International Business Management 9 (5): 998-1006, 2015
ISSN: 1993-5250

Search of Solutions with the Variable Strategy in Static Expert Systems

Maxim P. Denisov and Arnold M. Jurin
Kazan (Volga Region) Federal University, Kremlevskaya Street 18, Kazan, Russia

Abstract: This study describes the developed methods for finding the solutions based on the direct and reverse output by the example of an expert tool system ExPRO. The method of finding solutions with a variable strategy is proposed which is considered to improve the level of automation increase, the creation of a knowledge base and to address potential inconsistencies instrumental system strategies by the methods of an expert solutions. Formally, the opportunities for direct and reverse output in the search for solutions with a variable strategy are described and proven by setting certain parameter values. The conditions of a strategy selection and the methods for finding the solutions to determine the parameters of the search process depending on these conditions are developed. For one of the conditions the statement of possible selection for any of the strategies considered and its proof also presented the advantages and disadvantages of each strategy selection in this case. For all conditions the examples of knowledge bases are brought created in the system ExPRO as a graph of rules related on the basis of input and output variables. The results of solutions obtaining tests are shown with the variable strategy and the comparison of its effectiveness with direct and reverse output by the percentage of solved problems and the time of goal search.

Key words: Expert systems, knowledge base, production systems, the search for solutions, direct conclusion, opposite conclusion, tools

INTRODUCTION

Nowadays there are actively developing methods of Artificial Intelligence (AI) which allow to solve hardly formalized tasks and to increase the level of automation in various spheres of human activity. One of such areas under the AI are the Expert Systems (ES) (Novikov, 2010; Gavrilova and Khoroshevsky, 2001; Popov *et al*., 1996).

The distinctive features of ES are: the presence of a knowledge base (Novikov, 2010; Gavrilova and Khoroshevsky, 2001), the subsystems of solution search (Novikov, 2010; Russell and Norvig, 2006) and the ability to explain a decision (Popov *et al*., 1996). It allows to accumulate knowledge of qualified professionals for ES to consult and solve the problems of a particular subject area. The intelligent decision process depends on the knowledge base and the solution search subsystem. The most popular and easy for experts knowledge representation model is a production model (Novikov, 2010; Gavrilova and Khoroshevsky, 2001). The process of finding solutions for this model consists of determination the manner of rules implementation by generating an algorithm for an ultimate goal achievement.

The search of solutions is performed with iteration under the management of a definite strategy (Novikov, 2010; Popov *et al*., 1996). The strategy determines the choice of an output direction (direct output from the data to the target, the reverse output from the purpose to the source data) and the selection of a rule at each iteration.

As the consequence, the process of DB creation also depends on a strategy used in a system because as the used method of search can not get the right solution for the knowledge base created independently. However, this dependence often leads to the use of fictitious goals, the direct challenge to the rules and other actions requiring a greater algorithmization of a task from the designer of acknowledge base which reduces the level of problem solution automation by using ES. Consequently, there is a need for a method of finding solutions which allows to create a knowledge base as independently as possible and to receive the right solutions for these databases within the maximum number of cases.

DIRECT AND INVERSE OUTPUT IN EXPRO SYSTEM

The studies are conducted using an expert tool system ExPRO (Yurin and Denisov, 2013, 2014a, b), created in the KFU. The working version of the system implemented by the forward and reverse output (Fig. 1 and 2). The knowledge base is necessary to set an output strategy. The strategy is not changed during the process of a solution search.

Corresponding Author: Maxim P. Denisov, Kazan (Volga Region) Federal University, Kremlevskaya Street 18, Kazan, Russia

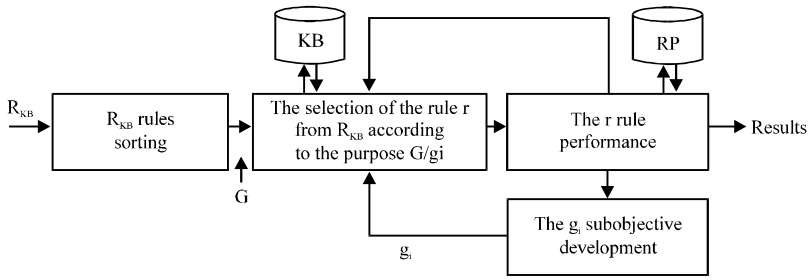


Fig. 1: Reverse output in ExPRO system

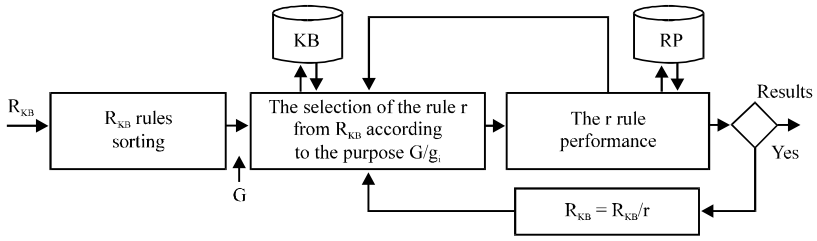


Fig. 2: Direct output in ExPRO system

To consider the search for solutions it is necessary to provide a formal description of some knowledge base elements. KB is knowledge base. KB = {G, V, R_{KB}} is the knowledge base consists of a purpose, variable entities and a set of rules. $r_i \in R_{KB}$ = {pr, C_i, A_i} is each rule consists of a set of conditions and a set of actions and also of a priority. $c_{ij} \in C_i = F_{cij}:I_{cij} \rightarrow$ {true, false} is each condition requires a set of entities, variables for an input and implements the function of their values transformation into a single value of a logical type. $a_{ij} \in A_i = F_{aij}:I_{aij} \rightarrow O_{ij}$ is every action implements a transformation function for a set of inputs to a set of outputs and their assignment to entity-variables. The working memory WM is also used in the process of solutions. The $x \in WM \rightarrow x \in V$ is the part of a working memory containing the essence of variables is considered of the $x \in WM \rightarrow x$. The checking of a definite variable (objectives, sub-objectives) is performed by testing its presence in the working memory.

The presented option of reverse output was developed as the result of selection methods study for the rules at the reverse output in static expert systems (Yurin and Denisov, 2014a) and at the present moment is the most effective among the studied ones. It consists of two main stages: preparation and the search cycle.

During the preparation phase of all the rules of the knowledge base (R_{KB}) are sorted: according to decreasing priority, the decrease of conditions number and the sequence number increase.

At each iteration of a search cycle the only rule r is chosen from the rules R_{KB} which will be considered on this iteration. The rule is chosen according to the target g and the initial search index n in the following way:

$$r = r_i \in R_{KB} : \in U_j O_{ij}, i \geq n; \nexists r_k : g \in U_j O_{kj}, n < k < i$$

This principle is implemented through a syntax check of the target variable presence in the left part of the expression: <object property>|<variable>|<object> = <mathematical expression>. An example of a rule from the knowledge base of image segmentation: IF PixelMark>0, the currentMark = 0, then PixelMark = currentMark. In the provided, the variable objectives are: mark, currentMark. At the same time, the following variables are required to perform this rule: PixelMark, currentMark. If there are several such rules, the first rule is selected due to the fact that the entire set of rules is already ordered by the necessary parameters.

When the rule r is selected for its review its performance takes place in two stages: the test of conditions, the implementation of actions. During these stages if a variable occurs which has no value, it becomes then a sub-goal variable and a search cycle runs for it recursively. When the variable sub-goal is found, the implementation of the considered rule continues. If the condition is not true, then the next opportunity is selected to review a rule or the error is displayed: “the goal is not found”. The subsequent selection is implemented by specifying an initial search index which is stored at the selection of a previous rule and is local for each recursive call.

The direct output also starts with a preparation phase (currently the same sorting is used as in reverse output) and also in the process of a rule selection (conflict resolution) the first suitable rule is selected by the original ordering. The right to choose is performed only once per search.

The right to choose not by a target but by certain variables that occur in the conditions and in the right part of expressions within the actions of a rule and also by the validity of the rule conditions:

$$r = r_i \in R_{KB} : \forall x \in U_j I_{cij} U_l (I_{ail} \setminus U_{m=1}^{t=1} O_{aim}) \rightarrow x \in WM,$$
$$\Lambda_j F_{cij}(I_{cij}) = true, i \geq n; \nexists r_k \in R_{KB} : \forall x \in U_j I_{ckj} U_l$$
$$(I_{akl} \setminus U_{m=1}^{t=1} O_{akm}) \rightarrow x \in WM, \Lambda_j F_{ckj}(I_{ckj}) =$$
$$true, n < k < i$$

The modified example of a rule from the knowledge base according to image segmentation: IF PixelMark>0, the currentMark = 0, THEN the coeff. = 1.001, the mark = 0, he currentMark = the PixelMark*coeff. In this case to define the inputs and outputs the pass will be made under the terms and actions of the rule:

- All the variables in use are the input ones with the condition:
  - I_c = {PixelMark, currentMark}
- First action analysis: I_a = {}, O_a = {coeff.}
- Second action analysis: I_a = {}, O_a = {coeff.,mark}
- Third action analysis:
  - I_a = {Pixel Mark, coeff.}\O_a = {Pixel Mark}, O_a = {coeff., mark}

Thus, this rule may be selected in direct output only when all input variables have the following value ($I_c \cup I_a$ = {pixelMark, coeff., PresentMark}$\subset$WM) at that the term of the rule is true ($\wedge_j F_{cj}(I_{cj})$ = (PixelMark>0$\wedge$ Presentmark = 0) = true).

During the rule performance the conditions are not tested, as the rules with right terms are chosen. After the performance the certainty of a variable-goal is checked. If it is determined the system operation comes to an end if not, the performed rule is excluded from the knowledge base list of rules and the next rule is selected.

The error "the goal is not found" will be available if you can not choose the following rule and the goal is not determined.

SEARCH OF SOLUTIONS WITH VARIABLE STRATEGY

According to the abovementioned studies and the search methods thereof and also on the basis of solution search in the systems G2 Gensym (Popov *et al*., 1996) and CLIPS (Dzharratano and Riley, 2007; Chastikov *et al*., 2003) the search of solutions was proposed (Fig. 3) with the base direct output and the possibility of an inverse run for separate entities-variables.

The variable is complemented with the property back (x$\in$V = {val, back}) which determines whether a chain of reverse output is developed for the variable. This feature during the creation of a knowledge base may be set manually as "yes" or "no". Also, the techniques that will automatically select the value of the property shall be developed. During the creation of a knowledge base the value "auto" may be set.

Also for this conclusion a double priority rule is necessary: pr1; for direct output, pr2; for reverse output. The search is performed iteratively without a preparatory stage. At each iteration, the following steps are performed:

- The creation of a set of rules, each of which may be used in the current iteration. The rules are chosen at which all input variables are selected or determined for the search through a reverse output $R_c = \{r_i \in R_{KB} \setminus R_U | \forall x \in U_j I_{cij} U_l (I_{ail} \setminus U^{l-1}_{m=1} O_{aim}) \rightarrow x \in WM \vee back_x = true\}$

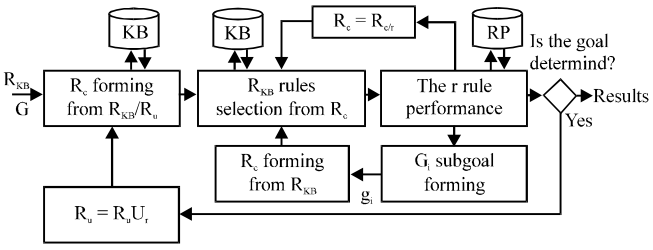


Fig. 3: Search of solutions with a variable strategy

- The selection of a single rule $r \in R_c$. If the set is empty, the output will be with the error "target not found". The selection is made in two stages: sorting R_c, the selection of the first one. Sorting is carried out only if a set was not ordered yet and as follows: priority by descending (for the backward chain pr2, otherwise pr1) by the presence of a target/subtarget in the outputs of rules by descending (present-1, missing-0), according to the number of conditions at descending and by serial number at ascending
- The implementation of the rule is similar to the opposite conclusion: the checking of conditions, the implementation of actions. If the condition is false, then $R_c = R_c \backslash r$ and the transition to the stage 2. When you check the conditions and the implementation of actions for the variables with the missing value a subtask g_i search is operated recursively:
  - The creation of a set of rules for the consideration at the current iteration. The rules are chosen which have only the variables with predefined values at input variables or the property back "yes" or "auto" and the output variables have the following purpose: $R_c = \{r_i \in R_{KB} | g_i \in U_j O_{ij}, \forall x \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}_{m=1} O_{aim}) \rightarrow x \in WM \vee back_x = true\}$
  - The transition to the stage 2 for the subpurpose g_i and after its obtaining the return to the search of a previous target/subtarget
- Check of particular purpose/subpurpose. If it is found, then the search is complete. If it not found, then go to step 1

To determine the features of this method in comparison with the direct and reverse output it is necessary to determine the possibility of their implementation through it.

Then, the implementation of the direct output is described. The property back for all the variables is set in the value of "no": $x \in V back_x$ = false. In this case at the stage 1: $R_c = \{r_i \in R_{KB} | \forall x \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}_{m=1} O_{aim}) \rightarrow x \in WM \vee back_x = true\} = \{r_i \in R_{KB} | \forall x \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}_{m=1} O_{aim}) \rightarrow x \in WM\}$.

At the stage 3, the call for sub-purpose search may not be performed, since all variables are defined. There will also be the rules $r \in R_c$ with the true terms of the rules which is equivalent to the formation of an initial set of rules according to the following rules: $Rc = \{r_i \in R_{KB} | \forall x \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}_{m=1} O_{aim}) \rightarrow x \in WM, \wedge_j F_{cij} (I_{cij} = true)\}$ that corresponds to the choice of a conflict set with the direct output.

The choice of a set of rules from a conflict set would not be equivalent to a direct conclusion due to the sorting by the presence of a target/subtarget in the outputs of rules. This may change the process of finding a solution only on one of the iterations because the implementation of a rules with a target variable will lead to the completion of the search. If necessary, execute the rules which do not define the goal but necessary for the solution of a problem. The use of ordering by priorities is possible.

Thus, the use of all the possibilities for a direct output is allowed but in some cases it is necessary to replace the use sequencing rules by a serial number with the use of priorities.

Then the implementation of the reverse output is described. The property back for all the variables is set in the value of "yes": $x \in V back_x$ = true, the priority for direct output (pr1) is presented the same for all the rules.

In this case, all the rules of the knowledge base will be chosen at the stage 3: $R_c = R_{KB}$. If this set also gets the rules governing the purpose if they are in the knowledge base (if $R_g = \{r_i \in R_{KB} | g \in U_j O_{ij}\} \neq \emptyset$, then $R_g \subset R_c$). If $R_g = \emptyset$, then the purpose won't be obtained as there are no rules, determining a target.

During the step 3, the sorting is performed by the presence of a target in the rule output. The direct output priorities of all of all the rules are the same, so the target rules will be the first ones in the list. If none of these rules are executed, the target won't be found. This is equivalent to choosing an initial set $R_c = \{r_i \in R_{KB} | g \in U_j O_{ij}\}$ that corresponds to the opposite conclusion. However, the sorting of the set will be different from the reverse output you can not use the priorities, only the number of conditions and a serial number. A subsequent search of sub-goals is completely similar to the opposite conclusion, as the step 3.1 provides $R_c = \{r_i \in R_{KB} | g \in U_j O_{ij}\}$ and the step 2 uses the priority pr2 at a reverse chain which does not impose conditions.

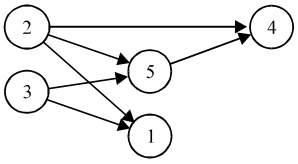
Thus, it is allowed to all the possibilities of reverse output but it is impossible to arrange the rules of a main goal search with the use of priorities but we may use the ordering according to a serial number.

Based on the information stated above, the search for solutions with a variable strategy allows you to implement the direct and reverse conclusion with some conditions and allows you to expand their capabilities.

DIRECT OUTPUT SELECTION CONDITION

The rules of knowledge base may be combined in a semantic network, where the nodes are the rules and the ribs connecting them inputs and outputs (Fig. 4). This model is used in the system Sprut ExPro for the modules of knowledge (Yevgenev, 2009).

For the knowledge base, all the rules which form such a network without cycles the following statement may be



Description of the rules (network nodes)

No	Names	Inputs	Outputs
1	Tracing a rectangle	X, Y, A, B	
2	The input of a figure parameters		A, B, R
3	Figure location		X, Y
4	Tracing a circle	X0, Y0, R	G
5	The circle position	X, A, Y, B	X0, Y0

Fig. 4: Example of semantic network rules for the display task of a compound shape. The goal is G

put forward: if there is a strategy for the choice of rules and the setting of rule priorities is such that a target value may be found using a reverse output, there is such a selection strategy and the and the arrangement of priorities for rules that it may also be found using a direct output. It is necessary to consider that during the direct display a greater number of rules may be performed than in reverse process as direct conclusion allows the execution of the rules which do not have a direct impact on a main target.

The reverse conclusion under these conditions has 4 cases, when a target/sub-target g is not found:

- $\nexists r_i \in R_{KB}: g \in U_j O_{ij}$ there is no rule defining a predetermined target/sub-target
- $\forall r_i \in R_{KB}: g \in U_j O_{ij} \rightarrow \exists c_{ij} \in C_i = F_{cij}(I_{cij}) =$ false for any rule defining a target/sub-target after determination of all unknown variables there is a false condition
- $\forall r_i \in R_{KB}: g \in U_j O_{ij} \rightarrow \exists g_i \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}_{m=1} O_{aim}): g_i$ "goal is not found" for any rule which determines a goal/subgoal there is an input variables that can not be obtained
- When for the part $r_i \in R_{KB}: g \in U_j O_{ij}$ the paragraph 2 is performed and for the part the paragraph 3 is performed

It is necessary to introduce additional marks for direct output:

- $R_n \in R_{KB}$ the solution subgraph if for all $r_i \in R_n$ at least one of the following conditions is performed:
  - $\forall p \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}_{m=1} O_{aim}) \rightarrow \exists k \neq i: p \in U_j O_{kj}$, all inputs of a rules are provided with th outputs for other knowledge base rules (the network of rules is a connected graph)
  - $\exists p \in U_j O_{ij} \rightarrow \exists k \neq i: p \in U_j I_{ckj} U_l (I_{akl} \backslash U^{l-1}_{m=1} O_{akm})$ at least one output of a rule provides the inputs for other regulations of this set
- $R_{pn} \in R_{KB}$ is the complete subgraph of the solution, i.e., the solution subgraph which can not be added by any other rule of a KB to leave a subgraph as a solution subgraph ($\nexists r \in R_{KB}, r \notin R_{pn}$: implemented which given in above bullets)
- $R_{pu} \in R_{KB}$ the source solution subgraph, i.e., the solution subgraph which comes from the initial rules. For all $r \in R_n$ a. is always fulfilled for paragraph 1. and b. may also be fulfilled

For the direct output under these conditions, there are 2 cases when the target/sub-target g is not to be found:

- The $\nexists$ of a complete source subgraph solution $R_{pnu} = \{r_i\}: g \in U_i U_j O_{ij}$ there is no a paragraph of rules, bringing to a purpose from the source data
- The $\exists$ a complete source subgraph of the solution $R_{pnu} = \{r\}_i: g \in U U O_{ij}$ but $\nexists$ of the initial solution subgraph $R_{pu2} = \{r_i\}: g \in U_i U_j O_{ij}$, $\forall_i, j$ $c_{ij} \in C_i = F_{cij}(I_{cij}) =$ ture, the subgraph of rules resulting from the raw data to a target exists but at the exclusion of any such subgraph rules with false conditions, it ceases to lead to a goal

Proof of approval: Let it is not so and for some KB there is a realization of a return output that may find a target. But for this KB there is no realization of a direct output which would allow to find a target.

In this case, at least one of the two cases will be performed when a target/sub-target g can not be found at the direct output. Analysis of the case 1; the $\nexists$ of a complete source subgraph of the solution $R_{pnu} = \{r_i\}: g \in U_i U_j O_{ij}$. In this case two options are possible:

- $\nexists r_i: g \in U_j O_{ij}$ there is no rule that determines (has at an output) a target/sub-target g. Then the case 1 is implemented concerning the impossibility of finding the target g with an opposite conclusion, contradiction

- The $\exists r_i: g \in U_j O_{ij}$ from the condition follows that any such $r_i \notin R_{pnu}$. Therefore, $\forall r_i: g \in U_j O_{ij} \rightarrow \exists p \in U_j I_{cij} U_l (I_{ail} \backslash U^{l-1}{}_{m=1} O_{aim}): \nexists R_{pnu} = \{r_k\}: p \in U_k(U_j O_{kj})$ in any rule defining the goal/subgoal g. There is an input that is not defined by any full source subgraph

For the part $p \nexists r_i: p \in U_j O_{ij}$. Then for p the case 1 is performed concerning the impossibility of finding a target with a reverse output. In its turn, the third case will be performed for g, contradiction.

For those p for which the abovementioned is not performed and which are also not detected by any of a complete graph it is necessary to continue the operation iteratively described in the beginning of this paragraph. In this case the case 3 will be carried out concerning the impossibility of finding a goal by an opposite conclusion and hence the case 3 will also be performed for g, contradiction.

Analysis of the case 2: the $\exists$ is a source subgraph of the solution $R_{pnu} = \{ri\}: g \in U_i U_j O_{ij}$ but $\nexists$ of the complete source subgraph of the solution $R_{pu2} = \{r_i\}: g \in U_i U_j O_{ij}$, $\forall_{i,j} c_{ij} \in C_i = F_{cij}(I_{cij}) = true$.

The first option is possible in this case all the rules directly determining the goal/subgoal have false conditions: $\forall r_i \in R_{pnu}: g \in U_i U_j O_{ij} \rightarrow \exists j F_{cij}(I_{cij}) = false$. This corresponds to the case 2 or 4 (if $\exists r_i \notin R_{pnu}: g \in U_j O_{ij}$) the impossibilities of p objective obtaining with a reverse conclusion, contradiction.

Second option; let's fix an arbitrary source subgraph of the solution $R_{pnu} = \{r_i\}: g \in U_i U_j O_{ij}$. The $\exists R_1 \in R_{pnu} \forall r_i \in R_1 \exists j$: $F_{cij}(I_{cij}) = false$ and also corresponds to a below-described items a and b:

- a: $R_{pnu2} = R_{pnu} \backslash R_1 = \{r_i\}$ set to (pu)$\rightarrow g \notin U_i(U_j O_{ij})$
- b: $\forall r \in R_1$, $R_{pnu2} = \{r_i\}$ based on $R_{pnu} \backslash (R_1 \backslash r) \rightarrow g \notin U_i(U_j O_{ij})$

R_1 is the basis of rules, without which a full source subgraph of a solution will not determine a target/sub-target.

From this condition and marks $R_1 = \{r_i\}$, $R_{pnu2} = \{r_i\}$ (from the item a.) one may obtain the following: $\exists P_1 \in U_i(U_j O_{ij})$: $(\forall p_1 \in P_1 \rightarrow (p \notin U_l(U_j O_{ij}), \exists R_2 = (R_{pnu} \backslash R_{pun2}) \backslash R_1 = \{r_k\}: p_1 \in U_k(U_j I_{ckj} U_m(I_{akm} \backslash U^{m-1}{}_{n=1} O_{akn})), \exists P_2 ... \exists R_n = (R_{pnu} \backslash R_{pun2}) \backslash (R_1 \cup ... \cup R_{n-1}) = \{r_z\}: p_{n-1} \in U_z(U_j I_{czj} U_{azm}(I_{azm} \backslash U^{m-1}{}_{n=1} O_{azn})), p \in U_z(U_j O_{zj})) ...))$ at that $\forall r_i \in R_{pnu}: (g \in U_i(U_j O_{ij}))$, $\forall j F_{cij}(I_{cij}) = true) \rightarrow r_i \in R_n$.

Any $p_0 \in P_1$ false under the case 2 concerning the impossibility of obtaining a sub-task by a reverse conclusion. Then for all $p_0 \in U_{i=\overline{2,n-1}} P_i$ the cases 2, 3 or 4 will be performed.

Then for a desired target/subtarget g will the case 3 or 4 will be performed concerning the impossibility of finding a target by a reverse conclusion, contradiction.

Thus, the implementation of any of the cases concerning the impossibility of finding a target with a direct output which implies the impossibility of finding an objective with a reverse conclusion which contradicts the assumption that proves the assertion.

Under the conditions described the converse is also true, it is proved in a similar way.

Based on the abovementioned facts, the condition of a direct output choice is the creation by rules related to the inputs and outputs in a semantic network, an acyclic graph and the lack of a direct call of rules, since in this case:

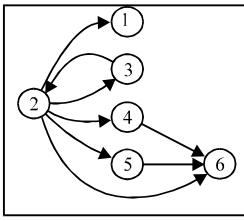
- If a target may be found for a Knowledge Base with the use of a reverse output, then it can be be found also by a direct output
- A direct output allows you to follow the rules which do not relate to an objective, i.e., the solution of a greater number of sub-tasks without entering the dummy variables is possible by the ordering of rules and the setting of priorities (it is possible in an automatic mode)
- A direct output is understood better at the explanation of a decision and at the design of a knowledge base, as the rules are executed in a natural order

In this case, we recommend the use of direct output during the whole search process. In exceptional cases, the designer of a knowledge base may set the search for required variables are manually through the opposite conclusion.

CHOICE CONDITIONS FOR OPPOSITE CONCLUSION

Currently two main reasons for the need of a reverse output selection during the search for solutions are identified empirically:

- The formation of cyclic relations between the rules in the semantic web of rules (Fig. 5). This means that there are two rules, none of which may be executed completely before the other one. This condition may also be obtained for the group of rules by combining the part of rules in the modules and examining the ties between two modules
- In the case of dynamic systems such as G2 Gensym another similar reason is revealed: one of the rules should be carried out but there are no variables necessary for its implementation. This happens with the rules, such as of whenever type. In static systems, this problem may only occur through a direct hand call of a rule



Description of rules (network nodes)			
No	Names	Inputs	Outputs
1	Calculation of P	U, I	P
2	Input of parameters and I calculation	R_{common}	n, R, U, I
3	General R calculation	R, n	R_{common}
4	Additional calculation	I	e
5	U2 calculation	R, I, n	U2
6	Display of results	U2, P, I	Report

Fig. 5: Example of parameter calculation subtask in the circuit of generator power for the tubes with circular connection between the rules (Goal-report)

The first case (of the rule 2.3 on Table 1) in ExPRO system at reverse output is solved with a recursive call of the rule 3 from the rule 2 at the calculation of I. In the second case, similar to the first one is the Rule 2 is called from the Rule 1.

In the system G2 Gensym the rules of the first type could not be performed neither by direct nor by reverse output and the rules of the second one are carried out by setting the flags on variables "U", "I" and the Rule 2 for possible start of a reverse chain.

This example may also be easily modified by breaking the rule 2 on two rules which are suitable for direct search. However, if the actions of the rule 2 will be in the cycle or the knowledge base will have a complex structure, the changes will not be apparent. In the knowledge bases created by students and professionals using the system ExPRO and also within the studied knowledge bases created using CLIPS and G2 Gensym such situations occur, so the need to automate the choice of a strategy in such situations is necessary.

To solve the problems with such rules during the search of solution with a variable strategy in the first case, it is necessary to note with a search flag output using a reverse output "R" and "n" and in the second case "U" and "I". Thus, it is necessary to describe formally the method of such cases determination.

Describing the situations when you need to use a reverse conclusion:

- The formation of cyclic relations between rules or modules: $\exists r_i$, a, b, c:$a \in O_{aip}$, $b \in I_{ail}$, $c \in O_{ail}$, $j<l$ $(\forall r_k \in R_l = \{r_k | b \in O_{akn}\} \rightarrow a \in I_{akm})$, $(\exists r_k \in R_l : m \leq n)$

If such rules are found and $back_b$ = auto, then in the process of finding solutions let's consider that $back_b$ = true. If 3 rules are cyclically dependent, any two of these are considered a single unit and the same verification is used. For a larger number of rules the similar actions are performed.

At this point, the remaining tested cyclic dependencies of the rules that are different from the present description was impossible to solve, since they led to an infinite loop or the error "target is not found".

Table 1: The example rules in the subtask of parameter calculation within the scheme of lamp supply by power generator: 2, 3) Cyclically related rules; 1) If called directly, there is the need of the rule 2 call

No.	Name	IF	THEN
2	Input of parameters and calculation of I	-	VVODBL (Enter the initial data of a task) $I=U/R_{common}$
3	R gen. calculation	-	$R_{o6m} = R/n$
1	P calculation	-	$P = U \times I$

Also for this approach, it is important to expose the parameter only for the variable "b", if the return search is turned on also for "a", "c", then the second rule may be performed twice, instead of one and the use of certain commands such as DELETE may cause an infinite cycle. Also, be aware that the inputs may be defined by other regulations that are not in a cyclic connection with the considered ones while the exhibiting of the search parameters through a reverse output is not necessary.

Direct call of a rule with unknown inputs in advance: For all $r_i \in R_{KB}$:r_i called directly for all $x \in U_j I_{cij} U_l (I_{ail} \setminus U^{l-1}_{m=1} O_{aim})$:$back_x$ = auto during the process of solution search consider $back_x$ = true. If a rule is called directly at the current moment is checked by syntax analysis the existence of the rule name in the arguments of the rule direct call function.

COMPARISON OF RESULTS EFFECTIVENESS

We consider the knowledge base, the semantic web of rules which is shown on Fig. 5. In case of this problem solution using an inverse output one can not perform normally the rule of "Additional payment" without changing the knowledge base and the introduction of fake objectives. The solution to this problem using a direct output will result in the error "The goal is not found" since the rule "of parameters and the calculation of I" can not be executed. When we use the search for solutions with a variable strategy and automatic placement of parameters for variables the problem will be solved and the ability to exercise the rule of "additional payment" will appear.

The prototype for solution search with a variable strategy was tested in the system ExPRO on previously established knowledge bases. At that the search parameter of variables through, a reverse output was set to "auto" for all variables. The results of the new solution search and its comparison with a forward and a reverse output are shown in Fig. 6.

A higher number of knowledge bases, the right solution using reverse output is caused by the fact that the direct output appeared in the system later and most of the knowledge bases were created for a reverse output with the use of its features (a recursive search call). However, the knowledge bases were found that were not fully addressed by a reverse output (the target was found but not all necessary rules were fulfilled) and correctly by using direct output. The difference in the number of solved problems between a direct and a reverse output was 5%. The search of solutions with a variable strategy allowed to solve 95% of issues which is 10% more than a reverse output and 15% more than a direct one. Thus, at this point, there was no evidence of problems that were solved with the use of a direct or a reverse output and were not resolved with the use of solution search with a variable strategy. Therefore, there is a suggestion that the unresolved knowledge bases contain enough knowledge to work with them.

The search with the use of a variable strategy is slightly slower than the search time using the direct output (the difference is <0.3%). A reverse output showed the worst result, the difference with the other methods was 5%. This result was achieved due to multiple performance of the same rules for several tests of knowledge bases (at least 4 tests).

SUMMARY

During the search process an algorithm for solving a task is generated by selecting the order of rules application use in accordance with the used strategy. The use of effective strategies and the automation of their choice expands the range of tasks and reduces the range of tasks under solution and decreases the level of knowledge base algorithmization. Using the search for solutions with a variable strategy allows to reduce the number of cases when it is necessary to modify the knowledge base for a search method or enter fake targets because it realizes two lines of output and the ability to change strategies during the search. The terms of a direct and a reverse output selection in the study may reduce the number of a strategy manual selection and set the parameters of elements that influence the search process. In this regard, we may talk about the automation level increase concerning the process of knowledge base creation.

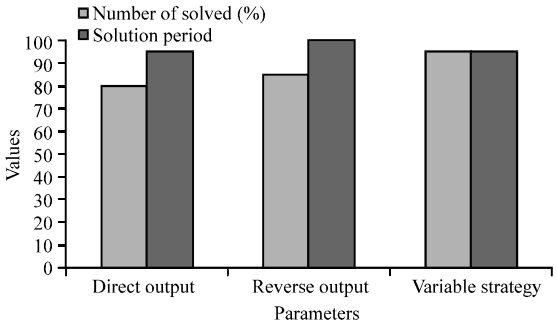


Fig. 6: Comparison of methods for finding solutions. For the experiment, 20 knowledge bases were selected, 36 tests were performed (different versions of input data, changing the course of a decision)

In the framework of this study, the issue of cyclicity at the selection of a reverse output output during the search remains unsolved. The strategy efficiency increase is required, in particular the effective implementation of conflict resolution at the direct output. The research continues in terms of search strategy selection.

CONCLUSION

Using the search for solutions with a variable strategy allows you to increase the percentage of solved knowledge bases. This is due to the possibility of variable strategy conversion to a direct or a reverse output and the possibility of their mutual use. At the same time, the results of testing do not increase the time of solution search.

ACKNOWLEDGEMENT

The research is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

REFERENCES

Chastikov, A.P., T.A. Gavrilova and D.L. Belov, 2003. "Development of expert systems. CLIPS environment". St. Petersburg: "BHV-Petersburg", pp: 608.

Dzharratano J. and H. Riley, 2007. Expert Systems: Design and Programming Principles. 4th Edn. Translation from English. Moscow: "ID Williams" LLC, pp: 1152.

Gavrilova, T.A. and V.F. Khoroshevsky, 2001. Knowledge bases of intelligent systems. SPb.: Peter, pp: 384.

Novikov, F.A., 2010. Artificial Intelligence: knowledge representation and methods of finding solutions. Textbook. SPb.: Publishing House of the Polytechnic University, pp: 240.

Popov, E.V., I.B. Fominyh, E.B. Kissel and M.D. Shapot, 1996. Static and dynamic expert systems. Textbook. M.: Finance and Statistics, pp: 320.

Russell, S. and P. Norvig, 2006. Artificial Intelligence. Modern Approach. 2nd Edn. M.: Williams, pp: 1408.

Yevgenev, G.B., 2009. Intelligent systems of engineering. Textbook. Moscow: Publishing House of MSTU named after N.E. Bauman, pp: 334.

Yurin, A.M. and M.P. Denisov, 2013. Development tools of expert systems. System ExPRO. In: V.A. Reichlin (Eds.). Proceedings of the the Republican Scientific Seminar "Methods of Modeling", No. 5, pp: 19-46.

Yurin, A.M. and M.P. Denisov, 2014a. Methods of rules selection rules at reverse output in static expert systems. Kazan Univ. Textbook. Physics and Mathematics Series, 156 (3): 142-151.

Yurin, A.M. and M.P. Denisov, 2014b. Development of knowledge production base with the use of an expert tool system ExPRO 4. Scientific Notes of the Institute of Social and Humanitarian Knowledge, 1 (12, Part 2): 299-305.