OK A LOCAL APPROACH TO REPRESENTATION IN PROBLEM SOLVING

V.L.Stefanuk
Institute for Information
Transmission Problems
Academy of Sciences U.S.S.R.
Ul. Aviamotoraaya 8, a
Moscow E-24
U.S.S.R.

Set of Descriptive Terms: Representation, "Tough Nut" Problem, Local control, Invariant description, Simplification of description, Reduction of names

Abstract

The common feature of all of the *problems* discussed in this paper is that, in their initial formulation, they have a large number of elements with different names, and for this reason the problems seem to be difficult. However, the first attempts to solve the problems give the necessary heuristic information for simplification of the problem description and a reduction in the number of different names for the elements. We could say that the desire to cut down the number of names is a "global idea', but the choice of names and the algorithm of naming are contained in the first attempts.

It is important to stress that the reduced description obtained is not always useful for the solution *of* the original problem, but it turns out to be useful for some of such problems.

This paper shows that local attempts to begin to solve a problem can be very useful, as sometimes they let one simplify the problem description and thus facilitate the solution of the original problem.

We feel that each time some heuristic is formulated (for instance, "try to occupy and to keep the center" in chess), it is formulated not because we know how to solve the total problem (how to win in chess), but as a result of studing some local subproblems. Of course, as this heuristic is formulated on the basis of some local consideration, its usefulness for the whole problem should be separately proved (theoretically or experimentally).

Tough Nut and Other Similar Problems

One of the most interesting problems showing the importance of a good representation is the "Tough But" problem.1.2 The problem, briefly, is to prove the impossibility of fully covering a chessboard, with two diagonally opposite squares removed, using domino pieces. The black/white coloring of the board leads to the simple solution.

One could reason that the successful coloring of the board in this case is simply an "insight" which is a rather natural thing for a person familiar with the chess-*Qoaro.* from hie childhood.

Here we will try, however, to look at this matter from a different point of view and to show how one could "make a guess" about the board coloring, without appeal to purely human experience.

The difficulty of the Tough Nut problem in its initial formulation lies in the fact that one is given with the board description simply as a two-dimensional field $\| C_{ij} \|$, i,j = 1. . . .8, of square cells (with C_{11} , C_{88} removed), and the right decision about placing the first pieces is completely unclear because the end result is not easily seen.

Indeed, the only way to describe our actions (positioning of the pieces on the board) is to display the names (C.■) of covered cells. in the original formulation of the problem all of the cells have different names. Such a naming allows only the strategy of solving the problem using a complete search and does not serve as a base for pruning the search tree.

From this point of view, the idea of somehow reducing the number of different names seems to be very reasonable, as it could bring out some information about "identical" positionings. In other words, we see that the only way to achieve a more or less uniform description of our actions is to reduce the number of names used.

The extreme case: one name is used for all of the cells. This gives ua a highly uniform description because our different actions (positionings) differ only in the number of covered cells. This naming obviously could help one only in solution of the simplest problems.

Using two names one could get a less trivial but nevertheless a very uniform description of the following situation: "a domino is put somewhere on the board." It is easy to see that it could be achieved only for a certain assignment of the two names to cells—namely chess assignment. Then we indeed come **invariant description** the board.

To clarify these ideas and to display the limitations inherent in this approach we consider a number of other similar problems.

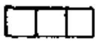
Other Examples of Similar Problems (The Problems of Enlargement)

In all the problems below we will follow our chain of reasoning—using a small number of different names for elements of a problem and then getting a certain assignment of the names to different elements in a way which will give us an invariant description of our actions. We'll see that the procedure of assignment has a very simple "algorithmic" character.

1. Let us decide to present a certain

array of square cells as a sum of elements not of the "domino" form

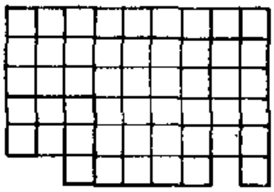
as in the Tough Hut problem, but of the form



Here the chessboard consideration can not be directly applied. However, using two names 0,1, one could achieve, for instance, the following invariant description:

0	0	1	0	0	1	0	0	1
1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1
1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0

With this description it is easy to show the impossibility of building the array

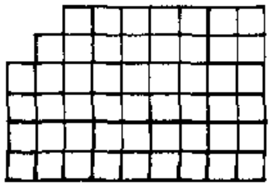


using a combination of "enlarged cells"

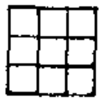
Obviously, we could not get an invariant description in this case using more than three names *tor* the cells. .For three names 0,1,2 one can get the following invariant "coloring"

0	1	2	0	1	2	0	1	2
1	2	0	1	2	0	1	2	0
2	0	1	2	0	1	2	0	1
0	1	2	0	1	2	0	1	2
1	2	0	1	2	0	1	2	0
2	0	1	2	0	1	2	0	1

which also indicates that a combination of such cells could not be used to construct the following array

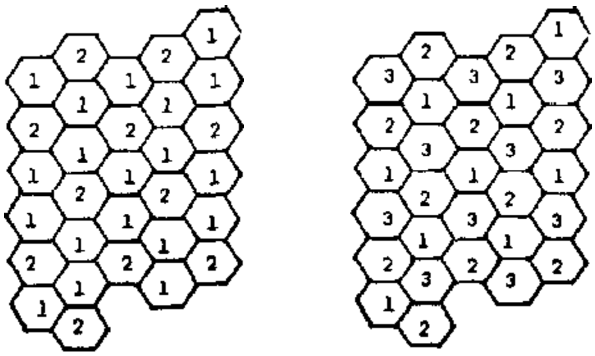


It can be easily seen that to get a suitable coloring in this case, it is necessary and sufficient to get a corresponding coloring of the following 3 x 3 array:



2, In the same manner it can be shown that in case of hexagonal cells it is impossible, using the blocks

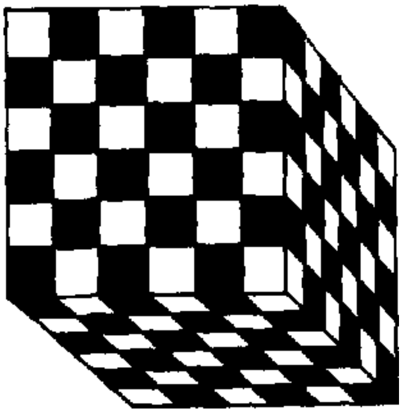
"to build" from them the following array of such cells, both using two or three names:

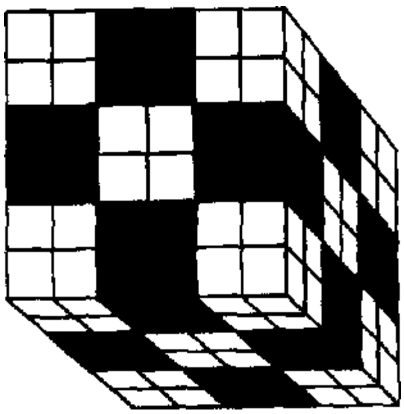


3. Suppose we are to prove that:
"A 6 x 6 x 6 cube cannot be constructed from bricks that are uniformly 1 x 2 x 4 in size."

Obviously we need not consider bricks so that they are not parallel to the aides of the cube. Similarily, a brick should be positioned only in an n-unit distance from the sides (n = 0,1,2,3). Taking this into account we can consider an assertion equivalent to the previous one:
"A 6 x 6 x 6 cube, composed from 1 x 1 x 1 subcubes cannot be presented as a sum of blocks of the subcubes which are 1 x 2 x 4 in size."

Using two names ("black" and "white") for subcubes we easily come, for instance, to the following Invariant descriptions (again, it is sufficient to consider a 4 x 4 x 4 system of the subcubes) —



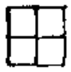


The latter description obviously does prove the assertion, as the number of black and white subcubes are not equal.

An approach to this problem in [5] is *one* that we call "insight." Indeed, the solution of the problem there is introduced with the phrase: "Imagine the cube to be made up of *2x2x2* cubes. Give each such cube the coordinates i,J,k (1 i,j,k 3), in the usual way. Color black the cubes whose coordinates add up to be an odd number; color the remaining cubes white." (It is interesting to note that quite similarly the "Tough Nut problem" in [5] from the very beginning is referred to as "the checkerboard -domino problem.")

4. We saw that the received description does not always help one in decision making. Moreover, does not always exists even one such invariant (simplified) description. As an example, consider the problem of building an array of squared cells as a sum of enlarged elements

It is obvious that with such blocks one could not find a non-trivial invariant description of the array

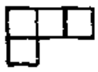


So in this way we unable to show that the array

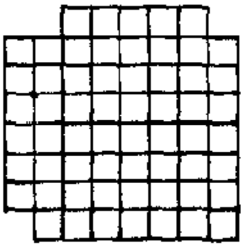
could not be presented as a sum of such blocks.

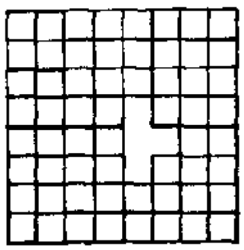
The last example is a case when local considerations do not permit one to get the "global answer." Bote, that in all examples used in this paper we used the idea of getting an invariant description in the Btage of assignment of small number of names to different elements. Of course, some other considerations(like simmetry) could be used too. Though these considerations are beyond of the scope of present paper, and will be given somewhere else.

5. Anticipating one of the possible questions concerning the last case, we will give one example more* Considering the following enlarged blocks



with the same local approach we could achieve as an invariant description the chessboard coloring that shows that it is impossible to build from such blocks either of the two following arrays;





Note on the "Chessboard Coloring"

The following discussion is based mainly on the author's intuition, and is not vital to the rest of the paper.

Previously we have mentioned that it is natural to think that it is the chessboard that gave us a hint of how to solve the Tough Nut problem[1]. We tried above to show that one could come to the idea of proper coloring from the purely local consideration in writing down the results of our actions while problem solving.

Maybe it would be interesting to think a little about why the chessboard was painted in the usual manner.

One can think that this coloring is not very necessary. Indeed, in all modern chess playing computer programs 3 this black-and -white coloring usually appears only on a computer display for the convenience of human beings and it is not taken into account in the computer program itself, even though in chess theory there exists notions of, for example, the black-field bishop and white- -field bishop. However, the black-field bishop always will keep track of the black field even if it doesn't see the color!

Below we will give *one of* the possible considerations, explaining how such a coloring could follow again from the local consideration of reducing the number of different names used for the chessboard fields. (In computer programs we have 64 names for the cells. In common chess notation, used, for instance, in the USSR, each row is numbered (1,2,...,8), and each column has a letter name (a,b,c,d,e,f,g,h).)

Note first that the white player in chess has an advantage of one movel However, it would not be true to say that white usually wins.

We think that the reason for it lies in the fact that the first-move-advantage is partially compensated for by the initial positioning of chess pieces, which is not perfectly symmetrical. The initial position is only mirror simmetrical (the kings are opposite each other).(The author discussed that hypothesis with an experienced chess player, who noticed that he felt that in a central symmetrical chess game the move of the white king pawn could almost surely lead to a win. Unfortunately, at this time, we haven't any experimental verification of this supposition.)

So, we beleive that we gave an unswer why in chess there is only mirror symmetry.

Note, now that this mirror symmetry requires a certain convention on how each player should put his pieces on the chessboard. It is not difficult to see,that for black- -and-white boards this convention has the most simple form, allowing partners to put their pieces on the board independently (otherwise, we have to use more than two nameB for board cells)•

Local Control of Taxis in a City

We are now going to consider the representation problem for quite a different area— that of traffic in the city. We will see that the idea of using a small number of names for elements is applicable here too,and that concrete assignment of the names could be again founded from local attemta to solve the problem (like positioning of a piece in Tough Nut) and a desire to get uniform description of our actions. The main difference from the previous problems we see in the fact that here "names" have quite an obvious physical interpretation. This makes the traffic problem more related to other problems of local control.4

In the last paragraph we will give a table of correspondence between the two classes of problems.

1. Imagine a city with cars of two kinds— taxis and others, A taxicab that knows apriori the exact location of demands (therefore for him there is no problem of"search for a demand") will be considered to be a non- -taxi. That is, in accordance with our definition, a free taxicab must Bearch for demands and be ignorant of demand locations. We will consider below only this idealistic taxi problem.

The most detailed description for thiB problem (analogous to that for the Tough Nut problem) is the case where all the carB in the city (taxi and non-taxi) have individual names (for instance , their registration numbers).

In the other extreme case (trivial description), all the cars are indistinguishable. Limiting them to two to three names, let us think about their appropriate assignment. The action here is the choice of direction for a free taxi when it approaches a street crossing. Let us consider one croee-street. Locally, it is obvious that if one of the taxis has Just left the place where it received a passenger and it comes across a free taxi, it is very important for the latter to learn, that the car he has just met is a taxi, and, that the taxi is occupied.

So from this local consideration, it follows that for the choice of actions (direction) one might use the following names:

* = non-taxi
0 = free taxi
1 = occupied taxi

Taking into account these names, we could hope that the taxi might find a demand quicker than in a case of a blind search. It is only a heuristic idea, gained from purely local considerations, and it is not clear now whether it would work for the problem as a whole.

2. This problem was simulated on a SIGMA 5 computer of the University of Washington. We will not present here the details of the program. As an example , downtown Seattle was used, with appropriate ("natural") locations of demand (figures near street crossings at the picture at the end of present paper).

The number of demands was initially fixed and the total time for their full satisfaction was computed (no new demands allowed).

The total time up to the moment of picking up the last passenger was calculated for different cases:

A, All the cars in the City have the same name (*) (each taxi moves in a purely random way).

B, There are non-taxis (*) and taxis (each taxi moves in a direction from where it sees the minimal number of taxis).

C. There are two namea: cars (*) and non-occupied taxis (0). (Taxis move in the direction from where they see the minimal number of free taxis.)

D. There three names: *, 0, and 1. (Taxi moves as in Case C, but if there are ties, it goes in the direction of the maximum number of occupied taxis.)

Initially all the taxis were placed in a certain location in the city (marked 4 at the picture). In the following table we give the total time (in a fixed measuring system) for different numbers of taxicabs in the city.

K	A	B	C	D
25	16.8	12.8	14.3	12.0
50	11.70	6.95	6,61	6.54
100	9.38	4.45	5.25	4.02
200	5.73	2.67	3.25	2.70

Thus we can see from the table that it is reasonable to use the names that we have gotten from purely local considerations, having in mind the global problem of minimization of total time required for fulfillment of all demands.

3. Perhaps it is appropriate to make several general remarks concerning this taxi problem itself. We are making use of the names of cars for local control of their movement around the city, but the very same names are convenient for passengers waiting for taxis. For this reason we have special signs *on* the care— taxi, free/occupied. Such signs are different for different countries and even for different cities and companies.

In London, a taxi has a very special "ancient" shape, quite different from the shapes of other cars. In New York a taxi ie a bright yellow color. In Seattle, Washington, the color and means of showing whether or not e given taxi is free depends to a certain degree on the company, and there is no standart rule. How we can see that these signs could be important for taxi drivers themselves and therefore it may be desirable to standardise them.

Conclusion

The distinction between two classes of problems, considered in first and the third paragraphs of this paper, is an obvious one. The first covers exclusively problems of mathematics of combinatorial nature; the second—a "real-life problem." Below we give the correspondence between them.

<u>Elements</u>

Squares of a board (cubes,hexagonals)	Cars

<u>Elementary Actions</u>

Positioning of a piece	Choice of the direction for a car to go at a street crossing

<u>One name—Trivial Description</u>

It is of interest only for the simplest problems (i.e.,to prove the impossibility of covering with two square dominos a board containing an odd number of squares).	The only possible decision—a random choice of the direction—Case A (the largest total time).

<u>Two or More Kames</u>

The local considerations of positioning for a piece is used for assignment of the names There are no "physical hints" for how to do it, instead we have a desire to get a uniform description of an elementary action, independent of the place on the board (invariant).	The thoughtful consideration of local situations and "physical hints" let us achieve a good assignment for the names. We need not consider any "stupid" assignment. Though, in automatical problem solving,considering several possible local situations and requiring the uniform description for them, we could automatically see that all of the busy cars should have the same name, which is different from names of *free* taxis. (Eventually, we have come to a uniform description in the sence that locally equivalent situations in different points of the City have the same description and the same choice of direction.)

<u>Acknowledgement</u>

We wish to express our thanks to Professor A.D.C.Holden and Professor M.Damborg, University of Washington, and to the members of Electrical Engineering and Computer Science Departments. They were always ready to help during my research visit to the University.

<u>References</u>

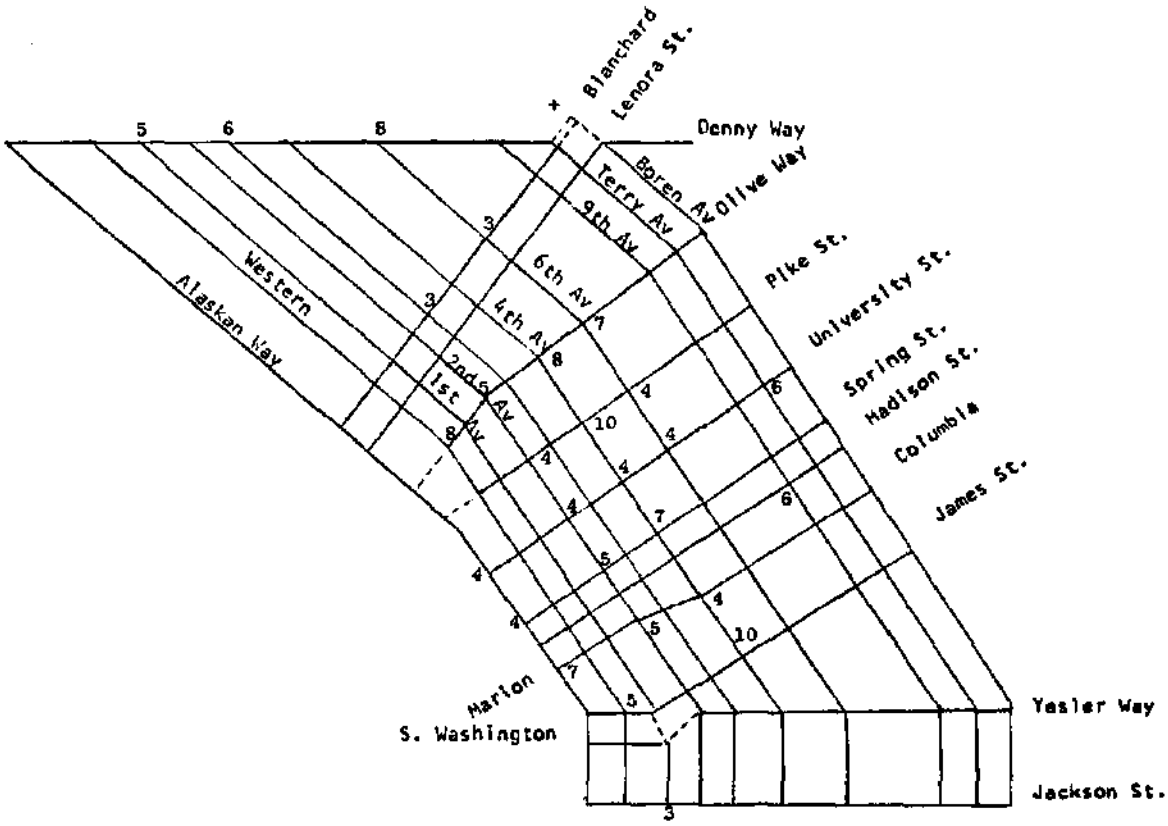
1. McCarthy, J., "A Tough Hut for Proof Procedures," Stanford University AI Project Memo No. 16, 1964.

2. Nilsson, N. "Problem Solving in Artificial Intelligence," 1971 McGraw-Hill.

3. SIGART Newsletter, No. 36 October 1972, pp 9-27.

4. Stefanuk, V.L., "Collective Behaviour of Automata and the Problems of Stable Local Control of a Large Scale System," Proceedings of the II IJCAI, London, 1971.

5. Simon, H.A.; Siklosey L.A. (editors), "Representation and Meaning," Prentice-Hall, Inc., 1972, pp 423-425.



Downtown

Seattle

(Simplified Plan)