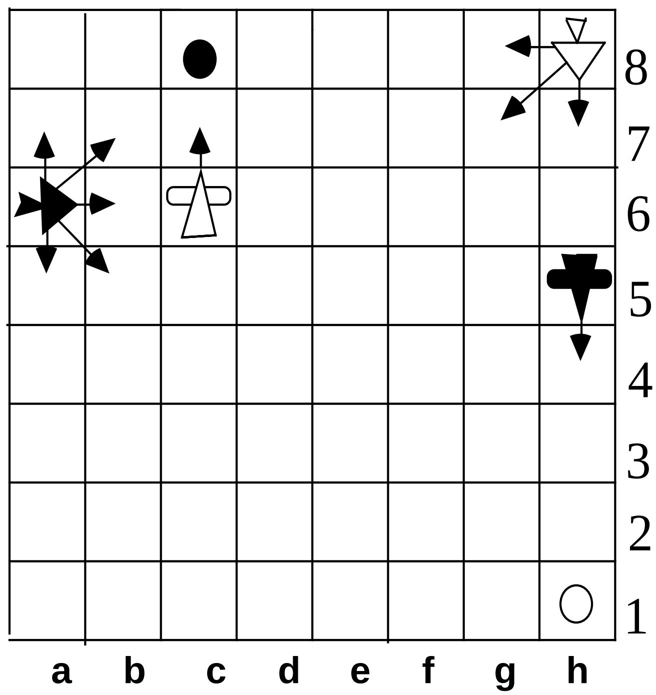
Optimization problem for aerospace robotic vehicles with serial alternating motions (2D/4A - Chapter 3)

2D Problem: SEARCH



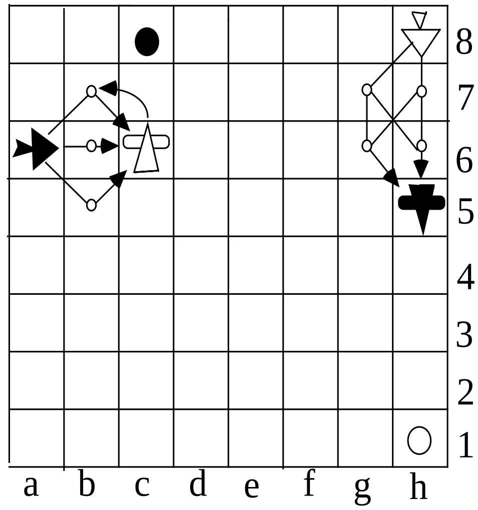
***Is there a strategy for the White to make a draw*?**

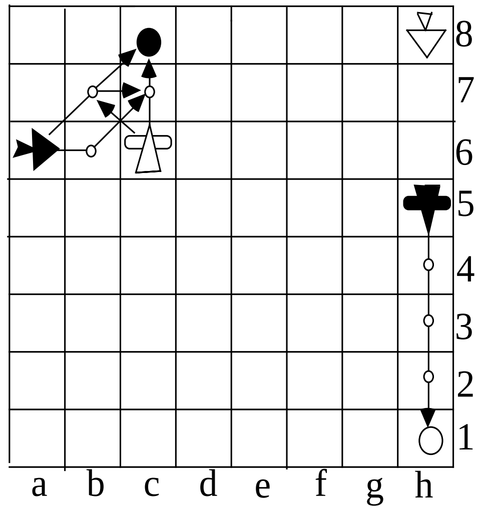
The specific question is as follows.
Is there an optimal strategy that provides one of the following?

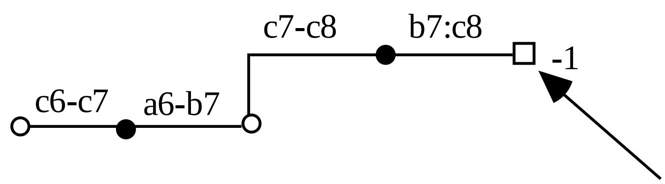
1. **Both BOMBERs hit their targets on subsequent time increments and stay safe for at least one time increment.**

2. **Both BOMBERs are destroyed before they hit their targets or immediately after that.**

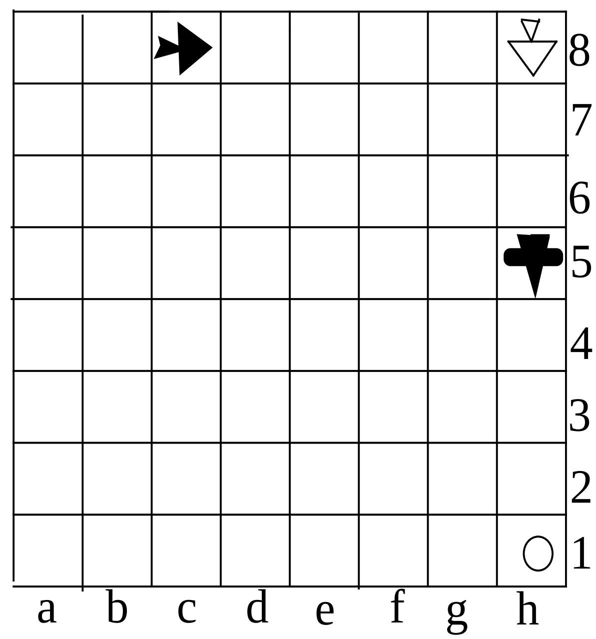
Zones in the Start State of the Robot Control Model

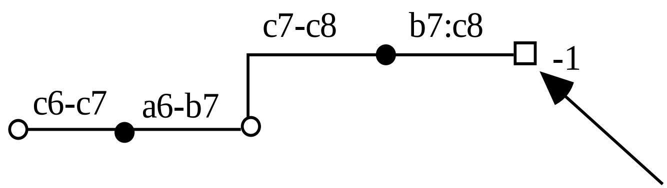




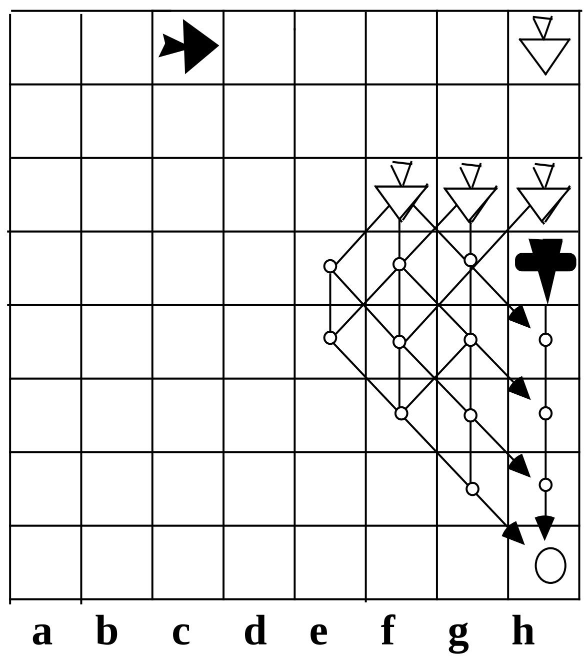


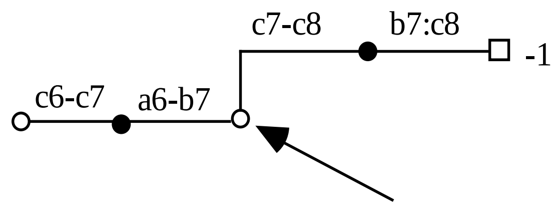
State where the branch was terminated and the domination Zone from h8 to c8 was detected.



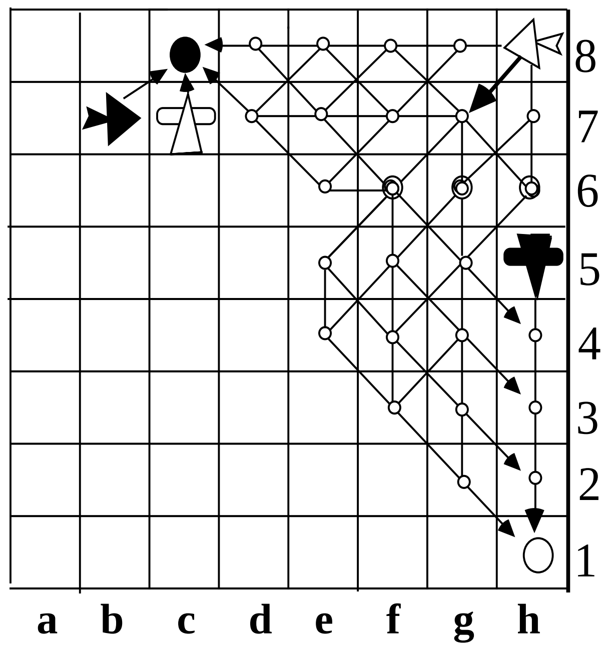


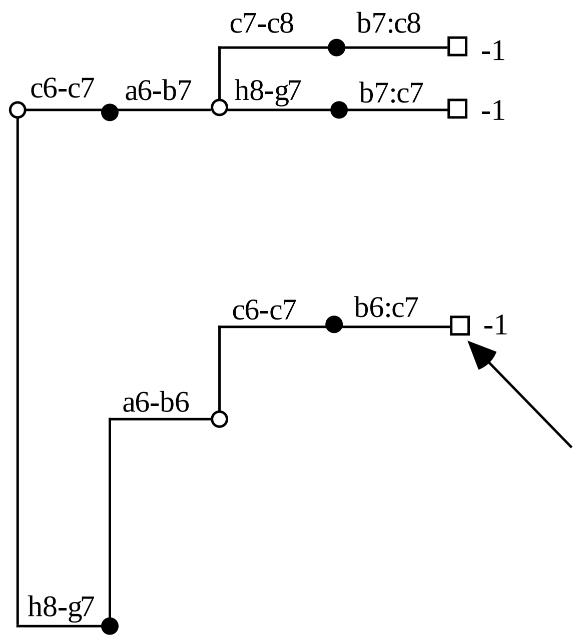
State where the branch was terminated and the domination Zone from h8 to c8 was detected.



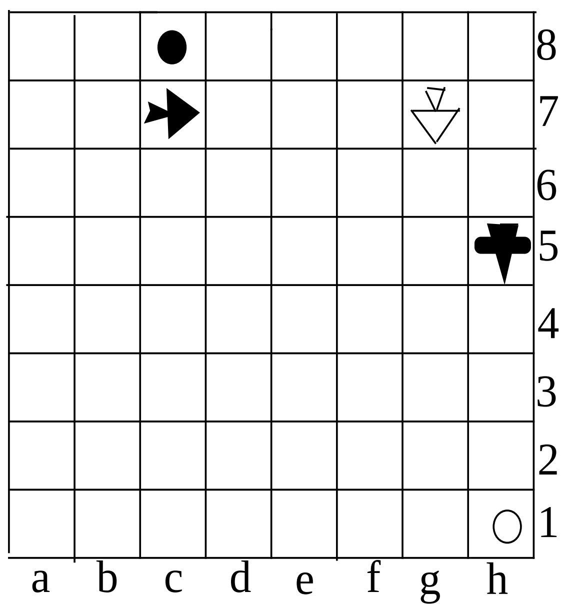


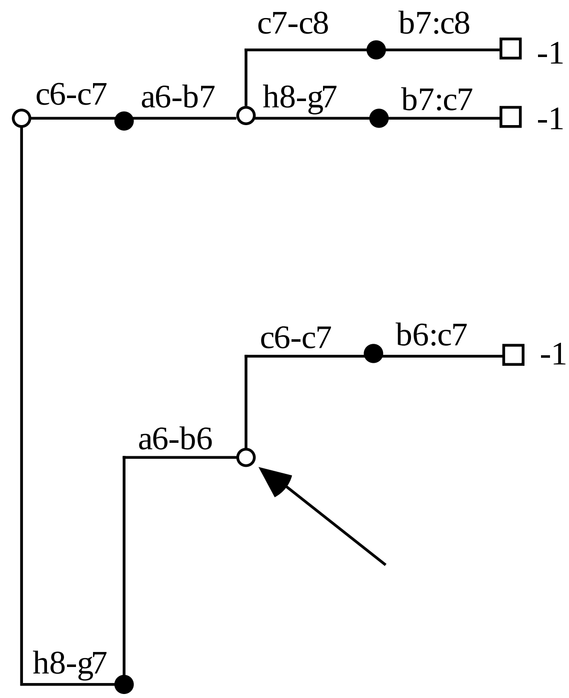
State where the domination Zone from h8 to c8 was included into the search



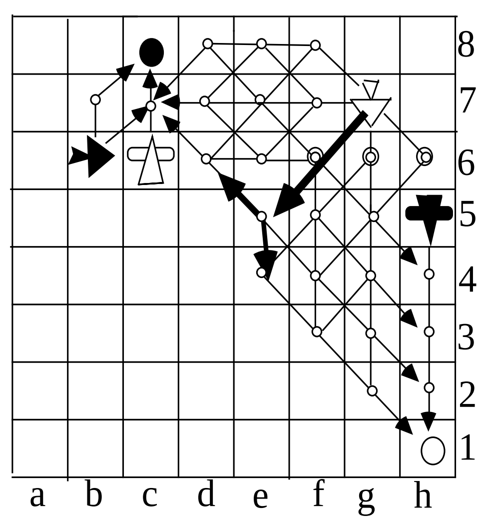


State where the domination Zone from g7 to c7 was detected

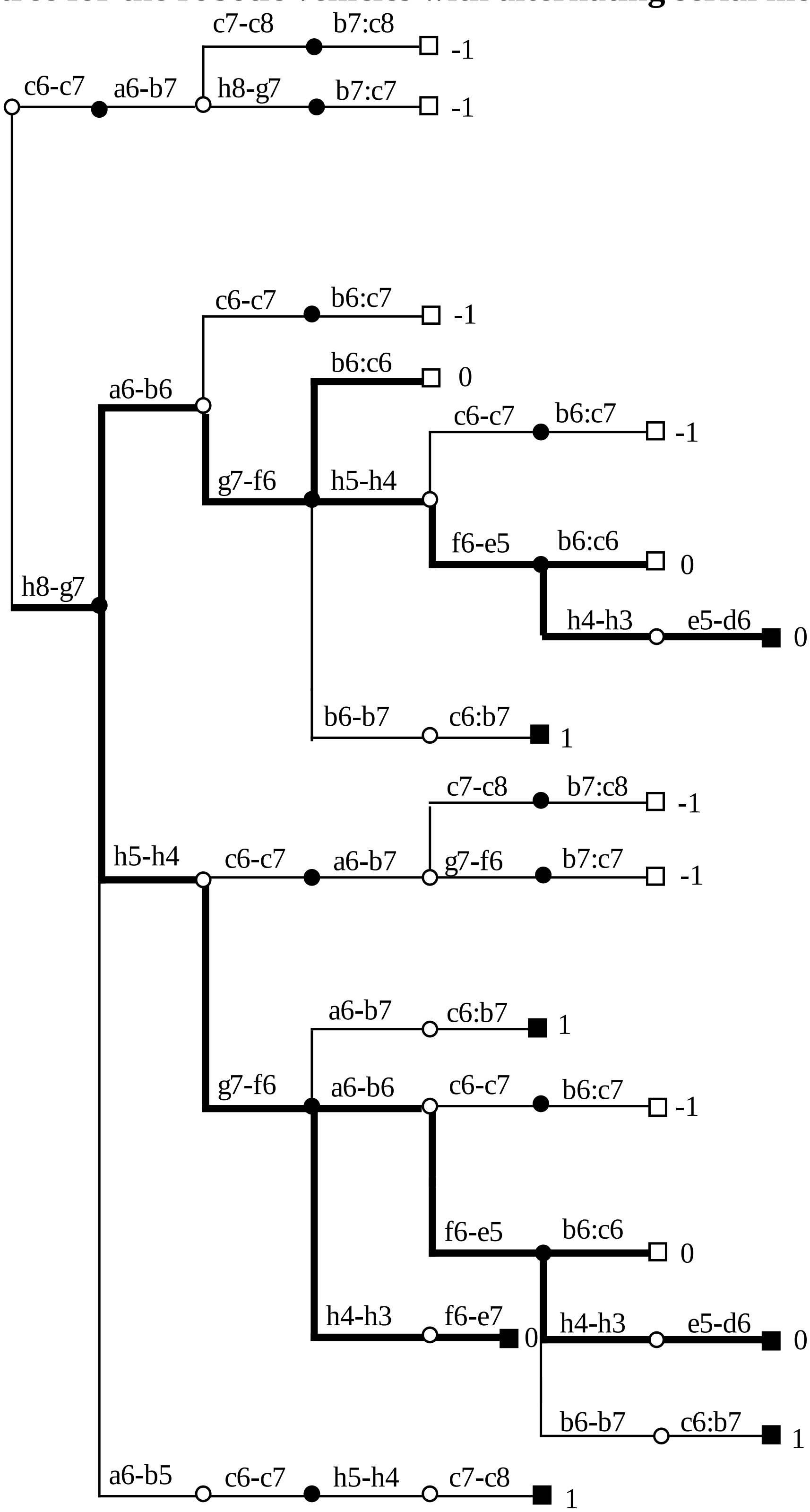




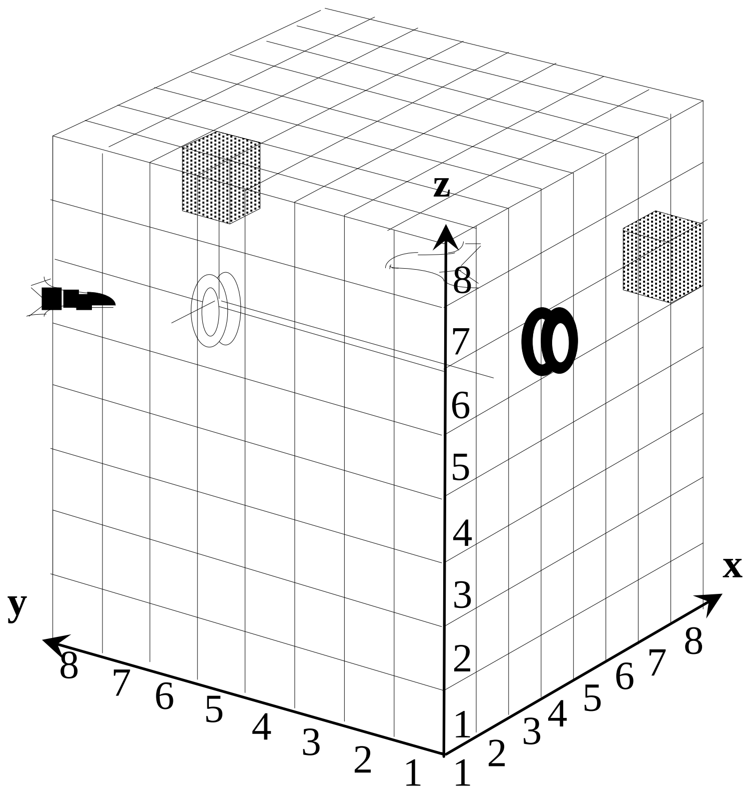
State where the domination Zone from g7 to c7 was included into the search



Search tree for the robotic vehicles with alternating serial motions.



Space Navigation Problem



Measurement of Distances in Abstract Board Games (ABG)

A Class of Problems

ABG

is the following eight-tuple

$$< X, P, R_p, \{ON\} , v, S_i, S_t, TR>$$

$X = \{x_i\}$ is a finite set of ***points***;

$P = \{p_i\}$ is a finite set of ***elements***; $P=P_1 \cup P_2$, $P_1 \cap P_2 = \emptyset$;

$R_p(x, y)$ is a family of binary relations of ***reachability*** in X ($x \in X$, $y \in Y$, $p \in P$); y is ***reachable*** from x for p;

ON(p) = x is a partial function of ***placement*** of elements P into X;

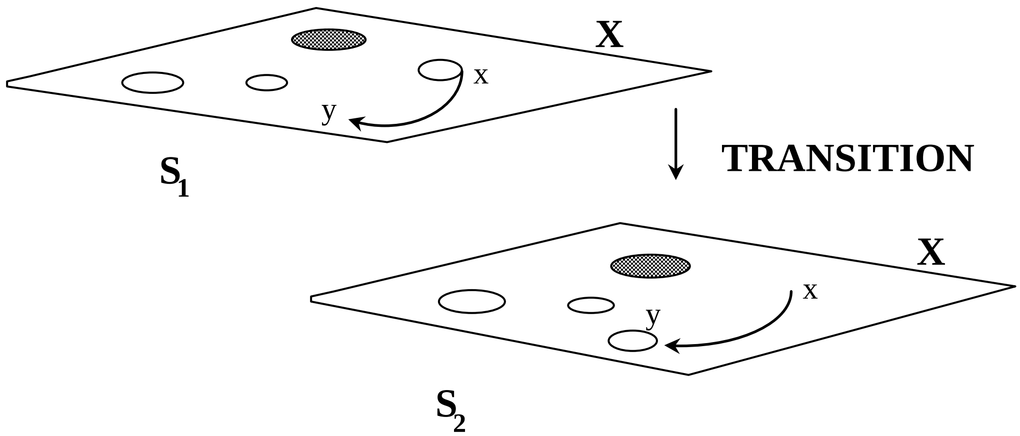
v > 0 is a real function, v(pi) are the ***values*** of elements;

S_i is a set of ***initial*** states of the system, a certain set of formulas $\{ON(p_i) = x_i\}$;

S_t is a set ***target*** states of the system (as S_i);

TR is a set of operators **TRANSITION(p, x, y)** for transition of the system from one state to another described as follows

precondition:	$ON(p) = x \wedge R_p(x, y)$
delete:	$ON(p) = x$, $ON(q) = y$
add:	$ON(p) = y$



X
x
y
S1
TRANSITION
X
x
y
S2

Example of Abstract Board Game (ABG)

$< X, P, R_p, ON, v, S_i, S_t, T >$	***The Chess Model***
$X=\{x_i\}$ is a finite set of ***"points"***;	64 **squares** of the board
$P=\{p_i\}$ is a finite set of ***"elements"***; $P=P_1 \cup P_2$, $P_1 \cap P_2 = \emptyset$;	**White** and **Black** pieces
$R_p(x, y)$ is a family of binary relations of ***"reachability"*** in X ($x \in X$, $y \in Y$, $p \in P$): **y** is ***reachable*** from **x** for **p**;	Given by rules of the game, permitting or forbidding a piece **p** to make a move from **x** to y
ON(p) = x is a partial function of ***"placement"*** of elements P onto X ;	Piece **p** stands on the square **x**
$v : P \rightarrow Z_+$, $v(p_i)$ are ***"values"*** of elements;	The values of pieces: N-3,B-3,R-5,Q-9,K-200
S_i is the description of the set of ***initial states*** of the system by a certain collection of Well Formed Formulas (**WFF**) of the first order predicate calculus: $\{ ON(p_i)=x_i \}$;	The initial position for for analysis or the starting position
S_t is the description of the set of ***target states*** of the system;	Positions which can be obtained from all possible mate positions by capture of the King
T is the description of operators **TRANSITION(p, x, y)** for transition of the system from one state to another in the form of the WFF of the ***applicability*** of this operator and the WFF of the ***transition***:	The move of the piece **p** from the square **x** to the square **y.**
remove ON(p)=x , ON(q)=y;	Capture of the piece **q**
add ON(p)=y.	

Interpretation of ABG for the robot control model

X represents the operational district which could be the area of combat operation broken into squares, e.g., in the form of the table 8 x 8, n = 64. It could be a space operation, where **X** represents the set of different orbits, etc.

P is the set of robots or autonomous vehicles. It is broken into two subsets $\mathbf{P_1}$ and $\mathbf{P_2}$ with opposing interests;

$\mathbf{R_p(x, y)}$ represent moving capabilities of different robots: robot **p** can move from point **x** to point **y** if $\mathbf{R_p(x, y)}$ holds. Some of them move fast and can reach point y (from x) in "one step", i.e., $\mathbf{R_p(x, y)}$ holds, and others can do that in *k* steps only, and many of them can not reach this point at all.

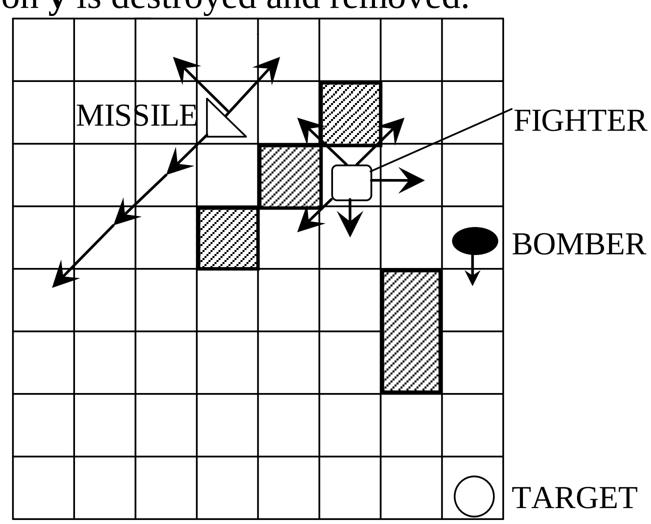
ON(p) = x, if robot p is at the point x;

v(p) is the value of robot p. This value might be determined by the technical parameters of the robot or by the immediate value of this robot for the given operation;

$\mathbf{S_i}$ is an arbitrary initial state of operation for analysis, or the starting state;

$\mathbf{S_t}$ is the set of target states. These might be the states where robots of each side reached specified points or $\mathbf{S_t}$ can specify states where opposing robots of the highest value are destroyed. The set of WFF $\{\mathbf{ON(p_j) = x_k}\}$ corresponds to the list of robots with their coordinates in each state.

TRANSITION(p, x, y) represents the move of the robot **p** from square **x** to square **y**; if a robot of the opposing side stands on **y**, a removal occurs, i.e., robot on **y** is destroyed and removed.



Measurement of Distances

A **map of the set X**

with respect to the point x and element p for the ABG is the mapping:

$$\mathbf{MAP_{x,p}: X \longrightarrow Z_+},$$

(where x is from X, p is from P), which is constructed as follows.

We consider a ***family of reachability areas*** from the point x, i.e., a finite set of the following nonempty subsets of X $\{\mathbf{M}^{\boldsymbol{k}}_{\mathbf{x,p}}\}$**:**

$\boldsymbol{k} = 1$: $\mathbf{M}^{\boldsymbol{k}}_{\mathbf{x,p}}$ is a set of points *m* ***reachable in one step*** from x: $R_p(x, m)=T$;

$\boldsymbol{k} > 1$: $\mathbf{M}^{\boldsymbol{k}}_{\mathbf{x,p}}$ is a set of points ***reachable in k steps and not reachable in k*-1 *steps***, i.e., points *m* reachable from points of $M^{\boldsymbol{k}-1}_{x,p}$ and not included in any $M^{\boldsymbol{i}}_{x,p}$ with numbers ***i*** less than ***k***.

Let $\mathbf{MAP_{x,p}(y)} = \boldsymbol{k}$, for y from $\mathbf{M}^{\boldsymbol{k}}_{\mathbf{x,p}}$ (***number of steps from x to y***).

In the remainder points let $\mathbf{MAP_{x,p}(y) = 2n}$, if $y \neq x$**;** $\mathbf{MAP_{x,p}(y) = 0}$, if y=x.

It is easy to verify that **map of the set X** for the element p from P defines an ***asymmetric distance function*** on X:

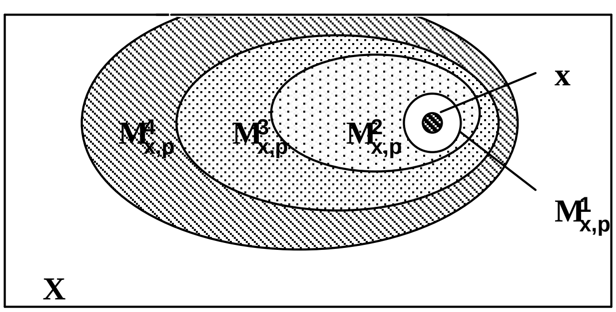
1. $MAP_{x,p}(y) > 0$ for $x \neq y$; $MAP_{x,p}(x)=0$;
2. $MAP_{x,p}(y) + MAP_{y,p}(z) \geq MAP_{x,p}(z)$.

If R_p is a symmetric relation,

3. $MAP_{x,p}(y) = MAP_{y,p}(x)$,

In this case each of the elements p from P specifies on X its ***own metric*.**

Reachability areas



Values of $\mathbf{MAP_{f6,FIGHTER}}$
(robot control model)

5	4	3	2	2	2	2	2
5	4	3	2	1		1	2
5	4	3	2		0	1	2
5	4	3		1	1	1	2
5	4	3	2	2	2		2
5	4	3	3	3	3		3
5	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5

Tables 15×15 (see p. 219)

The game of chess can be interpreted as ABG. To generate trajectories we have to investigate the geometry of the chess system. In chess function $MAP_{x,p}(y)$ yields the number of moves necessary for the piece p from square x to reach square y along the shortest path. Because of the symmetry of the relation R_p in this model, $MAP_{x,\,p}(y)$ specifies the *metric* on the chess board characteristic of each kind of piece. Even for a Pawn with more complex symmetry,

$$R_p(x, y) = R_q(y, x),$$

where p and q are the black and white Pawns, a sophisticated symmetry holds:

$$MAP_{x,\,p}(y) = MAP_{y,\,q}(x).$$

Hence, MAP as a function can be used as a *ruler* to measure *distances* in this system for arbitrary elements (pieces).

When implementing the Language of Trajectories for the chess problem, it was found necessary to specify the function MAP by a table in order to increase the efficiency of the program PIONEER. The tables are 7 in number, of size 15×15 each. The entries are constructed as follows: 0 is entered into the central square; the remaining squares are filled with numbers equal to the number of moves necessary for the piece to reach the given square from the central square along the shortest path.

							0							

7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	6	6	6	6	6	6	6	6	6	6	6	6	6	7
7	6	5	5	5	5	5	5	5	5	5	5	5	6	7
7	6	5	4	4	4	4	4	4	4	4	4	5	6	7
7	6	5	4	3	3	3	3	3	3	3	4	5	6	7
7	6	5	4	3	2	2	2	2	2	3	4	5	6	7
7	6	5	4	3	2	1	1	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	1	1	2	3	4	5	6	7
7	6	5	4	3	2	2	2	2	2	3	4	5	6	7
7	6	5	4	3	3	3	3	3	3	3	4	5	6	7
7	6	5	4	4	4	4	4	4	4	4	4	5	6	7
7	6	5	5	5	5	5	5	5	5	5	5	5	6	7
7	6	6	6	6	6	6	6	6	6	6	6	6	6	7
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

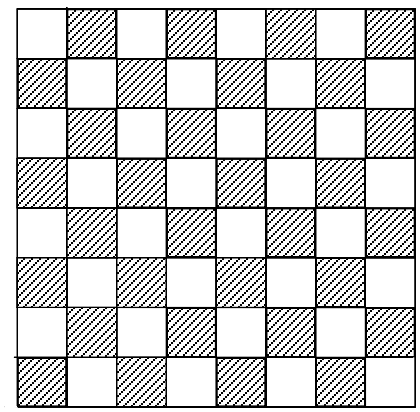
15x15 table for a King

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

15x15 table for a Bishop.

7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	6	6	6	6	6	6	6	6	6	6	6	6	6	7
7	6	5	5	5	5	5	5	5	5	5	5	5	6	7
7	6	5	4	4	4	4	4	4	4	4	4	5	6	7
7	6	5	4	3	3	3	3	3	3	3	4	5	6	7
7	6	5	4	3	2	2	2	2	2	3	4	5	6	7
7	6	5	4	3	2	1	1	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	1	1	2	3	4	5	6	7
7	6	5	4	3	2	2	2	2	2	3	4	5	6	7
7	6	5	4	3	3	3	3	3	3	3	4	5	6	7
7	6	5	4	4	4	4	4	4	4	4	4	5	6	7
7	6	5	5	5	5	5	5	5	5	5	5	5	6	7
7	6	6	6	6	6	6	6	6	6	6	6	6	6	7
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

15x15 table for a King



Chess Board

Imagine the following computation procedure. Assume an 8×8 table superimposed on the 15×15 table in such a way that square x = (x_1, x_2) coincides with the central square of the 15×15 table. Next, assume that the 8×8 table be transparent, then from the corresponding squares we read off the values of $MAP_{x,p}$, i.e., the values of the actual distances (in number of moves) of these squares from square x.

An example of such a superimposition of tables for x = (3, 2), being c2 and p = Rook is shown below.

2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	15
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	14
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	13
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	12
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	11
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	10
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	9
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	8
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	7
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	6
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	5
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	4
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	3
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Figure 2. Superimposition of 8x8 and 15x15 tables for a Rook on c2.

These tables may be gathered for uniformity, into a single three-dimensional array T15(v_1, v_2, f) of size 15×15×7. For all x in X, x = (x_1, x_2), x_1 = 1, 2,..., 8, x_2 =1, 2,..., 8, where x_1 and x_2 are the numbers of the files and ranks of the chess-board. Values of v = (v_1, v_2) are the numbers of files and ranks of the respective 15×15 table. Then

$$MAP_{X,\,p}(y) = T15(v_1, v_2, f), \tag{1}$$

where

$$x = (x_1, x_2),$$
$$y = (y_1, y_2),$$

What are the formulas for v_1 and v_2?

$$MAP_{x,\,p}(y) = T15(v_1, v_2, f), \qquad (1)$$

where

$$x = (x_1, x_2),$$
$$y = (y_1, y_2),$$

$$v_1 = 8 - x_1 + y_1,$$
$$v_2 = 8 - x_2 + y_2,$$

f = f(p) is the type of the piece p (King, Rook, etc.).

Seven 15×15 tables specify on X *seven different metrics*.