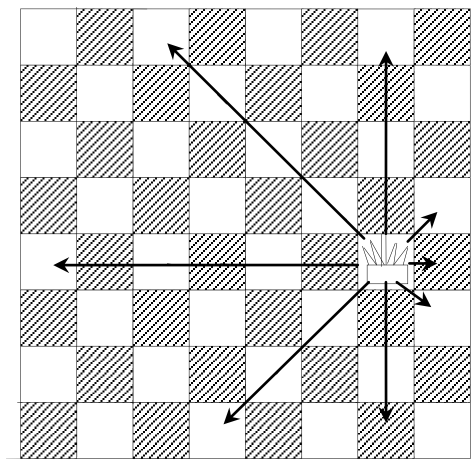
Assignment 12
Due: Saturday, April 30, 2011 (important for the Final)

(a) Show generation of the shortest trajectories for the King from a4 to h4. Show at least **four steps** of the generation of one of the trajectories **in details** (show all sets and functions).

(b) How many shortest trajectories from a4 to h4 exist (**extra credit**)?

(c) Does the grammar $\mathbf{G_t^{(1)}}$ generate all of them? Explain.

(d) Generate (in details) all the shortest trajectories for the Queen for the following cases (reachability relations for the Queen are shown below):
(d1) from d2 to b7;
(d2) from e2 to h5.



Final Review: Sample Problems

1. Construct an indexed grammar generating the following language: $\{0^n \mid n$ is a power of 2$\}$.

---

2. Construct an attribute grammar G = (V_T, V_N, P, S) with the following <u>terminals</u>

$$V_T = \{c_e , *, + , (,) , ANSWER_d\},$$

<u>nonterminals</u>

$$V_N = \{E_a, T_b, P_c, S \}$$

and <u>generated attributes</u> **a, b, c, d, e**, which can compile the following expressions:

$$(c_{e1} + c_{e2})*(c_{e3} + c_{e4})$$

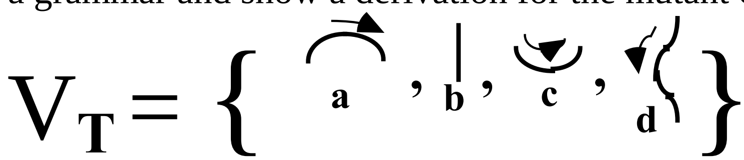
Compile the expression

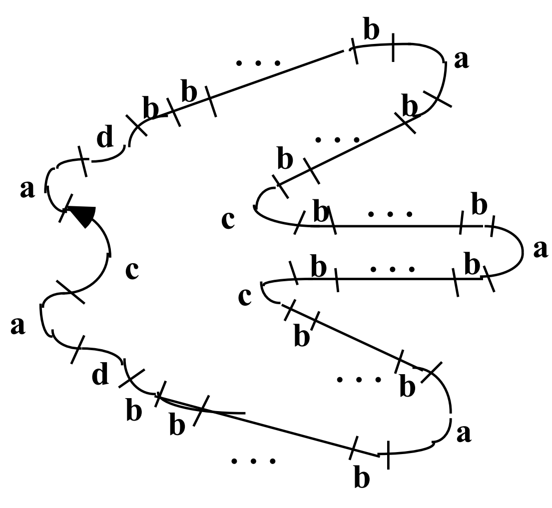
$$(c_2+c_5)*(c_{11}+c_3)ANSWER_d,$$

show derivation tree, and find the value of attribute d.

---

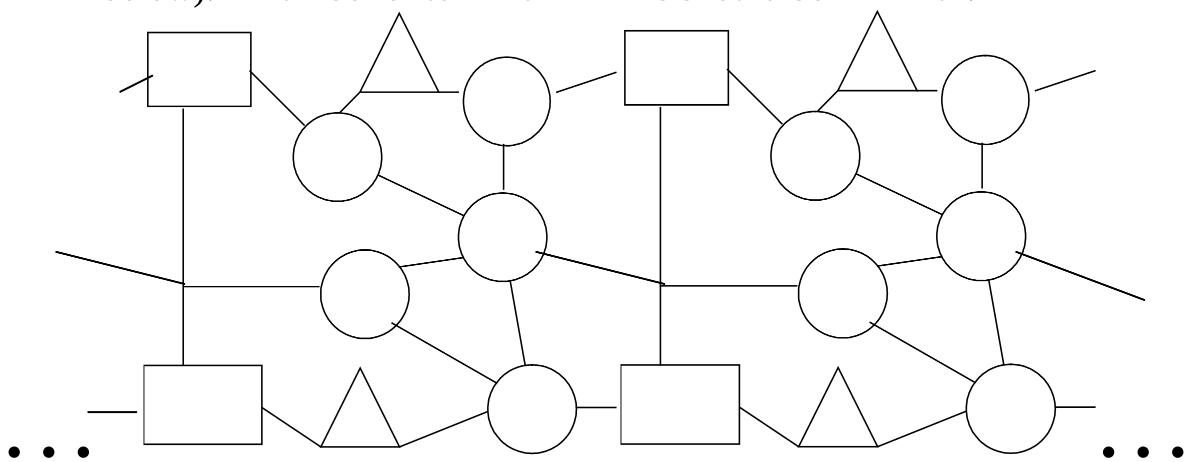
3. Construct a grammar and show a derivation for the mutant chromosome:





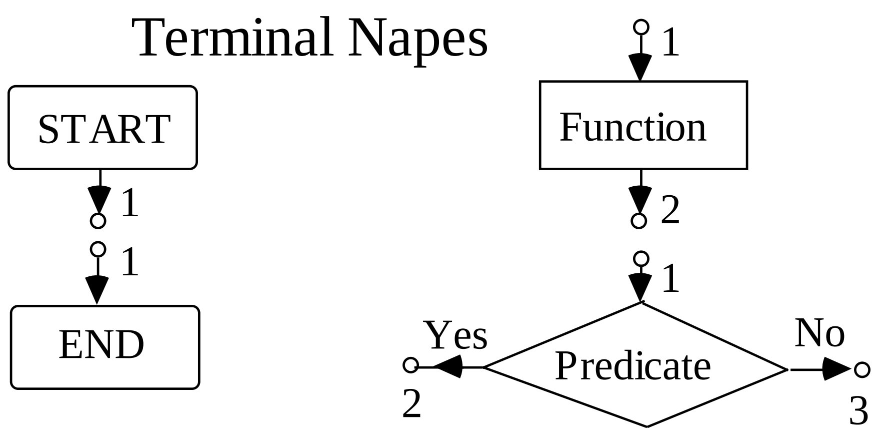
---

4. Construct terminal NAPEs for generating this kind of networks. They are called Chains of Data Flow Diagrams of arbitrary length. Construct a Plex Grammar generating the following Chain of Data Flow Diagrams (see below). A number of terminal NAPEs should be ***minimal***.



---

5. Show all the NAPEs which can be generated by the following plex grammar.



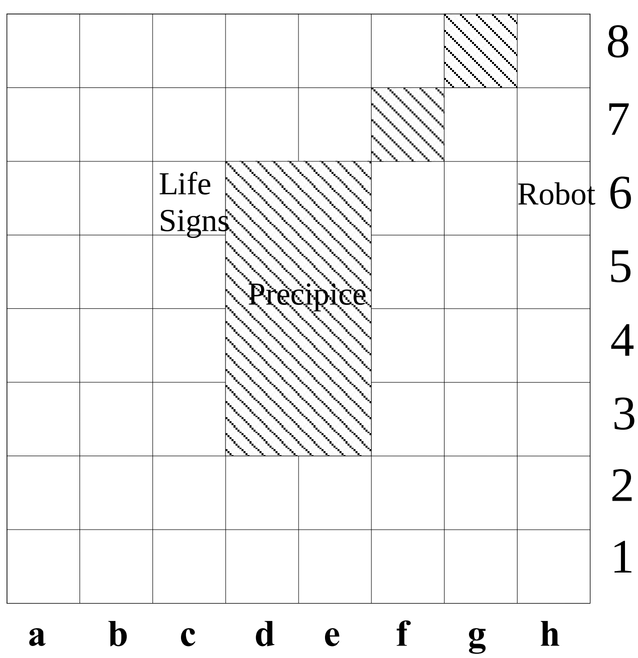
<P>(1,2) —> <FUNCT><FUNCT>(21)(10, 02)
<P>(1,2) —> <PRED><P>(21, 12)(21, 30)
<P>(1,2) —> <PRED><P><P>(210, 301, 022)(100, 022)
<PROGRAM> —> <START><P><END>(110, 021)

---

6. Generate all the shortest trajectories for the robot called <u>Bishop</u> from **a1** to **b6** employing the Grammar of Shortest Trajectories. Show values of all the important functions and sets. (Table TAB15 for Bishop is shown below.)

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

7. Assume that robot **Explorer** landed on Mars at the square **h6**. It has a map of Mars (shown below) taken from the orbit. It can reach any next square in one step (including diagonal moves). **Explorer** cannot move through the squares, which are in the precipice (shaded), but it can cross the edges of the precipice. The robot received a radio-message from the Earth to investigate <u>**all** the shortest paths</u> to the location of possible life signs (point **c6**). Its battery has limited capacity and it cannot recharge. Find **all** these paths employing the Grammar of Shortest Trajectories. Show values of all the important functions and sets.



Introduction to Linguistic Geometry

Class of Problems

Abstract Board Game (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***;

$P = P_1 \cup P_2$ is a finite set of ***pieces***, $P_1 \cap P_2 \neq \varnothing$ (opposing sides);

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

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

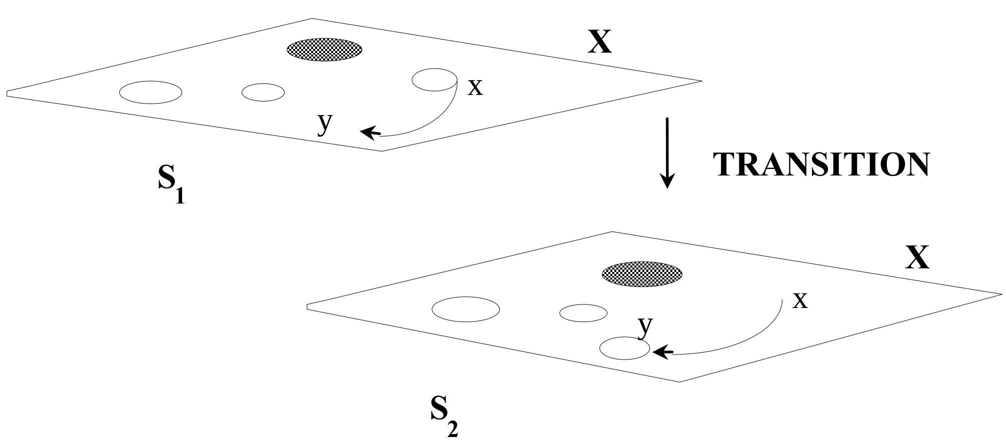
v > 0 is a real function, **v**(p) are the ***values*** of pieces;

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



Measurement of Distances in the ABG

A **map of the set X**

relative 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 $\{\mathbf{M^{k}_{x,p}}\}$**:**

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

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

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

In the remainder points let $\mathbf{MAP_{x,p}(y) = 2n}$, if y≠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:

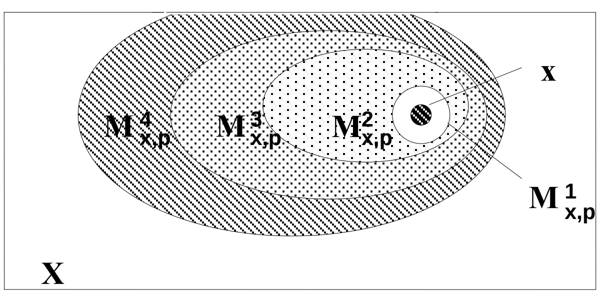
1. $MAP_{x,p}(y) > 0$ for $x \neq y$; $MAP_{x,p}(x) = 0$;
2. $MAP_{x,p}(y) + MAP_{y,p}(z) \geq MAP_{x,p}(z)$.

If R_p is a symmetric relation,

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

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

Reachability areas



Values of $\mathbf{MAP_{f6,\ King}}$

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

Language of Trajectories. The Shortest Path.

A *trajectory*
for a piece p of P with the beginning at x of X and the end at y of X (x≠y) with a length ***l*** is the following string of symbols with parameters, points of X:

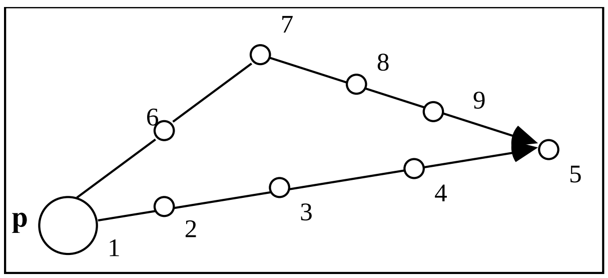
$$t_o = \boldsymbol{a}(x)\boldsymbol{a}(x_1)\ldots\boldsymbol{a}(x_l).$$

Here each successive point x_{i+1} is reachable from the previous point x_i:

$$R_p(x_i, x_{i+1}) \text{ holds for } i = 0, 1, \ldots, \boldsymbol{l}-1;$$

element p stands at the point x: ON(p) = x.
We denote $t_p(x, y, \boldsymbol{l})$ the set of trajectories in which p, x, y, and ***l*** are the same. $\boldsymbol{P}(t_o) = \{x, x_1, ..., x_l\}$ is the set of parameter values of the trajectory t.

A ***shortest trajectory*** t
of $t_p(x, y, l)$ is the trajectory of minimum length for the given beginning x, end y and element p.



Interpretation of shortest and admissible trajectories

Reasoning informally, an analogy can be set up: the shortest trajectory is an analogous to a straight line segment connecting two points in a plane. Let us consider an analogy to a k-element segmented line connecting these points.

An ***admissible trajectory of degree k***
is the trajectory which can be divided into k shortest trajectories; more precisely there exists a subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{k-1}}\}$ of $\boldsymbol{P}(t_o)$, $i_1 < i_2 < \ldots < i_{k-1}$, $k \le l$, such that corresponding substrings

$$\boldsymbol{a}(x_o)\ldots\boldsymbol{a}(x_{i_1}),\ \boldsymbol{a}(x_{i_1})\ldots\boldsymbol{a}(x_{i_2}),\ \ldots,\ \boldsymbol{a}(x_{i_{k-1}})\ldots\boldsymbol{a}(x_l)$$

are the shortest trajectories.

A ***Language of Trajectories*** $\mathbf{L_t^H(S)}$
for the ABG in a state S is the set of all the trajectories of the length less or equal H.

Controlled grammar of shortest trajectories $G_t^{(1)}$

L	Q	Kernel	F_T	F_F
1	$\boldsymbol{Q_1}$	$\boldsymbol{S}(x, y, l) \to \boldsymbol{A}(x, y, l)$	*two*	ø
$2_{\boldsymbol{i}}$	$\boldsymbol{Q_2}$	$\boldsymbol{A}(x, y, l) \to \boldsymbol{a}(x)\boldsymbol{A}(\boldsymbol{next_i}(x, l), y, \boldsymbol{f}(l))$	*two*	3
3	$\boldsymbol{Q_3}$	$\boldsymbol{A}(x, y, l) \to \boldsymbol{a}(y)$	ø	ø

$\boldsymbol{V_T} = \{\boldsymbol{a}\}$
$\boldsymbol{V_N} = \{\boldsymbol{S}, \boldsymbol{A}\}$
$\boldsymbol{V_{PR}}$

Pred = $\{\boldsymbol{Q_1}, \boldsymbol{Q_2}, \boldsymbol{Q_3}\}$,

$\boldsymbol{Q_1}(x, y, l) = (MAP_{x,p}(y) = l) \quad (0 < l < n)$

$\boldsymbol{Q_2}(l) = (l \geq 1) \quad \boldsymbol{Q_3} = T$

Var = $\{x, y, l\}$

$F = \{\boldsymbol{f}, \boldsymbol{next}_1, \ldots, \boldsymbol{next}_n\}$ $(n = |X|)$,

$\boldsymbol{f}(l) = l - 1$, $D(f) = \mathbf{Z}_+ - \{0\}$, $\boldsymbol{next_i}$ is defined below;

$\boldsymbol{E} = \mathbf{Z}_+ \cup X \cup P$
Parm: $S \to Var$, $A \to Var$, $\boldsymbol{a} \to \{x\}$
$\boldsymbol{L} = \{1, 3\} \cup two$, $two = \{2_1, 2_2, \ldots, 2_n\}$

At the beginning of derivation: $x=x_o$, $y=y_o$, $l=l_o$, $x_o \in X$, $y_o \in X$, $l_o \in \mathbf{Z}_+$, $p \in P$

Function $\boldsymbol{next_i}$ is defined as follows:

$D(\boldsymbol{next_i}) = X \times \mathbf{Z}_+ \times X^2 \times \mathbf{Z}_+ \times P$

$MOVE_l(x)$ is the intersection of the following sets:

$ST_1(x)$, $ST_{l_o-l+1}(x_o)$ and SUM, where

$SUM = \{v \mid v \text{ from } X, MAP_{x_o,p}(v) + MAP_{y_o,p}(v) = l_o\}$,

$ST_k(x) = \{v \mid v \text{ from } X, MAP_{x,p}(v) = k\}$,

If

$MOVE_l(x) = \{m_1, m_2, \ldots, m_r\} \neq \varnothing$

then

$\boldsymbol{next_i}(x, l) = m_i$ for $i \leq r$;

$\boldsymbol{next_i}(x, l) = m_r$ for $r < i \leq n$,

otherwise

$\boldsymbol{next_i}(x, l) = x$.

Interpretation of the algorithm for $next_i$ for the grammar $G_t^{(1)}$.

$MOVE_l(x)$ is the intersection of the following sets:

$ST_1(x)$, $ST_{l_o-l+1}(x_o)$, and SUM, where

$SUM = \{v \mid v \text{ from } X, MAP_{x_o,p}(v) + MAP_{y_o,p}(v) = l_o\}$,

$ST_k(x) = \{v \mid v \text{ from } X, MAP_{x,p}(v) = k\}$,

If

$MOVE_l(x) = \{m_1, m_2, \ldots, m_r\} \neq \emptyset$

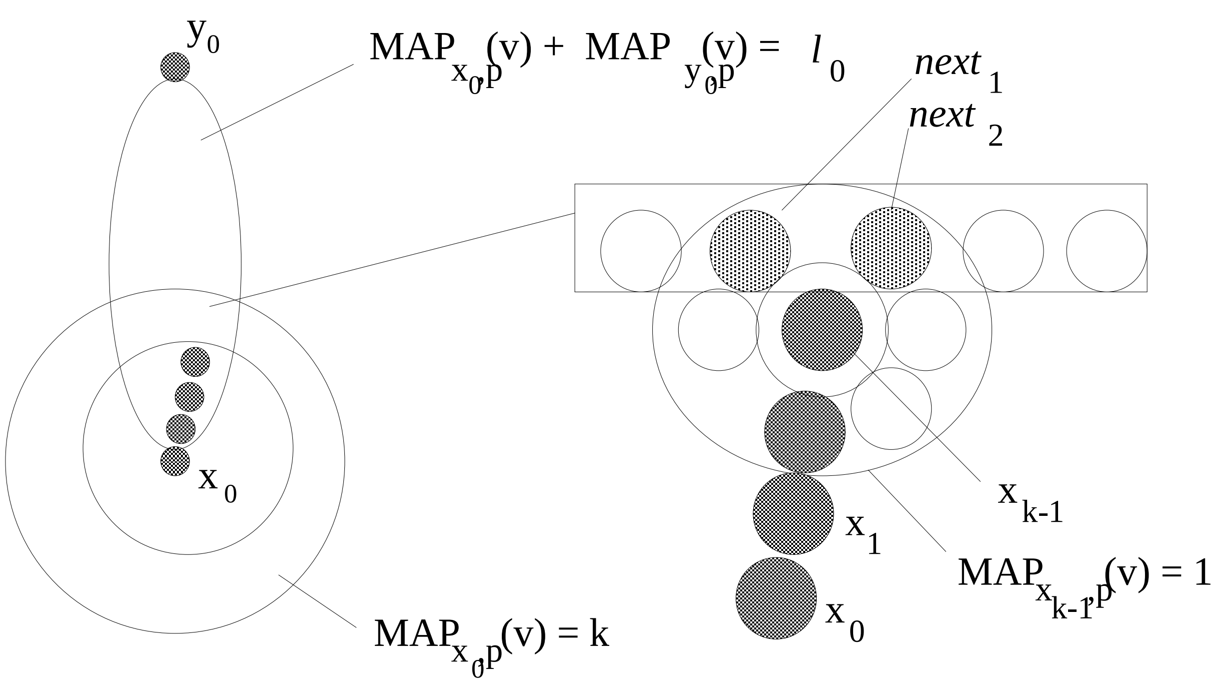
then

$next_i(x, l) = m_i$ for $i \le r$;

$next_i(x, l) = m_r$ for $r < i \le n$,

otherwise

$next_i(x, l) = x$.



Theorem about shortest trajectories

The shortest trajectories from point x_o to point y_o of the length l_o for the element p on x (i.e., ON(p) = x) exist if and only if the distance of these points is equal l_o:

$$MAP_{x_o,p}(y_o) = l_o,$$

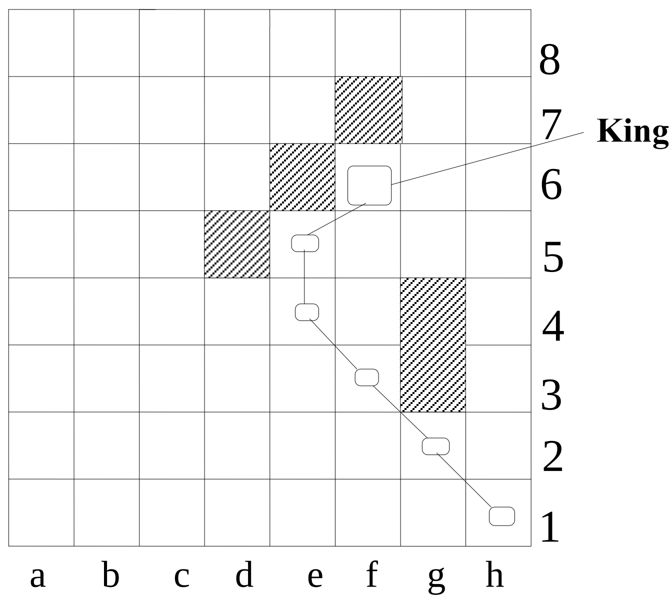
where $l_o < 2n$, n is the number of points in X.

If the relation R_p is symmetric, i.e., for all x from X, y from X and p from P

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

then all the shortest trajectories $t_p(x_o, y_o, l_o)$ can be generated by the grammar $\mathbf{G_t^{(1)}}$.

Generation of the shortest trajectory
a(f6)***a***(e5)***a***(e4)***a***(f3)***a***(g2)***a***(h1)
for the robot King



Generation of the shortest trajectory

$$\boldsymbol{S}(\mathbf{f6,\ h1,\ 5})\ ^{\mathbf{1}}\!\!\longrightarrow \boldsymbol{A}(\mathbf{f6,\ h1,\ 5})^{\mathbf{2_1}}\!\!\longrightarrow \boldsymbol{a}(\mathbf{f6})\boldsymbol{A}(\boldsymbol{next_1}(\mathbf{f6,\ 5}),\ \mathbf{h1,\ 4})$$

$$\mathbf{SUM} = \{v \mid v \in \mathbf{X},\ \ \mathbf{MAP}_{\mathbf{f6,\ King}}(v) + \mathbf{MAP}_{\mathbf{h1,\ King}}(v) = \mathbf{5}\}$$

$MAP_{f6,KING}$

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

$MAP_{h1,KING}$

8	7	7	7	6	7	7	7
7	7	6	6	6		6	6
7	6	6	5		5	5	5
7	6	5		4	4	4	4
7	6	5	4	3	3		3
7	6	5	4	3	2		2
7	6	5	4	3	2	1	1
7	6	5	4	3	2	1	0

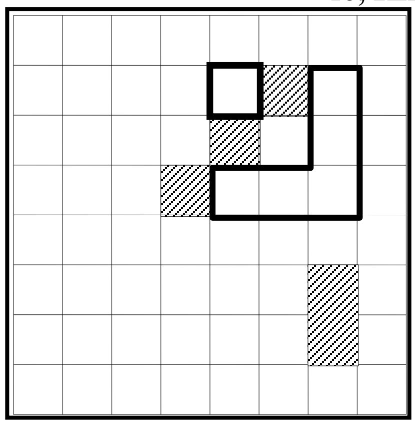
SUM

					5		
				5	5	5	
				5	5		5
					5		5
						5	5
							5

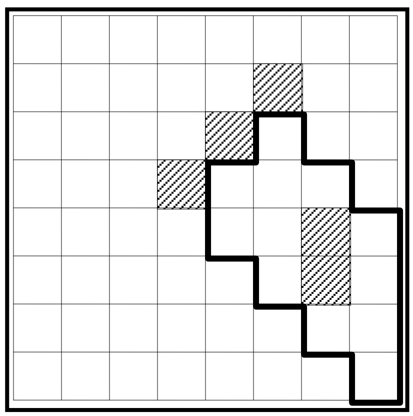
Generation of the shortest trajectory (continued)

$MOVE_5$(f6) is the intersection of
ST_1(f6), ST_{5-5+1}(f6) = ST_1(f6) and SUM

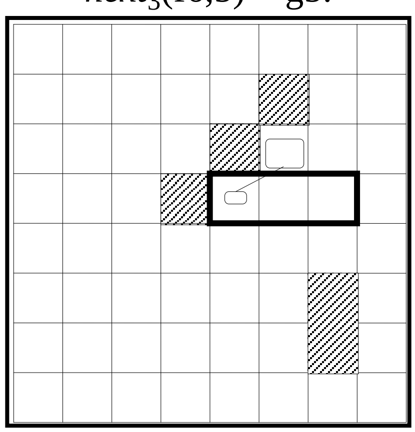
ST_1(f6) ={v | v from X, $MAP_{f6, King}(v)$=1}



SUM



$MOVE_5$(f6)={e5, f5, g5}
$next_1$(f6, 5) = e5, $next_2$(f6, 5) = f5,
$next_3$(f6,5) = g5.



Generation of the shortest trajectory (continued)

***a*(f6)*A*(e5,h1,4) ^{2}1—>*a*(f6)*a*(e5)*A*(*next*$_1$(e5,4),h1,3)**

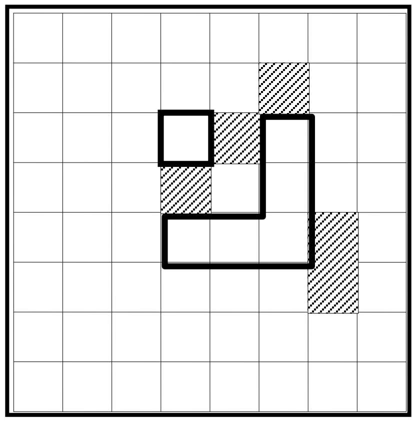
$MOVE_4$(e5) is the intersection of

SUM,

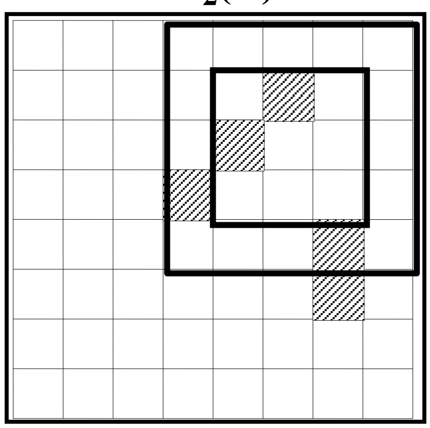
$ST_1(e5)=\{v \mid v \in X, MAP_{e5,\ King}(v)=1\}$,

and $ST_{5-4+1}(f6)= ST_2(f6)=\{v \mid v \in X, MAP_{f6,\ King}(v)=2\}$

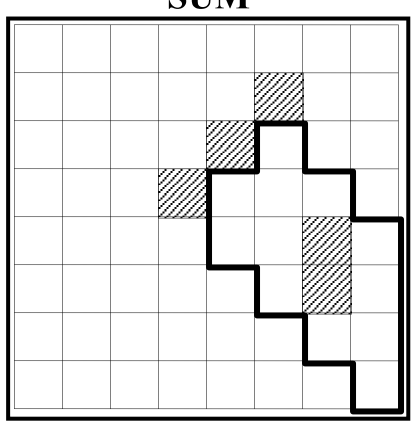
ST_1(e5)



ST_2(f6)



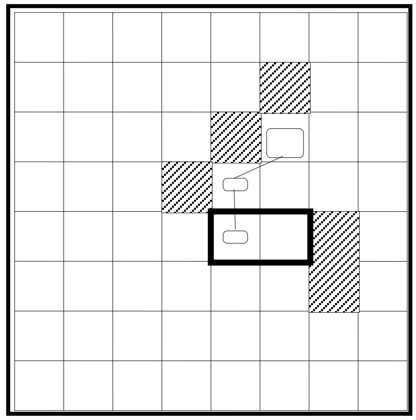
SUM



Generation of the shortest trajectory (continued)

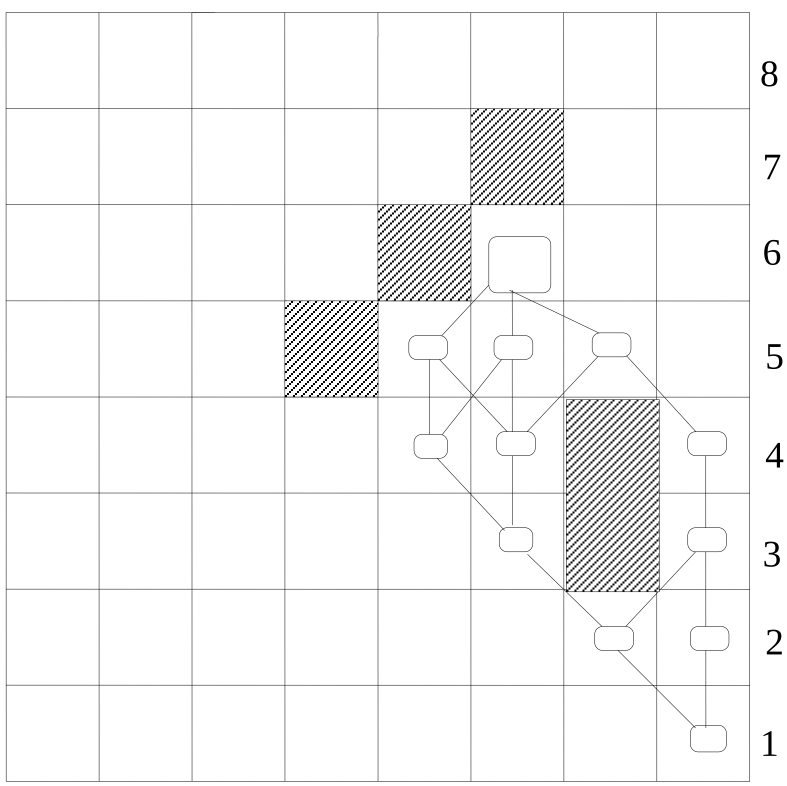
$MOVE_4(e5) = \{e4, f4\}$

$next_1(e5,4) = e4$; $next_2(e5, 4) = f4$.



a(f6)***a***(e5)***A***(e4, h1, 3) $\overset{21}{\longrightarrow}$ ...

a(f6)***a***(e5)***a***(e4)***a***(f3)***a***(g2)***a***(h1).



Tables 15×15

The game of chess can be interpreted as an 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 different for 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).

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

Fig. 1. 15x15 table for a King.

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	

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

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 were 7 in number, of size 15×15 each. The entries were 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. A number of 15×15 tables are shown in Fig. 1, 2, 3.

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 $x = (x_1, x_2)$, $y = (y_1, y_2)$, $v_1 = 8 - x_1 + y_1$, $v_2 = 8 - x_2 + y_2$, $f = f(p)$ is the type of the piece p (King, Rook, etc.). The seven 15×15 tables specify on X *seven different metrics*.

In order to explain (1) we can 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 (Fig. 2). Next, assume that the 8×8 table is 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 in Fig. 2.

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

Fig. 3. 15x15 table for a Bishop.