CHAPTER 5
FORMATION OF SOLUTIONS ON CONTROL

If it were so, then that would be alright,
But if alright then that is how it would be.
But because it is not so, then it is not so.
Such is the logic of things.

Lewis Carroll

5.1 Goals, Situations and Plans

In the present chapter we shall look at procedures used during semiotic control for forming a sequence of solutions with the aid of which it is possible to convert a current situation into a certain goal one. In the general scheme of situational control this problem is solved jointly by the Correlator, the Classifier, and the Extrapolator, shown in Figure 1.7. Systems that assure the solution of the given problem are customarily called planning systems or planners. Planners first form a plan, then check its executability and effectiveness (efficiency), select among those formed the best plan, begin its execution and if necessary correct the plan upon entry of additional information from the object of control and the environment.

Usually two types of planning are distinguished: planning by states and planning by subtasks. In the first case the concept of state is introduced which is composed from the state of the object of control, the state of the surrounding medium, and the state of the control system. The construction of the plan takes place in the statespace in such a manner that each one time solution on control converts the whole system from one state to another in the statespace. The plan is represented in this case by a certain trajectory in the statespace.

In the second case it is assumed, that there exists a certain selection of moduli, capable of solving definite problems. The planning process consists in the search of such a decomposition of the original problem of which the elements would turn out to be ready-made moduli. Then the plan is a tree-like structure, moving along which it is possible to gradually obtain a solution of the original problem from solutions of more local problems.
In spite of a certain difference in the formulations of these two problems of planning, they can be described by a certain common model, shown in Figure 5.1. On it is shown a network consisting of 15 apexes. The cross-hatched apex 1 is the initial one. It symbolizes either a current situation or a state in the statespace, or the initial problem that must be solved. The apexes 13, 14 and 15 (on the figure they are cross-hatched) symbolize either those goal situations or the final states in the statespace into which it is required to convert the object of control and the control system, or else ready-made moduli for the solution of the problem, which are already available in the system. During planning in the statespace it is necessary to find a path that leads from apex 1 to one of the cross-hatched apexes. In setting up an optimization problem it is possible, for instance, to look for the shortest path of this type. Or, if the transitions along the arcs require a certain expenditure of resources ascribed to these arcs, it is possible to pose the problem of attaining a final situation with a minimal expenditure of resources or to pose more complex problems: to find a path with a minimal expenditure of resources, but such, that the number of steps (time on the way) would be not greater than specified, or, to find a path, minimal in the number of steps, but not minimal by resources, although satisfying the upper limitations on resources.

During planning by problems it is necessary to go through the entire spread of paths leading from the initial apex labelled 1 to modular cross-hatched apexes that assure the solution of the initial problem.
In this manner, while searching for a plan in the statespace, all branchings in the apexes are considered as alternative. It is necessary to select one (any one) continuation of motion. Whereas while working in the problem-space, all arcs issuing from a given apex must be gone through, if they are interpreted as ratios of the type part-whole. Naturally, even then it is possible to consider not the only one decomposition of the initial problem into subproblems, but an alternative decomposition of it. In that case are needed special directives on the character of the arcs issuing from the apexes: are they alternative (OR-arcs) or are they mandatory for joint passage through (AND-arcs). A network with both types of arcs is commonly called an AND-OR network or an AND-OR graph. On Figure 5.1 small double arclets are used which unify among each other the AND-arcs. This means that the initial problem, corresponding to Apex 1, is decomposed into subproblems 2 and 5. Problem 2 reduces to problem 3 or to problem 4, which reduces simply to problem 5. Problem 5 can be reduced either to problem 6 or to problem 9. Problem 9 decomposes into problems 10 and 13. Problem 10 reduces to problem 14 that is already solved in the system, while problem 13 is decomposed into already solved problems 12 and 15.

Thus, in the case of a space of subtasks the plan of solution of problem 1 has the following form: using the already solved problem 14, problem 10 is solved. Using the solved problems 10, 12, 15, problem 9 is solved. Using the solved problem 9, problem 5 is solved. Using the solved problem 4, problem 2 is solved. Using the solved problems 2 and 5, problem 1 is solved. For the case of the statespace on the same network, the plan of attaining the goal-state might be as follows: 1-5-9-10-14. On this example is seen the difference of the above-considered problems of planning. Therefore we shall speak about the problem of planning in general, without indicating its type.

Before considering the problems of planning themselves, it is useful to introduce certain classes of networks, of the type shown in Figure 5.1. Firstly, let us divide such networks into closed ones and open ones. A network is closed, if the number of its apexes and arcs is fixed and does not change in the process of planning. Such networks are the most simple, but unfortunately, occur rarely in practice. The possibility of constructing a closed network for systems of situational control would signify

274
that the number of complete situations is finite and is enumerated from the start. Then, while analyzing the network based on these situations, it would be possible to search for a plan of transforming a current situation into the one that is of interest to us. Then the arcs could be interpreted as controlling actions under the influence of which the transition is accomplished. Clearly, such a finite model is not realistic. Our knowledge about the object of control and the surrounding medium is usually incomplete and therefore, does not permit us to assume, that the network for planning has been given to us a priori.

In open networks new apexes may appear because of the occurrence of new situations. New arcs may also occur if in the process of functioning of the control system we are able to find new links among the control decisions and the changes of situations.

It is possible to introduce one other characteristic of the planning network which subdivides such networks into deterministic and nondeterministic ones. In deterministic networks the selected solution, corresponding to some arc, realizes exactly the transition to a new apex, indicated in the network. In nondeterministic networks this is absent. Upon realization of some single-step solution the transition to one or another state is determined either by a certain distribution of probabilities, or is estimated qualitatively with the aid of frequency quantifiers or distribution functions. Such a situation is most typical for control systems of complex objects, when the technology of control depends on the level of training of the controller or on the completeness of the knowledge about control, stored in the memory of the control system.

The classification of the planning networks cited by us is also true for the planning in the problem-space. Openness in such a system is related, for example, to the introduction of new methods of decomposition and a conversion of some problems to type moduli, the solution of which is known, i.e., to the expansion of the library of standard moduli. Nondeterminacy may be treated, for example, as an inexact reduction of the decomposed problem to the totality of these or other subproblems. But, while for planning in the state-space the introduced classes of networks are most important for planning methods, for planning in the space of subproblems such a subdivision is less significant.
It is possible, finally, to divide the problems of planning into single-level and multilevel ones. Multilevel planning occurs then, when a totality of plans is built on the basis of information on the statespace or the space of subproblems of a different level of detailing. Before constructing an exact and detailed plan, it is always desirable to have a certain assurance that the final goals of planning are attainable. If they are not attainable, then it is desirable not to waste time on planning. It can turn out to be very great. This is where the multilevel plans can be of use. First, at the very highest level, the possibility of planning is established. If it is possible, then a variant of the most crude plan is constructed, the steps of which must be defined more precisely at lower levels, operating with more detailed descriptions of situations or problems. If the initial crude plan passes all levels of planning, then we obtain at the very lowest level the final plan. If however, it is found at some level that the transition planned high-up from crude generalized information cannot be realized, then a return upwards is made where another variant of the plan is selected (if it exists).

In systems of situational control, hierarchical planning occurs in a natural manner because of the hierarchy of the Classifier. The layer cake in the Classifier demands, for the possibility of working with it, a similar layer cake in the Correlator. Only in that case will all the information stored in the multi-level system of generalized descriptions be successfully utilized. We will describe below how this is done.

5.2 Strategy of Planning

We shall describe here some universal strategies of planning, which were utilized in different planners, both for systems of situational control as well as for solving problems on automation of constructing programs for ECM* or for finding of a goal-oriented succession of actions for robots. For questions of planning all these problems have much in common, which allows us to consider the problem of constructing a planner in a certain detachment from the pithy formulation of a problem.

---

* Electronic Computing Machines
1. Method of the Direct Wave. It represents the simplest of all methods of planning. Its essence boils down to the following. At the initial moment all apexes are excited which correspond to the initial conditions of the problem (for example the description of the initial situation). Next, if the planning is carried out in the state-space, all the arcs issuing from the excited apexes are excited. If however, planning is carried out in the problem space, then all arcs issuing from excited apexes are excited, but the arcs themselves are marked with special markings. Namely, all arcs related to one alternative variant are tagged with the number of that variant. For example, for the planning network shown in Figure 5.1, upon excitation of apex 1, both arcs are excited. They are labeled with the number of the alternative 1. Next the excitation continues to propagate along the planning network. In the case of planning in the state-space the propagation of excitation will either terminate at the goal vertices, or will cease to propagate, if it is impossible to excite new vertices. For planning in the problem space, the planning will cease upon reaching the same conditions. However, while for the state-space the attainment of even one goal apex serves as a signal about the solution of a problem, for the problem space it is still necessary to check whether all the paths are excited that are related to even one alternative and lead from the initial apexes to the goal one. The searched-for plan is formulated by a reverse check of the excited paths leading from the goal apex, which had been excited, to the initial apexes. In the case of state-space this usually is the path (any excited path, unless an optimization problem is posed), while for the problem space - it is the totality of paths that corresponds to a certain alternative.

Example 5.1. Propagation of excitation by the method of the direct wave for the case of planning in the problem space is shown in Figure 5.2. The planning network is the same as on Figure 5.1. The number of the alternative variant is formed by a gradual increase from left to right. The goal apexes have been excited by alternatives with numbers 121 and 122. For alternative 121 all apexes are excited which are related to it, and hence for this alternative the plan is the sequence of carrying out problems 13, 12, 15, 11, 14, 10, 9, 5, 4, 2, 1. For alternative 122 are also excited all apexes that correspond to it. The plan by this alternative has the form of the
following sequence: 14, 10, 8, 7, 6, 5, 4, 2, 1. If for any one alternative, apexes would become excited that happen to be hanging, i.e., from which there is no further path toward the cross-hatched apexes, then such an alternative would not correspond to the plan of the solution of the problem. On Figure 5.2 this case is related to alternative 11.

The weakness of the direct wave method is that the excitation propagates along the entire network, which may require a large expenditure of time.

2. Method of the Reverse Wave. This method is completely similar to the preceding one. Except that excitation begins to propagate from those goal apexes which it is required to attain. Cessation of excitation propagation occurs upon attaining the initial apexes. In planning by states, this occurs upon the first such attainment, and in planning by problems, this occurs upon attainment of initial apexes by all branches of any one alternative.

3. Method of Counter-Waves. For accelerating the search process for the path between the initial and the goal apexes it is possible to simultaneously utilize the methods of direct and the reverse waves. Here the process of excitation in the statespace terminates, when there occurs the first encounter of the excited fronts, while in the case of the problem space such an encounter must occur for all the branches of some one alternative. The weakness of the method of counterwaves may manifest itself in the case, when the fronts of the waves diverge, and the problem is posed of determining
the moment of cessation of the excitation process. This situation is shown on Figure 5.3.

4. Method of Local Improvements. This method has been for many years the basis of many planning systems. Its idea consists in that a certain metric is introduced into the planning network. In each apex of the planning network, in the presence of alternative selection of the continuation of motion along the network, it turns out to be possible to estimate, how effective one or another selection leads closer to the goal apex. If such an estimate were true, then the motion from the initial apex to the goal one would not cause any difficulties. It would be sufficient, in each apex with alternative selection, to move along the arc that had the best estimate. Such a solution is possible only in small deterministic closed planning networks. However, for such networks the problem of planning practically vanishes. For any initial and any final apex it is possible to find ahead of time the best path and to write it into the system memory. Upon excitation of the corresponding pair of apexes or of a set of such pairs, the answer is obtained in one step of addressing the information kept in the system memory.

For the case of open networks, however, or for closed networks whose dimensionality does not permit enumeration of all the paths available in the system, there occurs the problem of the type of a searching for a path in
a labyrinth* with availability of local information on approach to the goal apex. This idea was used for the first time in the well known program General Solver of Problems (GSP). The method of local improvements is based on the introduction of a special metric in the situation space. With its aid it turns out to be possible to measure the distances between situations, to estimate the measure of their nearness. And, in particular, to estimate the measure of nearness of the situation corresponding to the apex of the network in which we find ourselves, to the goal situations attainable from that apex. For each transformation permitted in the planning network (i.e., transition from one situation to another along an arc characterizing this transformation) are indicated those changes in the degree of similarity with the goal situations, which occur after the transformation. And at each step of the method of local improvements is selected that transformation, which maximally decreases the differences of the obtained situation from the goal ones. This method is applicable in planning both in the statespace as well as in the problem space. It can also be used as a basis of establishing the equivalence of descriptions of different situations in the sense of their transformality, one into another, with the utilization of a given set of operators.

Example 5.2. In the program LOGICIAN-THEORETICIAN, that served as prototype of the program GSP, is introduced a system of differences and a system of operators for their removal for transforming the descriptions made in the language of formulas of sentential calculus. Altogether there have been introduced eight distinctions (differences) among these formulas.

1. Distinction $P_1$. In the tested equivalence of two expressions of sentential calculus $z_1 = z_2$ (i.e., while trying to establish the equivalence of these two descriptions) there is a distinction between $z_1$ and $z_2$ in that either $z_1$ or $z_2$ contains a propositional variable which is absent in the other compared expression. Such a distinction exists for example, in the

* In §1.3 such problems were called labyrinth ones. There too, were discussed the psychological assumptions that formed the bases of the strategy of local improvements. An finally, there too was characterized the General Solver of Problems (GSP) about which we are yet to speak. Systems based on the modeling principle we shall discuss a little later.
descriptions a and ab. It is necessary to select a transformation which adds this variable.

2. Distinction \( P_2 \). It is analogical to \( P_1 \), but it is necessary to select a transformation which removes the excess variable.

3. Distinction \( P_3 \). In one of the formulas \( w_1 \) or \( w_2 \) a certain propositional variable has \( \xi \) entries more than the other. Such distinction is observed, for instance, between the descriptions: \( a\overline{v}b \) and \( ab \). It is necessary to select a transformation which decreases the number of entries of this variable.

4. Distinction \( P_4 \). It is analogical to \( P_3 \), but it is necessary to select a transformation which increases the number of entries of this variable into that description where there are fewer entries.

5. Distinction \( P_5 \). A distinction in the outermost operation contained in the formula. It is observed, for instance, in the descriptions \( (a\overline{v}b)(\overline{a}\overline{v}c) \) and \( ab\overline{v}c \).

6. Distinction \( P_6 \). One of the descriptions contains a common sign of negation over the entire description, while the other description does not have it. As an example may serve the descriptions \( a\overline{v}b \) and \( \overline{a}\overline{v}b \).

7. Distinction \( P_7 \). A distinction in the character of grouping of the propositional variables in \( w_1 \) and \( w_2 \), as for example, in the descriptions \( a\overline{v}(b\overline{v}c) \) and \( (a\overline{v}b)Vc \).

8. Distinction \( P_8 \). A distinction in the position of subformulas with respect to the external operation. As an example of such a distinction may serve the distinction in the descriptions \( ab+c \) and \( c+ab \).

The distinctions between descriptions introduced by us, can, of course, be combined among each other. But it is important that the above-cited system of distinctions forms a complete base system. Any distinction in
two descriptions in the language of sentential calculus is expressed by a certain combination of these base distinctions.

Let us enumerate now those transformation operators serving for removal of distinctions, which were used in the program LOGICIAN-THEORETICIAN. Altogether, ten groups of such operators are available. Operators entering into the first eight groups are symmetrical, they can be applied in any direction. Operators of the last two groups are unsymmetrical, and their use is possible only in the direction shown by the arrow. In the descriptions are adopted the following standard designations:

\[
\begin{align*}
O_1: & \begin{cases} a \lor b \equiv b \lor a, \\ ab \equiv ba; \end{cases} & O_2: & \begin{cases} \overline{a} \rightarrow b \equiv \overline{b} \rightarrow a, \\ a \rightarrow \overline{b} \equiv b \rightarrow \overline{a}; \end{cases} \\
O_3: & \begin{cases} a \lor a \equiv a, \\ aa \equiv a; \end{cases} & O_4: & \begin{cases} a \lor (b \lor c) \equiv (a \lor b) \lor c, \\ a(bc) \equiv (ab)c; \end{cases} \\
O_5: & \begin{cases} a \lor b \equiv \overline{ab}, \\ \overline{ab} \equiv \overline{a} \lor \overline{b}; \end{cases} & O_6: & \begin{cases} a \lor b \equiv \overline{a} \lor \overline{b}, \\ a \lor \overline{b} \equiv \overline{ab}; \end{cases} \\
O_7: & \begin{cases} a \lor bc \equiv (a \lor b)(a \lor c), \\ a(b \lor c) \equiv ab \lor ac; \end{cases} & O_8: & \begin{cases} a \equiv a \lor b \overline{b}, \\ a \equiv a(b \lor \overline{b}); \end{cases} \\
O_9: & \begin{cases} ab \Rightarrow a, \\ ab \Rightarrow b; \end{cases} & O_{10}: & a \Rightarrow a \lor b.
\end{align*}
\]

The transformations of the first eight operator groups do not alter the truth of the expressions of sentential calculus. The last two groups of transformations, serving for introduction and removal of propositional variables are such, that from the truth of the expression to the left of the arrow there follows the truth of the expression to the right of it, but not the reverse.

After introduction of the selection of distinctions and selection of operators for their removal, a special table is set up, in which are listed in the order of priorities assigned to them the operators that remove these or those distinctions. Among the distinctions themselves are also set up priorities according to the importance of one or another of them. In our
example the numeration of distinctions defines these priorities, while for the operators there are no priorities. The table itself that relates the distinctions and the operators that remove them, is shown below (Table 5.1). Crosses on it mark the positions in which the application of the respective operator removes the existing distinction.

Table 5.1

<table>
<thead>
<tr>
<th></th>
<th>(O_1)</th>
<th>(O_2)</th>
<th>(O_3)</th>
<th>(O_4)</th>
<th>(O_5)</th>
<th>(O_6)</th>
<th>(O_7)</th>
<th>(O_8)</th>
<th>(O_9)</th>
<th>(O_{10})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_3)</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_4)</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_5)</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_6)</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_7)</td>
<td></td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>(P_8)</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

As an example of utilization of the program LOGICIAN-THEORETICIAN let us consider the problem of establishing the equivalence of the descriptions \((a \land b)c\) and \(\overline{a \lor b}c\). For definiteness let us transform the second description, attempting to reduce it to the first one. The main observable distinction consists in the presence of a mutual negation over the formula, i.e., the distinction \(P_6\). For its removal we use the operator \(O_6\) and as a result we obtain the description \((\overline{a \lor b})c\). Comparing the obtained expression with the untransformable original one, we find a distinction in the fact that a subformula appears that contains a mutual negation, i.e., the distinction \(P_6\) is present again. We apply to this subformula the same operator \(O_6\) and obtain \((\overline{a \lor b})c\). Now the distinction \(P_6\) is observed to exist for the same subformula, because the more external operation on the compared descriptions coincides (conjunction). To remove the distinction we apply the operator \(O_6\). As a result we obtain \((a \land b)c\). The coincidence of the descriptions proves the equivalence of both initial descriptions.
A problem for the program LOGICIAN-THEORETICIAN, and in general for all planning methods, based on the idea of local improvements, is the determination of the nonequivalence of the compared descriptions. In that case, at a certain state of transformations, it is necessary to terminate them as being non-prospective. This may not occur automatically either, inasmuch as the operators, as it happened in the example cited above, may yield cyclically repeated descriptions. This situation is similar to one in which, while wandering about in a labyrinth, the people entrapped in it repeat many times the same path. As is known, a way out by marking the traversed path (the thread of Ariadne in the well-known Greek myth about Theseus who searches for a way out in the labyrinth of the Minotaur). For the planning algorithm this means, that it is necessary to remember the pre-history and to execute the operation of return (back-tracking) upon finding a repetition of the transformation in the form of a certain cycle.

In the program LOGICIAN-THEORETICIAN the table of distinctions and of operators of their removal was given in advance. However there exist programs which formulate the distinctions in the description obtained at a given stage, and also in the goal-description*.

5. Method of Constructing the Proof. In this approach the descriptions of the initial and the goal situations are viewed as formulas of a certain calculus. All elements entering into the initial descriptions are treated as axioms of this calculus, while all elements forming the goal description, --as formulas which need to be derived from the given selection of axioms with the aid of an available system of rules of deduction. Usually, the calculus of the predicates of the first degree is used as such a calculus. The rules of derivation are divided into two types: rules of derivation of a universal character, used in the traditional calculus of predicates, and the rules of derivation that are true only within the framework of that problem region which describes the given object of control, the peculiarities of its functioning, and the specifics of controlling it. The rules of the second type are customarily called the heuristic rules of derivation. As procedure

* Such a program is, for example, the program FDS of Quinlan and Hunt, which is discussed in the Commentary for the present chapter.
for the search of derivation use is made of some sufficiently powerful procedure of the type of the method of resolutions or of the reverse method of derivation of Maslow. These procedures will be described below.

Usually the technology of planning by the method of proof construction has the following form. First, an attempt is made to find the derivation of the goal description from the initial one. If such derivation is found, then the plan has been constructed. If it cannot be done, as determined on the basis of some principle of stopping, similar to the one indicated for the program LOGICIAN-THEORETICIAN, then a certain heuristic rule of derivation is applied to the initial description, and the process of proof is repeated now with a new initial description. Methods of this type are presently the fundamental ones in constructing the planners in the most diverse areas.

6. Method of Forestalling Planning. This strategy of planning is finding ever-increased application in systems of semiotic control and in planning the actions of intellectual robots. In its basis lies the idea of multilevel planning, about which we spoke already. Before realizing the final planning, for example by searching for a derivation (which may not exist, too), a generalized planning is carried out. It differs from ordinary planning by the utilization for the construction of plans not of the complete descriptions of possible paths of attaining goals, but of truncated descriptions or of descriptions obtained from original ones by aggregation, by merging of individual apexes into generalizes apexes. If generalized planning allows construction of a plan, then a transition is made to a precise level of planning, where this generalized plan serves as an initial approximation to the precise plan.

Let us make one other important observation. Two types of planning are possible: the monotonic (linear) and the nonmonotonic (non-linear). During planning of the first type we approach, at each step of planning, the goal apex, so to speak continually locally improving our position in the network. Procedures of planning of the second type do not possess this property. At individual steps they kind of permit a worsening of the plan and the next apex may turn out to be more remote from the goal apex one than the one previously found. But then later it turns out to be possible to sharply
improve the plan, i.e., to approach closer to the goal apex. In our life experience we will find numerous examples when such nonlinear plans turned out to be the only possible means for reaching the goal. Local non-success (sometimes created consciously) allowed them to quickly move towards the designated goal.

In the conclusion of the present paragraph we will note, that the planner can utilize not one fixed strategy, but can have at his disposal a selection of planning strategies, which can assure a more effective functioning of it.

5.3 Utilization of the Method of Resolution During Planning**

The method of resolution was proposed in the mid-1960s. It belongs to that group of methods of searching for a derivation which is usually called methods of refutation. This appellation is caused by the fact that in these methods, instead of searching for a derivation of a certain postulate \( F \), is being searched a proof of the non-derivability of the negation of the given assertion \( \neg F \). The equivalence of both problems follows from the closed nature of the calculus of predicates of the first order, in which is used the method of resolution, i.e., from the rule of the eliminated third in such calculations the truth of \( F \) follows from the falseness of \( \neg F \).

For applying the method of resolution it is necessary to translate the description of the statement that interests us into the language of logic of predicates of the first order and to represent it in a certain standard canonical form. Such representation is usually called a standardized one. It does not contain quantifiers (they are replaced by Skolem functions known from mathematical logic) and represent a conjunction of disjunctions of predicates. As arguments in the predicates may be used constants, variables, and Skolem functions. Standardization is carried out by means of using the system of equivalent transformations of the calculus of predicates of the first order.
Example 5.3. Let there be the description

\[ \forall x \exists \tau \left( [(x r_{s2} b) V (x r_{s3} b)] (t \rho \tau) \right) \]

Here \( x \) - some man whose name is not fixed, \( b \) - a letter, \( t \) - time, \( r_{s2} \) - to be the sender, \( r_{s3} \) to be the receiver. The meaning of this description is that each man at some moment of time \( \tau \) (there will always be at least one such moment) is either a sender of the letter, or its receiver. Let us depart from this notation in the language of situational control to a notation in the language of predicates. Expressions in round parentheses we will replace by corresponding predicate, whose interpretation is defined by the interpretation of the relations within the parentheses. We then obtain

\[ \forall x \exists \tau (P_1(x, b, \tau) V P_2(x, b, \tau)) \]

In the predicate \( \tau \) is entered as a variable. The quantifier of existence in this description is in the zone of action of the quantifier of generality. Therefore, in eliminating the quantifier we introduce the function of Skolem \( f_2 \), while the quantifier of generality is, as always, eliminated without any change in notation. After that our description takes on the form

\[ P_1(x, b, f_1(x)) V P_2(x, b, f_2(x)) \].

Example 5.4. Consider the predicate notation

\[ (P_1(x) \rightarrow P_2(x)) (P_3(x, y) \rightarrow P_4(y)) \].
Using equivalence relation of predicate logic, we transform it so as to eliminate from the notation the implication operations

\[
(F_1(x) \lor F_2(x)) \lor (F_3(x, y) \lor F_4(y)).
\]

Both representations obtained in the last two examples are already standardized. For the resolution method it is also customary to omit the signs of conjunction and to write instead of the conjunction form the set of disjuncts that form it. In Example 5.3 there is only one such disjunct, while in example 5.4 there are two. The set of disjuncts for a formula \( F \) we will denote by \( S_F \). For negating the formula \( F \) it is sufficient to show, that at least one of its disjuncts is underviable. Per force of the conjunctive relation of the disjuncts here, the formula \( F \) itself will be underviable. The meaning of all above-said is that there arises the possibility to replace the check of the contradiction of some formula \( F \), which may be quite complicated, by a successive check for contradiction of rather structurally simple formulas-disjuncts. But the method of resolution allows to make even this process, which with a large number of disjuncts can turn out to be cumbersome, more uniform and effective.

In what consists the basic difficulty of checking the falseness or the truth of a predicative expression? In that it is necessary to check the truth of such expressions during combinations of variables, taken from regions of predicate definitions containing such variables. These regions themselves can be infinite. Hence the prospect of direct checking of all possible interpretations of predicate meanings and of their combinations appears to be most sad. Actually, the situation is not so bad. It has been shown that there exists such a region of definition of all predicates entering into a description called the universum of Erbran which possesses an extremely important characteristic for us, namely: if in the universum of Erbran \( S_F \) is contradictory, then \( F \) is derivable.

The Erbran universum is constructed with the aid of a special procedure. If in \( S_F \) there are \( \ell \) constants, then a zero set \( H_0 \) is formed that consists of these constants. If there are no constants in \( S_F \), then some
arbitrary constant is adopted as the zero set of the Erban base. Next, the sets of the Erban base are constructed inductively. The \( H_{m+1} \) set is obtained by a unification of \( H_m \) and these propositional variables \( f(w_1, w_2, \ldots, w_n) \), for which \( w_2 \) are the elements of the set \( H_m \).

Example 5.5. Consider the predicative description, obtained in example 5.3. If an \( S_F \) is constructed for it, consisting of one disjunct, then it is seen that there is no constant in this disjunct. Let us introduce an arbitrary constant, \( x = a \). Then the Erban universum for this disjunct takes on the form

\[
\begin{align*}
H_0 &= \{a\}, \\
H_1 &= \{a, f_1(a), f_2(a)\}, \\
H_2 &= \{a, f_1(a), f_2(a), f_1(f_1(a)), f_1(f_2(a)), f_2(f_1(a)), f_2(f_2(a)), \ldots \}
\end{align*}
\]

If constant expressions from the Erban universum are substituted into all disjuncts from \( S_F \) instead of the variables, then an Erban base is formed as a result.

Example 5.6. For \( S_F \) from example 5.3 and with consideration of the result of example 5.5, we obtain the following Erban base:

\[
\begin{align*}
P_1(a, b, f_1(a)), P_2(a, b, f_2(a)), \\
P_1(f_1(a), b, f_1(f_1(a))), P_2(f_1(a), b, f_2(f_1(a))), \\
P_1(f_2(a), b, f_1(f_2(a))), P_2(f_2(a), b, f_2(f_2(a))), \\
P_1(f_1(f_1(a)), b, f_1(f_1(f_1(a)))) 
\end{align*}
\]

If, instead of the variables, some element from the universum of Erban is substituted into the disjuncts, then a certain interpretation \( S_F \) is obtained. In our example, in explicit form, are written out three interpretations and is given one expression selected to a fourth interpretation. In this manner, the Erban base contains all interpretations of the disjuncts of \( S_F \) on the Erban
universum. The number of these interpretations can be finite if the Erbran
universum is finite, or can be infinite, as in our example.

To each Erbran base can be single-valuedly juxtaposed to a certain
semantic tree. Its construction we shall illustrate on an example.

Example 5.7. For the Erbran base built in the preceding example,
the semantic tree has the form shown on Figure 5.4.

\[ \text{Figure 5.4} \]

On it \( P_j \) and \( P_k \) designate the predicates \( P_1 \) and \( P_2 \) with ordinal numbers \( j \), \( k \),
\( \text{assigned to them by the order of recording in example 5.6. Thus, the} \)
\( \text{predicate } P_1 \) has the form

\[ P_1(f_1(a), b, f_1(f_1(a))). \]

In the semantic tree the paths, leading from the root apex
downward, correspond to interpretations \( S_F \) on the universum of Erbran. The
complete tree contains all such interpretations.

Example 5.8. Let \( S_F = \{P_1(x), P_2(a) \lor P_3(y), \overline{P}_1(x) \lor \overline{P}_2(x), \overline{P}_1(x) \lor \overline{P}_3(a)\} \).
For this set of disjuncts the Erbran universum is very simple. It consists of
only one constant, \( H_a = \{a\} \). Then the Erbran base is also finite and has the
form

\[ I = \{P_1(a), P_2(a) \lor P_3(a), \overline{P}_1(a) \lor \overline{P}_2(a), \overline{P}_1(a) \lor \overline{P}_3(a)\}. \]
Or, omitting for ease of uniting the a's

\[ I = \{ P_1, P_2 \lor P_3, \overline{P}_1 \lor \overline{P}_2, \overline{P}_1 \lor \overline{P}_3 \}. \]

To this Erbran base corresponds the semantic tree shown in Figure 5.5.

![Figure 5.5](image)

Let us consider those eight interpretations which this tree specifies for \( S_F \). For interpretations with numbers 5-8, the set of disjuncts is unsatisfiable already because, with a false \( P_1 \), the first disjunct of \( S_F \) will be false regardless of interpretations \( P_2 \) and \( P_3 \). For this reason, the right portion of the semantic tree, shown on Figure 5.5, needs to be analyzed no further (i.e., no further motion is needed downward along the branches of this tree from this apex), which allows to sharply decrease the examination and checking of the realizability for the set of allowable interpretations \( S_F \). thus, it is only necessary to check out those interpretations in which \( P_1 \) assumes a true value. With this condition, the first disjunct of \( S_F \) is real. For trueness of the third and fourth disjuncts in \( S_F \), it is necessary that \( P_2 \) and \( P_3 \) should be false simultaneously. But, fulfillment of that condition leads to the falseness of the second disjunct in \( S_F \), whence it follows that all eight allowable interpretations are such that not one of them does not lead to the satisfying of \( S_F \). Therefore, \( F \) is non-derivable, while \( \overline{F} \) is a derivable formula.
The apexes of the semantic tree in which is first discovered the uselessness of a given interpretation for derivation of \( F \) (satisfying \( F_F \)) are usually called unfavorable. If all arcs issuing from some apex of the tree end up in unfavorable apexes, then it is said that for the given apex the tree is closed. If it is closed for the root apex, then this witnesses the non-derivability of \( F \).

The method of resolution coexists in the availability of a procedure of a gradual ascent up the semantic tree, a result of which the unfavorable apexes move up among the levels of the tree. The ascent continues until such time as this is possible. If in this manner it becomes possible to reach the root apex (i.e., to make the entire initial semantic tree closed), then this will prove the non-derivability of \( F \).

This procedure is based on the search of resolvents. A resolvent of two disjuncts is called that part of a description which remains if from these two disjuncts a complementary pair is removed. A complementary pair is two expressions in which are utilized identical predicates, except that in one expression they enter without negation, and in the other with negation. Arguments of the predicate play no role. As an example of a complementary pair may serve the following two disjuncts:

\[
P_1(x) \lor P_2(y) P_3(z); \quad P_4(x,y) \lor \overline{P}_2(f(x)) P_3(z).
\]

The resolvent of this complementary pair has the form \( P_1(x) \lor P_4(x,y) \). What good is a resolvent? If two disjuncts containing a complementary pair are such that they lead to a non-satisfying of \( S_F \), then their resolvent possesses the same property. It is this that is taken into account in the process of proving the non-derivability of \( F \). If a pair of disjuncts does not contain a complementary pair, then the resolvent can be absent. If a pair of disjuncts is such that they both completely form a complementary pair, then their resolvent is equal to zero. The possibility of ignoring, during a search for complementary pairs, their arguments, is based on the fact that a change of variables is the predicate, if it is false in a certain range of values, cannot make it true upon changing the variable and keeping or decreasing this range of values.
Now it becomes possible to describe the essence of the procedures of the method of resolution. Among a set of disjuncts are searched such two disjuncts which contain a complementary pair. To the set $S_F$ is added a resolvent of these two disjuncts. The process is repeated until in the set $S_F$ there appears an empty disjunct (i.e., a zero resolvent). This will witness the termination of the process of proving the non-derivability of $F$.

**Example 5.9.** Let us consider the set $S_F = \{\overline{P_1(x)}VP_2(x), P_1(f(z))VP_2(x); P_2(y)VP_3(f(y)), \overline{P_3(x)}\}$. For two of the disjuncts the resolvent is $P_2(x)$. Together with the third disjunct this resolvent generates a new resolvent, $P_3(f(y))$. Finally, the resolvent of this secondary resolvent together with the last disjunct gives the zeroth disjunct. Now we can state that the $F$ in our example is non-derivable.

We have presented only the essence of the method of resolution. While carrying out the procedure of consecutive search for resolvents, various unpleasantnesses are possible, when the next-in-line resolvent can't be generated or when the process becomes cyclic. Moreover, with a large number of initial disjuncts (and they may even be infinitely many) special measures are required for accelerating the process of search for an empty disjunct. And, finally, as in all methods of refutation, for the method of resolution there occurs the problem of cessation of search for the zero disjunct, when in reality it is impossible to obtain a refutation because of the derivability of $F$ in the set $S_F$. There exists a large literature devoted to all these problems, but the discussion of such questions, important for increasing the effectiveness of the work of the planners based on the realization of the given method of derivation search, goes beyond the limits of our book. Those interested in a detailed study of these problems may familiarize themselves with the literature given in the commentary to the present chapter.

In conclusion of the present paragraph, let us note some peculiarities of systems of decision planning based on derivation in a formal system. Firstly, the plan itself, that is formed with the aid of a logical derivation, is realized in a real time and space. And the characteristics of this real environment must be considered during planning. the next example illustrates those unpleasantnesses which may occur if they (the characteristics) are ignored.
Assume that if a certain moment of planning, while analyzing a given situation, it turned out to be possible to work out two mutually independent solutions: \( U_1 \) - to move a crane 200 meters forward along the dock, and \( U_2 \) - to move the boom of the crane 90° to the left. However, from the derivation of these two solutions there is no way follows the conclusion of the possibility of their simultaneous realization, although in ordinary deduction systems there follows, from the derivability of \( U_1 \) and \( U_2 \), the derivability of their conjunction \( U_1 \& U_2 \). To move the crane 200 meters forward along the dock, while simultaneously rotating its boom 90° to the left. In a real situation this may lead to a disaster. It may happen, that along the path of motion of the crane, 50 m from the point where it was, there is another crane, whose body can be touched by the boom of our crane during turning. In other words, planning demands the consideration of the derivability of one or another solution with ties to the dynamics of change of situation resulting from realization of another demand solution (decision).

There is yet another remark, selected to function if plans based on the idea of local improvement of motion toward the goal. In many problems, the goal can be reached only in the case, when at a certain stage of solution it is necessary to kind-of back off from the goal, to make one's situation worse than it was, which would allow at the next step to get considerably closer to the final goal. The theory of such non-monotonic plans is still in an embryonic condition and not anything like general approaches have been developed for their formation.

The method of resolution, of course, is not the only method of deduction derivation that can be used in systems of semiotic control. There exist also other planning methods based on logical derivation. Commentaries to the present chapter discuss them in particular. But in all similar methods there is one common defect. It consists in that it is necessary to have available a sufficiently complete description of the problem area to serve as initial axioms with the aid of which derivations will be made. And this causes great difficulties and presumably leads to uncorrectible errors due to the incompleteness of the initial description. Besides, the earlier noted difficulties related to the dynamic nature of the problem areas, also limit the possibilities of using purely deductive systems of forming solutions.
Below, we shall consider that path along which have proceeded systems of planning solutions in practical control systems.

5.4 Functional Models for Planning

In searching for a solution related to the control of some object, it is natural to use those data which are known to the control system, both on the object of control itself as well as on the processes occurring in it. Those data, as was already mentioned before, are kept in the memory of the control systems in the form of a data base. Data on each current situation must be filled with concrete data that characterize it. The base of knowledge is sort of immersed in the base of data, it is fed by the data. And depending on the filling by concrete data, the planner works out some or other solutions, expressed in plans of influencing the object of control. On Figure 5.6, all said is represented in the form of a scheme, that has now become standard, of planning the search of solutions. On this figure are not shown the connections that the data base has with the object of control, and the connections of the planner with other subsystems of the control system.

![Diagram](Figure 5.6)

To illustrate how the model of the problem area is arranged and how the planner can utilize it in searching a solution to attain the posed control goal, let us consider one class of such models frequently occurring in practice. These are the so-called functional models. Every functional model represents a network without orientation including in itself apexes of two types: descriptors and specifiers. In graphic representation of functional
models, the descriptors are commonly designated by circles and the specifiers by rectangles. Each specifier is juxtaposed to a certain functional expression of the type \( f(x_1, x_2, \ldots, x_n) = 0 \), where it is assumed that \( f \) allows an explicit solution with respect to at least one of the arguments, \( x_i \). In other words, for at least one \( x_i \), this equation can be rewritten in the form: \( x_i = f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \). Such an operation is called an allowable solution of the specifier. It is assumed that for one specifier there may exist from 1 to \( n \) allowable solutions. With the specifier \( f(x_1, x_2, \ldots, x_n) = 0 \) are connected \( n \) ribs (edges) of the functional model each of which is connected with one of the descriptors of sets \( \{x_1, x_2, \ldots, x_n\} \). Two specifiers in the functional model are connected with each other by a certain descriptor if the latter defines an argument common to both descriptors.

Example 5.10. On Figure 5.7 is shown a fragment of a functional model which is related to the subject area, Planimetry of a Triangle. On the figure the elements of the triangle are designated as it is customary in a school course: \( \alpha, \beta, \text{ and } \gamma \) are the angles of the triangle; \( a, b, \text{ and } c \)--its sides (the selection between the designations of angles and of sides is illustrated by the sketch in the upper part of the figure); \( R \) and \( r \) are, respectively, the radii of the circumscribed and the inscribed circles of the triangle; \( p \)--the semiperimeter of the triangle; and \( s \)--its area. Inside each specifier is written the relation connecting various descriptors. The maximal number of possible solutions of each specifier corresponds to the number of edges (ribs) connected with it. For instance, for the specifier with the relation \( s - rp = 0 \), such possible and allowable solutions are three: \( s = rp; r = s/p, \text{ and } p = s/r \).

Functional models similar to the one just considered in example 5.10 allow the description of an entire complexus of procedures related to the determination of some descriptors through others. For this, it is sufficient to only specify the initial set of descriptors and the goal descriptors and to find, with the aid of the planner, such allowable solutions for the specifiers which would form an oriented path, leading from the initial descriptors to the goal ones. The search for and a path (at least one of the possible ones) constitute the fundamental problem of the planner.
Example 5.11. Assume that the task to find the area of a triangle, if its sides $a$, $b$, and $c$ are given has been inputted to the entry of the planning system. According to the text of the task, the planner excites in the functional model the descriptors that correspond to $a$, $b$, and $c$ and begins the search for a path leading to the goal descriptor $S$. The search proceeds in accordance with one of the strategies about which we spoke in section §5.2. Assume, for definitiveness, that this shall be the strategy of the direct wave. Then the process of search of the plan will proceed as is shown in Figure 5.8. At the first step, the planner becomes convinced that the allowable solution is provided only by the specifier marked by number 1 on the figure. A solution for it gives the possibility to find the descriptor $P$, if values for $a$, $b$, and $c$ are known. Next, the solution for the specifier marked by the number 2 becomes possible, thus allowing a determination of the value of descriptor 2. Finally, at the third stage of planning is resolved the specifier marked by the number 3, which leads to the determination of the
value of descriptor $S$. Since the descriptor $S$ is the goal one, the planner concludes his work.

Figure 5.8

After having found the oriented path, leading from the initial descriptors to the goal ones, it is necessary to formulate a computing program. This is possible if in the memory of the control system there is an earlier prepared selection of standard program moduli, each of which corresponds to one or another allowable solutions for a defined specifier. With availability of a selection of such moduli, the path found by the planner indicates the sequence of carrying out the moduli one after another.

The described procedure of planning solves only the first part of the problem of solution search—it formulates a program, after which it is necessary to carry out the filling of the program with concrete data. At that stage it also becomes necessary to turn to the base of knowledge about the problem area, for it is only in it that the regularities are contained that allow judgment on the allowability of one or another combination of data. For example, if we utilize the functional model described in examples 5.10 and 5.11, then upon input at the entry of the planning system of the problem to find the area of a triangle, if its sides are given as $a = 100$, $b = 1$, and $c = 20$, a program will be constructed corresponding to the path found in example 5.11. Next, it will begin to be carried out for concrete values of
the sides of the triangle, specified in the task. But, the execution of this program will not be concluded successfully. An emergency stoppage will occur, because during the calculation of the radius of the inscribed circle, in accordance with the specifier marked by the number 2 on Figure 5.8, a negative expression will be obtained under the radical sign.

In order to exclude such cases, regularities are allowed for ahead of time in the model of the problem area; the carrying out of these must be obligatory in work with concrete data. In our case, in the model of the problem area, should be stored the following regularities: The sum of the lengths of two sides of a triangle is always greater than the length of the third side; the difference of the lengths of two sides of a triangle is always (less) than the length of the third side; the sum of the angles of a triangle is always equal to 180°, and so forth. These preliminary checks should expose all cases of unallowability of carrying out programs with such fixed initial data, which violate the fundamental regularities of the problem area.

The totality of the deductive conclusion assured by the planner on a functional model, the description of the model itself and its related program moduli and regularities of the given problem area along with procedures for checking them form what is now customary to call intellectual package of applied programs. It is namely as such a package that the correlator manifests itself in structural control. Its main component is a solution of logico-transformational rules:

\[ S_1; S_2 \Rightarrow S_3. \]

Here, \( S_1 \) is a description of a fragment of the current situation the presence of which defines the applicability of the logico-transformational rule; \( S_2 \) is a description of the fragment being transformed; and \( S_3 \) is the resultant description of the new fragment of description. If \( S_1, S_2, \) and \( S_3 \) are considered as descriptors, and \( \Rightarrow \) as a certain specifier, then it is easy to establish a correspondence between the functional models and the selection of logico-transformational rules. Therefore, it can be assumed that traditional methods for situational control have anticipated the idea of working with functional models.
In real problems are often found cases, when transitions between states in the state space or between subproblems in the subproblem space are undetermined, which reflect the incompleteness of our knowledge or the possibilities of such transitions. In this case, the arcs of the network, or which planning is carried out, are weighted, for example, by values of a function of affiliation. A general evaluation of the formal found plan is calculated from the estimates of these values or different arcs by various methods. In particular, a general evaluation may coincide with a minimal evaluation encountered on the given path, or may be the arithmetic mean of these evaluations.

5.5 Scenarios and Logics of Action*

In searching for decisions in systems of situational control are used not only functional models but also other methods of finding solutions. In §5.4 we had already mentioned logico-transformational rules, laying in the base of many real systems of situational control. The use of similar rules assumes that all necessary information for deriving a solution is stored in a specially organized base of knowledge (see Figure 5.6). This organization consists in breaking up the entire base of knowledge into separate, possibly overlapping, regions called spheres. Each sphere contains knowledges among which exists a close semantic, pragmatic, or situational connection. A definite name is assigned to a sphere. Each fact, stored in the memory of the control system, contains information on the names of those spheres of which it is a part. Moreover, the fact has its own, belonging only to it, name. Finally, for the parts are given lists of selections by which it is tied to the other parts, and of the types (names) of these selections.

Depending on the concrete utilized base of data, into which is immersed the base of knowledge, the formalization of the fact may be different. For example, it can be described in the form of a role frame.

Example 5.12. In creating a control system for loading and unloading railroad cars at a railroad station, it might be necessary to describe in the form of a fact the knowledge on the technology of unloading the cars. This knowledge can be represented by the following role frame:

300
\{<UNLOADING><PLACE><Value of slot><TYPE OF CAR><Value of slot> 
<TYPE OF CARGO><Value of slot><PRIORITY><Value of slot> 
<TIME OF ARRIVAL><Value of slot><RECEIVER><Value of slot>\}.

Such a role frame may have an inlaid structure. In other words, 
individual slots corresponding to roles defined by their names, may themselves 
be the names of other role frames. For instance, the Slot PLACE can itself be 
specified in the form of the following frame:

\{<PLACE FROM WHERE><Value of slot><WHERE TO><Value of slot> 
<TYPE OF CAR><Value of slot><MEANS><Value of slot>\}.

In its turn, the slot TYPE OF CAR can also have the form of a 
certain role frame.

\{<TYPE OF CAR><MEANS OF LOADING><Value of slot><MEANS OF UNLOADING> 
<Value of slot><CONDITIONS><Value of slot>\}.

In filling the slots of the cited role frames with concrete data 
from the data base, concrete frames are formed that result from those 
prototype frames. For example,

\{<UNLOADING PLACE><Platform No. 2><TYPE OF CAR><Type Δ> 
<TYPE OF CARGO><Sand><PRIORITY><First><TIME OF ARRIVAL> 
<25 December 1985><RECEIVER><Brick factory No. 68?>\};

\{<PLACE FROM WHERE><Platform No. 1><WHERE TO><Platform No. 2> 
<TYPE OF CAR><Type Δ><MEANS><Diesel locomotive 02652>\};

\{<TYPE OF CAR><MEANS OF LOADING><Any><MEANS OF UNLOADING> 
<Any><CONDITIONS><Length of platform not less than 50 m, 
height of platform not more than 0.5 m>\};

\{<SAND><MEANS OF LOADING-UNLOADING><Transporters, type 1,4,7; 
crane type 6; manual loading-unloading><LOCATION OF MECHANISMS> 
301
What do these descriptions provide? They contain in them the information that on 25 December 1985 a railroad car type 4 with sand which arrived for use at brick factory No. 68, is situated near platform No. 1. However, at that platform there are no mechanical means suitable for unloading sand. Therefore, with the aid of diesel locomotive 02652, the car can be brought up to platform No. 2, at which there are available transporters of types 1 and 4 suitable for unloading sand. In carrying out this decision (solution) it is further necessary to check the fulfillment of a condition related to the type of the car and the length of the platform No. 2 and its height. If these conditions are fulfilled, then a solution of the problem has been found. If they are not fulfilled, then it is necessary to find another variant of unloading related, for example, to the moving of the car with sand to platform No. 7.

As is seen from the cited example, two types of descriptions are stored in the system's memory: intentional and extentional. The intentional descriptions are descriptions of pure knowledges about the object of control and of the processes occurring in it. A role frame in which the roles are only named but not filled represents an example of such an intentional description. In filling out the roles, however, there originate a number of extentional descriptions—concrete facts which in their totality give a description of a current situation.

Both intentional as well as extentional descriptions may be connected among each other by various selections. In our example, for greater clearness, these selections, as well as the names of spheres into which these facts enter, were not pointed out. Most important for intentional descriptions are the classifying relations, characteristics for the classifier (i.e., the selections of the type class-element of class). These relations describe the structure of the object of control and the structure of the processes of decision-making on the control of the given object. On the other hand, the relations on the set of extentional descriptions are situative. They connect individual concrete facts among each other into a single situation.
Connecting together descriptions of two types, it is possible to obtain a scenario for decision-making.

**Example 5.13.** On Figure 5.9 is shown a fragment of a scenario for control of water usage and water consumption in a certain region. On the first level of this scenario, seven classes of situations are singled out, during the occurrence of which control is required. These classes are interpreted in the following manner: 0 - a change in the operation of the system of water usage is required because of 1.1 - state of weather conditions, 1.2 - agricultural water use effects, 1.3 - water transport effects, 1.4 - daily communal use effects, 1.5 - effects of hydroelectric structures, 1.6 - industrial effects, 1.7 - effects of the tourism industry. Next is analyzed only that part of the scenario which is related to the necessity of control as a result of requirements posed to the control system by the daily communal usage. The remaining scenario apexes on Figure 5.9 are interpreted in the following manner: 2.1 - increase in quantity of water used, 2.2 - increase in quantity of effluent water and deterioration of its

![Diagram](image-url)
quality, 3.1 - analysis of reasons of balance violation, 4.1 - prognosis of further changes, 5.1 - changes have a temporary character, 5.2 - changes have a permanent character, 6.1 - modeling of the effect of changes on the state of the environment, 6.2 - determination of the leaking quantity of water, 6.3 - modeling of the effect of change of quality of run-offs or the state of the environment, 7.1 - evaluation of the modeling results, 7.2 - analysis of the effect of water economy balance on the perspective (outlook), 8.1 - estimate of possibility of increasing water usage, 8.2 - evaluation of modeling results, 9.1 and 9.2 - determination of solutions (decisions) on the basis of the carried-out analysis.

Next follow the solutions (decisions) themselves, indicated on Figure 5.9 by numbers: 1 - Introduction of a water economy regime, 2 - increase of load or water economy facilities, 3 - putting into action of reserve powers, 4 - construction of water conduits, 5 - modernization of the equipment of available water purification facilities, 6 - construction of new facilities for purification, 7 - redistribution of run-offs and their further utilization, 8 - increase of load or purification facilities and utilization of reserve powers.

Thus, the scenario contains in itself the most heterogeneous elements related to the checking of conditions, requiring the intervention of a control system, analysis of the current situation, forecasting of its development, and the adoption proper of one or another solution (decision). But the scenario examined in our example 5.13 is only the upper level of the decision-making system. At the lower levels the solution must be defined concretely. Concretization may require several steps, including procedures of analysis, invitational modeling (extrapolation), or optimization. In a number of systems of situational control created in our country, systems of decision-making have three levels of concretization. At the uppermost level there occurs the elicitation of the conflict situation that requires making decisions on control. At the second level are divided into parts all allowable solutions. An extrapolation is carried out for them and, if possible, an optimization. Finally, at the third level the found solution scheme (intentional description of the solution scenario of the type of the role frame from example 5.12 or of a logico-transformational rule) is filled with a concrete content from a data base and is transformed into an
extensional description of the scenario of decision-making, which is realized
command by command. The transition from the second level to the third is
attended with a possible nonsingle-valuedness of the concrete scheme of
realization of an intentional description because of the nonsingle-valuedness
of the working sequence of logico-transformational rules. To prevent these
conflicts, special metarules are introduced with the aid of which are selected
allowable sequences of application of logico-transformational rules.

Let us consider one more approach related to the making of
multistep decisions, when the logic of actions is utilized on the apparatus
for making decisions. Let us consider a fragment of such a logic related to a
partial orderliness in time of actions that constitute in their totality the
realization of a certain solution. Let $d_i$ be elementary actions, for which
the control system has standard techniques of realization; $D_j$ - complex
actions consisting of chains of actions, $d_i$; $A_k$ - subjects capable of carrying
out actions $d_i$ from a certain totality of such actions. Here the role of
subjects can be played by both humans as well as various possible mechanisms
and devices. By $T_k$ we shall designate the time intervals during which subject
$A_k$ acts. Finally, $O_m$ are objects that experience the effect. Let us now
introduce the main basic triplets characteristic for the logic of actions in
time. In labeling the relations entering into them we will make use of the
relations from Table 2.1 and Table 2.2, namely:

\begin{align*}
  d_i & : r \rightarrow D_j & \text{the elementary act $d_i$ is a part of act $D_j$,} \\
  A_k & : r \rightarrow d_i & \text{subject $A_k$ carries out act $d_i$,} \\
  A_k & : r \rightarrow D_j & \text{subject $A_k$ carries out act $D_j$,} \\
  d_i & : r \rightarrow T_k & \text{act $d_i$ is realized in interval $T_k$,} \\
  D_j & : r \rightarrow T_k & \text{act $D_j$ is realized in interval $T_k$,} \\
  A_k & : r \rightarrow T_k & \text{subject $A_k$ acts in time interval $T_k$,} \\
  O_m & : r \rightarrow T_k & \text{object $O_m$ experiences effect over interval $T_k$,} \\
  d_i & : r_2 \rightarrow d_i & \text{acts $d_i$ and $d_i$ can be carried out in arbitrary order and} \\
  & & \text{without being adjacent,} \\
  d_i & : r_2 \rightarrow d_i & \text{acts $d_i$ and $d_i$ must be carried out adjacent to each other} \\
  & & \text{and in the order indicated,}
\end{align*}

305
D_jr^1D_m \ - \ acts \ D_j \ and \ D_m \ can \ be \ carried \ out \ adjacently \ in \ any 
order,
D_jr^2D_m \ - \ acts \ D_j \ and \ D_m \ must \ be \ carried \ out \ adjacently \ and \ in \ the 
order \ indicated,
d_i\gamma_iO_m - \ acts \ d_i \ and \ d_1 \ are \ carried \ out \ simultaneously,
D_jr^2D_m \ - \ acts \ D_j \ and \ D_m \ are \ carried \ out \ simultaneously,
d_i\gamma_iO_m \ - \ act \ D_i \ is \ carried \ out \ on \ object \ O_m.

Let us also mention examples of rules of deduction used in such logic of actions.

1. Each subject may carry out a certain action (d_i or D_j) only over that time interval on which it can do so. For actions d_i, this rule assumes the form

\( (A_kr_1T^*)(A_kr_2d_i)(d_i\gamma_1T**) => (T**r_2T^*) \).

2. The time during which a subject does an act on some object, and the time during which an object experiences the effect of this act (d_i or D_j) must be equal. For action d_i this rule takes on the form

\( (d_i\gamma_1T^*)(A_kr_2d_i)(d_i\gamma_1O_m)(O_m\gamma_1T**) => (T**r_2T**). \)

3. Elementary acts that enter into a complex action must have intervals of realization that are contained in the interval of realization of the complex action. For act d_i this rule has the form

\( (d_i\gamma_8D_j)(d_i\gamma_1T^*)(D_j\gamma_1T**) => (T**r_2T**). \)

4. For immediately adjacent actions (d_i or D_j) the intervals of their realizations should be in relation of adjacency. For d_i this rule has the form
\[(d_1 r_{26} d_1)(d_1 r_{17} T^*)(d_1 r_{17} T**)) \Rightarrow (T^* r_{26} T**).

5. If a certain action \((d_i \text{ or } D_j)\) is completed before another action \((d_i \text{ or } D_m)\) then the same relation exists also between the intervals on which they are realized. For \(d_i\) this rule has the form

\[(d_1 r_{22} d_1)(d_1 r_{17} T^*)(d_1 r_{17} T**)) \Rightarrow (T^* r_{22} T**).

6. If two actions \((d_i \text{ or } D_j)\) are carried out simultaneously, then their intervals of execution coincide. For \(d_i\) this rule has the form

\[(d_1 r_{21} d_1)(d_1 r_{17} T^*)(d_1 r_{17} T**)) \Rightarrow (T^* r_{21} T**).

Similarly, it is possible to construct a fragment of the logic of actions related to the spatial realization of actions. For this, it is necessary to select from Table 2.1 and Table 2.2 the corresponding spatial relationships.

Such fragments of the logic of actions can be utilized in constructing complex solutions from elementary ones and for checking the conditions of their combination (matching) in time and space.

5.6 Extrapolation of Solutions

The complexity of the objects, for which normally are used methods of situational control, requires the making of one or another decisions during the search of a solution or the control of possible consequences. The problem of imitative modeling is not related to any particular problem tied in specifically with situational control. There exist many methods of such modeling which allow extrapolation of the dynamics of the states of an object and of the processes occurring in it. Special languages of imitative modeling have been developed and program packages have been created, allowing realization and imitation.
However, in utilizing methods of the semiotic type and of those languages of situation descriptions that are used in them, special requirements arise toward modeling systems. These requirements are tied to the fact that it would be desirable to obtain in the ideal a forecast of the development of events at the level of descriptions of those situations which may occur in the future. In other words, it is desired to obtain an extrapolation in the form of an inverted tree, as shown in Figure 5.10. Its root corresponds to a situation (description of a situation) at the object of a given moment of time. If the solution is planned as $D^* = d_1d_2d_3$, then the subsequent levels of the tree show those situations into which the object can get a result of realization of that particular solution. The branching of the tree corresponds to that uncertainty with which it is possible to conceive the process of development of events. Around each situation located at the end branches of the tree, estimates $g_i$ are written that characterize the possibility of such a consequence (they are indicated by shaded circles). These estimates may have an export, a probability, or a diffused character.

If, in the initial situation, in addition to solution $D^*$, it is possible to use certain other solutions, then an imitational process is constructed for all of them, generating its own tree of the same type as in Figure 5.10. Next, following some decision rule, are evaluated the estimates $g_i$ obtained as a result of modeling and that solution $D^*$ is selected for which the decision rule gives the best result.

Figure 5.10

An important characteristic of the above cited method during modeling is that each time we have a description of the obtained situation, and hence can classify it by means of the classifier and to evaluate its conflictness or nonconflictness for controlling the object. This leads to
imitational modeling as though on two levels: at the level of the dynamics of the controlled object itself and at the level of description of operations forming on it with consideration of possibilities (for instance, resources) of the control system itself. It is in this two-step manner that the values of estimates \( g_i \) are formed.

The following peculiarity of imitative processes in systems of decision-making on during control, in the base of which lie logico-linguistic models, consists in that of interest are not only the statistical data accumulated in the process of modeling (for instance, average times of expectation or lengths of queues of objects waiting for servicing), but also the dynamics of one concrete situation, and here are varied not only (and not how much) some numerical parameters, but also structural descriptions (for example, relations realized among separate elements entering into the description of the situation). All this leads to the necessity in constructing an extrapolator for systems of situational control to utilize possibilities contained not only in traditional languages for imitative modeling, but also in languages characteristic for describing and transforming the descriptions of situations.

And, finally, one more peculiarity of the extrapolator--its operation must be closely tied with the operation of other blocks of the system of situational control and first of all with the operation of the classifier and of the system of selection of solutions. This indicates that the imitative process in systems of situational control proceeds in such a way that motion along a tree of the type shown in Figure 5.10 is being carried out with consideration of not only the object itself, but also with involvement of procedures of situation classification that come up in the course of modeling, and with utilization of procedures of solution forming based on the results of this classification.

In §2.7 we introduced the concept of objects of control in the form of a discrete situational network (DSN). Such a model allows a very simple demonstration of the essence of the process of imitative modeling in the extrapolator of the system of situational control.

Let us recall that a DSN includes in itself elements of four types: sources, risks, passive solvers, and active solvers. Along a DSN are moving, during discrete moments of time, objects, each of which is specified by a
selection of characteristics. Characteristics can be of two types: static, characterizing the essence of the given object, and dynamic, related to temporal estimates, ascribed to the object during a given concrete process that is being realized or a DSN. Each element entering into the DSN can be represented by some deterministic or probabilistic automaton. These automata can even be schematically realized, but in systems of situational control it is preferred to have their programmatically realized models.

Example 5.14. Let us consider the DSN shown in Figure 5.11. As we see, the DSN has one source, the functioning of which generates objects of two types: stock_1 and stock_2. The objects enter into the passive solver and then into a solver in which are combined the passive and the active solvers AR+R.

![Figure 5.11](image)

Finally, after that the objects get into one more passive solver and flow off. The first and the last passive solvers imitate the transporters, Transp-1 and Transp-2, feed the stocks to the solver AR+R, which imitates a machine tool, and pass the ready parts to the bunker, shown on the DSN as the sink C. The processing of the object stock_1 and stock_2 on C machine tool requires a certain time, which alters the dynamic characterization of these objects. The working on the machine tool itself changes their static characteristics.

In imitating the process of motion of the stock and their working on the machine tool we shall introduce the following processes.

1. Setting Up. Corresponding to this process is the logico-transformational rule (LTR) of the following form: If the machine tool is free and the stock_1 is on transp-1, then it is necessary to place the object stock_1 on the machine tool after taking it off the transp-1, and then switch the machine tool into the occupied position.

310
2. Switching On. The corresponding LTR has the form if the machine tool is occupied, then switch it to the switched-on state. Note that the unoccupied position has an indefinite, in time, interval of actualization. The occupied state, however, depending on the type of the object stock \( i \) \((i = 1, 2\) for our example) is actualized for a definite period of time (for example, for \( i = 1 \) for 10 minutes, and for \( i = 2 \) for 45 minutes).

3. Feeding the Stock. The corresponding LTR has the form: If on transp-1 there are not objects of the type stock \( i \), then transfer the control to the source and wait for the appearance of the first object of stock \( i \), after which the functioning of the source ceases until the vacating of transp-1.

Such conditions of operation of the LTR for the stock feed process may, of course, not be fulfilled in practice. The source may function independently of the presence or absence of stock on the transporter which supplies the parts to the machine tool.

4. Switching Off. The corresponding LTR has the form: If the counter of the astronomical time, for the occupied position, has reached a limit corresponding to the object stock \( i \), then switch the machine tool into the free position and place the ready part onto the transporter transp-2.

5. Feeding the Part. The corresponding LTR has the form: If the part is on transp-2, then switch it on and, after a specified astronomical time, switch it off.

This rule corresponds to the transposition of the part from the machine tool to the exit bunker (sink of the DSN).

The above-described five LTR contain in them the conditions of applicability, the fulfillment of which permits changes in the situation, described in these LTR. The conditions of applicability are introduced by the marker If, and the operators of transformation—by the marker then.

The above-cited very simple example shows that the description of a current situation and the description of the LTR for a slightly complex real object will be extremely cumbersome, while the work itself on the formulation of the needed list of LTR represents a very uneasy task. In real objects there will be moving along the DSN many very multitype objects, and the LTR
for their processing in active solvers will be numerous and interconnected by the conditions of their applicability.

Example 5.16. For illustrating the above, we will cite as an example Table 5.2, in which are enumerated types and numbers of objects moving along a DSN and forming its active solvers, for the Kalinnigred Sea fishing port.

Here we do not separate the objects and the active solvers, because depending on desire they can be transformed one into the other. An analysis of this table shows how complex may turn out to be the LTR system which takes into account all kinds of objects, resources, and process-synchronizing parameters. An estimate of the number of LTR, characteristic for modeling systems of such objects, lies within the bounds of 200-400. The number of decisions adopted in processing ships in a port of the sea-fishing port type is also very large. In practice, each such decision (solution) is a multistep one. The following fundamental operations are parts of it: loading, unloading, trans-shipment, towing; dislocational operations; positioning by the first board, positioning by the second board, pulling over, manipulating, mooring, remooring; filling operations: fuel, oil, fuel-oil, water, settling, conveying; auxiliary operations: carbonation, deactivation, disinfection, surface conditioning, washing and drying, packaging, ship repair, cleaning, fire prevention, diver operations, passenger operations. For cargoes, there is also an entire spectrum of operations. A number of operations (for example, loading or towing) themselves consisting of a number of simpler operations. These operations are interconnected by the scenarios that determine the necessary and allowable sequences of execution of individual operations. As an example may serve the scenario shown on Figure 5.12.

![Figure 5.12](image)
<table>
<thead>
<tr>
<th>Objects and Active Solvers</th>
<th>Number Encountered in the Framework of a Single Situation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pier Cranes</td>
<td>2</td>
</tr>
<tr>
<td>Gantry Cranes</td>
<td>36</td>
</tr>
<tr>
<td>Moorings</td>
<td></td>
</tr>
<tr>
<td>Industrial Harbor</td>
<td>8</td>
</tr>
<tr>
<td>Lumber Harbor</td>
<td>7</td>
</tr>
<tr>
<td>Oil Base</td>
<td>2</td>
</tr>
<tr>
<td>Integrated Works</td>
<td>1</td>
</tr>
<tr>
<td>Gasification or Aeration</td>
<td>1</td>
</tr>
<tr>
<td>Electric Loaders</td>
<td>115</td>
</tr>
<tr>
<td>Automotive Loaders</td>
<td>28</td>
</tr>
<tr>
<td>Ships of the Harbor Service</td>
<td>45</td>
</tr>
<tr>
<td>Refrigerator</td>
<td>3</td>
</tr>
<tr>
<td>Covered Storage</td>
<td>13</td>
</tr>
<tr>
<td>Open Storage</td>
<td>30</td>
</tr>
<tr>
<td>Railroad Track</td>
<td>7</td>
</tr>
<tr>
<td>Workers</td>
<td>2459</td>
</tr>
<tr>
<td>Brigades</td>
<td>32</td>
</tr>
<tr>
<td>Ships of the Kaliningrad Registry</td>
<td>more than 400</td>
</tr>
<tr>
<td>Types of Ship Winches</td>
<td>25</td>
</tr>
<tr>
<td>Types of Twin Decks</td>
<td>126</td>
</tr>
<tr>
<td>Types of Cargo Spaces</td>
<td>63</td>
</tr>
<tr>
<td>Types of Railroad Cars</td>
<td>17</td>
</tr>
<tr>
<td>Assortment of Cargoes</td>
<td>about 500</td>
</tr>
<tr>
<td>Types of Automotive Vehicles</td>
<td>7</td>
</tr>
</tbody>
</table>
Here, as always, the relation $z_{21}$ is the relation simultaneously, while the relation $z_{22}$—to be earlier. Numbers denote the following operations: 1—transmittal of the cargo plan, the date of arrival, 2—preparation for the processing of the ship, 3—transference of the ship into the port, 4—familiarization of arrival, 5—processing of the ship, 6—formalization of the cargo documents and of departure. Of course, each apex of the scenario is itself the name of some scenario, and such a nesting of scenarios can be sufficiently deep. As parameters that change the form of the scenarios or their sequence, act the concrete characteristics of the types of ships, moorings, and other elements enumerated in Table 5.2.

Besides solutions which are connected with the consideration of the totality of scenarios defining the technological requirements toward the realization of solutions and entering, as a rule, into conditions of applicability of some or other LTR, there are also imitated and adopted solutions determined only by those control criteria which the control system itself is guided by. These criteria, as a rule, are contradictory, because they touch upon the interests of different administrations. The plan of unloading and of cargo shipment is always born as a result of a certain compromise, evaluated in the process of imitation of the consequences of the adopted decisions. Behind the simplicity of the final solution, as illustrated for example by Figure 5.13, there is concealed the realization of a very cumbersome procedure of imitation and decision-making.

![Diagram](image)

**FIGURE 5.13**

314
5.7 Peculiarities of Reasoning in a Human

In the process of preliminary filling of the memory of a control system of semiotic type and in the process of teaching, when the technologist-controller forms for the system the fundamental strategies of control, and also the teaching samples, he involuntarily introduces into this information those peculiarities of his thinking which are inherent in him as a human.

In the present paragraph we will try to enumerate the methods of thinking readily utilized by man in the course of his reasonings and not fitting into the traditional framework of logic. The purpose of the present section is to alert the specialist on control, who frequently makes light of that level of strictness which in reality characterizes the position and the pronouncements of the expert-controller. A second purpose of what is expounded below—is to offer a selection of some recommendations on the construction of systems which mistake human methods of reasoning. The necessity in this is called for not specifically by problems of imitation of human abilities by artificial systems, but by the fact that in systems of communication between a human and an intellectual system it is necessary to create a level of mutual understanding and trust occurring during the communication of two people.

Subsequent exposition will have a concise (recapitulative) character. A more extensive consideration of some or other problems connected with the methods of reasoning of man, can be found in books and articles discussed in the commentaries to the present chapter. The order of discussion of the peculiarities of reasoning and of the conclusions made by man on the basis of these considerations, reflects that significance which can today be ascribed to them in solving management problems.

1. Scaleability of Deductions. In §4.2 we spoke much about scales forming the space in which is realized the classification of objects and phenomena of the surrounding world in a human. Understandably, his reasonings and conclusions made by him cannot ignore these scales. However, the scaling of the conclusions occurs, as a rule, sort of outside their verbal expression, remain outside the sequence, although it is recognized in a necessity. But sometimes such scaling manifests itself in an explicit form. Let us have two
scales: a scale of causes and a scale of consequences. On each scale is defined a certain relation of order. Let us designate by $\alpha_1 < \alpha_2$ and by $\beta_1 < \beta_2$ that consequences $\beta_1$ is less than cause $\beta_2$. Assume it to be known that the conclusion $\alpha_1 \Rightarrow \beta_1$ is justified. What can be said about the conclusion $\alpha_2 \Rightarrow \beta_2$, for which it is known that $\alpha_1 < \alpha_2$? We are interested in the case when from the information given it is possible to make either the conclusion $\beta_1 < \beta_2$ or the conclusion $\beta_2 < \beta_1$. In the first case we will say that the scale of causes and the scale of consequences are coordinated, and in the second that they are inversely coordinated. It appears that for man these two cases of coordination of the scales of causes and consequences occur practically always.

Let us consider the following example. Let scale $\alpha$ be the scale of criminal offenses. Then the relation less than may be interpreted as to be lighter from the standpoint of inflicted damage. Let scale $\beta$ be the scale of punishments determined for crimes. Clearly, the scales $\alpha$ and $\beta$ are coordinated. For a graver crime, a graver punishment is prescribed. If scale $\beta$ is taken as a scale on which is marked the probability of correction of the criminal, then it is possible to assume that it is inversely coordinated with scale $\alpha$.

2. Incorrect Use of Causes. In conclusions of the type $\alpha \Rightarrow \beta$ in man's reasonings, cases are quite often formed when their realization is modified by certain conditions. In other words, the conclusion has the form $\gamma: \alpha \Rightarrow \beta$, where the predicate $\gamma$ characterizes the possibility of the conclusion $\alpha \Rightarrow \beta$. However, very often $\gamma$ is forgotten and the conclusion $\alpha \Rightarrow \beta$ is made according to tradition. That is the layer of knowledge which it is customary to call parental. Such knowledge is absorbed by man uncritically without an evaluation of its reliability. In primitive reasoning, in myths, in rituals, such conclusions are used everywhere. And in our days' expressions like It is customary to consider that... or In accordance with the established tradition..., and so forth, demonstrate that some method of reasoning. But frequently such conclusions turn out to be false because of the falseness of the predicate $\gamma$ in a given situation.

Another example of an incorrect use of causes is the case when the childish layer of knowledge participates in a conclusion. Conclusions based
on such knowledge, as a rule, base themselves on reasonings of the type I desire \( \beta \) but in order for \( \beta \) to be, \( \alpha \) must be. Let \( \alpha \) be then.\(^*\) Next follows the conclusion \( \alpha \Rightarrow \beta \), although \( \alpha \) may not even exist.

Finally, very often as cause \( \beta \) is selected not that true reason which causes \( \beta \), but some quasi-reason that lies on the surface. This is the most frequently encountered human error. It is precisely because of it that Napoleon thought that steamships could not float: They must sink, because the iron from which they are made sinks in water.

3. Personal Disinterest in Consequences. Very often in the conclusion \( \alpha \Rightarrow \beta \) man is rather indifferent as to just what it is that is being concluded from \( \alpha \). Such a situation is in a certain cause juxtaposed to reasonings based on the childish layer of knowledge. Often in reply to the question What follows from this? or Why is this done?, there follows the answer I don't know. Rationalism that is usually ascribed to human thinking and to human reasonings is in fact not encountered so often. For this reason many conclusions \( \beta \), which follow from \( \alpha \), have actually a chance and nonpragmatic character.

4. Traductivity of Conclusions. In human practice the conclusion is most often a traductive one (i.e., a conclusion from particular to particular). In child reasoning, the percentage of such conclusions is close to 100. As their examples may serve reasonings of the type This uncle is bad; he has a rough voice. or This thing is red; that means it tastes good. But conclusions of this type are also found very often in reasonings of grownup people.

5. Subjective Utilization of Quantifiers. Quantifiers are always subjective and reflect the personal experiences of a person using them in reasonings. We spoke about this already when discussing quantifiers. Here, we will just point out a somewhat different aspect of this subjectivity, which plays an important role in evaluating the information obtained from experts-managers. In their utilization of quantified reasonings it is always

\* Conclusions of this type belong to the mathematical folklore.
necessary to remember that the evaluation of a quantifier is enormously affected by the personal gain aspect of the user. While, for instance, many people purchase tickets of a money-goods lottery or a sports-lotto, there would remain few takers if their conditions were altered as follows: With that probability, close to zero, with which the grand prize is won, the winner of that ticket would be subjected to a very severe penalty, while in all cases corresponding to loss, a small win were paid. Because, when the consequences for a person may be severe, the probability of the risk increases subjectively. Roughly speaking, with increase of possible unpleasantnesses the a priori estimates of the subjective probability of expectation of this event grows. The traducive conclusions with and quantifiers is a common phenomenon.

6. Scenario Conclusion. In human practice are often found conclusions based on certain temporal series, causal scenarios, or scenarios of other types. An example of such conclusions can serve The electric train is approaching a station, the platform is deserted, many people are crowded by the car doors. This means that there will be many people on the platform. The lecturer entered the auditorium and started to write formulas on the board. Hence, the students are writing in the abstracts that which is written on the board. We note that in such reasonings great importance is played by role structures about which is spoken in §2.2. In particular, it is precisely the knowledge of the role structure that explains such a conclusion: The man was chopping the wood with an axe; the axe was sharp. Hence, it was easy to chop.

7. Paradoxicality and Absurdity of Reasonings. This property of human reasonings is less significant for us than the others. Hence, we confine ourselves only with examples that show that a human in a number of cases commits a violation of even fundamental laws of logic. Most often this occurs in proverbs and sayings. Thus, in the well-known saying He didn't burn it, but scorched it or Not half of forty, but twenty is violated the logical law of identity. In the sayings He would have been a clever man had he not been a fool or Ulyss is full when she doesn't want to eat from the standpoint of logic shouldn't be contained any information, for it is asserted in them
that \( A = A \). For us are especially interesting the quasiconclusions based on the fact that with a false premise it is possible to describe the implication any consequence. If you find a man on a cow, then a mane on a cow, then a mane will have horns. An interesting modification of such conclusions is the saying The gelding is bay, but there is no fur on it and the beautiful text of D. Harms constructed on the same idea There was a redheaded man who didn't have any eyes and ears. He didn't have any hair either, so that he was called redheaded conditionally.
APPENDIX

HISTORICAL SKETCH OF THE DEVELOPMENT OF SITUATIONAL CONTROL

(pp. 254-258 of the book)

Cart after cart, the wagon train is huge,
And it's a wonder to behold
That which in one day, just one day
Became extinct and antiquated
They cart like rotten wooden logs
Like junk, like rusty metal scrap,
Erroneous representations
and piles of imaginary axioms.

L. Martynov

In the early 1960's bionics was in vogue. Its main purpose -- to use that which nature has realized in plants and in animals for creating technical devices -- seemed very attractive. Conferences and symposia on bionics were convened. Biologists, engineers, mathematicians and cyberneticists learned to listen to and to understand each other. Formation of specialists went on who would be capable to work at the juncture of very diversified scientific disciplines.

And against this background there naturally arose the question of cooperation of cyberneticists and psychologists. To utilize the knowledge accumulated by the psychologists in the creation of intellectual systems was a no less interesting goal when compared with the purpose of bionics. In analogy with bionics the new direction was labeled psychonics. This appellation turned out to be not very successful and did not take root. But a seminar with that name functioned for about 10 years. It was organized at the Moscow Energetics Institute. Its main membership, the composition of which

320
varied with the passage of time, were specialists working at MEI and the
Institute of General and Pedagogical Psychology. The constant leader of this
seminar was the author of this book.

It was precisely at that seminar where for the first time were
formulated the principles of the modeling method of problem solution by man.
The renunciation of the theory of labyrinth reasoning prevailing in
psychology, which we talked about in Chapter 1 of our book, the transition to
a theory of thinking where was dominant the procedure of constructing a
labyrinth that lead often a search to a solution, was repeatedly declared in
the addresses of V.N. Pushkin and his students. Ideas of structurization of
the initial description of a problem and the interconnections of this
structure with a structure of the aimed-at situation were understandable to
cyberneticists. Instead of heuristic procedures of sorting, imitating a
search in the labyrinth of possibilities, the new approach required the
creation of procedures based on work with structurized descriptions. And this
in turn required the creation of new models of representation of the objects
of control and an elaboration of special languages for describing the
situations being formed at the object of control and in the systems of
controlling it. These two problems received their solution in the first
dissertations defended in the area of situational control in 1967. In the
work of Zh. Zhelezov was developed a theory of discrete situational networks
which served as a good model of objects of control for many subsequent
developments on situational control. Discrete situational networks were
described in Chapter 2 of the present book. Another graduate student of the
author of the present book, Yu.I. Klykov, basing himself on the then known
language of RX-codes (brief data on it are given in Chapter 2 of this book),
had developed a special model language, named by him later as the "language of
syntagmatic chains". This language is described in Chapter 2. For many years
it became the basic language of describing situations and decision making in
situational control.

The results of 1967 laid a firm foundation for the concept of model
control, actively developed by V.N. Pushkin and D.A. Pospelov. The appearance
in 1972 of their joint monograph "Thinking and Automata" completed the initial
stage of the development of the new approach.
The term "situational control" itself did not become formed suddenly. In the studies of the initial period the new approach to control was first called the modeling approach. Next, the term "situational model" was formed. And only with the appearance in 1971 of the article of D.A. Pospelov "Principles of Situational Control" (Izvestiya Akademii Nauka SSSR, series Technical Cybernetics, No. 2) did this term displace all the others. Simultaneously it became clear that the principles of situational control (described above) are not necessary and universal for the methods utilizing the logic-linguistic models. This served as an impetus for the origination of a new term: "semiotic control" (sometimes referred to as "semiotic modeling").

For the first time this term was used in the title of a seminar organized at the Moscow House of Scientific-technical Propaganda in 1971. The seminar, the leaders of which were L.T. Kuzin and D.A. Pospelov, had the title "Semiotic Methods of Control in Large Systems". In the introductory report of the seminar organizers "Problems of Semiotic Control" for the first time was formulated the statement that the use of logical and linguistic means in problems of control, which are considered in situational control, and problems arising in other fields, have much in common. In other words, the ideas used in situational control have actually a more general character.

A result of such a view on logical-linguistic models was the origination of two new scientific directions, different from situational control, but utilizing in their studies many ideas first originated in the midst of specialists in the area of situational control. One of these directions is the "characterizational control", actively developed by V.A. Gorbakov and his students. They created some principally new methods of solving traditional problems of synthesis of discrete control systems, based on the principle of "semantic equivalency", structured descriptions of schemes and demands for their functioning.

At the Moscow Engineering Physics Institute, under the guidance of L.T. Kuzin, began to be actively developed ideas of control in ACS* of different nature, based on logical-linguistic models different from situational control. Such studies naturally lead to the solution of problems

---

* Automated Control Systems

322
of building intellectual banks and languages of representation of knowledge and their manipulation.

By 1971 there was formed a body of investigators who saw in the method of situational control a means of overcoming those difficulties which they encountered in designing systems of control of complex objects. This body was not too big, for the unfamiliarity of the new approach based on other ideas, sharply different from traditional methods of automatic control, created a psychological barrier, which can scarcely be sensed now. But at that time there were many more critics than followers. In 1970, L.S. Zagadskaya defended a dissertation in which the new method served as a basis for the system for controlling cargo handling operation in a sea port. This program little resembled the simple programs of 1967 and 1968, in which the method of situational control was used for directing the movement of ships in sluice-lock portions of canals. In the program system of L.S. Zagadskaya there were already present all the basic blocks of situational control: analyzer, classifier, correlator and extrapolator.

There began to form local territorial schools in the area of situational control. In Odessa the school was headed by L.S. Zagadskaya, in Grozny - by L.A. Afonin, in Kaliningrad - by V.F. Ponomarev. Somewhat later appeared groups of specialists on situational control also in other cities: Riga (under the direction of Ya.A. Gelfandbeyn), Ustinov (under the direction of A.Yu. Leviatov), Baku (under the direction of R.E. Gadzhiyev). These bodies were aimed at utilizing the method of situational control for solving real problems posed before them. In Odessa and Kaliningrad this was the problem of operational dispatcher control of loading/unloading operations in a seaport, in Grozny - a problem of operational control of moving around among bore holes of special installations for carrying out plugging operations and of boring installations among points of boring, in Riga was being solved a complex of problems related to ACS in civil action, in Ustinov - operational control of production areas at instrument building facilities, in Baku - operational diagnostics of illnesses and prescribed therapy. Such a wide spectrum of applications naturally required in each concrete case to develop and modify those methods and models of situational control which were born in the initial body of investigators.
The participants of these studies were full of enthusiasm and optimism. It appeared to them that the principles of situational control would be realized in the systems created by them easily and effectively. And it is not their fault that the matter turned out to be considerably more complex than they thought. The then existing native ECM* in their technical possibilities and mathematical support did in no way come up to the requirements which were posed to them by methods of situational control. This did not become obvious at once, either. It was necessary to experience through practice what it was that was concealed behind this new approach which they were using. For this reason the first program systems remained purely experimental.

In the same 1971, so important for situational control, the resolution was adopted on conducting All-Union symposia on situational control of large systems. The first such symposium took place in Odessa in 1972. From that time on Odessa became for a long time the traditional place of conducting symposia; the latest (sixth by count) occurred in 1981.

In the plenary report, made at the first symposium by Yu.I. Klykov, D.A. Pospelov and V.N. Pushkin, which was titled "Why is Situational Control Needed?" its authors tried for the first time to clearly indicate the place of situational control among other methods of control of complex objects. Replying to the question posed in the title of the report, the authors formulated the thesis that situational control, in distinction from other methods of control, solves not only the problem of control on the basis of a certain already known model of the controlled object, but also builds that very same model. The construction of the model of the object of control, the construction of the procedure of controlling it, and the search for goal-oriented solutions on control are inseparably connected in the method of situational control. And it is precisely this characteristic of it that allows to hope for its applicability in those cases, when the complexity of the object of control does not permit to build its formal mathematical model and to pose the problem of control in the traditional spirit.

After the first symposium followed two more, in 1973 and 1974. At the second symposium the central problem discussed was the problem of

* Electronic Computing Machines
construction of a multi-level Classifier. The idea of a "layered cake" that originated back in the model of a gyromat (a device capable of tuning its structure to the structure of the problem being solved), first proposed at a seminar on psychonics in the mid-1960s, received at that symposium support and approval on the part of its participants. There were proposed several principles of formulating such a Classifier, described in the pages of the present book.

At the third symposium at the center of attention were problems related to the creation of mathematical support for systems of situational control. Precisely this question was then the most urgent, because on its solution depended the very possibility of applying the ideas of situational control in control systems of real objects. The participants of the discussions on these problems pointed out the necessity of using languages capable of working with data having a complex internal structure and oriented toward such progressive for the time ECM as the BESM-6 and YeSRYAD. Not a few suggestions were voiced, many of which were realized considerably later in languages of representation of knowledge of the type of relational and frame languages.

The necessity in expanding the sphere of application of the method of situational control, the transmission to other organizations of the methods and programs created by that time, posed the problem of development of special typical techniques. This work, actively conducted in the early 1970s, was crowned with success. There were created the first typical techniques for the design of situational models of control (L.S. Zagadskaya, O.V. Sokolova, 1973), formation of the model of describing the object of control in the language of syntagmatic chains (V.F. Ponomarev, A.V. Kolesnikov, 1973), the development of program support for models of situational control (O.V. Sokolova, L.S. Zagadskaya, Yu.I. Klykov, 1973). These first techniques, in spite of their imperfection, played a definite role in the development of the method and its extension to practical program models.

In 1974 appeared the first collection of articles on situational control, after which followed three more collections (1975, 1977, and 1980).
All the collections were published by the Scientific Council on the Complex Problem "Cybernetics" under the Presidium of the ASUSSR.

For coordination of all studies on situational control and utilization of it for solving practical problems the project "Situation" was established, which was organizationally included in the Scientific Council on the problem "Artificial Intelligence" of the Committee on System Analysis under the Presidium of the ASUSSR. Starting in 1975, the working council of this project has coordinated all studies in the area of situational control, carried out in the USSR.

In 1974, appeared the first monograph on situational control (Klykov, Yu.I. Situational Control of Large Systems - M. Energiya, 1974), which played along with collections on situational control an important role in the popularization and dissemination of the method.

In 1975 the method of situational control in its main parts was already formed, there appeared a stable, generally accepted terminology, was created a complex of basic methods and procedures. Beginning from that year the study groups, in which was being used the new approach, began to transition from experimental systems and model problems toward the construction of real systems of control, based on the ideas of situational control. This required the coordination of efforts of a number of study groups and an exchange of the experience accumulated during the creation of real systems. Thus was born the idea on conducting conferences, devoted to the "technology" of systems of situational control. Altogether were conducted five such conferences (Groznyy, 1975, Ustinov, 1977, Kaliningrad, 1979, Kuybyshev, 1982, Kaunas, 1982).

The number of organizations interested in applying the method of situational control began to grow rapidly. For popularizing the method are being carried out study schools, at which are read lectures on all basic problems, connected with situational control and of its use in developing various real systems (Moscow, 1974, Baku, 1975, Tashkent, 1977, Moscow, 1979, Kuybyshev, 1982). With the same purpose the publishing house Znaniye in 1975 published the popular book of D.A. Pospelov "Large Systems, Situational Control".

* Academy of Sciences, USSR
By that time it became clear that aside from traditional problems of dispatching and operational control of technical and organizational systems, the situational control could be used even in those cases, when traditional approaches do not yield a solution because of the dimensionality of the solved problem. In those cases, instead of an exact solution, the obtainment of which becomes practically impossible, the method of situational control allows to obtain approximate solutions, acceptable in their quality. The first examples of such problems were the problem on the distribution of programs in complexes of ECM (D. Boyev, 1974) and the problem on the cutting of the graph with weighted edges into components with a minimization of the summary weight of the edges that fell on the section line (N. Georgiyeva, 1977). Both problems are well known in discrete mathematics. For them were suggested many methods of solutions. But because of the time wasted for the search for their solutions, these methods are not useful in operational systems, whose problems include a segmentation of the programs and a distribution of the segments onto parallel working ECM. The use of the methods of situational control allowed to create procedures for the solution of the indicated problems executable in the real time scale.

In the mid-1970s Yu.I. Klykov began to actively use the method of situational control during the modeling of the process of teaching humans. With time under his guidance there originated a school, in which there began to develop methods of modeling the learning activity of man and methods of controlling it.

A characteristic feature of the end of the 1970s became the essential expansion of the understanding of situational control. Factually situational control began to be considered from unified positions of semiotic modeling and of control. The reason for this was the impetuous development in the theory of artificial intelligence of that area which carries the name "representation of knowledge". Situational control had anticipated by ten years the development of that area, being first to begin work with structured information. In situational control were created the first models of representation of knowledge and of languages of representation and of manipulation of knowledge. For this reason new trends in the area of artificial intelligence and the active development of the theory of semantic networks and frame concepts were adopted by study groups which had collected a
large experience in the area of situational control "on the run" and allowed
to adopt the new ideas and to use them in their development. Equally painless
occurred the transition to programming languages of the new type (LISP and its
developments FRL). In this plan, the specialists who mastered the principles
of situational control, found themselves at the forefronts of studies in
artificial intelligence.

Beginning in 1980, it is practically impossible to separate the
development of the situational control proper from the development of semiotic
controls which organically included in itself its historical predecessor. And
most symbolic sound the titles of the plenary reports made at the VI World
Symposium on Situational Control of Large Systems (Odessa, 1981) "Situational
Analysis" (L.S. Zagadskaya), "Situational Programming" (V.S. Lozovskiy),
"Scheme of Outputs in Semiotic Models" (D.A. Pospelov).

The logic of science development is harsh and does not submit to
anyone's will. Situational control has played the role of a catalyst, having
evoked in life new ideas and principles of constructing models of the reality
and activity of man, that have so widely developed in the wide flow of studies
on artificial intelligence. But one wants to believe that in the theory of
semiotic models that is being born before our eyes, the results obtained in
the depths of situational control, shall occupy their proper place.
SITUATIONAL CONTROL
Theory and Practice

BIBLIOGRAPHY

Chapter 1.


Chapter 2.


Chapter 3.


Chapter 4.


338


Chapter 5.


