KYBERNETIKA — VOLUME 14 (1978), NUMBER 6

An Attempt to Formalize the Notion of Heuristics in State Space

IVAN KRAMOSIL

A proposition is suggested how to define formally the notion of heuristic procedure using the apparatus of state space. The suggestion is of discussive character.

The notions of heuristics and heuristic procedures or algorithms occur often in papers and considerations dealing with artificial intelligence or computer science. Strangely enough, the use of these notions is (almost always) informal and rather vague and it is based on a supposed "common sense" of the reader that he will understand these expressions in the same way as the author intended or at least in a way not substantially different from the intended one. Some authors emphasize several aspects which can be taken as attributes of heuristics or at least as typical features of heuristic procedures, but these properties are far from being exhaustive and categoric as well as far from being mathematically formalized or even formalizable. Also some counterexamples, i.e., examples of procedures or solutions which cannot be or should not be considered as heuristic ones are far from being so rich and exact as to be able to serve as a ground of a "negative definition" of heuristics.

A heuristic procedure is one which goes very well, quickly and easily in some "typical" or "usual" cases, but does not go or goes very badly in other cases. The examples of both the groups of inputs are easily obtainable, but the boundaries between them are very unsharp and vague. Deciding to use some heuristics or proposing a heuristic procedure we usually profit of a side information, however, it is very difficult or even impossible to classify the pieces of information being at our disposal into the "direct" and "side" ones.

Here we propose a possibility how to formalize the concept of heuristics using the notion of state space and the apparatus of mathematical logic. In no case we would like to pretend as if the formalization proposed here were the only possible or the best one, our only intention is to present our ideas as a subject for further discussion and critique.

Let us start with the notion of state space. Consider a nonempty set $\mathcal{S}$, the elements of which are denoted by s (possibly indexed) and called *states*. Intuitively, states are to formalize various states or configurations or situations of the environment in which a subject (human being, robot, automaton) is to solve a problem, to execute a procedure or computation, to take a decision. The environment is not static and it may transit from a state to another one; either according to its internal laws of development, or according to interventions (actions, operations) executed by the subject, or according to some random influences if we want or need to distinguish them from the internal laws. In order to formalize the possibilities of the subject to intervene in the environment and change its instantaneous state let us introduce the notion of operator. Operator φ is a partial mapping defined in $\mathcal{S}$ and taking its values again in $\mathcal{S}$; by *Dom* (φ) we denote the set of states for which it is defined. The partial character of the mapping φ corresponds to the fact, that real, physical operators are applicable only in certain states of the environment, not in all states (there is usually a complex of conditions ascribed to an operator, under which it is applicable). The pair $\langle \mathcal{S}, \Phi \rangle$ is usually called the *state space*, $\Phi = \{\varphi\}$.

The next part of our construction is a language $\mathcal{L}$ which enables to speak about states of the state space, about their properties and relations among them. To be able to express also the dynamics of the environment let us parametrize the particular statements by the name of the state to which the statement is referred. Formally it is done by replacing $\mathcal{L}$ by the two-sorted (or more than two-sorted if $\mathcal{L}$ itself was many-sorted) language $\mathcal{L}^*$ by enriching each elementary formula by a new variable or constant (term, in general) of a new, situation type and considering only those formulas in which just one situation term occurs; in listing the variables, constants or terms occuring in a formula of $\mathcal{L}^*$ this situation one is listed as the last. As a rule, and this will be the typical situation for our further considerations, the means of the language $\mathcal{L}^*$ do not enable to distinguish any two states $s, s' \in \mathcal{S}$, i.e., the information about states, expressible in $\mathcal{L}^*$, is not complete from the point of view of distinguishing among states.

We shall need a formalization of what we have just said. Let Ax be a set of formulas of $\mathcal{L}^*$, called *axioms*, and expressing the known state-independent true assertions (corresponding to general laws governing the basic features and dynamical development of the environment in question) as well as the assertions concerning particular states and containing names of these states as situation terms. Moreover, consider the usual deduction rules of mathematical logic corresponding to the type and order of the language $\mathcal{L}^*$. Instead writing $Ax \vdash A$ for denoting the fact that A (a formula of $\mathcal{L}^*$) can be derived from the formulas contained in Ax via the deduction rules we shall write simply $\vdash A$, as Ax will be supposed to be fixed and, hence, to play the same role as logical axioms.

The level of distinguishing among states on the grounds of $\mathcal{L}$ and Ax is formalized in the following definition. If A is a formula of $\mathcal{L}$ and $\bar{s}$ a situation variable or term, then $A[\bar{s}]$ denotes the formula of $\mathcal{L}^*$ resulting when A is parametrized by s.

Definition 1. Let $s_1, s_2 \in \mathscr{S}$, let $\bar{s}_1, \bar{s}_2$ be situation constants or terms corresponding to these two states (their names). Then s_1, s_2 are equivalent with respect to $\mathscr{L}$ and Ax, formally $s_1 \approx s_2(\mathscr{L}, Ax)$, iff for all formulas $A \in \mathscr{L}$

$$\vdash A[\bar{s}_1] \quad \text{iff} \quad \vdash A[\bar{s}_2] \, .$$

As $\mathscr{L}$ and Ax will be fixed in what follows we write simply $s_1 \approx s_2$. Clearly, $\approx$ is an equivalence relation on $\mathscr{S}$. Denote by $\mathscr{S}_0$ the quotient algebra $\mathscr{S}/\approx$; its elements will be, if useful, considered also as sets of states, i.e., as subsets of $\mathscr{S}$.

Let $\mathscr{I}$ be a system of subsets of $\mathscr{S}$ (i.e., $\mathscr{I} \subset \mathscr{P}(\mathscr{S})$) satisfying the only condition that $I_1 \in \mathscr{I}$ and $I_2 \supset I_1$ imply $I_2 \in \mathscr{I}$: Elements of $\mathscr{I}$ are called *important sets* (*of states*). In general, important sets need not be definable in the language $\mathscr{L}^*$, they even need not be expressible as unions of elements from $\mathscr{S}_0$. This "independence" of $\mathscr{L}^*$ and the other language in which important sets are described is of crucial significance in what follows.

A *procedure* (for solving a problem, taking a decision, reaching a goal) is, in the most general form, a mapping p ascribing to each $s \in \mathscr{S}$ an infinite sequence of operators from Φ, formally, $p : \mathscr{S} \to \Phi^\infty$ (finite sequences of operators can be always considered as special cases of the infinite ones supposing there is an "empty operator" Λ in Φ). Clearly, because of the limited abilities in distinguishing the states of $\mathscr{S}$ by the means of $\mathscr{L}$ we are not able to define a procedure explicitly, rather we define a mapping from $\mathscr{P}(\mathscr{S}_0)$ into $\mathscr{P}(\Phi^\infty)$, i.e., to a set of states distinguishable from others by $\mathscr{L}$ we ascribe a set of operator sequences from Φ^∞, i.e., a branching plan. In other words, not being able to separate a particular state, we must take into consideration simultaneously several possibilities how to proceed; just in the process of executing the environment will force us to follow the way ascribed by the procedure in question to the actual initial state.

As a rule, the subject's activity in the environment is not senseless, it aims to reach a *goal*. The goal usually consists in transforming of the state (configuration) of the environment into a desired state, more formally, into a state satisfying some a priori given goal conditions. Let G be a formula of $\mathscr{L}^*$ with one free variable $\bar{s}$, let $\{s : s \in \in \mathscr{S}, G[\bar{s}]\}$ be just the set of desired goal states. A pair $\langle s, G\rangle$, $s \in \mathscr{S}$, is called a *problem* (in the state space $\langle \mathscr{S}, \Phi\rangle$).

Definition 2. Let $\langle \mathscr{S}, \Phi\rangle$ be a state space, let p be a procedure, let $\langle s, G\rangle$ be a problem. Procedure p *solves the problem* $\langle s, G\rangle$, iff, denoting $p(s)$ by $\varphi_1\varphi_2\varphi_3 \ldots$, there is an index i such that

(i) for all $j < i$, $\varphi_j(\varphi_{j-1} \ldots f_1(s) \ldots) \in Dom\,(\varphi_{j+1})$, $s \in Dom\,(\varphi_1)$.

(ii) $\vdash G(\varphi_i(\varphi_{i-1} \ldots \varphi_1(\bar{s}) \ldots))$.

The minimal i satisfying (i) and (ii) will be denoted by $l(s, G, p)$, for other s put $l(s, G, p) = \infty$. Denote

$$p(G) = \{s : s \in \mathscr{S},\ l(s, G, p) < \infty\} \, .$$

Definition 3. Let $\langle \mathscr{S}, \Phi \rangle$ be a state space, let p be a procedure, let $\langle s, G \rangle$ be a problem. Procedure p is called *heuristic in the state* s, iff there are $s_1, s_2 \in \mathscr{S}$, $s_1 \approx s_2 \approx s$, such that $s_1 \in p(G)$, $s_2 \in \mathscr{S} - p(G)$. Procedure p is called *heuristic*, if it is heuristic in all states of $\mathscr{S}$ and if, at the same time, $p(G) \in \mathscr{I}$, i.e., $p(G)$ is an important set. Procedure p is called *semi-heuristic*, if there is at least one $s \in \mathscr{S}$ in which p is heuristic and if again, at the same time, $p(G) \in \mathscr{I}$.

Very often we speak about heuristic procedures not only in the "absolute" sense, but in comparison with other procedure or procedures. Also this notion can be formalized by the means of our apparatus.

Definition 4. Let $\langle \mathscr{S}, \Phi \rangle$ be a state space, let p_1, p_2 be procedures, let $\langle s, G \rangle$ be a problem. Denote

$$\langle p_1, p_2 \rangle (G) = \{s : s \in \mathscr{S},\ l(s, G, p_1) < l(s, G, p_2)\} .$$

Procedure p_1 is called *heuristic with respect to* p_2 *in the state* s, iff there are $s_1, s_2 \in \mathscr{S}$, $s_1 \approx s_2 \approx s$, such that $s_1 \in \langle p_1, p_2 \rangle (G)$, $s_2 \in \mathscr{S} - \langle p_1, p_2 \rangle (G)$. Procedure p_1 is called *heuristic with respect to* p_2, if it is heuristic with respect to p_2 in all states of $\mathscr{S}$ and if, at the same time, $\langle p_1, p_2 \rangle (G) \in \mathscr{I}$. Procedure p_1 is called *semi-heuristic with respect to* p_2, if there is at least one $s \in \mathscr{S}$ in which p_1 is heuristic with respect to p_2 and if, at the same time, $\langle p_1, p_2 \rangle (G) \in \mathscr{I}$.

Intuitively said, a procedure is heuristic if it solve the problem (or solves it sooner than another procedure) for an important, e.g., great enough or probable enough set of states and if, however, for any such state there is an indistinguishable state in which the procedure fails or is at least less effective than the other procedure with which the former one is compared. In the case of a semi-heuristic procedure this uncertainty takes place only in some cases, on the other hand, there are states distinguishable from the others and such that we can be sure that the procedure goes or goes well for these states. Of course, the crucial role is played by the formalized theory $\langle \mathscr{L}, Ax, \vdash \rangle$, enriching the language or the set of axioms we make the quotient algebra $\mathscr{S}/\approx$ refined. If it becomes refined enough to separate the states with $l(s, G, p) < \infty$ from the other states, the procedure ceases to be heuristic or even semi-heuristic, in spite of the fact that it has not been changed, as far as considered from the platonistic point of view, i.e., as mapping from $\mathscr{S}$ into Φ^∞. We do not solve here the problem which are the relations among non-heuristic, non-semi-heuristic and algorithmical procedures.

Let us finish this explanation by an example. Consider a procedure consisting in driving a nail into a wall with a hammer. For the sake of simplicity we may identify the set $\mathscr{S}$ of states with the set of all walls supposing a nail and a hammer to be at our disposal in all states. The only operator φ being at our disposal is "strike the nail one time with the hammer", i.e., $\Phi = \{\varphi\}$. Consider the only possible procedure $p : \mathscr{S} \to \Phi^\infty$, $p(s) = \varphi^\infty$ for all $s \in \mathscr{S}$. Let $\mathscr{L}$ be a formalized language which is able to express the visually verifiable properties of walls, e.g., colour, quality of surface.

eventually temperature, etc., but $\mathscr{L}$ has no means to express the internal properties of walls, their construction, chemical quality of the raw material used, etc. Namely, $\mathscr{L}$ is not able to express the fact whether there is or is not a metallic objection in the wall which could prevent the nail from driving in the wall. Hence, our procedure p is heuristic, as there are two walls, not differing from each other from the observational point of view, but such that in one case the procedure successes (if there is no metallic objection, a finite number of strikes will do), in the other case the same procedure fails (if there is a metallic objection inside the wall). Considering, e.g., the set of all purely wooden or brick walls as an important one (remember, that this property is not expressible in $\mathscr{L}$!) we can see that our procedure goes for all states inside this important set, i.e., it is a heuristic procedure in our sense.

Below we refer some works which may be useful for an orientation in the problem domain connected with heuristics or heuristic procedures. The author kindly asks the readers to send their opinions, comments, remarks or objections either in the form of a discussion contribution to Kybernetika, or in the form of a private communication to the author.

(Received June 16, 1978.)

REFERENCES

[1] G. Pólya: How to solve it. Princeton Univ. Press, Princeton. N.J. 1954. Russian translation: Mir, Moscow 1961.

[2] H. Gelertner, N. Rochester: Intelligent behavior in problem-solving machines. IBM Journal Res. Dev., 1958, pp. 336—345.

[3] N. J. Nilsson: Problem-solving methods in artificial intelligence. Mc Graw-Hill, New York 1971. Russian translation: Nauka, Moscow 1973.

[4] J. R. Slagle: Artificial intelligence: the heuristic programming approach. Mc Graw-Hill, New York 1971. Russian translation: Nauka, Moscow 1973.

[5] Artificial intelligence and heuristic programming. N. V. Findler and B. Meltzer, Eds. Edinburgh 1971.

[6] Е. А. Александров: Основы теории эвристнческих решений. Советское радио, Москва 1975.

Dr. Ivan Kramosil, CSc., Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 182 08 Praha 8. Czechoslovakia.