KYBERNETIKA — VOLUME 13 (1977), NUMBER 6

Incidental and State-Dependent Phenomena in Robot Problem Solving*)

OLGA ŠTĚPÁNKOVÁ, IVAN M. HAVEL

An attempt at a unifying, formal, approach to robot problem solving is outlined. Various new ideas are presented and analyzed. Important among them are the concept of incidental phenomena, which enables us to cope with side-effects of the robot's actions, and an approach dealing with objects identifiable only in certain specific situations.

1. INTRODUCTION

After years of development of specific methods and programs in artificial intelligence it is worthwhile to assume a more theoretical viewpoint and try to use mathematical tools for unifying, generalizing and comparing the work that has been up to now. Such attempts may eventually result in a "mathematical theory of artificial intelligence" which would be able to formulate and prove "metatheorems" about various methods used in AI, as well as to yield "metaheuristics" helping to discover good ideas for new methods.

The aim of our paper is to discuss a particular attempt for a unifying formal approach to robot problem solving, especially to methods based on predicate-calculus representations and reasoning. We have proved some metatheorems elsewhere [11] and in the present paper we demonstrate how our formal approach may suggest certain new ideas and extensions.

The main contribution of this paper are neither new results nor simplified proofs. Instead of that we have concentrated on describing some ideas and giving them a sound intuitive background. We hope that the reader, faced with these ideas, may be moved to expand the theoretical development as well as to launch new experimental research.

*) This paper is an extended and modified version of a contribution presented at the Summer Conference on Artificial Intelligence and Simulated Behaviour, University of Edinburgh, July 12—14, 1976.

Conceptually this paper is a successor of [11], which is a rather extensive paper with many technical details. Therefore we have decided to include here also an exposition of the main concepts developed there (the image space, branching plans etc.). This makes the present paper less dependent and, we hope, easier to read.

In general, by a *robot* we mean a computer-controlled integrated system capable of autonomous interaction with real environment accoding to general goals given by a man. This entails three important cognitive faculties: (1) the ability to perceive the environment, (2) the ability to maintain and update an internal representation of the environment, and (3) the ability to plan its own behaviour using the internal representation as a basis for imagination.

We shall be concerned with the third ability, which consists, in essence, in converting the knowledge of a current and a goal states of the environment into a plan how to achieve the goal using a finite number of elementary actions. This forms the first, *planning stage* of the robot's activity. The second stage is then the *execution* of the plan in the actual environment.

The study of efficient procedures that start with an initial state and a goal state descriptions (i.e., a problem specification), and end with an appropriate plan (a solution to the problem) is the main subject of the theory of problem solving (Nilsson's book [7] is a good introduction to this theory). One of the simplest formal approaches to this study is based on the concept of a *state space*: a set of *states*, representing concrete instantaneous situations in the world (e.g. configurations of objects), and a set of operators, partial functions mapping states into states and representing the changes of the world caused by the robot's actions. Technically the state space can be identified with an oriented graph with edges labeled by operator names. A *problem* is given by an *initial state* and a set of *goal states*. The sequence of operators spelled out by this path forms a *plan* for the problem in question. (We use the term 'operator' ambiguously both for a partial function on the set of states and for the symbolic name of such function which has the role of an instruction in the plan. By 'executing an operator' we actually mean performing the corresponding action in the real world.)

2. THE IMAGE SPACE

The state-space representation is useful in rather restricted cases when a complete description of any instantaneous situation in the problem world is feasible (it is, e.g., the case of some games and puzzles). However, in the case of a general environment it is necessary to restrict ourselves only to partial descriptions, which we call *images* (in STRIPS [1] they are called 'world models"; we have reasons for avoiding the term 'model' in this sense). Formally an image is a formalized theory (or just a set of well-formed formulas, its *axioms*) in a first-order predicate calculus with a fixed language. Our formal structure for robot world representation and problem solving

will be thus based on a generalization of the state space, called the *image space*.

It is convenient to take apart axioms that represent state-independent facts and to group them into a single theory T_I, called the *core* (of the image space I). A particular image T is then obtained by adding a *specific axiom*, say F, to T_I; we write

$$T = T_\mathrm{I}[F] = T_\mathrm{I} \cup \{F\}\ .^*)$$

The image space is defined implicitly as a pair

$$\mathrm{I} = \langle T_\mathrm{I}, \Phi_\mathrm{I} \rangle$$

where T_I is the core of I and Φ_I is a set of *operator schemata* of I. An operator schema $\varphi \in \Phi_\mathrm{I}$ may be understood just as a name of one of robot's capabilities, e.g. "*push*", "*goto*", etc. With each $\varphi \in \Phi_\mathrm{I}$ we associate a pair $\langle \mathrm{C}_\varphi, \mathrm{R}_\varphi \rangle$ of formulas called the *condition of* φ and the *result of* φ, resp. The free variables in C_φ and R_φ (if there are any) are called the *parameters of* φ. We write φ also in the form $\varphi[x_1, \ldots, x_m]$ to exhibit the parameters $x_1, \ldots, x_m$ of φ. We then obtain *operators* of I as *instances of operator schemata* by considering a variable-free term (in particular a constant) instead of each parameter and making the corresponding substitutions in C_φ and R_φ. If $\varphi \in \Phi_\mathrm{I}$ and ψ is an instance of φ, we denote by C_ψ and R_ψ the obtained instances of C_φ and R_φ, resp. The operators represent particular actions of the robot, e.g. "push box a from room 1 into room 2".

The application of an operator ψ can be explained as follows: if $T \vdash \mathrm{C}_\psi$ (C_ψ is provable in T) for an image T then ψ is *applicable in* T and yields a new image $T_\mathrm{I}[\mathrm{R}_\psi]$. Thus the problem solver can in its imagination wander through the image space until it finds a path (a sequence of operators) that solves the problem, i.e. leads from an initial to a goal image. Formally a *problem* is specified by a pair $\langle X, Y \rangle$ of variable-free formulas; the initial image is $T_\mathrm{I}[X]$ and a goal image is any extension T of T_I such that $T \vdash Y$.

3. OPERATORS WITH INCIDENTAL PHENOMENA

The image space in its "pure" form described above is suitable and more or less sufficient for theoretical investigation (cf. [11]). It is also relevant to some questions suggested by Simon ([8], p. 415). However, from the point of view of practical applications, it is not very satisfactory for two main reasons. First, the specific knowledge comprised in a new image depends only on the operator result and all other specific facts known in the previous image are lost even if they may not be influenced by the operator (this leads to the well-known "frame problem"). Second, all known side effects of any operator ψ, whether relevant to the problem in question or not, have to be permanently included in the formula R_ψ, and moreover, if these side

*) If convenient we treat a finite set of formulas as a single formula — their conjunction.

effects have their own extra conditions, the operator has to be split, in general, into several new operators with distinct conditions C_ψ.

These drawbacks can be avoided by associating with each operator schema $\varphi \in \Phi_I$ a special well-defined (syntactical) procedure that generates – or recognizes – certain pairs of formulas $\langle A, B\rangle$, called the *incidental phenomena of* φ. The formulas A (the *antecedent*) and B (the *consequent*) are assumed to have no free variables except possibly the parameters of φ and if an operator ψ is an instance of φ we obtain the incidental phenomena of ψ by substituting the corresponding variable-free terms for these parameters in incidental phenomena of φ.

The application of an operator ψ is now defined as follows. If

$$T \vdash C_\psi \,\&\, A$$

for a given image T and an incidental phenomenon $\langle A, B\rangle$ of ψ, then the application of ψ yields a new image

$$T_I[R_\psi \,\&\, B]\,.$$

Any incidental phenomenon $\langle A, B\rangle$ represents a possible side effect changing the world (if A an B are distinct), or an element of the "frame" (if A equals to B)*), and can – but may not – be taken into consideration during the plan formation according to the problem solver's knowledge of the particular problem and his anticipation of useful facts for later stages of planning.

The collection (let us denote it by Inc_φ) of all incidental phenomena for a given operator (schema) φ, in practice the procedure for their recognition, depends on the nature of φ, on the environment, and on the expected class of problems. Typically it is specified by certain syntactical restrictions on participating formulas (cf. Example 1 below). To allow using more incidental phenomena at the same time it is useful to require that for any $\langle A, B\rangle$ and $\langle A', B'\rangle \in \mathrm{Inc}_\varphi$ we have also $\langle A \,\&\, A', B \,\&\, B'\rangle \in$ $\in \mathrm{Inc}_\varphi$.

Let $\langle X, Y\rangle$ be a problem in I. We define the (*straight-line*) *plan* for $\langle X, Y\rangle$ as a sequence

$$\gamma = (\psi_1, \psi_2, \ldots, \psi_n)$$

of operators for which there exist incidental phenomena

$$\langle A^{(i)}, B^{(i)}\rangle \in \mathrm{Inc}_{\psi_i} \quad (i = 1, \ldots, n)$$

such that

(i) $\qquad T_I[X] \vdash C_{\psi_1} \,\&\, A^{(1)}\,,$

(ii) $\qquad T_I[R_{\psi_i} \,\&\, B^{(i)}] \vdash C_{\psi_{i+1}} \,\&\, A^{(i+1)}\,,$

(iii) $\qquad T_I[R_{\psi_n} \,\&\, B^{(n)}] \vdash Y.$

*) The "image space with frames" of [11], Sec. 7, considers the latter case.

The pure image space of the previous section comes out as a special case with $\mathrm{Inc}_\varphi = \{\langle \textit{true}, \textit{true}\rangle\}$ for all $\Gamma \in \Phi_I$.

Let us illustrate our formalism by a simple example.

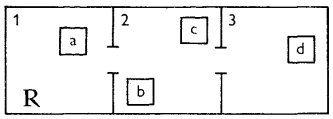

Fig. 1

Example 1. The robot's world consists of three rooms 1, 2, 3 (all connected) and four boxes a, b, c, d distributed in the rooms according to Figure 1 (formally described by the predicate IN, e.g. IN(b, 2)). Robot itself is also in one of the rooms (ROBOT-IN (1)). The state-independent properties of the environment like $\forall x\, \exists y\, \mathrm{IN}(x, y)$ or $(\mathrm{IN}(x, y_1) \,\&\, \mathrm{IN}(x, y_2)) \rightarrow y_1 = y_2$ compose the core theory T_I. There are two operator schemata:

$push[x, y, z]$ (robot pushes box x from room y into room z) with condition

$$\mathrm{IN}(x, y) \,\&\, \text{ROBOT-IN}(y)$$

and result

$$\mathrm{IN}(x, z) \,\&\, \text{ROBOT-IN}(z)\,;$$

$goto[y, z]$ (robot goes from room y into room z) with condition

$$\text{ROBOT-IN}(y)$$

and result

$$\text{ROBOT-IN}(z)$$

As the incidental phenomena of both operator schemata we take first of all the frame: all pairs $\langle A, A\rangle$ where A is any formula without an occurence of the predicate ROBOT-IN and, in the case of $push[x, y, z]$, also of the atomic formula $\mathrm{IN}(x, y)$; (thus e.g. $\langle \mathrm{IN}(\mathrm{b}, 2), \mathrm{IN}(\mathrm{b}, 2)\rangle$ is an incidental phenomenon of $push[\mathrm{c}, 2, 3]$ but not of $push[\mathrm{b}, 2, 3]$).

It can be easily seen that in such an image space one can solve the problem $\langle X, Y\rangle$ where X specifies the position of boxes and of the robot as indicated in Figure 1 and the goal Y has the form

$$\exists v\, (\mathrm{IN}(\mathrm{a}, v) \,\&\, \mathrm{IN}(\mathrm{b}, v) \,\&\, \mathrm{IN}(\mathrm{c}, v) \,\&\, \mathrm{IN}(\mathrm{d}, v))\,.$$

An example of a solution sequence is

$$\gamma = (push[\text{a}, 1, 2],\ goto[2, 3],\ push[\text{b}, 3, 2])$$

Note that the incidental phenomena were necessary in order to use the fact that the boxes b and c remained in their original position.

Another example of an incidental phenomenon of $push[x, y, z]$ may be

$$\langle y \neq z \,\&\, \forall u\,(\text{IN}(u, y) \rightarrow u = x),\ \text{EMPTY}(y)\rangle$$

(a room becomes empty after the robot pushed away the last box). While useless for the above problem it helps to solve a problem with the goal, say, EMPTY(2).

The described idea of using incidental phenomena suggests a new interesting research topic: while the usual application of heuristic methods in problem solving concerns the selection of operators (as e.g. the GPS-like strategy in STRIPS), here one can also investigate special heuristic rules of controlling the flow of information from one image into another by a clever choice of incidental phenomena. Note that it is not apparent from the final plan, which particular incidental phenomena were used: they just serve as catalysts for planning.

In a strict mathematical sense incidental phenomena can be always eliminated: any triple $\langle \varphi, A, B\rangle$, where $\langle A, B\rangle \in \text{Inc}_\varphi$, can be viewed as a new, *modified operator (schema)* $\varphi_{A,B}$ with $\text{C}_{\varphi_{A,B}} \equiv \text{C}_\varphi \,\&\, A$ and $\text{R}_{\varphi_{A,B}} \equiv \text{R}_\varphi \,\&\, B$. In general this would result in an enormous (possibly infinite) number of new operators in the image space; on the other hand, it may yield an interesting two-stage approach to problem solving: first constructing a suitable set of such modified operators, specifically tailored for certain problem or a class of problems, and then solving a given problem in the modified image space. A similar idea was used by Sintzoff in [9] with the intention to obtain a set of modified operators enabling a backtrack-free search of a solution. Another possibility is to modify operators on the basis of learning from past experience.

There are various reasons for keeping the condition-result pairs and the incidental phenomena conceptually apart. In addition to reasons already mentioned let us emphasize that we view the conditions and results of operators as their inherent attributes, in fact parts of their "definitions" provided by the designer. Therefore we make a basic assumption that the robot is, at the execution time, always able, if necessary, to test the validity of operator conditions (of course, this is the view of the problem-solver, not of the executor). Such an assumption cannot be made about the rather arbitrary formulas occuring as antecedents of incidental phenomena. Thus our formalism gives a special treatment to facts not only of an auxiliary nature but also of uncertain verifiability (this point comes out again in Section 6).

Let us note in passing that incidental phenomena may also serve as a natural basis for randomization, yielding a possible probabilistic approach to robot problem solving (cf. [5]).

R. Waldinger gives in [12] (Part 2.) an interesting overview of various ways of representing actions and their effects in contemporary problem-solving systems. We claim that a thorough analysis of most of these systems (requiring their reformulation in exact mathematical and logical terms – which in itself may not be an easy task, indeed) would reveal their reducibility to one or the other special case of the image space (with incidental phenomena), combined with a special search method.

Let us illustrate our claim on the particular cases of the STRIPS problem solver [1] and of Waldinger's regression technique [12].

To give a rigorous logical meaning to the set-theoretical operations in STRIPS we have to agree on a certain class of *basic formulas* (for instance all literals) which may be used as elements of world descriptions. Any operator φ specified in STRIPS by the triple C_φ (precondition), Del_φ (the set of basic formulas to be deleted) and Add_φ (the set of basic formulas to be added) can be in the image-space formalism specified by the condition-result pair $\langle C_\varphi, R_\varphi \rangle$ where R_φ is the conjunction of formulas in Add_φ, and by incidental phenomena of the form $\langle A, A \rangle$ where A is any basic formula not in Del_φ, or a conjunction of such formulas.

In general, every STRIPS problem description can be in this way converted into an image space I such that the solution obtained by STRIPS to a problem is also a plan (for the same problem) in I.

Conversely, let us define a STRIPS-like image space as any image space with the set of incidental phenomena including only pairs of the form $\langle A, A \rangle$ and closed under conjunction (the set of all formulas that do not participate in any incidental phenomenon of φ may then serve as an analogy to Del_φ). For any image space $I = \langle T_I, \Phi_I \rangle$ one can construct a STRIPS-like image space $\tilde{I} = \langle T_I, \tilde{\Phi}_I \rangle$ by introducing a new modified operator schema $\varphi_{A,B}$ for each triple $\langle \varphi, A, B \rangle$, where $\langle A, B \rangle \in \in Inc_\varphi$, $A \neq B$ (see end of Section 3). It can be shown that a problem has a solution in I iff it has a solution in $\tilde{I}$.

As concerns the class of tractable problems, the image space is thus, in the above sense, equally powerful as STRIPS. Its conceivable advantage can be found in its generality (lesser dependence on a specific problem) and in the mentioned idea of selective choice of facts to be remembered from one image to another and thus giving the possibility of saving some memory space.

Waldinger's idea of regression, used for synthesis and modification of plans, can be in the framework of incidental phenomena expressed without difficulties: In order for a formula B to be true *after* the execution of an operator ψ, either (1) it must follow from R_ψ, or else (2) there must exist a formula A, which is true *before* the execution and such that $\langle A, B \rangle \in Inc_\psi$. Thus, when passed back over ψ, the formula B becomes ***true*** in the former case and turns into A in the latter case. As an application of the regression technique in multiple-goal planning one may transfer a given

additional goal requirement backwards through the plan to a position where the plan could be properly altered.

5. THE SITUATION CALCULUS

The question of relationship of STRIPS-like methods to historically older and more logically oriented situation calculus was rised several times (for instance by Nilsson [7], pp. 211–212). Due to the untractability of the frame problem the SRI group has after some experiments abandoned the situation calculus completely, while other authors (as Kowalski) strongly argue in favour of predicate logic as a tool for representing knowledge and for problem solving; Kowalski himself suggests a new variant of situation calculus for robot plan formation ([6], Chapter 3).

It is our belief that the situation calculus, and formal logical tools in general, are and will remain relevant to artificial intelligence, if for nothing else then for its important theoretical role. This will be partly documented in this and the following sections.

To make our arguments clearer we shall consider the situation calculus in the simple form used by Green [2] (cf. also [7], Section 7-7). Instead of giving its formal definition we shall only sketch how an image space can be converted into a formal theory in situation calculus. Consider an image space $\mathrm{I} = \langle T_\mathrm{I}, \Phi_\mathrm{I} \rangle$ as introduced in Section 2 and let L be the underlying logical language. We extend L in such a way that each formula A of L is "parametrized" by a special *situation argument*: a new formula, denoted by $A[\![s]\!]$ expresses the same as A but only for situation s. Furthermore, the operator schemata are used as additional function symbols in the new language and are interpreted as mappings from situations with objects as parameters into situations. Thus the term $\varphi(x_1, \ldots, x_m, s)$ corresponds to the operator schema $\varphi[x_1, \ldots, x_m]$ applied in situation s. This is an example of a *situation term*; the terms interpreted as objects are called *object terms*. Note that the new function symbols admit just one situation argument and consequently all situation subterms of the same situation term are nested in a monotoneous way – an important property for our purposes.

Two alternative axiomatic systems in situation calculus can be associated with the image space I, both with three basic types of axioms. The first system is denoted by K_I and involves:

(i) for each formula A in T_I a *core axiom* of the form

$$\forall s \; A[\![s]\!] \; ;$$

(ii) for each operator schema $\varphi[x_1, \ldots, x_m]$ with condition C_φ and result R_φ an *operator axiom* of the form

$$\forall s \; (\mathrm{C}_\varphi[\![s]\!] \rightarrow \mathrm{R}_\varphi[\![\varphi(x_1, \ldots, x_m, s)]\!]) \; ;$$

(iii) for each $\varphi[x_1, \ldots, x_m]$ and each $\langle A, B\rangle \in \mathrm{Inc}_\varphi$ an *incidental axiom* of the form

$$\forall s\ (C_\varphi[\![s]\!] \,\&\, A[\![s]\!] \to B[\![\varphi(x_1, \ldots, x_m, s)]\!])$$

(in particular, when A equals B we call it the *frame axiom*).

The second system, denoted Ins K_1, differs only in one point: axioms of type (ii) and (iii) are replaced by all their ground instances (variable-free object terms of L are substituted for all parameters of the axioms).

To avoid degenerate cases we shall assume all theories used in the sequel to be consistent.

Let $\langle X, Y\rangle$ be a problem in I and let K be either K_1 or Ins K_1. We shall say that the problem $\langle X, Y\rangle$ is *solvable in* K iff

$$K \vdash \exists s\,(X[\![s_0]\!] \to Y[\![s]\!])\,, \tag{1}$$

where s_0 is a constant denoting the initial situation.

Denote K' an open conservative extension of K. It appears as a consequence of Herbrand's theorem (cf. [11]), that (1) holds iff there is a finite set $\mathscr{T}$ of variable-free situation terms of K' such that

$$K' \vdash X[\![s_0]\!] \to \bigvee_{t\in\mathscr{T}} Y[\![t]\!]\,. \tag{2}$$

We shall call $\mathscr{T}$ a *solution set for* $\langle X, Y\rangle$ *in* K'.

In the case when K is Ins K_1, if (1) holds then there is a solution set consisting of terms of K only. This solution set can be interpreted as a plan for robot's behaviour. If it contains only a single term then it corresponds to a straight-line plan. For instance in our Example 1 we obtain the term

$$push(\mathrm{b}, 3, 2, goto(2, 3, push(\mathrm{a}, 1, 2, s_0)))\,.$$

6. BRANCHING PLANS

What is the meaning of the solution set $\mathscr{T}$ from (2) in the general case, when it consists of more than one term? It appears natural to interpret this set as a *branching plan*. For instance, if the disjunction in (2) has the form $Y[\![t_1]\!] \vee Y[\![t_2]\!]$ and t_3 is the largest common situation subterm of t_1 and t_2 (if nothing else, it is s_0) then the set $\{t_1, t_2\}$ represents a plan where the execution of the common part t_3 is followed by a decision whether to continue execution of t_1 or of t_2. This decision may involve perceiving, or testing, the actual state of the environment. This important generalization of plans enables to postpone certain decisions from the planning stage to the execution stage.

Actual reasons for this postponemenet can be easily demonstrated using the image-space approach. Let us consider first the case without incidental phenomena.

Assume that for a certain image T we have $T \vdash C_{\psi_1} \vee C_{\psi_2}$ for two distinct operators ψ_1 and ψ_2, but neither $T \vdash C_{\psi_1}$ nor $T \vdash C_{\psi_2}$ holds. According to the previous definition neither of the operators is applicable in T; nevertheless, the real world is logically complete and thus one of the two alternatives can be in execution time followed. A similar case occurs when $T \vdash C_{\psi} \vee Y$ (Y is the goal formula). In real world either Y holds, i.e. the goal has been achieved and the execution halts, or the execution may continue by the operator ψ.

The nature of branching plans and their role in problem solving is discussed in a more detail in [11], cf. also [6], p. 41. A method for generating branching plans was implemented by Warren [13]. An important metatheorem, proved in [11], shows the correspondence between plan formation in image space and theorem proving in situation calculus. It asserts that a problem $\langle X, Y\rangle$ has a branching plan in I if and only if it is solvable in the associated situation calculus Ins K_{I}. Moreover, the solution set can be effectively extracted from the proof in Ins K_{I}.

Let us now turn to the case of an image space involving both incidental phenomena and branching. The nature of a plan in this general case is best explained by defining it as a formal mathematical object. Let us denote by Σ the set of all operators of an image space I. So far the elementary objects of planning were operators of Σ; now it is better to work with triples of the form $\langle \psi, A, B\rangle$, where $\psi \in \Sigma$ and $\langle A, B\rangle \in \in \mathrm{Inc}_{\varphi}$. We call such triples *transitions*; finite sequences of them are *transition sequences* (in distinction from *operator sequences*). If τ denotes a transition we write $\tau = \langle \mathrm{op}(\tau), A_\tau, B_\tau\rangle$; obviously, each transition sequence $\alpha = (\tau_1, \tau_2, \ldots, \tau_n)$ defines uniquely an operator sequence $\mathrm{op}(\alpha) = (\mathrm{op}(\tau_1), \mathrm{op}(\tau_2), \ldots, \mathrm{op}(\tau_n))$. In both cases we consider also the "empty" sequence (for $n = 0$) denoted by Λ; clearly, $\mathrm{op}(\Lambda) = \Lambda$.

Let $\langle X, Y\rangle$ be a problem in I and let Γ be a finite nonempty set of transition sequences. For any transition sequence β define the set F_β of all transitions which are immediate successors of β in Γ:

$$\mathrm{F}_\beta = \{\tau \mid \beta\tau\beta' \in \Gamma \text{ for some sequence } \beta'\}$$

(in particular, F_Λ consists of all the first elements of sequences in Γ). By $(Y)_{\beta\in\Gamma}$ denote the formula Y if $\beta \in \Gamma$ and ***false*** otherwise. Assume that Γ satisfies

$$T_{\mathrm{I}}[X] \vdash \bigvee_{\tau\in \mathrm{F}_\Lambda} (C_{\mathrm{op}(\tau)} \,\&\, A_\tau) \vee (Y)_{\Lambda\in\Gamma} \tag{3}$$

and for each nonempty initial segment $\alpha = (\tau_1, \ldots, \tau_n, \tau')$ of any transition sequence in Γ,

$$T_{\mathrm{I}}[R_{\mathrm{op}(\tau')} \,\&\, B_{\tau'}] \vdash \bigvee_{\tau\in \mathrm{F}_\alpha} (C_{\mathrm{op}(\tau)} \,\&\, A_\tau) \vee (Y)_{\alpha\in\Gamma}\,. \tag{4}$$

Then the set $P = \{\text{op}(\alpha) \mid \alpha \in \Gamma\}$ is, by definition, a plan for the problem $\langle X, Y\rangle$ in I.

The set Γ conveys more information than the plan P: it explicitly indicates where and what incidental phenomena has been taken into account in the course of planning. Why then to define a plan as the set P rather than Γ? The main reason is the already mentioned assumption that the robot is able to test in real world only the conditions of its operators and not, in general, the antecedents of incidental phenomena. Thus the additional information in Γ would be of little use for the execution subsystem. (It may be useful for other purposes, for instance when plans are evaluated according to their reliability – cf. [5].)

The difference between hypothetical assumptions in the planning stage and actual knowledge during the execution may explain a "paradox" of conditional plans illustrated by the following example.

Example 2. In a world similar to Example 1, but with only one box a, consider the problem $\langle X, Y\rangle$, where X is

$$\text{ROBOT-IN}(2) \,\&\, (\text{IN}(a, 1) \vee \text{IN}(a, 3))$$

(cf. Fig. 2) and Y is

$$\exists x\, (\text{ROBOT-IN}(x) \,\&\, \text{IN}(a, x)),$$

representing the task "go to the room with the box".

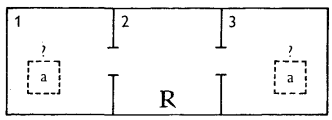

Fig. 2

The branching plan

$$P = \{(goto[2, 1]),\ (goto[2, 3])\},$$

consisting of two single-operator sequences, clearly solves the problem. Moreover, this plan is executable since the condition (ROBOT-IN(2)) for both the *goto* operators is true in the initial state. Yet we are somewhat reluctant to admit P as a reliable plan: its execution clearly fails to reach the goal if the improper branch is executed.

This anomaly is a consequence of using incidental phenomena. The set Γ, from which P is derived, consists of two transition sequences (each of length one):

$$\Gamma = \{(\langle goto[2, 1],\ \text{IN}(a, 1),\ \text{IN}(a, 1)\rangle),$$
$$(\langle goto[2, 3],\ \text{IN}(a, 3),\ \text{IN}(a, 3)\rangle)\},$$

where the second parameter of *goto* depends on the corresponding incidental phenomenon, a structural property that cannot be observed in P. We can deal with this situation in several ways. One possibility is to assume that the robot is able to test the validity of formulas IN(a, 1) and IN(a, 3) in the execution time. Then we can replace the operator schema $goto[x, y]$ by its modified version $goto'[x, y, z]$ with the condition ROBOT-IN(x) & IN(a, z). This reduces the decision at the branching point just to testing conditions of operators. On the other hand, if we cannot assume the ability of testing the relation IN, the only possibility is to reconcile with unreliable plans – which is after all a natural thing to do.

(Let us remark that there exists another plan for the above problem, which is reliable under the assumption that the goal formula is testable and that backtracking is allowed: (*goto*[2, 1]), (*goto*[2, 1], *goto*[1,2], *goto*[2, 3])).

The mechanism of incidental phenomena enables us to treat *tests* (mesurements, experiments, and even robot's questions to the user) in the same way as physical actions and represent them explicitly by operators in the image space. For instance in our last example we may invent a new operator schema *look-where-is*$[x]$ with condition, say, CAN-SEE(x) and with the incidental phenomena

$$\{\langle \text{IN}(x, y),\ \text{k-IN}(x, y)\rangle \mid y = 1, 2, 3\},$$

where k-IN(x, y) is a new predicate interpreted as "the robot knows that (the box) x is in (the room) y". A natural core axiom may be $\forall x\, \forall y\, (\text{k-IN}(x, y) \rightarrow \text{IN}(x, y))$ (but not the converse). Of course, all occurences of the predicate IN in operator conditions should be replaced by k-IN.

Finally, let us mention one further point concerning the relationship of the set Γ and the corresponding plan P. In general, op(α) = op(β) does not imply $\alpha = \beta$, even if $\alpha, \beta \in \Gamma$ or if both α and β are initial segments of some sequences in Γ. For instance, it may happen that the disjunction in the definition of a plan (in (3) or (4) above) has the form, say,

$$(5) \qquad (C_\psi \,\&\, A) \vee (C_\psi \,\&\, \neg A)\,.$$

Then a branching occurs in Γ but not necessarily in P. Let us call this phenomenon *concealed branching*. There are logical reasons to allow plans with concealed branching (first, theorem proving in the situation calculus does not avoid them, and second, they solve a strictly larger class of problems). A typical case of concealed branching occurs when an explicit test (in the just described sense) is involved in the plan. Thus the operator ψ in (5) may be a test whether A is true or not, the condition of the test being independent on which of the two alternatives really holds.

7. STATE-DEPENDENT FUNCTIONS AND THE EXPLICATION ABILITY

The theory Ins K_1 was shown to have the same problem-solving power as the original image space I. Are there cases when the stronger theory K_1 can be used as a tool for problem-solving? Is the class of problems solvable in K_1 generally wider then that of Ins K_1?

Before we try to answer these questions we shall somewhat modify Example 1 from Section 3.

Example 3. Let the robot's three room world contain an unknown number of boxes of a varying size. Suppose there is an ordering relation for all boxes represented by predicate SMALLER(x, y). The situation calculus K_1 for this world contains among its core axioms also the following one:

$$(6) \qquad \forall s\, \forall y\, ((\text{ROBOT-IN}(y, s) \,\&\, \neg \text{EMPTY}(y, s)) \rightarrow$$
$$\rightarrow \exists x\, (\text{IN}(x, y, s) \,\&\, \forall u\, (\text{IN}(u, y, s) \rightarrow \text{SMALLER}(x, u, s)))),$$

i.e., there is always one smallest box among the boxes in each nonemtpy room visited by the robot.

Consider a problem $\langle X, Y \rangle$ where X is ROBOT-IN(1) & EMPTY(3) and Y is

$$\exists x\, (\text{IN}(x, 3) \,\&\, \forall y\, (\text{IN}(y, 3) \rightarrow y = x) \,\&\, \forall y\, \text{SMALLER}(x, y)).$$

The robot is asked to find the smallest box from all the rooms and place it into the room 3, which was originally empty. This problem is represented in K_1 by the formula

$$(7) \qquad X[\![s_0]\!] \rightarrow \exists s\, Y[\![s]\!].$$

It is not difficult to find out, that the formula (7) is not provable in Ins K_1 as well as the problem $\langle X, Y \rangle$ is not solvable in the image space I. The reason for this failure is the lack of a concrete information about the robot's environment. The robot knows neither the names of boxes in a certain room nor their amount.

Let us try to prove the formula (7) in the theory K_1. This formula is provable in K_1 iff it is provable in the theory K_1', which is an open conservative extension of K_1. The theory K_1' is obtained from K_1 after elimination of all the existential quantifiers in the prenex form of the axioms of K_1 using Skolem functions.

In our example the theory K_1' is obtained by introducing a new function *smallest*(y, s) and by replacing the axiom (6) by

$$(8) \qquad \forall s\, \forall y\, (\text{ROBOT-IN}(y, s) \,\&\, \neg \text{EMPTY}(y, s) \rightarrow$$
$$\rightarrow (\text{IN}(\textit{smallest}(y, s), y, s) \,\&\, \forall u\, (\text{IN}(u, y, s) \rightarrow$$
$$\rightarrow \text{SMALLER}(\textit{smallest}(y, s), u, s)))).$$

Suppose there are no existential quantifiers in X and no universal quantifiers in Y. Then (7) holds iff there is a solution set $\mathscr{T}$ for $\langle X, Y\rangle$ that consists of variable-free situation terms of K_1' such that

$$K_1' \vdash X[\![s_0]\!] \rightarrow \bigvee_{t \in \mathscr{T}} Y[\![t]\!]$$

(a consequence of Hilbert-Ackermann's theorem).

For instance, a solution set for the problem in our running example may consist of two terms of the form

$$push(smallest(2, t), 2, 3, t)$$

where t is either $push(smallest(1, s_0), 1, 2, s_0)$ (if room 1 is nonempty) or $goto(1, 2, s_0)$ (if room 1 is empty). This describes the plan "if room 1 is nonempty push the smallest box from room 1 into room 2 (or else, if 1 is empty, just go to 2) and then push the box which happens to be smallest in 2 from 2 into 3".

Suppose that the robot has the ability to identify in any actual situation the smallest box, whose existence is claimed in the axiom (6). The solution set for the formula (7) then represents a sound plan for robot's behaviour in this case. Here we have an example of a problem solvable in K_1 and unsolvable in Ins K_1.

What makes the theories K_1 and Ins K_1 different in general? During a theorem-proving procedure, when the existential quantifiers from the prenex form of the axioms are being removed, the language of the theory under consideration is enriched by new functions (Skolem functions). In a solution suggested by Ins K_1 these new function symbols play only a passive auxiliary role and never appear in a solution set since they do not occur in the place of parameters of operators – cf. [11] for details.

On the other hand, the terms in a solution set in K_1 may easily contain object-valued Skolem functions dependent on situation. (In our above example it will be the function $smallest(y, s)$.)

How such a function should be interpreted? What are the robot's capabilities, which make it possible to use for planning K_1 instead of Ins K_1?

Consider a robot with a preprogrammed ability to find and identify in any actual state of the environment a specimen of the object, the existence of which is claimed for this state by an axiom (or equivalently, to determine the value of an object function with one situation argument) and remember it for a later use. In particular, the robot can evaluate its Skolem functions during execution of its plans. This is an activity oriented to a better understanding of the environment rather than to its actual change.

Suppose a robot has the described property for all existential quantifiers occuring in the prenex form of the core axioms and of the consequents of the operator and incidental axioms of K_1. Moreover let the antecedents of the operator and incidental axioms be verifiable by the robot, e.g. let they be open. Then we say that the robot is

endowed with the *explication ability* (for K_1). In the rest of the paper we shall consider only robots with this ability.

All possible actions of the robot are described by the theory K_1 or even better by a theory K_1' obtained from K_1 after the existential quantifiers in question are eliminated using the Skolem functions.

However, not every solution set can be immediately interpreted as a well-defined plan. The difficulty is caused by the new object functions depending on situation argument, which may appear in terms of $\mathscr{T}$ in a rather peculiar way. For example the situation term

$$push(smallest(3, goto(2, 3, s_0)), 3, 1, push(a, 2, 3, s_0))$$

does not represent a feasible command because the value of the object argument $smallest(3, goto(2, 3, s_0))$ can be explicated only in the situation obtained from s_0 by applying *goto* from 2 to 3 – but the robot never passes through this situation on his way towards the situation $push(a, 2, 3, s_0)$!

We shall call a situation term *regular* iff it represents a feasible command (a formal definition is in [10]). The following theorem can be proved.

Let K_1' describe the capabilities of a robot with the explication ability and let $\langle X, Y\rangle$ be a problem, where X is a formula without existential quantifiers. Then

$$K_1' \vdash X[\![s_0]\!] \to \exists s\ Y[\![s_0]\!]$$

iff a set $\mathscr{T}$ of regular terms of K_1' can be effectively found, such that $\mathscr{T}$ is a solution set for $\langle X, Y\rangle$ in K_1', i.e.,

$$K_1' \vdash X[\![s_0]\!] \to \bigvee_{t \in \mathscr{T}} Y[\![t]\!]$$

(the proof is in [10]).

Certain subterms of situation terms should be distinguished. We shall call them **-subterms* and define them by induction:

The only *-subterm of s_0 is s_0. Let t be a term of the form $\varphi(\ldots, t_1)$; a situation term t' is a *-subterm of t iff t' is a *-subterm of t_1 or $t' = t$.

An execution procedure for a solution set $\mathscr{T}$ from the above theorem may have the following form:

Step 1. Set PRESENT := s_0, HINT := $\mathscr{T}$.

Step 2. If Y is satisfied in the actual environment then exit with success, otherwise continue.

Step 3. For any term of the form $f(\ldots,$ PRESENT$)$ which is a subterm of some term in HINT and where f is a Skolem function, detect in the actual environment the value of f with the same arguments and remember it.

Step 4. Find an operator schema $\varphi[x_1, \ldots, x_m]$ and object terms $a_1, \ldots, a_m$ such that

(i) there is a term $t \in$ HINT for which $\varphi(a_1, \ldots, a_m, \text{PRESENT})$ is a $*$-subterm of t, and

(ii) the condition C_φ with parameters replaced by the values $\alpha_1, \alpha_2, \ldots, \alpha_n$ of the terms $a_1, \ldots, a_m$ (computed using the values of Skolem functions determined in previous steps) is met in the actual environment.

If such an operator schema and terms do not exist then exit with failure; otherwise set

OPERATION:= (the operator obtained from φ by considering $\alpha_1, \alpha_2, \ldots, \alpha_n$ in the place of its parameters). Set NEXT := $\varphi(a_1, \ldots, a_m,$ PRESENT).

Step 5. Execute the OPERATION.

Step 6. Set PRESENT := NEXT, HINT := (the set of all $t \in$ HINT having PRESENT as a $*$-subterm), and go to Step 2.

This procedure ends with failure (in Step 4) only if the theory K_1, which yielded the solution set $\mathscr{T}$, was an inadequate representation of the world.

The state-dependent functions admit a more general interpretation than we considered above. It seems likely that they may serve as a proper tool for representation of changing environment even in cases when the robot is not the sole agent responsible for the changes.

8. PLANNING UNDER CONSTRAINTS

In the previous section we have observed that – as far as state-dependent functions were concerned – the situation calculus appeared as a more suitable framework than the image space even for practical planning. It is therefore worth looking at some other variants of problem solving and ask whether they can be in a suitable way expressed in the framework of situation calculus.

Let us here shortly mention, for instance, the class of problems which may be called *problems with constraints*. So far we have considered only problems expressible in terms of the initial situation and a goal situation. The problems like "go to room x" or "find the smallest box in the room" are all of this type – let us call them *unconstrained problems*. On the other hand, problems with constraints are specified, besides the initial and goal states of the environment, also by properties of all the states encountered during the execution of a plan. The Tower of Hanoi problem is a typical example of a problem with constraints.

How can a problem solver be constructed, which is able to find solutions even of problems with constraints?

Of course, a state-space or an image-space problem solver can be used to solve even constrained problems by automatically limiting the considered states (images) only to those which meet the constraining conditions. It appears that also the situation calculus can be modified so that it can cope with these problems. Let S be a theory in situation calculus enriched by a new predicate $\leqq$, defining a partial ordering of situations.

Let X, Y be formulas describing the initial and the final states of a problem and let D be a formula specifying the constraints on all situations encountered during the solution. This problem can be represented by the following formula

$$X[\![s_0]\!] \rightarrow \exists s\,(Y[\![s]\!] \,\&\, \forall s'\,(s' \leqq s \rightarrow D[\![s']\!]))\,.$$

The question, whether this formula is provable in S iff there is a solution to the problem $\langle X, Y\rangle$ with constraints D, will be a subject of our future studies.

9. RELATIONSHIP TO AUTOMATA THEORY

This last section is just a note on an existing abstract counterpart to the image space within the conceptual framework of automata theory. Aware of the many distinctions between the approaches of automata theory (which studies general behaviour, global properties, and algorithmic means) and of the theory od problem solving in artificial intelligence (interested in particular solutions, local search, and heuristic methods), we can nevertheless advocate the former approach for certain particular questions of a mathematical nature in the latter area. A global mathematical treatment of the image space in its full generality would enable treating plans as mathematical objects suitable for mutual comparison, composition, and grouping into sets of all possible plans for a given goal.

The notion of an *abstract automaton* (specified by a set Q of *states*, by a *transition function* $\delta : Q \times \Sigma \rightarrow Q$, where Σ is a set of abstract symbols called the *input alphabet*, by a *initial state* q_0 and by a set of *final states* $F \subseteq Q$) is a natural and obvious abstraction of the the state space. One just interprets the symbols of Σ as operators and the pair $\langle s_0, F\rangle$ as a problem; the so called ***behaviour*** of such an automaton then represents the set of all straight-line plans for a given goal.

The situation is a bit more complicated for the case of branching plans. In [3] an extension of the concept of a finite automaton was suggested with a special branching relation associating with each state a collection of alternative branchings. Each plan obtains a mathematical form of a set of finite strings over Σ; this set, in distinction to the formalism based on predicate logic (as in Section 4), is allowed to be also infinite.

A further step can be made when one wants to give an automata-theoretic counterpart to an image space with incidental phenomena. From a fixed state (image) using a fixed input symbol (operator) one may obtain several distinct new states if different

incidental phenomena are employed. The adequate formalization leads to the concept of a *nondeterministic branching automaton* [4], where the transition function is many-valued. It appears that – unlike in the case of classical automata – the nondeterminism of this form yields a strictly greater class of representable plans: a mathematical formulation of the fact that allowing incidental phenomena is a nontrivial extension.

ACKNOWLEDGEMENT

The second author wishes to acknowledge the influence of stimulating conversations with workers of related interests during his visit to the Department of Artificial Intelligence, University of Edinburgh, in summer 1976.

(Received July 21, 1977.)

REFERENCES

[1] R. E. Fikes, N. J. Nilsson: STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence *2* (1971), 189–208.

[2] C. Green: Application of theorem proving to problem solving. Proc. IJCAI'69, Washington, D.C., 1969.

[3] I. M. Havel: Finite branching automata. Kybernetika *10* (1974), 281–302.

[4] I. M. Havel: Nondeterministic finite branching automata. Res. Report No 623, ÚTIA-ČSAV 1975.

[5] I. M. Havel, I. Kramosil: A stochastic approach to robot plan formation. Submitted for publication.

[6] R. Kowalski: Logic for problem solving. Memo No. 75, Department of Computational Logic, University of Edinburgh 1974.

[7] N. J. Nilsson: Problem-Solving Methods in Artificial Intelligence. McGraw-Hill, New York 1971.

[8] H. A. Simon: On reasoning about actions. In Representation and Meaning: Experiments with Information Processing Systems (eds. H. A. Simon & L. Siklóssy), Prentice-Hall, Englewood Cliffs 1972, pp. 414–430.

[9] M. Sintzoff: Eliminating blind alleys from backtrack programs. In: Automata, Languages, and Programming (S. Michaelson and R. Milner, Eds.), Edinburgh University Press 1976, pp. 531–557.

[10] O. Štěpánková: Skolem functions and the planning in the situation calculus. A collection of papers 1975 Inst. of Computation Techniques, Technical University of Prague 1975.

[11] O. Štěpánková, I. M. Havel: A logical theory of robot problem solving. Artificial Intelligence *7* (1976), 129–161.

[12] R. Waldinger: Achieving several goals simultaneously. In: Machine Intelligence *8* (E. W. Elcock and D. Michie, Eds.), Ellis Horwood, Chichester 1977, pp. 94–136.

[13] D. H. D. Warren: Generating conditional plans and programs. In: AISB Conf. Proceedings, University of Edinburgh 1976, pp. 344–354.

RNDr. Olga Štěpánková, Ústav výpočtové techniky ČVUT (Institute of Computation Technique – Technical University of Prague), 128 00 Praha 2, Horská 3. Czechoslovakia.

Ing. Ivan M. Havel CSc., Ph.D., Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation – Czechoslovak Academy of Sciences), 182 08 Praha 8, Pod vodárenskou věží 4. Czechoslovakia.