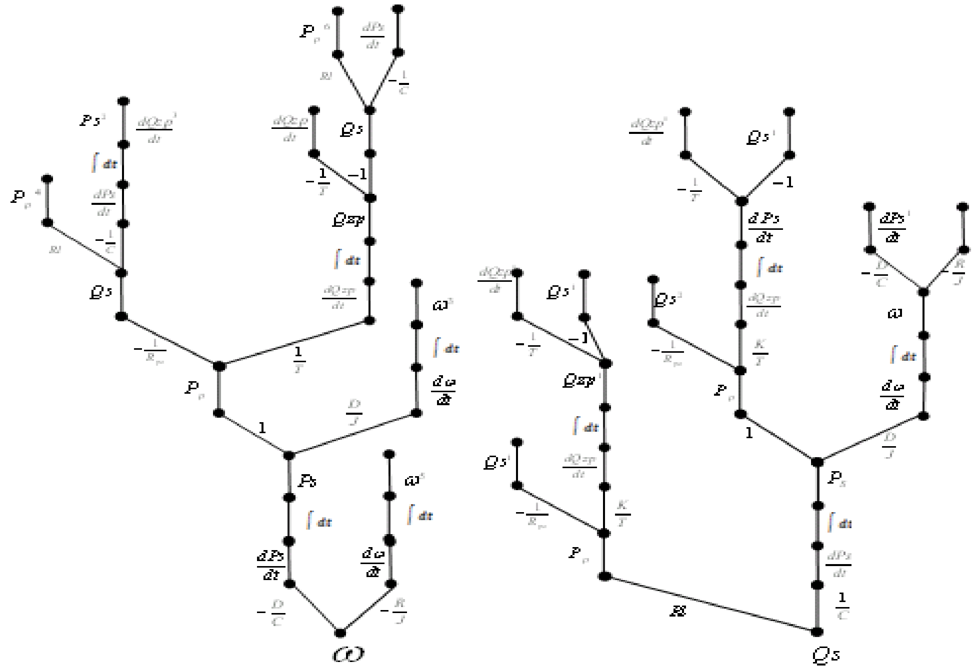

Figure 12. Tree game structures $G(H)^{++}_{\omega}$ i $G(H)^{++}_{Qs}$
Source: own study based on [5, 6, 11]

6. Conclusion

Game graphs make it possible to analyse the so-called “connected” decisions. Results obtained after the first decision have an influence on subsequent decisions. This is why they make it possible to make dynamic models. The complexity coefficient makes it possible to choose a structure of the smallest decision making degree. There are many methods of looking for ideas and problem structuration in the knowledge engineering e.g. a morphological method, an analysis of connected AIDA decision areas, a decision tables method, etc. Game graphs are among these tools. It is worth highlighting that there is a dependency between the complexity coefficient value and the size of the multi-valued logical decision tree.

REFERENCES

[1] *Allard* C. R., Ouwersloot H., Lemmink, J. (2006) *Antecedents Of Effective Decision-Making: A Cognitive Approach*, The IUP Journal of Managerial Economics, IUP Publications, vol. 0(4), pp. 7-28.

[2] *Brunsson* N. (2007) *The Consequences of Decision-making*, New York: Oxford University Press.

[3] Deptuła A. (2011) *Determination of game-tree structures complexity level in discrete optimization of machine systems*, in: Proceedings of International Masaryk Conference for Ph. D. Students and Young Researches, Hradec Kralove, Czech Republic, pp. 2250-2259.

[4] *Deptuła* A. (2013) *Kompleksowy współczynnik rozgrywający parametrycznie z grafu zależności przepływu sygnałów*, XLII Konf. Zast. Mat., Zakopane 2013, Inst. Mat. PAN, Warszawa.

[5] *Deptuła* A., Partyka M.A. (2011) *Application of dependence graphs and game trees for decision decomposition for machine systems*, Journal of Automation, Mobile Robotics & Intelligent Systems, vol. 5, No. 3, pp.17-26

[6] *Deptuła* A., Partyka M.A. (2010) *Application of game graphs in optimization of dynamic system structures*, International Journal of Applied Mechanics and Engineering, vol.15, No.3, pp. 647-656.

[7] *Deptuła* A., Partyka M.A. (2013) *Discrete optimization of a gear pump after tooth undercautting by means of complex multi-valude logic tree*s. XVI Konferencja Innowacje w Zarządzaniu i Inżynierii Produkcji, Zakopane 2013, ;Pol. Towarz. Zarz. Prod. PTZP. (in Polish)

[8] *Deptuła* A. (2012) *Współczynnik złożoności struktury dla minimalizacji wielowartościowych funkcji logicznych*; XLI Konf. Zast. Mat., Zakopane 2012, Inst. Mat. PAN, Warszawa.

[9] Hall J.G. (2012) *Engineering knowledge engineering,* Expert Systems, Volume 29, Issue 5, pp. 427–525.

[10] http://www.decision-making-solutions.com

[11] Kazimierczak J. (1978) *System cybernetyczny*, Wiedza Powszechna, Omega, Warszawa.