

Project 1 – Distances in ABG

Sean G Meyer

Reachability Rules

I created my program so that every reachability rule is input as a simple function, with parameters of x-distance and y-distance between the first and second positions, along with the the actual x/y values for 1st and 2nd position (only used if necessary, in this case only for the pawn). Below are the reachability rules I created for each piece (different ones can easily be input manually when running program).

```
(defn king
  [x-dist y-dist & xy]
  (and
    (<= x-dist 1)
    (<= y-dist 1)))

(defn knight
  [x-dist y-dist & xy]
  (or
    (and
      (= x-dist 2)
      (= y-dist 1))
    (and
      (= x-dist 1)
      (= y-dist 2))))

(defn queen
  [x-dist y-dist & xy]
  (or
    (= x-dist 0)
    (= y-dist 0)
    (= x-dist y-dist)))

(defn rook
  [x-dist y-dist & xy]
  (or
    (= x-dist 0)
    (= y-dist 0)))

(defn bishop
  [x-dist y-dist & xy]
  (= x-dist y-dist))
```

```
(defn pawn
  [x-dist y-dist x1 y1 x2 y2]
  (and
    (= x-dist 0)
    (= (- y1 y2) 1)))
```

Sample Tables

Below are tables, one per chess piece, with sample locations computed. Each chess piece was placed on a different board of varying size with varying obstacles.

Pawn on an 8x8 board with no obstacles, starting at position c2

	a	b	c	d	e	f	g	h
8	x	x	6	x	x	x	x	x
7	x	x	5	x	x	x	x	x
6	x	x	4	x	x	x	x	x
5	x	x	3	x	x	x	x	x
4	x	x	2	x	x	x	x	x
3	x	x	1	x	x	x	x	x
2	x	x	0	x	x	x	x	x
1	x	x	x	x	x	x	x	x

King on an 8x8 board with obstacles at positions marked []

	a	b	c	d	e	f	g	h
8	1	1	1	2	3	4	5	6
7	1	0	1	2	3	4	5	6
6	1	1	1	2	[]	[]	5	6
5	2	2	2	2	[]	[]	6	6
4	3	3	3	3	[]	[]	6	7
3	4	4	4	4	4	5	6	7
2	5	5	5	5	5	5	6	7
1	6	6	6	6	6	6	6	7

Knight on a 10x10 board with obstacles

	a	b	c	d	e	f	g	h	i	j
15	4	3	4	3	4	3	4	3	4	3
14	3	4	3	4	3	2	3	2	3	4
13	4	3	2	3	□	□	2	3	2	3
12	3	2	3	4	□	□	1	4	3	2
11	4	3	2	1	□	□	4	1	2	3
10	3	2	3	2	3	0	3	x	3	2
9	4	3	2	1	2	3	2	x	2	3
8	3	2	x	4	1	2	1	4	3	4
7	4	3	x	3	2	3	2	3	2	3
6	3	4	3	2	3	2	3	2	3	4

Queen on a 10x10 board with no obstacles

	a	b	c	d	e	f	g	h	i	j
10	1	2	2	2	2	1	2	2	2	2
9	2	1	2	2	2	1	2	2	2	1
8	2	2	1	2	2	1	2	2	1	2
7	2	2	2	1	2	1	2	1	2	2
6	2	2	2	2	1	1	1	2	2	2
5	1	1	1	1	1	0	1	1	1	1
4	2	2	2	2	1	1	1	2	2	2
3	2	2	2	1	2	1	2	1	2	2
2	2	2	1	2	2	1	2	2	1	2
1	2	1	2	2	2	1	2	2	2	1

Bishop on a 10x10 board with obstacles

	a	b	c	d	e	f	g	h	i	j
10	x	2	x	2	x	2	x	2	x	1
9	3	x	2	x	2	x	2	x	1	x
8	x	3	x	2	x	2	x	1	x	2
7	2	x	3	x	2	x	1	x	2	x
6	x	3	x	□	x	1	x	2	x	2
5	3	x	4	□	0	x	2	x	2	x
4	x	3	x	□	x	1	x	2	x	2
3	3	x	3	x	2	x	1	x	2	x
2	x	3	x	2	x	2	x	1	x	2
1	4	x	2	x	2	x	2	x	1	x

Rook on a 10x10 board with obstacles

	a	b	c	d	e	f	g	h	i	j
10	2	2	2	2	1	2	2	2	2	2
9	2	2	2	2	1	2	2	2	2	2
8	2	2	2	2	1	2	2	2	2	2
7	2	2	2	2	1	2	2	2	2	2
6	3	3	3	□	1	2	2	2	2	2
5	3	3	3	□	0	1	1	1	1	1
4	3	3	3	□	1	2	2	2	2	2
3	2	2	2	2	1	2	2	2	2	2
2	2	2	2	2	1	2	2	2	2	2
1	2	2	2	2	1	2	2	2	2	2

15x15 Tables

Below are the 15x15 tables my program generated for each piece. These are generated when the input ABG is 8x8 with no obstacles.

King 15x15

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
15	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
14	7	6	6	6	6	6	6	6	6	6	6	6	6	6	7
13	7	6	5	5	5	5	5	5	5	5	5	5	5	6	7
12	7	6	5	4	4	4	4	4	4	4	4	4	5	6	7
11	7	6	5	4	3	3	3	3	3	3	3	4	5	6	7
10	7	6	5	4	3	2	2	2	2	2	3	4	5	6	7
9	7	6	5	4	3	2	1	1	1	2	3	4	5	6	7
8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
7	7	6	5	4	3	2	1	1	1	2	3	4	5	6	7
6	7	6	5	4	3	2	2	2	2	2	3	4	5	6	7
5	7	6	5	4	3	3	3	3	3	3	3	4	5	6	7
4	7	6	5	4	4	4	4	4	4	4	4	4	5	6	7
3	7	6	5	5	5	5	5	5	5	5	5	5	5	6	7
2	7	6	6	6	6	6	6	6	6	6	6	6	6	6	7
1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

Knight 15x15

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
15	6	5	6	5	4	5	4	5	4	5	4	5	4	5	6
14	5	6	5	4	5	4	5	4	5	4	5	4	5	4	5
13	6	5	4	5	4	3	4	3	4	3	4	3	4	5	4
12	5	4	5	4	3	4	3	4	3	4	3	4	3	4	5
11	4	5	4	3	4	3	2	3	2	3	2	3	4	3	4
10	5	4	3	4	3	2	3	2	3	2	3	2	3	4	3
9	4	5	4	3	2	3	4	1	2	1	4	3	2	3	4
8	5	4	3	4	3	2	1	2	3	2	1	2	3	4	3
7	4	5	4	3	2	3	2	3	0	3	2	3	2	3	4
6	5	4	3	4	3	2	1	2	3	2	1	2	3	4	3
5	4	5	4	3	2	3	4	1	2	1	4	3	2	3	4
4	5	4	3	4	3	2	3	2	3	2	3	2	3	4	3
3	4	5	4	3	4	3	2	3	2	3	2	3	4	3	4
2	5	4	5	4	3	4	3	4	3	4	3	4	3	4	5
1	6	5	4	5	4	3	4	3	4	3	4	3	4	5	4

Queen 15x15

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
15	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
14	2	1	2	2	2	2	2	1	2	2	2	2	2	1	2
13	2	2	1	2	2	2	2	1	2	2	2	2	1	2	2
12	2	2	2	1	2	2	2	1	2	2	2	1	2	2	2
11	2	2	2	2	1	2	2	1	2	2	1	2	2	2	2
10	2	2	2	2	2	1	2	1	2	1	2	2	2	2	2
9	2	2	2	2	2	2	1	1	1	2	2	2	2	2	2
8	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
7	2	2	2	2	2	2	1	1	1	2	2	2	2	2	2
6	2	2	2	2	2	1	2	1	2	1	2	2	2	2	2
5	2	2	2	2	1	2	2	1	2	2	1	2	2	2	2
4	2	2	2	1	2	2	2	1	2	2	2	1	2	2	2
3	2	2	1	2	2	2	2	1	2	2	2	2	1	2	2
2	2	1	2	2	2	2	2	1	2	2	2	2	2	1	2
1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Rook 15x15

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
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	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2
8	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
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	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2

Bishop 15x15

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
15	1	x	2	x	2	x	2	x	2	x	2	x	2	x	1
14	x	1	x	2	x	2	x	2	x	2	x	2	x	1	x
13	2	x	1	x	2	x	2	x	2	x	2	x	1	x	2
12	x	2	x	1	x	2	x	2	x	2	x	1	x	2	x
11	2	x	2	x	1	x	2	x	2	x	1	x	2	x	2
10	x	2	x	2	x	1	x	2	x	1	x	2	x	2	x
9	2	x	2	x	2	x	1	x	1	x	2	x	2	x	2
8	x	2	x	2	x	2	x	0	x	2	x	2	x	2	x
7	2	x	2	x	2	x	1	x	1	x	2	x	2	x	2
6	x	2	x	2	x	1	x	2	x	1	x	2	x	2	x
5	2	x	2	x	1	x	2	x	2	x	1	x	2	x	2
4	x	2	x	1	x	2	x	2	x	2	x	1	x	2	x
3	2	x	1	x	2	x	2	x	2	x	2	x	1	x	2
2	x	1	x	2	x	2	x	2	x	2	x	2	x	1	x
1	1	x	2	x	2	x	2	x	2	x	2	x	2	x	1

Pawn 15x15

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
15	x	x	x	x	x	x	x	7	x	x	x	x	x	x	x
14	x	x	x	x	x	x	x	6	x	x	x	x	x	x	x
13	x	x	x	x	x	x	x	5	x	x	x	x	x	x	x
12	x	x	x	x	x	x	x	4	x	x	x	x	x	x	x
11	x	x	x	x	x	x	x	3	x	x	x	x	x	x	x
10	x	x	x	x	x	x	x	2	x	x	x	x	x	x	x
9	x	x	x	x	x	x	x	1	x	x	x	x	x	x	x
8	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
5	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Implementation

The code was implemented in the programming language Clojure. Source code is included in a file alongside this report. There is no runnable executable at this time, but one can be created on request if needed.