Upper Bounds for Pursuer Speed in Rectilinear Grids

Technical Report OSU-CISRC-1/04-TR01

Christopher A. Bohn Paolo A.G. Sivilotti
Department of Computer & Information Science
The Ohio State University
Columbus, Ohio 43210–1277
Email: {bohn,paolo}@cis.ohio-state.edu

Abstract

We define a pursuit-evasion game played on a finite grid and establish the speeds sufficient for a pursuer to detect all evaders.

I. Introduction

In recent years, the US Department of Defense has taken an ever-growing interest in the use of unmanned aerial vehicles (UAVs) to perform potentially dangerous and potentially dull tasks such as reconnaissance; the current generation of tactical UAVs, however, are hampered by a field of view that has been compared to a soda-straw [1]. We therefore define a game of pursuers and evaders in which the pursuers cannot ascertain any evader's location except when the evaders fall within a narrow field of view. In this paper we establish the sufficient speeds for a single pursuer to detect all evaders in a simple form of the game. Establishing insufficiency is outside the scope of this paper.

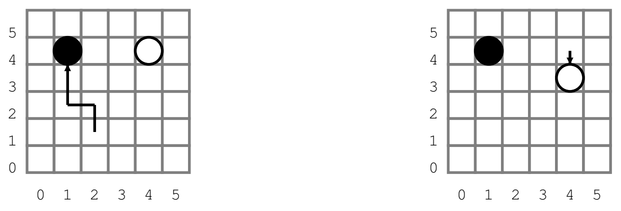
In Section II we formally define the game. In Section III we formally define a toy programming language which we shall use to express algorithms describing the pursuer's movements. In Section IV, we algorithms to provide proofs of existence of pursuer-winning search patterns at particular speeds. Finally, in Section V we express our conclusions.

II. Game Description

The game is played on a board covered by a rectilinear grid of m columns and n rows. The pursuer's objective is to occupy the same grid cell as the evader eventually, whereas the object for the evader is to prevent colocation with the pursuer indefinitely. The columns are numbered $0 \,..\, m-1$ starting at the leftmost column, and the rows are numbered $0 \,..\, n-1$ starting at the bottom row. The pursuer has the advantage that it can move s spaces per turn (where $s > 1$), but the evader can move only one space per turn (Figure 1). On the other hand, the evader has the advantage that it always knows the pursuer's location, whereas the pursuer is unable to determine the evader's location unless it occupies the evader's cell. They take turns moving: in each turn, the pursuer moves up to s spaces, then the evader moves one space. Movements are from the current cell to an adjacent cell. The four basic variations revolve around the definition of "adjacent": in all four variants, movement in the four cardinal directions are legal, and the variations are the cross-product of whether the pursuer can move diagonally and whether the evader can move diagonally.

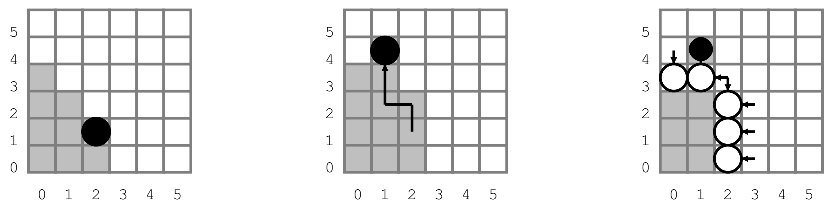
The restriction on the pursuer's knowledge suggests that if we wish to prove that the evader can be caught (and how it can be caught!), then we need to consider something other than the locations of the players. In this paper we shall consider the set of locations the evader cannot occupy and how that set changes. We call this set $Clear$, and its complement is the set of possible locations the evader may occupy. Consider Figure 2. Suppose it is known in Figure 2(a) that the shaded region cannot be occupied by the evader; *i.e.*, the evader must be in some cell in the unshaded region. In Figure 2(b), the pursuer makes the same move as in Figure 1(a). Observe that the pursuer, having passed through some cells, either encountered the evader in one of those

The views expressed in this article are those of the authors and do not necessarily reflect the official policy of the Air Force, the Department of Defense or the U.S. Government.



(a) Pursuer's turn (b) Evader's turn

Fig. 1. Examples of movements by the pursuer and the evader. Solid circle is the pursuer; hollow circle is the evader.



(a) Before pursuer moves (b) Pursuer's turn (c) Evader's turn

Fig. 2. Examples of changes in the possible locations for the evader. Evader is known to be in unshaded region.

cells (hence, won the game) or established that the evader cannot occupy those cells — the cells the pursuer visited are added to the set $Clear$.

Now it is the evader's turn to move. Since the pursuer does not know where the evader is located (other than that the evader must be located in the unshaded region), the set $Clear$ must be updated by all possible moves. If the evader is located in a cell that is not adjacent to a shaded cell, then it cannot move into the shaded region; such possible moves will not affect the set. If a shaded cell is adjacent only to other shaded cells, then it is not possible for the evader to move into that cell, and that cell remains in $Clear$. But, if a shaded cell and an unshaded cell are adjacent, then the possibility exists that the evader is in the unshaded cell, and the possibility also exists that the pursuer moves from the unshaded cell into the shaded cell; thus, that shaded cell is removed from $Clear$. The one exception is that the evader cannot move into the cell occupied by the pursuer, or the pursuer would know the evader's location and would win the game. In this scenario, the aggregate of all the ways in which the evader might move increases the size of the region the evader may occupy (Figure 2(c)) and reduces the cardinality of $Clear$.

III. Language Definition

A. Pseuocode Language

The algorithms presented in this paper use a statically-scoped, pass-by-value, Logo-like language in which translations have a duration. Movement of the pursuer is specified by the following commands:

move $dir\ spaces$	Move the pursuer a distance of $spaces$ in the specified direction $dir \in \{\mathrm{N}, \mathrm{NE}, \mathrm{E}, \mathrm{SE}, \mathrm{S}, \mathrm{SW}, \mathrm{W}, \mathrm{NW}\}$
wait $duration$	Do not move the pursuer for a duration equal to that of **move** $d\ duration$
evader-move	Do not move the pursuer for the remainder of its turn (equivalent to **wait** $duration$, where $duration$ is the number of moves left in the pursuer's turn), and then do nothing for a duration equal to that of **wait** 1 while the evader makes its move

m	Instantiation variable: the number of columns on the game board. $m \geq 2$
n	Instantiation variable: the number of rows on the game board. $n \geq 2$
speed	Instantiation variable: the maximum number of spaces the pursuer may move between moves by the evader. $\mathit{speed} \geq 1$
row	The row on the game board occupied by the pursuer. With col, uniquely identifies the pursuer's location. $0 \leq \mathit{row} < n, \mathit{row}_{init} = 0$.
col	The column on the game board occupied by the pursuer. With row, uniquely identifies the pursuer's location. $0 \leq \mathit{col} < m, \mathit{col}_{init} = 0$
time	The number of discrete units of time elapsed since the start of the game. $\mathit{time} \geq 0, \mathit{time}_{init} = 0$.
Ψ	The set of all cells on the game board. $\lvert\Psi\rvert = m \times n$
Clear	The set of cleared cells, as defined in Section III-B. $\mathit{Clear} \subseteq \Psi,\ \mathit{Clear}_{init} = \{(0,0)\}$
cycle	Constant value: the number of discrete clock ticks between the start of two consecutive rounds of movent. $\mathit{cycle} \triangleq \mathit{speed} + 1$

Fig. 3. Symbols used in Sections III and IV

More formally, using integer (non-modulus) arithmetic,

$$\left\{\begin{array}{lcll} \mathit{dir} \in \{\mathrm{SW}, \mathrm{S}, \mathrm{SE}\} & \Rightarrow & \mathit{row} = r + \mathit{spaces} & \wedge \\ \mathit{dir} \in \{\mathrm{W}, \mathrm{E}\} & \Rightarrow & \mathit{row} = r & \wedge \\ \mathit{dir} \in \{\mathrm{NW}, \mathrm{N}, \mathrm{NE}\} & \Rightarrow & \mathit{row} = r - \mathit{spaces} & \wedge \\ \mathit{dir} \in \{\mathrm{NW}, \mathrm{W}, \mathrm{SW}\} & \Rightarrow & \mathit{col} = c + \mathit{spaces} & \wedge \\ \mathit{dir} \in \{\mathrm{N}, \mathrm{S}\} & \Rightarrow & \mathit{col} = c & \wedge \\ \mathit{dir} \in \{\mathrm{NE}, \mathrm{E}, \mathrm{SE}\} & \Rightarrow & \mathit{col} = c - \mathit{spaces} & \wedge \\ \mathit{time} = t - \mathit{spaces} & & & \wedge \\ t \operatorname{div} \mathit{cycle} = \mathit{time} \operatorname{div} \mathit{cycle} & & & \end{array}\right\} \textbf{move}\ \mathit{dir}\ \mathit{spaces} \left\{\begin{array}{ll} \mathit{row} = r & \wedge \\ \mathit{col} = c & \wedge \\ \mathit{time} = t & \end{array}\right\} \tag{1}$$

$$\left\{\begin{array}{ll} t \operatorname{div} \mathit{cycle} = (\mathit{time} \operatorname{div} \mathit{cycle}) + 1 & \wedge \\ t \bmod \mathit{cycle} = 0 & \end{array}\right\} \textbf{evader-move}\, \{\mathit{time} = t\} \tag{2}$$

$$\left\{\begin{array}{ll} \mathit{time} = t - \mathit{duration} & \wedge \\ t \operatorname{div} \mathit{cycle} = \mathit{time} \operatorname{div} \mathit{cycle} & \end{array}\right\} \textbf{wait}\ \mathit{duration}\, \{\mathit{time} = t\} \tag{3}$$

B. Definitions

We now present some terms that we will use later in the paper when discussing the properties of the game.

cell	An ordered pair (r, c) is a unique location on the playing board located in the r^{th} row and the c^{th} column. A cell may be unoccupied, occupied by the pursuer, occupied by the evader, or occupied by both (if a cell is occupied by both, then the pursuer has won the game). For example, in Figure 1(b), the pursuer occupies cell $(4, 1)$.
move	A player's legal transition from one cell to another or the same cell.
turn	A sequence of moves; the length of the sequence is determined by the pursuer's speed advantage over the evader. In each turn, the pursuer has *speed* moves, and the evader has 1 move, for a total of $\mathit{cycle} \triangleq \mathit{speed} + 1$ moves.
$\mathrm{cleared}(r, c)$	is TRUE if and only if no undetected evader can occupy cell (r, c). *(pick one)*
$\mathrm{cleared}(r, c, t)$	is TRUE if and only if any evaders occupying cell (r, c) at time t must have been detected at time $\tau \leq t$. *(pick one)*
$\mathrm{e\text{-}adjacent}(C_1, C_2)$	Cell C_1 is e-adjacent to cell C_2 if and only if there is a legal move for the evader to move from C_1 to C_2 in a single move. The evader need not be in C_1 for C_1 to be $\mathrm{e\text{-}adjacent}$ to C_2, but it must be possible for the evader to occupy C_1. For example, in Figure 2, $(4, 0)$ is $\mathrm{e\text{-}adjanc ent}$ to $(3, 0)$, but there are no cells $\mathrm{e\text{-}adjacent}$ to $(0, 0)$.

```
CLEAR-BOARD
    preconditions
1   $col = 0$
2   $row = 0$
3   $time = 0$
    postconditions
1   $\forall \varrho < n, \kappa < m : (\varrho, \kappa) \in Clear$
```

Fig. 4. Specification for a winning pursuer algorithm.

p-adjacent(C_1, C_2) Cell C_1 is p-adjacent to cell C_2 if and only if there is a legal direction dir such that the pursuer can move from C_1 to C_2 by invoking **move** dir 1. The pursuer need not be in C_1 for C_1 to be p-adjacent to C_2.

These definitions permit us to enrich the axiomatic semantics of our language, to describe the effects of the pursuer's and evader's movements on the set of cleared cells. Axiom (1) becomes:

$$\left\{\begin{array}{lll l} dir \in \{\mathrm{SW}, \mathrm{S}, \mathrm{SE}\} & \Rightarrow & row = r + spaces & \wedge \\ dir \in \{\mathrm{W}, \mathrm{E}\} & \Rightarrow & row = r & \wedge \\ dir \in \{\mathrm{NW}, \mathrm{N}, \mathrm{NE}\} & \Rightarrow & row = r - spaces & \wedge \\ dir \in \{\mathrm{NW}, \mathrm{W}, \mathrm{SW}\} & \Rightarrow & col = c + spaces & \wedge \\ dir \in \{\mathrm{N}, \mathrm{S}\} & \Rightarrow & col = c & \wedge \\ dir \in \{\mathrm{NE}, \mathrm{E}, \mathrm{SE}\} & \Rightarrow & col = c - spaces & \wedge \\ time = t - spaces & & & \wedge \\ t \operatorname{div} cycle = time \operatorname{div} cycle & & & \wedge \\ dir = \mathrm{N} & \Rightarrow & \mathcal{F} \triangleq \{(r-s, c) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{NE} & \Rightarrow & \mathcal{F} \triangleq \{(r-s, c-s) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{E} & \Rightarrow & \mathcal{F} \triangleq \{(r, c-s) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{SE} & \Rightarrow & \mathcal{F} \triangleq \{(r+s, c-s) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{S} & \Rightarrow & \mathcal{F} \triangleq \{(r+s, c) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{SW} & \Rightarrow & \mathcal{F} \triangleq \{(r+s, c+s) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{W} & \Rightarrow & \mathcal{F} \triangleq \{(r, c+s) | s \in [0\,..\,spaces]\} & \wedge \\ dir = \mathrm{NW} & \Rightarrow & \mathcal{F} \triangleq \{(r-s, c+s) | s \in [0\,..\,spaces]\} & \wedge \\ Clear \cup \mathcal{F} = \mathcal{C} & & & \end{array}\right\} \textbf{move}\ dir\ spaces \left\{\begin{array}{ll} row = r & \wedge \\ col = c & \wedge \\ time = t & \wedge \\ Clear = \mathcal{C} & \end{array}\right\} \tag{4}$$

And Axiom (2) becomes:

$$\left\{\begin{array}{ll} t \operatorname{div} cycle = (time \operatorname{div} cycle) + 1 & \wedge \\ t \operatorname{mod} cycle = 0 & \wedge \\ \mathcal{F} \triangleq \{(r, c) | \exists(\rho, \kappa) \in \overline{Clear} : \text{e-adjacent}\,((\rho, \kappa), (r, c))\} & \wedge \\ (Clear \setminus \mathcal{F}) \cup \{(row, col)\} = \mathcal{C} & \end{array}\right\} \textbf{evader-move} \left\{\begin{array}{ll} time = t & \wedge \\ Clear = \mathcal{C} & \end{array}\right\} \tag{5}$$

IV. Properties of the Game

We begin by considering the specification of a winning algorithm for the pursuer, which is independent of the variation of the game. All variables in this section may be assumed to be natural numbers.

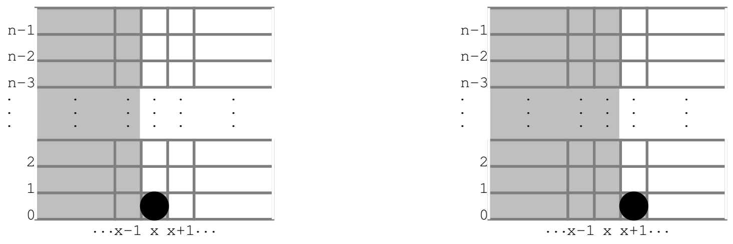
Theorem 1: If the pursuer follows an algorithm satisfying the specification in Figure 4, then the pursuer and evader eventually will be colocated.

Proof: Since an evader must occupy some cell, then by the definition of cleared, the postcondition of CLEAR-BOARD's specification can be true only if the pursuer was collocated with each evader at least once before the algorithm terminated. ■

Now consider the implementation of CLEAR-BOARD in Figure 5. In each iteration of the **while** loop, the algorithm clears all cells in column x. The effect of the call to CLEAR-COLUMN in

```
CLEAR-BOARD
1  $x \leftarrow 0$
2  while $x < m - 1$
3      do
4          CLEAR-COLUMN
5          $x \leftarrow x + 1$
6      end do
7  CLEAR-LAST-COLUMN
```

Fig. 5. Winning pursuer algorithm.



(a) Before CLEAR-COLUMN (b) After CLEAR-COLUMN

Fig. 6. Effect of calling CLEAR-COLUMN during the x^{th} iteration of the loop in Figure 5.

line 4 is depicted in Figure 6. After columns $0 \mathinner{..} m-2$ are cleared in the loop, a call to CLEAR-LAST-COLUMN clears the cells in column $m - 1$.

Lemma 2: The loop in Figure 5 terminates.

Proof: Consider the variant function $\mathfrak{f}(x) = m - x$. CLEAR-COLUMN does not alter x, and line 5 increments x; therefore, $\mathfrak{f}$ is strictly monotonically decreasing. When the loop test fails, $x \geq m - 1$, and $\mathfrak{f}(x) \leq 1$. Since $\mathfrak{f}(0) > 1$ and $\mathfrak{f}$ decreases strictly monotonically, eventually x must assume some value such that $\mathfrak{f}(x) \leq 1$, and the loop test fails. ■

Theorem 3: The algorithm in Figure 5 satisfies the specification of CLEAR-BOARD in Figure 4.

Proof: The derivation in Appendix B shows the partial correctness of the algorithm in Figure 5. By Lemma 2, the algorithm must terminate. Therefore we have total correctness. ■

Now consider the implementation of CLEAR-COLUMN in Figure 9. In each iteration of the **while** loop, the algorithm clears the cell in row y in the current column. The effect of the call to

```
CLEAR-COLUMN
   preconditions
1  $0 < c < m$
2  $time \bmod cycle = 0$
3  $col = c - 1$
4  $row = 0$
5  $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
   postconditions
1  $time \bmod cycle = 0$
2  $col = c$
3  $row = 0$
4  $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
```

Fig. 7. Specification for an algorithm that guarantees the evader is not in a column $col < m - 1$.

```
CLEAR-LAST-COLUMN
    preconditions
1   $\mathit{time} \bmod \mathit{cycle} = 0$
2   $\mathit{col} = m - 1$
3   $\mathit{row} = 0$
4   $\forall \varrho < n, \kappa < m-1 : (\varrho, \kappa) \in \mathit{Clear}$
    postconditions
1   $\forall \varrho < n, \kappa < m : (\varrho, \kappa) \in \mathit{Clear}$
```

Fig. 8. Specification for an algorithm that guarantees the evader is not in the rightmost column.

```
CLEAR-COLUMN
1   $y \leftarrow n - 1$
2   while $y > 0$
3       do
4           CLEAR-CELL($y$)
5           $y \leftarrow y - 1$
6       end do
7   CLEAR-LAST-CELL
```

Fig. 9. Algorithm that guarantees the evader is not in a column $col < m - 1$.

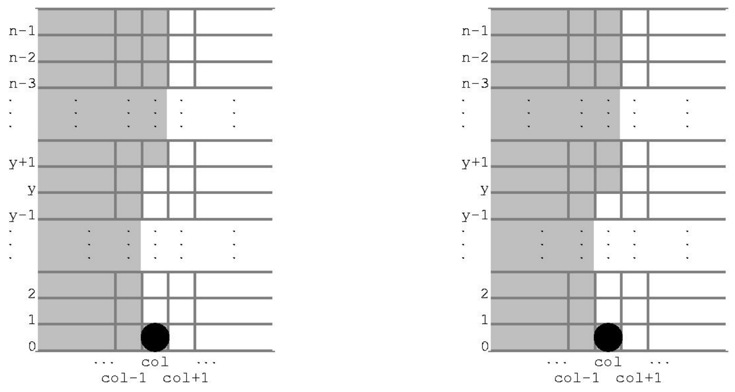
CLEAR-CELL in line 4 is depicted in Figure 10. After cells $(1, col) \mathinner{..} (n-1, col)$ are cleared in the loop, a call to CLEAR-LAST-CELL clears cell $(0, col)$.

Lemma 4: The loop in Figure 9 terminates.

Proof: Consider the variant function $\mathrm{f}(y) = y$. CLEAR-CELL does not alter y, and line 5 decrements y; therefore, f is strictly monotonically decreasing. When the loop test fails, $y \leq 0$, and $\mathrm{f}(y) \leq 0$. Since $\mathrm{f}(n-1) > 0$ and f decreases strictly monotonically, eventually y must assume some value such that $\mathrm{f}(y) \leq 0$, and the loop test fails. ■

Theorem 5: The algorithm in Figure 9 satisfies the specification of CLEAR-COLUMN in Figure 7.

Proof: The derivation in Appendix C shows the partial correctness of the CLEAR-COLUMN algorithm. By Lemma 4, the algorithm must terminate. Therefore we have total correctness. ■



(a) Before CLEAR-CELL (b) After CLEAR-CELL

Fig. 10. Effect of calling CLEAR-CELL during the y^{th} iteration of the loop in Figure 9.

```
CLEAR-CELL($y$)
    preconditions
1   $c < m - 1$
2   $0 < y < n$
3   $time \bmod cycle = 0$
4   $col = c$
5   $row = 0$
6   $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
7   $\forall \varrho : y + 1 \leq \varrho < n : (\varrho, col) \in Clear$
    postconditions
1   $time \bmod cycle = 0$
2   $col = c$
3   $row = 0$
4   $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
5   $\forall \varrho : y \leq \varrho < n : (\varrho, col) \in Clear$
```

Fig. 11. Specification for an algorithm that guarantees the evader is not in a cell.

```
CLEAR-LAST-CELL
    preconditions
1   $0 < c < m$
2   $time \bmod cycle = 0$
3   $col = c - 1$
4   $row = 0$
5   $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
6   $\forall \varrho : 1 \leq \varrho < n : (\varrho, col) \in Clear$
    postconditions
1   $time \bmod cycle = 0$
2   $col = c$
3   $row = 0$
4   $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
```

Fig. 12. Specification for an algorithm that guarantees the evader is not in the bottommost cell of a column.

So far, we have placed no restrictions on the pursuer's speed. We now present two algorithms which do require specific lower bounds on the value of $speed$. The first is an algorithm for CLEAR-LAST-COLUMN in Figure 13, which requires $speed \geq n - 1$. The second is an algorithm for CLEAR-LAST-CELL in Figure 14, which requires $speed \geq n$.

Theorem 6: If $speed \geq n - 1$, then the algorithm in Figure 13 satisfies the specification of CLEAR-LAST-COLUMN in Figure 8.

Proof: The proof follows directly from the semantics of **move**. The derivation in Appendix D shows that if the preconditions are met, then after CLEAR-LAST-COLUNM has completed, the postcondition will be satisfied. ■

Theorem 7: If $speed \geq n$, then the algorithm in Figure 14 satisfies the specification of CLEAR-LAST-CELL in Figure 12.

```
CLEAR-LAST-COLUMN
1   move N $n - 1$
```

Fig. 13. Algorithm that guarantees the evader is not in the rightmost column when $speed \geq n - 1$.

```
CLEAR-LAST-CELL
1  move E 1
2  move N $n - 1$
3  evader-move
4  move S $n - 1$
5  evader-move
```

Fig. 14. Algorithm that guarantees the evader is not in the bottommost cell of a column when $speed \geq n$.

```
CLEAR-CELL
1  move N $y$
2  move E 1
3  move N $n - y - 1$
4  evader-move
5  move S $n - y - 1$
6  move W 1
7  move S $y$
8  evader-move
```

Fig. 15. Algorithm that guarantees the evader is not in a cell when the evader cannot move diagonally and $speed \geq n$.

Proof: The derivation in Appendix E shows that if the preconditions are met, then after CLEAR-LAST-CELL has completed, the postconditions will be satisfied. ■

We now consider the specific variants of the game.

A. Evader and pursuer cannot move diagonally

Lemma 8: If $speed \geq n$, then when the evader cannot move diagonally (it can only use headings $\in \{\text{N}, \text{E}, \text{S}, \text{W}\}$), the algorithm in Figure 15 satisfies the specification of CLEAR-CELL in Figure 11.

Proof: The derivation in Appendix F shows that if the preconditions are met, then after CLEAR-CELL has completed, the postconditions will be satisfied. ■

Theorem 9: If neither the pursuer nor the evader can move diagonally, then to catch the evader the pursuer's minimum speed is at most $\min(m, n)$ spaces/turn.

Proof: Assume without loss of generality that $\min(m, n) = n$. By Theorem 7 and Lemma 8, we have correct implementations of CLEAR-LAST-CELL and CLEAR-CELL that can be used by the implementation of CLEAR-COLUMN when $speed \geq n$. By Theorems 5 and 6, we have correct implementations of CLEAR-COLUMN and CLEAR-LAST-COLUMN that can be used by the implementation of CLEAR-BOARD. By Theorem 3, we have a correct implementation of CLEAR-BOARD. Finally, by Theorem 1, that algorithm will assure that the pursuer will be collocated with each evader eventually. ■

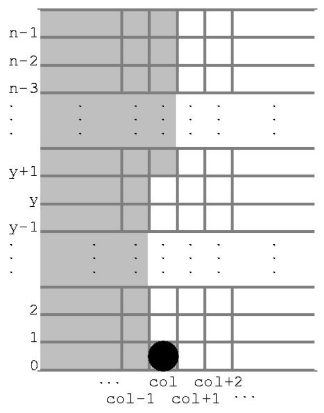
Conjecture 10: If neither the pursuer nor the evader can move diagonally, then to catch the evader the pursuer's speed must be at least $\min(m, n)$ spaces/turn.

Corollary 11: Assume neither the pursuer nor the evader can move diagonally. By Theorem 9, and if Conjecture 10 holds, the pursuer can catch the evader if and only if the pursuer can catch the evader when moving $\min(m, n)$ spaces/turn.

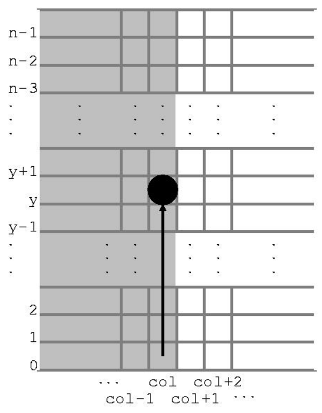
B. Evader and pursuer can move diagonally

Lemma 12: If $speed \geq n + 1$, then when both the pursuer and the evader can move diagonally (they can use all headings $\in \{\text{N}, \text{NE}, \text{E}, \text{SE}, \text{S}, \text{SW}, \text{W}, \text{NW}\}$), the algorithm in Figure 17 satisfies the specification of CLEAR-CELL in Figure 11.

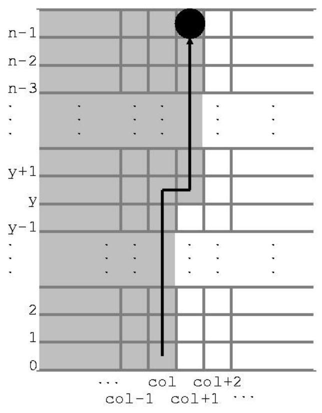
Proof: The derivation in Appendix G shows that if the preconditions are met, then after CLEAR-CELL has completed, the postconditions will be satisfied. ■



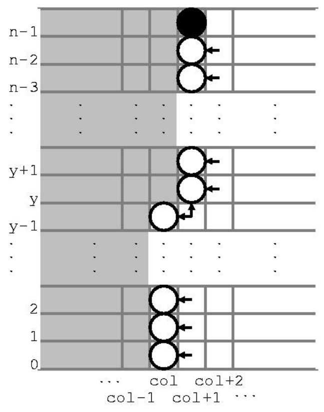
(a) Initial conditions



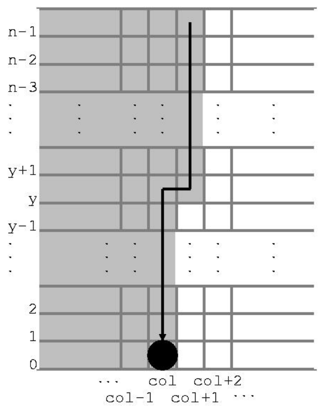
(b) After pursuer moves in line 1, cell (y, col) has been cleared



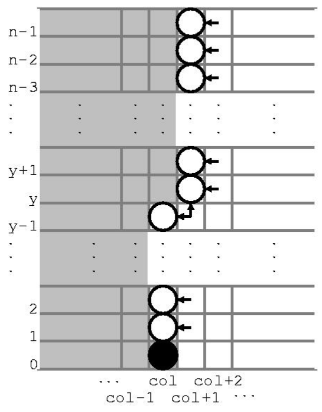
(c) Pursuer has moved n spaces, after line 3



(d) Evader moves in line 4, causing cells on the frontier to be uncleared *(better phrasing?) (clarify "frontier")*



(e) Pursuer moves next n spaces, in lines 5–7, returning to cell $(0, col)$

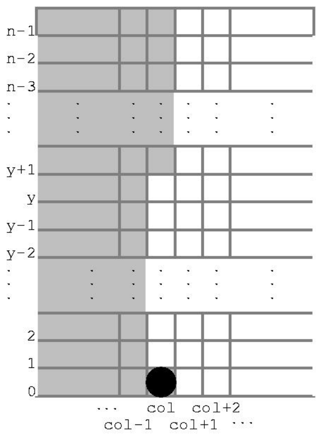


(f) Evader moves in line 8, causing cells on the frontier to be uncleared; the postconditions are now satisfied

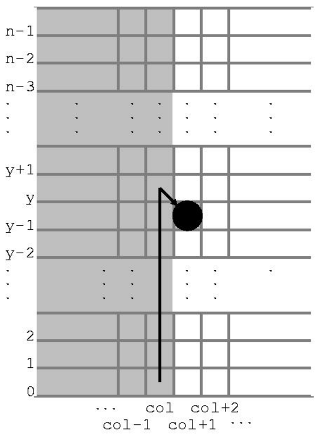
Fig. 16. Execution of the CLEAR-CELL algorithm in Figure 15.

```
CLEAR-CELL
1  **move** N $y$
2  **move** SE $1$
3  **move** N $n - y$
4  **evader-move**
5  **move** S $n - y$
6  **move** W $1$
7  **move** S $y - 1$
8  **evader-move**
```

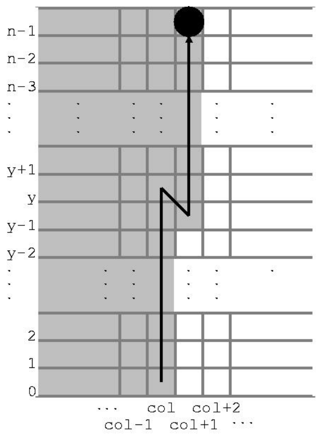
Fig. 17. Algorithm that guarantees the evader is not in a cell when both the pursuer and the evader can move diagonally and $speed \geq n + 1$.



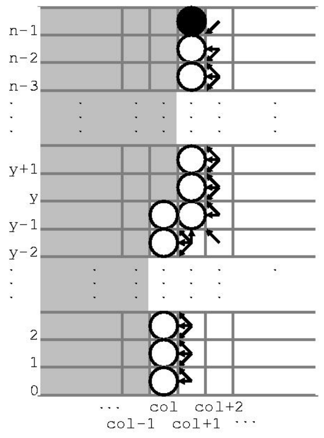
(a) Initial conditions



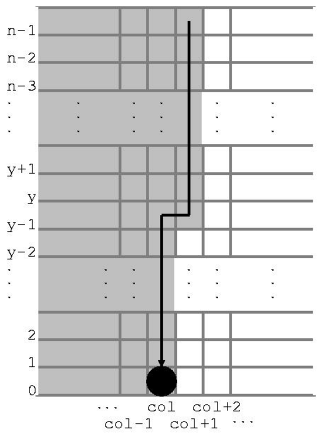
(b) After pursuer moves in lines 1, cell (y, col) has been cleared; the moves in line 2 will protect cell (y, col) from the evader's diagonal movement



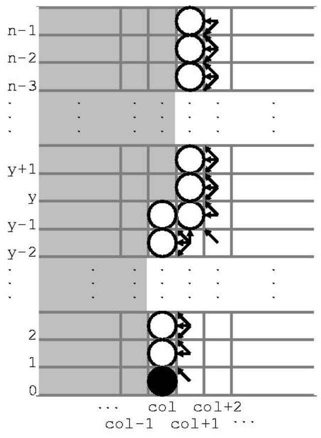
(c) Pursuer has moved $n + 1$ spaces, after line 3



(d) Evader moves in line 4, causing cells on the frontier to be uncleared *(better phrasing?) (clarify "frontier")*



(e) Pursuer moves next n spaces, in lines 5–7, returning to cell $(0, col)$



(f) Evader moves in line 8, causing cells on the frontier to be uncleared; the postconditions are now satisfied

Fig. 18. Execution of the CLEAR-CELL algorithm in Figure 17.

Theorem 13: If both the pursuer and the evader can move diagonally, then to catch the evader the pursuer's minimum speed is at most $\min(m, n) + 1$ spaces/turn.

Proof: Assume without loss of generality that $\min(m, n) = n$. By Theorem 7 and Lemma 12, we have correct implementations of CLEAR-LAST-CELL and CLEAR-CELL that can be used by the implementation of CLEAR-COLUMN when $speed \geq n + 1$. By Theorems 5 and 6, we have correct implementations of CLEAR-COLUMN and CLEAR-LAST-COLUMN that can be used by the implementation of CLEAR-BOARD. By Theorem 3, we have a correct implementation of CLEAR-BOARD. Finally, by Theorem 1, that algorithm will assure that the pursuer will be collocated with each evader eventually. ■

Conjecture 14: If neither the pursuer nor the evader can move diagonally, then to catch the evader the pursuer's speed must be at least $\min(m, n) + 1$ spaces/turn.

Corollary 15: Assume neither the pursuer nor the evader can move diagonally. By Theorem 13, and if Conjecture 14 holds, the pursuer can catch the evader if and only if the pursuer can catch

```
CLEAR-CELL
1  move N $y$
2  move E $1$
3  move S $1$
4  move N $n - y$
5  evader-move
6  move S $n - y$
7  move W $1$
8  move S $y - 1$
9  evader-move
```

Fig. 19. Algorithm that guarantees the evader is not in a cell when the evader can move diagonally but the pursuer cannot, and $\mathit{speed} \geq n + 2$.

the evader when moving $\min(m, n) + 1$ spaces/turn.

C. *Evader cannot move diagonally; pursuer can*

Theorem 16: If the pursuer can move diagonally but the evaders cannot, then to catch the evaders the pursuer's minimum speed is at most $\min(m, n)$ spaces/turn.

Proof: Assume without loss of generality that $\min(m, n) = n$. By Theorem 7 and Lemma 8, we have correct implementations of CLEAR-LAST-CELL and CLEAR-CELL that can be used by the implementation of CLEAR-COLUMN when $\mathit{speed} \geq n$. By Theorems 5 and 6, we have correct implementations of CLEAR-COLUMN and CLEAR-LAST-COLUMN that can be used by the implementation of CLEAR-BOARD. By Theorem 3, we have a correct implementation of CLEAR-BOARD. Finally, by Theorem 1, that algorithm will assure that the pursuer will be collocated with each evader eventually. ■

Conjecture 17: If neither the pursuer nor the evader can move diagonally, then to catch the evader the pursuer's speed must be at least $\min(m, n)$ spaces/turn.

Corollary 18: Assume neither the pursuer nor the evader can move diagonally. By Theorem 16, and if Conjecture 17 holds, the pursuer can catch the evader if and only if the pursuer can catch the evader when moving $\min(m, n)$ spaces/turn.

D. *Evader can move diagonally; pursuer cannot*

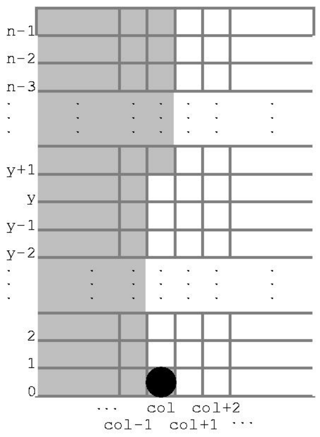
Lemma 19: If $\mathit{speed} \geq n + 2$, then when both the pursuer and the evader can move diagonally (they can use all headings $\in \{\text{N}, \text{NE}, \text{E}, \text{SE}, \text{S}, \text{SW}, \text{W}, \text{NW}\}$), the algorithm in Figure 19 satisfies the specification of CLEAR-CELL in Figure 11.

Proof: The derivation in Appendix H shows that if the preconditions are met, then after CLEAR-CELL has completed, the postconditions will be satisfied. ■

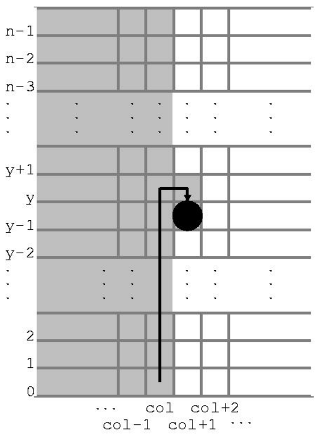
Theorem 20: If both the pursuer can move orthoganlly only and the evaders can move diagonally, then to catch the evaders the pursuer's minimum speed is at most $\min(m, n) + 2$ spaces/turn.

Proof: Assume without loss of generality that $\min(m, n) = n$. By Theorem 7 and Lemma 19, we have correct implementations of CLEAR-LAST-CELL and CLEAR-CELL that can be used by the implementation of CLEAR-COLUMN when $\mathit{speed} \geq n + 2$. By Theorems 5 and 6, we have correct implementations of CLEAR-COLUMN and CLEAR-LAST-COLUMN that can be used by the implementation of CLEAR-BOARD. By Theorem 3, we have a correct implementation of CLEAR-BOARD. Finally, by Theorem 1, that algorithm will assure that the pursuer will be collocated with each evader eventually. ■

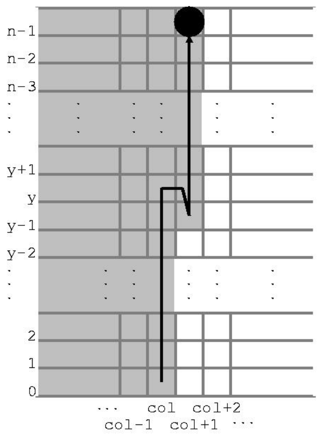
We have shown that when the evader can move diagonally but the pursuer can move only cardinally, a speed advantage of $n + 2$ is sufficient to assure the pursuer's victory, whereas $n + 1$ is sufficient when both the pursuer and evader can move diagonally. Contrast this with the two cases in which the evader cannot move diagonally – in those cases, the pursuer does not benefit from being able to move diagonally; in each case, a speed advantage of n is sufficient and, we



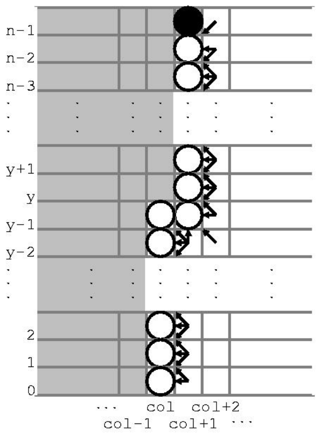
(a) Initial conditions



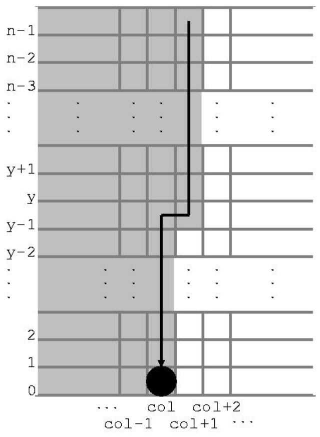
(b) After pursuer moves in lines 1, cell (y, col) has been cleared; the moves in lines 2–3 will protect cell (y, col) from the evader's diagonal movement



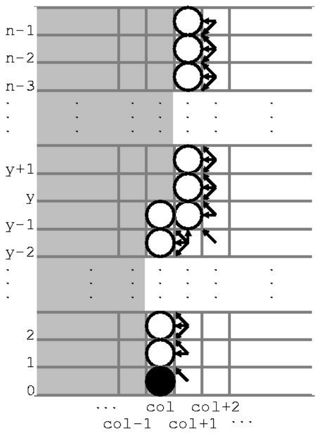
(c) Pursuer has moved $n + 2$ spaces, after line 4



(d) Evader moves in line 5, causing cells on the frontier to be uncleared *(better phrasing?) (clarify "frontier")*



(e) Pursuer moves next n spaces, in lines 6–8, returning to cell $(0, col)$



(f) Evader moves in line 9, causing cells on the frontier to be uncleared; the postconditions are now satisfied

Fig. 20. Execution of the CLEAR-CELL algorithm in Figure 19.

believe, necessary. The obvious question to ask at this point is whether the pursuer can win with a speed advantage of $n + 1$ in both cases in which the pursuer can move diagonally.

The answer is "yes", though the witness algorithm is considerably less straight-forward than those we have presented so far. First, we shall require some new subroutines, which we shall use to construct a new implementation of CLEAR-COLUMN, involving monotonically increasing the number of cleared cells in the column being cleared. Unlike the previous implementation, this new implementation alternates between clearing cells at the top and at the bottom of the column.

The next subroutine is TRANSITION, specified in Figure 23. After half of the column has been cleared, TRANSITION is used to rearrange the cleared cells to satisfy the conditions needed by GROW-TOP2 and GROW-BOTTOM2, which are specified in Figures 24 and 25, respectively. As with GROW-TOP1, GROW-TOP2 preserves the number of cleared cells by clearing cells at the top of the column, and then GROW-BOTTOM2 increases th enumber of cleared cells by clearing cells at the bottom of the column. With these subroutines specified, we can now implement the

```
GROW-BOTTOM1($y$)
    preconditions
1   $c < m - 1$
2   $y < \left\lfloor \frac{n}{2} \right\rfloor$
3   $\mathit{time} \bmod \mathit{cycle} = 0$
4   $\mathit{col} = c$
5   $\mathit{row} = y$
6   $\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}$
7   $\forall \varrho : n - y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}$
    postconditions
1   $\mathit{time} \bmod \mathit{cycle} = 0$
2   $\mathit{col} = c$
3   $\mathit{row} = n - y - 1$
4   $\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}$
5   $\forall \varrho \leq y : (\varrho, \mathit{col}) \in \mathit{Clear}$
```

Fig. 21. Specification for an algorithm that places cleared cells in the lower half of a column.

```
GROW-TOP1($y$)
    preconditions
1   $c < m - 1$
2   $y < \left\lfloor \frac{n}{2} \right\rfloor$
3   $\mathit{time} \bmod \mathit{cycle} = 0$
4   $\mathit{col} = c$
5   $\mathit{row} = n - y - 1$
6   $\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}$
7   $\forall \varrho : \varrho \leq y : (\varrho, \mathit{col}) \in \mathit{Clear}$
    postconditions
1   $\mathit{time} \bmod \mathit{cycle} = 0$
2   $\mathit{col} = c$
3   $y < \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow \mathit{row} = y + 1$
4   $y = \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow \mathit{row} = y$
5   $\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}$
6   $\forall \varrho : n - y - 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}$
```

Fig. 22. Specification for an algorithm that places cleared cells in the upper half of a column.

specification of CLEAR-COLUMN with CLEAR-COLUMN, shown in Figure code:clearColumnCD.

Theorem 21: The algorithm in Figure 26 terminates and satisfies the specification of CLEAR-COLUMN in Figure 7.

Proof: The derivation in Appendix I shows the total correctness of the CLEAR-COLUMN algorithm. ■

We now offer implementations of subroutines used by CLEAR-COLUMN. Figure 27 shows an implementation of GROW-BOTTOM1, and Figure 28 shows an implementation of GROW-TOP1; both require that $\mathit{speed} \geq n+1$. Figures 32 and 33 show representative uses of GROW-BOTTOM1 and GROW-TOP1 when $n = 8$ for the first and last iterations of the **while** loop in Lines 2–7 of Program 26.

Lemma 22: The algorithm in Figure 27 terminates and satisfies the specification of GROW-BOTTOM1 in Figure 21.

Proof: The derivation in Appendix J shows the total correctness of the GROW-BOTTOM1 algorithm. ■

Lemma 23: The algorithm in Figure 28 terminates and satisfies the specification of GROW-

```
TRANSITION
   preconditions
1  $0 < c < m$
2  $time \bmod cycle = 0$
3  $col = c - 1$
4  $row = \left\lceil \frac{n}{2} \right\rceil - 1$
5  $\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
6  $\forall \varrho : row < \varrho < n : (\varrho, col) \in Clear$
   postconditions
1  $time \bmod cycle = 0$
2  $col = c$
3  $row = \left\lfloor \frac{n}{2} \right\rfloor - 1$
4  $\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear$
5  $\forall \varrho < row : (\varrho, col - 1) \in Clear$
```

Fig. 23. Specification for an algorithm that repositions the pursuer from the postcondition of GROW-TOP1 to the precondition of GROW-TOP2.

```
GROW-TOP2($y$)
   preconditions
1  $0 < c < m$
2  $\left\lfloor \frac{n}{2} \right\rfloor \leq y < n - 1$
3  $time \bmod cycle = 0$
4  $col = c$
5  $row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1$
6  $\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear$
7  $\forall \varrho < y : (\varrho, col - 1) \in Clear$
   postconditions
1  $time \bmod cycle = 0$
2  $col = c$
3  $row = y + 1$
4  $\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear$
5  $\forall \varrho : 2\left\lfloor \frac{n}{2} \right\rfloor - y \leq \varrho < n : (\varrho, col - 1) \in Clear$
```

Fig. 24. Specification for an algorithm that places cleared cells in the upper half of a column.

TOP1 in Figure 22.

Proof: The derivation in Appendix K shows the total correctness of the GROW-TOP1 algorithm. ■

Figure 29 shows an implementation of TRANSITION when $speed \geq n+1$, and Figure 34 shows a representative use of TRANSITION when $n = 8$ in Line 8 of Program 26.

Lemma 24: The algorithm in Figure 29 terminates and satisfies the specification of TRANSITION in Figure 23.

Proof: The derivation in Appendix K shows the total correctness of the GROW-TOP1 algorithm. ■

Finally, Program 30 shows an implementation of GROW-TOP2, and Program 31 shows an implementation of GROW-BOTTOM2; both require that $speed \geq n + 1$. Figures 35 and 36 show representative uses of GROW-TOP2 and GROW-BOTTOM2 when $n = 8$ for the first and last iterations of the **while** loop in Lines 9–14 of Program 26.

Lemma 25: The algorithm in Figure 30 terminates and satisfies the specification of GROW-TOP2 in Figure 24.

Proof: The derivation in Appendix M shows the total correctness of the GROW-TOP2 algorithm. ■

```
GROW-BOTTOM2($y$)
    preconditions
1   $0 < c < m$
2   $\left\lfloor \frac{n}{2} \right\rfloor \leq y < n - 1$
3   $time \bmod cycle = 0$
4   $col = c$
5   $row = y + 1$
6   $\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear$
7   $\forall \varrho : 2 \left\lfloor \frac{n}{2} \right\rfloor - y \leq \varrho < n : (\varrho, col - 1) \in Clear$
    postconditions
1   $time \bmod cycle = 0$
2   $col = c$
3   $y < n - 2 \Rightarrow row = 2 \left\lfloor \frac{n}{2} \right\rfloor - y - 2$
4   $y = n - 2 \Rightarrow row = 0$
5   $\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear$
6   $\forall \varrho \leq y : (\varrho, col - 1) \in Clear$
7   $y = n - 2 \Rightarrow \forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear$
```

Fig. 25. Specification for an algorithm that places cleared cells in the lower half of a column.

```
CLEAR-COLUMN
 1  $y \leftarrow 0$
 2  while $y < \left\lfloor \frac{n}{2} \right\rfloor$
 3      do
 4          GROW-BOTTOM1($y$)
 5          GROW-TOP1($y$)
 6          $y \leftarrow y + 1$
 7      end do
 8  TRANSITION
 9  while $y < n - 1$
10      do
11          GROW-TOP2($y$)
12          GROW-BOTTOM2($y$)
13          $y \leftarrow y + 1$
14      end do
```

Fig. 26. Algorithm that guarantees the evader is not in a cell when the evader can move diagonally, implementing the specification of CLEAR-COLUMN in Figure 7.

```
GROW-BOTTOM1($y$)
1   move S $y$
2   move E $1$
3   move N $y + 1$
4   move W $1$
5   move N $n - 2y - 2$
6   evader-move
```

Fig. 27. Algorithm satisfying GROW-BOTTOM1 specification of Figure 21 when *speed* $\geq n + 1$.

```
Grow-Top1($y$)
move N $y$
move E $1$
move S $y+1$
move W $1$
if $n > 2y+3$
    then
        move S $n-2y-3$
    end if
evader-move
```

Fig. 28. Algorithm satisfying Grow-Top1 specification of Figure 22 when $speed \geq n+1$.

```
Transition
move S $\left\lceil \frac{n}{2} \right\rceil - 1$
move E $1$
move N $\left\lfloor \frac{n}{2} \right\rfloor$
move S $1$
evader-move
```

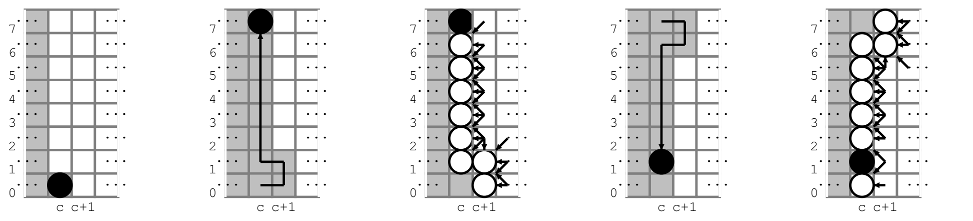
Fig. 29. Algorithm satisfying Transition specification of Figure 23 when $speed \geq n+1$.

```
Grow-Top2($y$)
move N $2\left(y - \left\lfloor \frac{n}{2} \right\rfloor\right) + 1$
move W $1$
move N $n-y-1$
move E $1$
move S $n-y-2$
evader-move
```

Fig. 30. Algorithm satisfying Grow-Top2 specification of Figure 24 when $speed \geq n+1$.

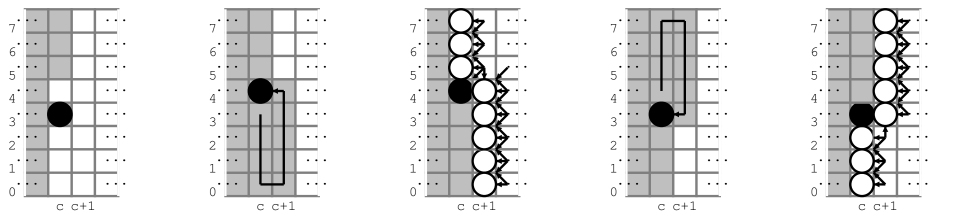
```
Grow-Bottom2($y$)
move S $2\left(y - \left\lfloor \frac{n}{2} \right\rfloor + 1\right)$
move W $1$
move S $2\left\lfloor \frac{n}{2} \right\rfloor - y - 1$
move E $1$
if $y < 2(\left\lfloor \frac{n}{2} \right\rfloor - 1)$
    then
        move N $2\left\lfloor \frac{n}{2} \right\rfloor - y - 2$
    end if
evader-move
```

Fig. 31. Algorithm satisfying Grow-Bottom2 specification of Figure 25 when $speed \geq n+1$.



(a) Before any movement (b) After pursuer's turn in Grow-Bottom1(0) (c) After evader's turn in Grow-Bottom1(0) (d) After pursuer's turn in Grow-Top1(0) (e) After evader's turn in Grow-Top1(0)

Fig. 32. Partial execution of Clear-Column algorithm in Figure 26 when $n = 8, y = 0$.



(a) Before any movement (b) After pursuer's turn in Grow-Bottom1(3) (c) After evader's turn in Grow-Bottom1(3) (d) After pursuer's turn in Grow-Top1(3) (e) After evader's turn in Grow-Top1(3)

Fig. 33. Partial execution of Clear-Column algorithm in Figure 26 when $n = 8, y = 3$.

Lemma 26: The algorithm in Figure 31 terminates and satisfies the specification of Grow-Bottom2 in Figure 25.

Proof: The derivation in Appendix N shows the total correctness of the Grow-Bottom2 algorithm. ■

We now can show that $speed = n + 1$ is a sufficient condition for the pursuer to detect all evaders when the evaders can move diagonally.

Theorem 27: If both the pursuer can move orthoganlly only and the evaders can move diagonally, then to catch the evaders the pursuer's minimum speed is at most $\min(m, n)+1$ spaces/turn.

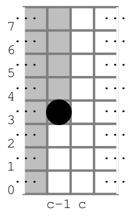
Proof: Assume without loss of generality that $\min(m, n) = n$. By Lemmas 22–26, we have correct implementations of Grow-Bottom1, Grow-Top1, Transition, Grow-Top2, and Grow-Bottom2 that can be used by the Clear-Column implementation of Figure 26 when $speed \geq n + 1$. By Theorems 21 and 6, we have correct implementations of Clear-Column and Clear-Last-Column that can be used by the implementation of Clear-Board. By Theorem 3, we have a correct implementation of Clear-Board. Finally, by Theorem 1, that algorithm will assure that the pursuer will be collocated with each evader eventually. ■

V. Conclusion

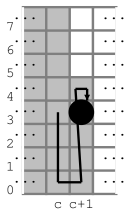
We have established that if the pursuer can move at speeds $s \geq n + 1$, where n is the shorter dimension of the grid, it has a search strategy that is guaranteed to locate the evader. Moreover, if the evader cannot move diagonally, then the pursuer also has a search strategy at speed $s = n$.

Acknowledgment

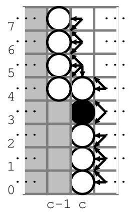
The authors gratefully thank the US Air Force and the Air Force Institute of Technology for their direct support of the primary author. This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-2-3154).



(a) Before any movement

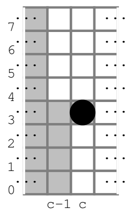


(b) After pursuer's turn in TRANSITION

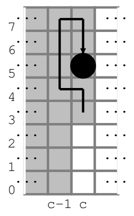


(c) After evader's turn in TRANSITION

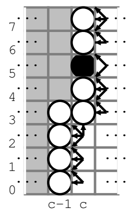
Fig. 34. Partial execution of CLEAR-COLUMN algorithm in Figure 26 at execution of TRANSITION.



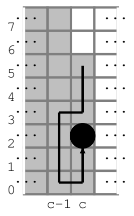
(a) Before any movement



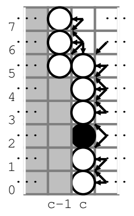
(b) After pursuer's turn in GROW-TOP2(4)



(c) After evader's turn in GROW-TOP2(4)



(d) After pursuer's turn in GROW-BOTTOM2(4)



(e) After evader's turn in GROW-BOTTOM2(4)

Fig. 35. Partial execution of CLEAR-COLUMN algorithm in Figure 26 when $n = 8, y = 4$.

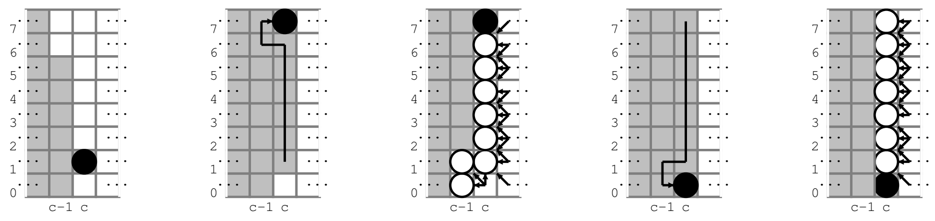
REFERENCES

[1] Office of the Secretary of Defense, "Unmanned aerial vehicles roadmap: 2002–2027," December 2002.

[2] C. Hoare, "An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, no. 10, pp. 576–583, October 1969.

[3] ——, "Procedures and parameters: An axiomatic approach," in *Lecture Notes in Mathematics 118*. Springer-Verlag, 1971, pp. 102–116.

[4] Z. Manna, *Mathematical Theory of Computation*. McGraw-Hill, 1974, ch. 3-3.



(a) Before any movement (b) After pursuer's turn in GROW-TOP2(6) (c) After evader's turn in GROW-TOP2(6) (d) After pursuer's turn in GROW-BOTTOM2(6) (e) After evader's turn in GROW-BOTTOM2(6)

Fig. 36. Partial execution of CLEAR-COLUMN algorithm in Figure 26 when $n = 8, y = 6$.

APPENDIX

A. Axiomatic Semantics

In this section we introduce the axioms and rules of inference used in the derivations in the other sections of the appendix. The notation used in this section is:

- $\mathcal{Q}$ is the set of program states
- $\sigma \in \mathcal{Q}$ is an arbitrary program state
- p, p', q, q', r are propositional formulae
- x is an arbitary variable name
- $expr$ is an arbitrary expression that is type-compatible with x
- b is a boolean expression
- $\mathrm{S}, \mathrm{S}_1, \mathrm{S}_2$ are program fragments: a program fragment is either an atomic command (*e.g.* an assignment or a **move** command) or a composition of program fragments
- PROC is an arbitrary procedure name
- *declare* PROC S indicates the declaration of PROC with produre body S
- *call* PROC indicates the use of PROC
- $\overrightarrow{params}$ is a list of non-local variables for PROC
- $\overrightarrow{args}$ is a list of expressions that are type-compatible with the variables in $\overrightarrow{params}$
- p^x_{expr} is the formula p with every occurance of x replaced by $expr$
- $\{p\}$ is the set of program states σ such that $\sigma \models p$
- $\{p\}\,\mathrm{S}\,\{q\}$ is a *Hoare triple* indicating that if the program state satisfies p before S executes, then q is satisfied after S executes; we assume that any state variables that are not part of the Hoare triple's specification are left unchanged by S
- $\{p\}\,\mathrm{S}\,\{q\} \downarrow$ is a Hoare triple that also indicates that S will terminate without generating a run-time error
- $\mathrm{f} : \mathcal{Q} \rightarrow \mathbb{N}$ is an arbitrary function mapping from program states to natural numbers

1) Axioms: We begin with five axioms, the Skip and Assignment Axioms and the three axioms unique to our language.

a) Skip Axiom: We don't explictly use the Skip Axiom in any of the algorithms, but we introduce it nonetheless to emphasize that a command that has no effect does not change the program state. Moreover, we shall use it to simply our derivations in later appendices.

$$\{p\}\,\textbf{no-op}\,\{p\} \tag{6}$$

b) Assignment Axiom: The Assignment Axiom is derived from that used by Hoare [2].

$$\left\{p^x_{expr}\right\} x \leftarrow expr\,\{p\} \tag{7}$$

Application of the Assignment Axiom is essentially the textual replacement of $expr$ for x in the precondition. For example: $\{2 < 3\}\, x \leftarrow 2\, \{x < 3\}$. Obviously, this axiom assumes that evaluation of $expr$ has no side-effects.

c) Move Lemma: The Move Lemma was introduced as (4) in the text. We consider it to be a lemma since it is Axiom (1) enhanced by definitions in Section III-B. The terms in the precondition and postcondition are described in Figure 3.

$$\left\{\begin{array}{lcll} \mathit{dir} \in \{\mathrm{SW}, \mathrm{S}, \mathrm{SE}\} & \Rightarrow & \mathit{row} = r + \mathit{spaces} & \wedge \\ \mathit{dir} \in \{\mathrm{W}, \mathrm{E}\} & \Rightarrow & \mathit{row} = r & \wedge \\ \mathit{dir} \in \{\mathrm{NW}, \mathrm{N}, \mathrm{NE}\} & \Rightarrow & \mathit{row} = r - \mathit{spaces} & \wedge \\ \mathit{dir} \in \{\mathrm{NW}, \mathrm{W}, \mathrm{SW}\} & \Rightarrow & \mathit{col} = c + \mathit{spaces} & \wedge \\ \mathit{dir} \in \{\mathrm{N}, \mathrm{S}\} & \Rightarrow & \mathit{col} = c & \wedge \\ \mathit{dir} \in \{\mathrm{NE}, \mathrm{E}, \mathrm{SE}\} & \Rightarrow & \mathit{col} = c - \mathit{spaces} & \wedge \\ \mathit{time} = t - \mathit{spaces} & & & \wedge \\ t \,\mathrm{div}\, \mathit{cycle} = \mathit{time} \,\mathrm{div}\, \mathit{cycle} & & & \wedge \\ \mathit{dir} = \mathrm{N} & \Rightarrow & \mathcal{F} \triangleq \{(r-s, c) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{NE} & \Rightarrow & \mathcal{F} \triangleq \{(r-s, c-s) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{E} & \Rightarrow & \mathcal{F} \triangleq \{(r, c-s) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{SE} & \Rightarrow & \mathcal{F} \triangleq \{(r+s, c-s) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{S} & \Rightarrow & \mathcal{F} \triangleq \{(r+s, c) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{SW} & \Rightarrow & \mathcal{F} \triangleq \{(r+s, c+s) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{W} & \Rightarrow & \mathcal{F} \triangleq \{(r, c+s) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{dir} = \mathrm{NW} & \Rightarrow & \mathcal{F} \triangleq \{(r-s, c+s) | s \in [0 \,..\, \mathit{spaces}]\} & \wedge \\ \mathit{Clear} \cup \mathcal{F} = \mathcal{C} & & & \end{array}\right\} \textbf{move}\ \mathit{dir}\ \mathit{spaces} \left\{\begin{array}{ll} \mathit{row} = r & \wedge \\ \mathit{col} = c & \wedge \\ \mathit{time} = t & \wedge \\ \mathit{Clear} = \mathcal{C} & \end{array}\right\} \tag{4}$$

The legal values for dir are $\{$NW,N,NE,W,E,SW,S,SE$\}$ if the pursuer can move diagonally, and $\{$N,W,E,S$\}$ if the pursuer can only move in the four cardinal directions. The pursuer moves a distance of $\mathit{spaces} \in \mathbb{N}$ in the specified direction, and so the pursuer's row and col values are changed by the movement. Each space moved increments time by one unit, but the pursuer is not permitted to move so far that the move cannot be completed before the evader's turn to move. Finally, because the pursuer clears each cell it occupies, the set of $\mathrm{cleared}$ cells changes.

d) Evader-Move Lemma: As with the Move Lemma, the Evader-Move Lemma was introduced as (5) in the text as the enhancement of Axiom (2) by definitions in Section III-B. The terms in the precondition and postcondition are described in Figure 3.

$$\left\{\begin{array}{ll} t \,\mathrm{div}\, \mathit{cycle} = (\mathit{time} \,\mathrm{div}\, \mathit{cycle}) + 1 & \wedge \\ t \bmod \mathit{cycle} = 0 & \wedge \\ \mathcal{F} \triangleq \{(r, c) | \exists (\rho, \kappa) \in \overline{\mathit{Clear}} : \text{e-adjacent}\,((\rho, \kappa), (r, c))\} & \wedge \\ (\mathit{Clear} \setminus \mathcal{F}) \cup \{(\mathit{row}, \mathit{col})\} = \mathcal{C} & \end{array}\right\} \textbf{evader-move} \left\{\begin{array}{ll} \mathit{time} = t & \wedge \\ \mathit{Clear} = \mathcal{C} & \end{array}\right\} \tag{5}$$

Invoking **evader-move** has two effects. The first is to advance the clock to the beginning of the pursuer's next turn. The other effect is to change the set of $\mathrm{cleared}$ cells. Any cell that is $\mathrm{e\text{-}adjacent}$ to an $\mathrm{uncleared}$ cell cannot be $\mathrm{cleared}$ after the evader has moved; the one exception is the cell currently occupied by the pursuer.

e) Wait Axiom: The Wait Axiom was introduced as (3) in the text. Its purpose is to provide a mechanism to increment time without moving the pursuer or evader.

$$\left\{\begin{array}{ll} \mathit{time} = t - \mathit{duration} & \wedge \\ t \,\mathrm{div}\, \mathit{cycle} = \mathit{time} \,\mathrm{div}\, \mathit{cycle} & \end{array}\right\} \textbf{wait}\ \mathit{duration}\, \{\mathit{time} = t\} \tag{3}$$

2) Inference Rules: We now cover seven rules of inference. Six of these rules are used when the derivation uses structured programming constructs; the other rule is can be used simplify derivations.

a) Consequence Rule: The Consequence Rule is the combination of Hoare's two Rules of Consequence [2].

$$\{p\}\,\mathrm{S}\,\{q\} \quad , \quad p' \Rightarrow p \quad , \quad q \Rightarrow q' \qquad \vdash \qquad \{p'\}\,\mathrm{S}\,\{q'\} \tag{8}$$

Put simply, the Consequence Rule permits us to begin the derivation step with a stronger precondition than the specified precondition and end the derivation step asserting a weaker postcondition than the specified postcondition. For example, if the specification is $\{\mathrm{TRUE}\}\,\mathrm{S}\,\{x = 3\}$ then our derivation step could instead show $\{x = 42\}\,\mathrm{S}\,\{x < 42\}$.

b) Sequential Composition Rule: The Sequential Composition Rule is Hoare's Rule of Sequential Composition [2]. It permits us to combine two program fragments as straight-line code.

$$\{p\}\,\mathrm{S}_1\,\{q\} \quad , \quad \{q\}\,\mathrm{S}_2\,\{r\} \qquad \vdash \qquad \{p\}\,\mathrm{S}_1;\mathrm{S}_2\,\{r\} \tag{9}$$

Note that because of the Consequence Rule, the specified postcondition of S_1 and the precondition of S_2 need not be identical.

c) Conditional Rules: When Hoare introduced axiomatic semantics [2], he initially omitted the **if-then-else** construct. Within a couple years, he had corrected this, and the first Conditional Rule is his Rule of Alternation [3].

$$\{p \wedge b\}\,\mathrm{S}_1\,\{q\} \quad , \quad \{p \wedge \neg b\}\,\mathrm{S}_2\,\{q\} \qquad \vdash \qquad \{p\}\,\textbf{if}\ b\ \textbf{then}\ \ \mathrm{S}_1\ \ \textbf{else}\ \ \mathrm{S}_2\ \ \textbf{end if}\,\{q\} \tag{10}$$

Our second Conditional Rule is a special case of the first, where S_2 can be considered to be **no-op**:

$$\{p \wedge b\}\,\mathrm{S}\,\{q\} \quad , \quad p \wedge \neg b \Rightarrow q \qquad \vdash \qquad \{p\}\,\textbf{if}\ b\ \textbf{then}\ \ \mathrm{S}\ \ \textbf{end if}\,\{q\} \tag{11}$$

d) Iteration Rule: The Iteration Rule comes from Hoare's Rule of Iteration [2].

$$\{p \wedge b\}\,\mathrm{S}\,\{p\} \qquad \vdash \qquad \{p\}\,\textbf{while}\ b\ \textbf{do}\ \ \mathrm{S}\ \ \textbf{end do}\,\{p \wedge \neg b\} \tag{12}$$

A derivation involving a loop requires us to establish a loop invariant, that is, a proposition that will be satisfied at the beginning and end of every loop iteration. If the loop invariant p holds before the **while** loop, then p and the negation of the loop condition will hold after the **while** loop. Note that this Iteration Rule can only be used to establish partial correctness: the rule says nothing about whether the loop will terminate, only that if the loop terminates then $p \wedge \neg b$ are satisfied after it terminates.

e) Invocation Rules: Our first Invocation Rule is Hoare's first Rule of Invocation [3], for procedures that do not use parameters.

$$\mathit{declare}\ \textsc{Proc}\ \ \mathrm{S} \quad , \quad \{p\}\,\mathrm{S}\,\{q\} \qquad \vdash \qquad \{p\}\ \mathit{call}\ \ \textsc{Proc}\,\{q\} \tag{13}$$

We combine Hoare's second Rule of Invocation with his Rule of Substitution for our second Invocation Rule. $\overrightarrow{params}$ is the list of formal parameters for the procedure, and $\overrightarrow{args}$ is a list of expressions that correspond to the formal parameters.

$$\begin{gathered} \mathit{declare}\ \textsc{Proc}(\overrightarrow{params})\ \ \mathrm{S} \quad , \quad \{p\}\,\mathrm{S}\,\{q\} \quad , \quad \|\overrightarrow{params}\| = \|\overrightarrow{args}\| \\ \vdash \qquad \left\{p^{\overrightarrow{params}}_{\overrightarrow{args}}\right\}\ \mathit{call}\ \ \textsc{Proc}(\overrightarrow{args})\ \left\{q^{\overrightarrow{params}}_{\overrightarrow{args}}\right\} \end{gathered} \tag{14}$$

While our language is pass-by-value, we make the simplifying assumption in this proof rule that no terms in the expressions in $\overrightarrow{args}$ are assigned new values in the procedure body. Of course doing so would have no effect on the *values* in $\overrightarrow{args}$, but it would complicate the substitution portion of our second Invocation Rule.

We have not used all the rules from Hoare's axiomatic treatment of procedures [3]. For example, Invocation Rules (13) and (14) are insufficient for recursive calls, but we do not use recursive calls in this paper. We also have not introduced Hoare's Rule of Declaration, since we do not use local variable names inside procedure bodies that are also the names of variables in a greater scope.

3) Total Correctness: In the previous two sections, we covered the material that can be used to demonstrate program correctness *if the program terminates error-free*. We will now cover the material that can be used to demonstrate that the program will terminate.

a) Skip Lemma: We begin with the trivial: by definition, **no-op** will always terminate without generating a run-time error.

$$\{p\}\ \textbf{no-op}\ \{p\}\downarrow \tag{15}$$

b) Assignment Rule: The Assignment axiom is not, in of itself, sufficient to establish that an assignment command will terminate error-free. As we did with the Skip Axiom, we could define assignments as always terminating, but that would be inappropriate. Manna did not alter his Assignment Axiom *check name!* [4], there are still two ways in which the assignment would fail to terminate correctly ***reference!***. If $expr$ contains a function call, then the function might not terminate; as our language does not include functions, this is not an issue for us. The other concern is that the evaluation of $expr$ might generate an error: it might involve a divide-by-zero error, or it might evaluate to a value outside x's range.

$$\text{“}expr \text{ is error-free”} \;\;,\;\; \left\{p^x_{expr}\right\} x \leftarrow expr \,\{p\} \quad \vdash \quad \left\{p^x_{expr}\right\} x \leftarrow expr \,\{p\} \downarrow \tag{16}$$

c) Move Rule: Similar to the concern with assignments that an illegal value not be assigned, we must be cautious that when the pursuer moves, it does not move off the board.

$$\begin{gathered}\text{“}dir \text{ and } spaces \text{ are error-free”} \;\;,\;\; p \Rightarrow (0 \le row < n) \wedge (0 \le col < m) \;\;, \\ q \Rightarrow (0 \le row < n) \wedge (0 \le col < m) \;\;,\;\; \{p\}\, \textbf{move}\; dir\; spaces\, \{q\} \quad \vdash \quad \{p\}\, \textbf{move}\; dir\; spaces\, \{q\} \downarrow\end{gathered} \tag{17}$$

d) Evader-Move Lemma: As with **no-op**, we can safely define **evader-move** $\downarrow$:

$$\left\{\begin{array}{ll} t \operatorname{div} cycle = (time \operatorname{div} cycle) + 1 & \wedge \\ t \operatorname{mod} cycle = 0 & \wedge \\ \mathcal{F} \triangleq \left\{(r,c) | \exists (\rho,\kappa) \in \overline{Clear} : \text{e-adjacent}\left((\rho,\kappa),(r,c)\right)\right\} & \wedge \\ (Clear \setminus \mathcal{F}) \cup \{(row, col)\} = \mathcal{C} & \end{array}\right\} \textbf{evader-move} \left\{\begin{array}{ll} time = t & \wedge \\ Clear = \mathcal{C} & \end{array}\right\} \downarrow \tag{18}$$

e) Wait Rule: The **wait** command can safely be assumed to terminate, provided the precondition is satisfied and expression within the command is cleanly evaluated:

$$\text{“}duration \text{ is error-free”} \;\;,\;\; \{p\}\, \textbf{wait}\; duration\, \{q\} \quad \vdash \quad \{p\}\, \textbf{wait}\; duration\, \{q\} \downarrow \tag{19}$$

f) Consequence, Sequential Composition, and Conditional Rules: These rules require no special treatment *(double check!)*; if the program fragments terminate, then so do the constructs *(probably ought to cite that)*:

$$\{p\}\, \mathrm{S}\, \{q\} \downarrow \;\;,\;\; p' \Rightarrow p \;\;,\;\; q \Rightarrow q' \quad \vdash \quad \{p'\}\, \mathrm{S}\, \{q'\} \downarrow \tag{20}$$

$$\{p\}\, \mathrm{S}_1\, \{q\} \downarrow \;\;,\;\; \{q\}\, \mathrm{S}_2\, \{r\} \downarrow \quad \vdash \quad \{p\}\, \mathrm{S}_1; \mathrm{S}_2\, \{r\} \downarrow \tag{21}$$

$$\{p \wedge b\}\, \mathrm{S}_1\, \{q\} \downarrow \;\;,\;\; \{p \wedge \neg b\}\, \mathrm{S}_2\, \{q\} \downarrow \quad \vdash \quad \{p\}\, \textbf{if}\; b\; \textbf{then}\; \mathrm{S}_1\; \textbf{else}\; \mathrm{S}_2\; \textbf{end if}\, \{q\} \downarrow \tag{22}$$

$$\{p \wedge b\}\, \mathrm{S}\, \{q\} \downarrow \;\;,\;\; p \wedge \neg b \Rightarrow q \quad \vdash \quad \{p\}\, \textbf{if}\; b\; \textbf{then}\; \mathrm{S}\; \textbf{end if}\, \{q\} \downarrow \tag{23}$$

g) Iteration Rule: The Iteration Rule (24) in Section A.2 uses a loop invariant to establish that a loop has correct behavior if it terminates. We now introduce a variant function to establish that the loop terminates. When Manna introduced this concept, he required the variant function ranges over a well-founded set [4]; however for simplicity's sake, we shall limit that to the natural numbers.

$$\begin{gathered}\exists\, \mathrm{f} : \mathcal{Q} \rightarrow \mathbb{N} \;\; \forall k \in \mathbb{N}^+ \;\; :: \;\; \{p \wedge b \wedge (\mathrm{f}(\sigma) = k)\}\, \mathrm{S}\, \{p \wedge (\mathrm{f}(\sigma) < k)\} \downarrow \;\;, \\ (\mathrm{f}(\sigma) = 0) \Rightarrow \neg b \quad \vdash \quad \{p\}\, \textbf{while}\; b\; \textbf{do}\; \mathrm{S}\; \textbf{end do}\, \{p \wedge \neg b\} \downarrow\end{gathered} \tag{24}$$

Notice that the requirement for f is that each iteration reduce its value. Since f ranges over $\mathbb{N}$, it will eventually reach 0, and the loop must terminate when $\mathrm{f}(\sigma) = 0$.

h) Invocation Rules: Total correctness of the first Invocation Rule requires no special treatement; if the procedure body terminates, then the procedure call terminates *(reference)*:

$$\textit{declare}\; \textsc{Proc}\;\; \mathrm{S} \;\;,\;\; \{p\}\, \mathrm{S}\, \{q\} \downarrow \quad \vdash \quad \{p\}\; \textit{call}\; \textsc{Proc}\, \{q\} \downarrow \tag{25}$$

Finally, total correctness of the second Invocation Rule requires not only that the procedure body terminate error-free, but also that the expressions in $\overrightarrow{args}$ evaluate error-free *(reference)*.

$$\begin{gathered}\textit{declare}\; \textsc{Proc}(\overrightarrow{params})\;\; \mathrm{S} \;\;,\;\; \{p\}\, \mathrm{S}\, \{q\} \downarrow \;\;,\;\; \|\overrightarrow{params}\| = \|\overrightarrow{args}\| \;\;, \\ \forall\, expr \in \overrightarrow{args} : \text{“}expr \text{ is error-free”} \quad \vdash \quad \left\{p^{\overrightarrow{params}}_{\overrightarrow{args}}\right\}\; \textit{call}\; \textsc{Proc}(\overrightarrow{args})\, \left\{q^{\overrightarrow{params}}_{\overrightarrow{args}}\right\} \downarrow\end{gathered} \tag{26}$$

B. *Derivation of* CLEAR-BOARD *in Figure 5*

We make use of the following loop invariant:

$$(x < m) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = x < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \tag{27}$$

And the following loop variant function:

$$\mathrm{f}(\sigma) = m - x \tag{28}$$

```
  $\{\ (time = 0) \wedge (0 \leq col = 0 < m) \wedge (0 \leq row = 0 < n)\ \}$ ▷ precondition
1 $x \leftarrow 0$
  $\{\ (x = 0) \wedge (time = 0) \wedge (0 \leq col = 0 < m) \wedge (0 \leq row = 0 < n)\ \}$
  $\Rightarrow$
  $\{\ (x = 0) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = x < m) \wedge (0 \leq row = 0 < n)\ \}$
  $\triangleright\ 0 < m - 1 < m$
  $\triangleright\ \{(\varrho, \kappa) | \varrho < n, \kappa < 0\} = \emptyset$
  $\triangleright\ \mathrm{f}(\sigma) = 0 \Rightarrow x = m \geq m - 1$
2 while $x < m - 1$
3     do
  $\left\{ \begin{array}{l} (x < m - 1 < m) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = x < m) \wedge \\ (0 \leq row = 0 < n) \wedge (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \end{array} \right\} \triangleright \begin{array}{l} (27) \wedge (x < m - 1) \\ \mathrm{f}(\sigma) = m - x \triangleq k \geq 1 \end{array}$
4         CLEAR-COLUMN
  $\left\{ \begin{array}{l} (x < m - 1) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = x + 1 < m) \wedge \\ (0 \leq row = 0 < n) \wedge (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \end{array} \right\}$
5         $x \leftarrow x + 1$
  $\left\{ \begin{array}{l} (x < m) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = x < m) \wedge \\ (0 \leq row = 0 < n) \wedge (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \end{array} \right\} \triangleright \begin{array}{l} (27) \\ \mathrm{f}(\sigma) = m - x - 1 < k \end{array}$
6     end do
  $\left\{ \begin{array}{l} (x = m - 1 < m) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = m - 1 < m) \wedge \\ (0 \leq row = 0 < n) \wedge (\forall \varrho < n, \kappa < m\text{-}1 : (\varrho, \kappa) \in Clear) \end{array} \right\} \triangleright (27) \wedge \neg(x < m - 1)$
7 CLEAR-LAST-COLUMN
  $\{\ (\forall \varrho < n, \kappa < m : (\varrho, \kappa) \in Clear)\ \}$ ▷ postcondition
```

C. *Derivation of* CLEAR-COLUMN *in Figure 9*

We make use of the following loop invariant:

$$(0 < c < m) \wedge (y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y + 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \tag{29}$$

And the following loop variant function:

$$\mathrm{f}(\sigma) = y \tag{30}$$

Note that the c in this invariant is the c "magic" variable[1] in CLEAR-COLUMN's specification, which is distinct from the c "magic" variables in CLEAR-CELL and CLEAR-LAST-CELL. The significance of its appearance in the loop invariant is that the pursuer must begin every iteration in the same column.

```
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \end{array} \right\} \triangleright$ precondition
1    $y \leftarrow n - 1$
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = n - 1) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \end{array} \right\}$
     $\triangleright\ 0 < n - 1 < n$
     $\triangleright\ \{(\varrho, \mathit{col}) | n - 1 + 1 \leq \varrho < n\} = \emptyset$
     $\triangleright\ \mathrm{f}(\sigma) = 0 \Rightarrow y \leq 0$
2    while $y > 0$
3        do
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (0 < y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y + 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\} \triangleright \begin{array}{l} (29) \wedge (y > 0) \\ \mathrm{f}(\sigma) = y \triangleq k \end{array}$
     $\triangleright\ c_{\text{CLEAR-CELL}} = c_{\text{CLEAR-COLUMN}} - 1$
4            CLEAR-CELL$(y)$
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (0 < y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$
     $\Rightarrow$
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (y - 1 < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y - 1 + 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$
5            $y \leftarrow y - 1$
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y + 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\} \triangleright \begin{array}{l} (29) \\ \mathrm{f}(\sigma) = y - 1 < k \end{array}$

6        end do
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = 0 < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y + 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\} \triangleright (29) \wedge \neg(y > 0)$
     $\Rightarrow$
     $\left\{ \begin{array}{l} (0 < c < m) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$
     $\triangleright\ c_{\text{CLEAR-LAST-CELL}} = c_{\text{CLEAR-COLUMN}}$
7    CLEAR-LAST-CELL
     $\left\{ \begin{array}{l} (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \end{array} \right\} \triangleright$ postcondition
```

[1]is there a better name? "parameter"? "specification variable"?

D. Derivation of CLEAR-LAST-COLUMN *in Figure 13*

We assume the time is initially t_0.

```
$\left\{\begin{array}{l} (\mathit{speed} \geq n-1) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge \\ (0 \leq \mathit{col} = m-1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\forall \varrho < n, \kappa < m-1 : (\varrho, \kappa) \in \mathit{Clear}) \end{array}\right\}$ ▷ precondition
$\Rightarrow$
$\left\{\begin{array}{l} (\mathit{cycle} > n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = m-1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < m-1\} \subseteq \mathit{Clear}) \wedge ((t_0 + n - 1) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}) \end{array}\right\}$
1  **move** N $n-1$
$\left\{\begin{array}{l} (\mathit{cycle} > n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n - 1) \wedge (0 \leq \mathit{col} = m-1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < m\} \subseteq \mathit{Clear}) \end{array}\right\}$
$\Rightarrow$
$\left\{ (\forall \varrho < n, \kappa < m : (\varrho, \kappa) \in \mathit{Clear}) \right\}$ ▷ postcondition
```

E. Derivation of CLEAR-LAST-CELL *in Figure 14*

We assume the time is initially t_0.

$$\left\{\begin{array}{l} (\mathit{speed} \geq n) \wedge (0 < c < m) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c-1 < m) \wedge \\ (0 \leq \mathit{row} = 0 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array}\right\} \triangleright \text{precondition}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c-1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho, c-1) | 1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge ((t_0+1) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

1 **move** E 1

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + 1) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(0, c)\} \subseteq \mathit{Clear}) \wedge ((t_0+n) \operatorname{div} \mathit{cycle} = (t_0+1) \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

2 **move** N $n-1$

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \subseteq \mathit{Clear}) \end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n-1 < \\ (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \subseteq \mathit{Clear}) \wedge (\forall \varrho < n : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge \\ ((t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + n) \operatorname{div} \mathit{cycle} + 1) \end{array}\right.$$

3 **evader-move**

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle}) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n-1 \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(n-1, c)\} \subseteq \mathit{Clear}) \wedge ((t_0 + \mathit{cycle} + n - 1) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle}) \end{array}\right.$$

4 **move** S $n-1$

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle} + n - 1) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \subseteq \mathit{Clear}) \end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge ((t_0 + 2\,\mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle} + n - 1) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \subseteq \mathit{Clear}) \wedge (\forall \varrho < n : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge \\ ((t_0 + 2\,\mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n - 1) \operatorname{div} \mathit{cycle} + 1) \end{array}\right\}$$

5 **evader-move**

$$\left\{\begin{array}{l} (\mathit{cycle} > n) \wedge (0 < c < m) \wedge ((t_0 + 2\,\mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + 2\,\mathit{cycle}) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(0, c)\} \subseteq \mathit{Clear}) \end{array}\right\}$$

$\Rightarrow$

$$\left\{ (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \right\} \triangleright \text{postcondition}$$

F. Derivation of CLEAR-CELL *in Figure 15*

We assume the time is initially t_0.

$$\left\{\begin{array}{l}(speed \geq n) \wedge (c < m-1) \wedge (0 < y < n) \wedge (time \bmod cycle = 0) \wedge (time = t_0) \wedge (0 \leq col = c < m) \wedge \\ (0 \leq row = 0 < n) \wedge (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho : y+1 \leq \varrho < n : (\varrho, col) \in Clear)\end{array}\right\} \triangleright \text{precondition}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod cycle = 0) \wedge (time = t_0) \wedge \\ (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y+1 \leq \varrho < n\} \subseteq Clear) \wedge \\ ((t_0 + y) \operatorname{div} cycle = t_0 \operatorname{div} cycle)\end{array}\right\}$$

1 **move** N y

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod cycle = 0) \wedge (time = t_0 + y) \wedge \\ (0 \leq col = c < m) \wedge (0 \leq row = y < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \subseteq Clear) \wedge ((t_0 + y + 1) \operatorname{div} cycle = (t_0 + y) \operatorname{div} cycle)\end{array}\right.$$

2 **move** E 1

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod cycle = 0) \wedge (time = t_0 + y + 1) \wedge \\ (0 \leq col = c+1 < m) \wedge (0 \leq row = y < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \cup \{(y, c+1)\} \subseteq Clear) \wedge \\ ((t_0 + n) \operatorname{div} cycle = t_0 + y + 1 \operatorname{div} cycle)\end{array}\right\}$$

3 **move** N $n - y - 1$

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod cycle = 0) \wedge (time = t_0 + n) \wedge \\ (0 \leq col = c+1 < m) \wedge (0 \leq row = n-1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y \leq \varrho < n\} \subseteq Clear)\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod cycle = 0) \wedge (time = t_0 + n) \wedge \\ (0 \leq col = c+1 < m) \wedge (0 \leq row = n-1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y \leq \varrho < n\} \subseteq Clear) \wedge \\ (\forall \varrho : \varrho < y : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : y \leq \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ ((t_0 + cycle) \operatorname{div} cycle = (t_0 + n) \operatorname{div} cycle + 1)\end{array}\right\}$$

4 **evader-move**

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle) \wedge \\ (0 \leq col = c+1 < m) \wedge (0 \leq row = n-1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y \leq \varrho < n\} \subseteq Clear) \wedge \\ ((t_0 + cycle + n - y - 1) \operatorname{div} cycle = (t_0 + cycle) \operatorname{div} cycle)\end{array}\right\}$$

5 **move** S $n - y - 1$

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle + n - y - 1) \wedge \\ (0 \leq col = c+1 < m) \wedge (0 \leq row = y < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, \kappa) | y \leq \varrho < n, c \leq \kappa \leq c+1\} \subseteq Clear) \wedge \\ ((t_0 + cycle + n - y) \operatorname{div} cycle = (t_0 + cycle + n - y - 1) \operatorname{div} cycle)\end{array}\right\}$$

6 **move** W 1

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle + n - y) \wedge \\ (0 \leq col = c < m) \wedge (0 \leq row = y < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, \kappa) | y \leq \varrho < n, c \leq \kappa \leq c+1\} \subseteq Clear) \wedge \\ ((t_0 + cycle + n) \operatorname{div} cycle = (t_0 + cycle + n - y) \operatorname{div} cycle)\end{array}\right\}$$

7 **move** S y

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle + n) \wedge \\ (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y \leq \varrho < n\} \subseteq Clear)\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle + n) \wedge \\ (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y \leq \varrho < n\} \subseteq Clear) \wedge \\ (\forall \varrho : \varrho < y : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : y \leq \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ ((t_0 + 2\,cycle) \operatorname{div} cycle = (t_0 + cycle + n) \operatorname{div} cycle + 1)\end{array}\right\}$$

8 **evader-move**

$$\left\{\begin{array}{l}(cycle > n) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + 2\,cycle) \bmod cycle = 0) \wedge (time = t_0 + 2\,cycle) \wedge \\ (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y \leq \varrho < n\} \subseteq Clear)\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho : y \leq \varrho < n : (\varrho, col) \in Clear)\end{array}\right\} \triangleright \text{postcondition}$$

G. *Derivation of* CLEAR-CELL *in Figure 17*

We assume the time is initially t_0.

$$\left\{\begin{array}{l} (\mathit{speed} \geq n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = 0 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y+1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array}\right\} \triangleright \text{ precondit}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y+1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + y) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

1 **move** N y

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \subseteq \mathit{Clear}) \wedge ((t_0 + y + 1) \operatorname{div} \mathit{cycle} = (t_0 + y) \operatorname{div} \mathit{cycle}) \end{array}\right.$$

2 **move** SE 1

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y + 1) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = y-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(y-1, c+1)\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + n + 1) \operatorname{div} \mathit{cycle} = (t_0 + y + 1) \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

3 **move** N $n-y$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n + 1) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n + 1) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ (\forall \varrho : \varrho < y : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : y-1 \leq \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ (\text{e-adjacent}((c+1, y-2), (c, y-1))) \wedge ((t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + n + 1) \operatorname{div} \mathit{cycle} + 1) \end{array}\right\}$$

4 **evader-move**

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle}) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + \mathit{cycle} + n - y) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

5 **move** S $n-y$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle} + n - y) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, \kappa) | y \leq \varrho < n, c \leq \kappa \leq c+1\} \cup \{(y-1, c+1)\} \subseteq \mathit{Clear}) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = y-1 < n) \wedge ((t_0 + \mathit{cycle} + n - y + 1) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n - y) \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

6 **move** W 1

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + \mathit{cycle} + n - y + 1) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y-1 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, \kappa) | y-1 \leq \varrho < n, c \leq \kappa \leq c+1\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + \mathit{cycle} + n) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n - y + 1) \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

7 **move** S $y-1$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\mathit{time} = t_0 + \mathit{cycle} + n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \end{array}\right.$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle} + n) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ (\forall \varrho : \varrho < y : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : y-1 \leq \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ (\text{e-adjacent}((c+1, y-2), (c, y-1))) \wedge ((t_0 + 2\,\mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n) \operatorname{div} \mathit{cycle} + 1) \end{array}\right\}$$

8 **evader-move**

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + 2\,\mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + 2\,\mathit{cycle}) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y \leq \varrho < n\} \subseteq \mathit{Clear}) \end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array}\right\} \triangleright \text{ postcondition}$$

H. *Derivation of* CLEAR-CELL *in Figure 19*

We assume the time is initially t_0.

$$\left\{\begin{array}{l}(\mathit{speed} \geq n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = 0 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y+1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear})\end{array}\right\} \triangleright \text{ precondit}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y+1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + y) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle})\end{array}\right\}$$

1 **move** N y

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \subseteq \mathit{Clear}) \wedge ((t_0 + y + 1) \operatorname{div} \mathit{cycle} = (t_0 + y) \operatorname{div} \mathit{cycle})\end{array}\right.$$

2 **move** E 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y + 1) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = y < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(y, c+1)\} \subseteq \mathit{Clear}) \wedge ((t_0 + y + 2) \operatorname{div} \mathit{cycle} = (\end{array}\right.$$

3 **move** S 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y + 2) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = y-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(y-1, c+1), (y, c+1)\} \subseteq \mathit{Clear}) \wedge ((t_0 + n\end{array}\right.$$

4 **move** N $n - y$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n + 2) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear})\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n + 2) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ (\forall \varrho : \varrho < y : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : y-1 \leq \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ (\text{e-adjacent}((c+1, y-2), (c, y-1))) \wedge ((t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + n + 2) \operatorname{div} \mathit{cycle} + 1)\end{array}\right\}$$

5 **evader-move**

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle}) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + \mathit{cycle} + n - y) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

6 **move** S $n - y$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle} + n - y) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, \kappa) | y \leq \varrho < n, c \leq \kappa \leq c+1\} \cup \{(y-1, c+1)\} \subseteq \mathit{Clear}) \wedge \\ (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = y-1 < n) \wedge ((t_0 + \mathit{cycle} + n - y + 1) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n - y) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

7 **move** W 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + \mathit{cycle} + n - y + 1) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y-1 < n) \wedge \\ (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, \kappa) | y-1 \leq \varrho < n, c \leq \kappa \leq c+1\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + \mathit{cycle} + n) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n - y + 1) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

8 **move** S $y - 1$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\mathit{time} = t_0 + \mathit{cycle} + n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear})\end{array}\right.$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle} + n) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ (\forall \varrho : \varrho < y : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : y-1 \leq \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ (\text{e-adjacent}((c+1, y-2), (c, y-1))) \wedge ((t_0 + 2\,\mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + \mathit{cycle} + n) \operatorname{div} \mathit{cycle} + 1)\end{array}\right\}$$

9 **evader-move**

$$\left\{\begin{array}{l}(\mathit{cycle} > n+2) \wedge (c < m-1) \wedge (0 < y < n) \wedge ((t_0 + 2\,\mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + 2\,\mathit{cycle}) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho, \kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | y \leq \varrho < n\} \subseteq \mathit{Clear})\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear})\end{array}\right\} \triangleright \text{ postcondition}$$

I. Derivation of Clear-Column *in Figure 26*

We make use of two loop invariants. For the loop in lines 2–7:

$$(0 < c < m) \wedge (y \leq \lfloor \tfrac{n}{2} \rfloor) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (y < \lceil \tfrac{n}{2} \rceil \Rightarrow \mathit{row} = y) \wedge$$
$$(y = \lceil \tfrac{n}{2} \rceil \Rightarrow \mathit{row} = y - 1) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : n - y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \tag{31}$$

For the loop in lines 9–14:

$$(0 < c < m) \wedge (\lfloor \tfrac{n}{2} \rfloor \leq y < n) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (y < n - 1 \Rightarrow \mathit{row} = 2 \lfloor \tfrac{n}{2} \rfloor - y - 1) \wedge$$
$$(y = n - 1 \Rightarrow \mathit{row} = 0) \wedge (\forall \varrho < n, \kappa < \mathit{col} - 1 : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho < y : (\varrho, \mathit{col} - 1) \in \mathit{Clear}) \wedge$$
$$(y = n - 1 \Rightarrow (n - 1, \mathit{col} - 1) \in \mathit{Clear}) \tag{32}$$

Both loops use the following loop variant function:

$$\mathrm{f}(\sigma) = n - y \tag{33}$$

```
  $\left\{ \begin{array}{l} (0 < c < m) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \end{array} \right\}$ $\triangleright$ precondition
1 $y \leftarrow 0$
  $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = 0) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \end{array} \right\}$
  $\triangleright\ 0 < \lfloor \frac{n}{2} \rfloor$
  $\triangleright\ \{(\varrho, \mathit{col}) | n - 0 \leq \varrho < n\} = \emptyset$
  $\triangleright\ \mathrm{f}(\sigma) = 0 \Rightarrow y = n \geq n - 1 \geq \lfloor \frac{n}{2} \rfloor$
2 while $y < \lfloor \frac{n}{2} \rfloor$
3     do
  $\left\{ \begin{array}{l} (0 < c < m) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = y < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : n - y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$ $\triangleright$ $\begin{array}{l} (31) \wedge (y < \lfloor \frac{n}{2} \rfloor \\ \mathrm{f}(\sigma) = n - y \triangleq \end{array}$
  $\triangleright\ c_{\text{Grow-Bottom1}} = c_{\text{Clear-Column}} - 1$
4         Grow-Bottom1$(y)$
  $\left\{ \begin{array}{l} (0 < c < m) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = n - y - 1 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho \leq y : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$
  $\triangleright\ c_{\text{Grow-Top1}} = c_{\text{Clear-Column}} - 1$
5         Grow-Top1$(y)$
  $\left\{ \begin{array}{l} (0 < c < m) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (y < \lceil \frac{n}{2} \rceil - 1 \Rightarrow \mathit{row} = y + 1) \wedge \\ (y = \lceil \frac{n}{2} \rceil - 1 \Rightarrow \mathit{row} = y) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : n - y - 1 \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$
  $\Rightarrow$
  $\left\{ \begin{array}{l} (0 < c < m) \wedge (y \leq \lfloor \frac{n}{2} \rfloor - 1) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c - 1 < m) \wedge (y < \lceil \frac{n}{2} \rceil - 1 \Rightarrow \mathit{row} = y + 1) \wedge \\ (y = \lceil \frac{n}{2} \rceil - 1 \Rightarrow \mathit{row} = y) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : n - (y + 1) \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$
```

```
6           $y \leftarrow y + 1$
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (y \leq \lfloor \frac{n}{2} \rfloor) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c - 1 < m) \wedge (y < \lceil \frac{n}{2} \rceil \Rightarrow row = y) \wedge \\ (y = \lceil \frac{n}{2} \rceil \Rightarrow row = y - 1) \wedge (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho : n - y \leq \varrho < n : (\varrho, col) \in Clear) \end{array} \right\}$ $\rhd$ (31)
                                                                                        $\mathrm{f}(\sigma) = n$
7       end do
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = \lfloor \frac{n}{2} \rfloor) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c - 1 < m) \wedge (y < \lceil \frac{n}{2} \rceil \Rightarrow row = y) \wedge \\ (y = \lceil \frac{n}{2} \rceil \Rightarrow row = y - 1) \wedge (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \wedge \\ (\forall \varrho : n - y \leq \varrho < n : (\varrho, col) \in Clear) \end{array} \right\}$ $\rhd$ (31) $\wedge \neg(y <$
    $\rhd$ is-odd$(n) \Rightarrow y = \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil - 1 < \lceil \frac{n}{2} \rceil \Rightarrow row = y = \lceil \frac{n}{2} \rceil - 1$
    $\rhd$ is-even$(n) \Rightarrow y = \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil \Rightarrow row = y - 1 = \lceil \frac{n}{2} \rceil - 1$
    $\rhd$ $n - y = n - \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil > \lceil \frac{n}{2} \rceil - 1 = row$
    $\Rightarrow$
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = \lfloor \frac{n}{2} \rfloor) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c - 1 < m) \wedge \big(0 \leq row = \lceil \frac{n}{2} \rceil - 1 < n\big) \wedge \\ (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho : row < \varrho < n : (\varrho, col) \in Clear) \end{array} \right\}$
    $\rhd$ $c_{\text{TRANSITION}} = c_{\text{CLEAR-COLUMN}}$
8   TRANSITION
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = \lfloor \frac{n}{2} \rfloor) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge \big(0 \leq row = 2 \lfloor \frac{n}{2} \rfloor - y - 1 < n\big) \wedge \\ (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho < y : (\varrho, col - 1) \in Clear) \end{array} \right\}$
9   while $y < n - 1$
10      do
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n - 1) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge \big(0 \leq row = 2 \lfloor \frac{n}{2} \rfloor - y - 1 < n\big) \wedge \\ (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho < y : (\varrho, col - 1) \in Clear) \end{array} \right\}$ $\rhd$
    $\rhd$ $c_{\text{GROW-TOP2}} = c_{\text{CLEAR-COLUMN}}$
11          GROW-TOP2$(y)$
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n - 1) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (0 \leq row = y + 1 < n) \wedge \\ (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge \big(\forall \varrho : 2 \lfloor \frac{n}{2} \rfloor - y \leq \varrho < n : (\varrho, col - 1) \in Clear\big) \end{array} \right\}$
    $\rhd$ $c_{\text{GROW-BOTTOM2}} = c_{\text{CLEAR-COLUMN}}$
12          GROW-BOTTOM2$(y)$
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n - 1) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (y < n - 2 \Rightarrow row = 2 \lfloor \frac{n}{2} \rfloor - y - 2) \wedge \\ (y = n - 2 \Rightarrow row = 0) \wedge (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho \leq y : (\varrho, col - 1) \in Clear) \wedge \\ (y = n - 2 \Rightarrow \forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \end{array} \right.$
    $\Rightarrow$
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor - 1 \leq y < n - 1) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (y < n - 2 \Rightarrow row = 2 \lfloor \frac{n}{2} \rfloor - y - \\ (y = n - 2 \Rightarrow row = 0) \wedge (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho < y + 1 : (\varrho, col - 1) \in Clear) \wedge \\ (y = n - 2 \Rightarrow (n - 1, col - 1) \in Clear) \end{array} \right.$
13          $y \leftarrow y + 1$

    $\left\{ \begin{array}{l} (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (y < n - 1 \Rightarrow row = 2 \lfloor \frac{n}{2} \rfloor - y - 1) \wedge \\ (y = n - 1 \Rightarrow row = 0) \wedge (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho < y : (\varrho, col - 1) \in Clear) \wedge \\ (y = n - 1 \Rightarrow (n - 1, col - 1) \in Clear) \end{array} \right\}$ $\rhd$
14      end do
    $\left\{ \begin{array}{l} (0 < c < m) \wedge (y = n - 1 < n) \wedge (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < col - 1 : (\varrho, \kappa) \in Clear) \wedge (\forall \varrho < y : (\varrho, col - 1) \in Clear) \wedge \\ ((n - 1, col - 1) \in Clear) \end{array} \right\}$ $\rhd$ (32) $\wedge \neg(y <$
                                                                                        $\mathrm{f}(\sigma) = n -$
    $\Rightarrow$
    $\left\{ \begin{array}{l} (time \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge \\ (\forall \varrho < n, \kappa < col : (\varrho, \kappa) \in Clear) \end{array} \right\}$ $\rhd$ postcondition
```

J. *Derivation of* Grow-Bottom1 *in Figure 27*

We assume the time is initially t_0.

$$\left\{\begin{array}{l}(\mathit{speed} \geq n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = y < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : n-y \leq \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear})\end{array}\right\} \triangleright \text{preconditio}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | n-y \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + y) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle})\end{array}\right\}$$

1 **move** S y

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + y + 1) \operatorname{div} \mathit{cycle} = (t_0 + y) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

2 **move** E 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y + 1) \wedge (0 \leq \mathit{col} = c+1 < m) \wedge \\ (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y \leq \varrho < n\} \cup \{(0, c+1)\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + 2y + 2) \operatorname{div} \mathit{cycle} = (t_0 + y + 1) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

3 **move** N $y+1$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + 2y + 2) \wedge (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = y+1 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y \leq \varrho < n\} \cup \{(\varrho, c+1) | \varrho \leq y+1\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + 2y + 3) \operatorname{div} \mathit{cycle} = (t_0 + 2y + 2) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

4 **move** W 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + 2y + 3) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y+1 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y+1\} \cup \{(\varrho, c) | n-y \leq \varrho < n\} \cup \{(\varrho, c+1) | \varrho \leq y+1\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + n + 1) \operatorname{div} \mathit{cycle} = (t_0 + 2y + 3) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

5 **move** N $n - 2y - 2$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + n + 1) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n - y - 1 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq n - y - 1\} \cup \{(\varrho, c) | n-y \leq \varrho < n\} \cup \{(\varrho, c+1) | \varrho \leq y+1\} \subseteq \mathit{Clear})\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n + 1) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n - y - 1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c\} \cup \{(\varrho, c+1) | \varrho \leq y+1\} \subseteq \mathit{Clear}) \wedge \\ (\forall \varrho : y + 2 \leq \varrho < n : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge (\forall \varrho : \varrho \leq y+1 : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge \\ (\text{e-adjacent}((c+1, y+2), (c, y+1))) \wedge ((t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + n + 1) \operatorname{div} \mathit{cycle} + 1)\end{array}\right\}$$

6 **evader-move**

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle}) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n - y - 1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho < y+1\} \subseteq \mathit{Clear})\end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n - y - 1 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho \leq y : (\varrho, \mathit{col}) \in \mathit{Clear})\end{array}\right\} \triangleright \text{postcondition}$$

K. *Derivation of* GROW-TOP1 *in Figure 28*

We make use of the observation in Appendix A.2.c that Rule (11) can be treated as Rule (10) in which the **else** block consists only of a **no-op**; we use this to simplify the derivation by expanding the algorithm's **if** construct to include a **no-op**-only **else** block.

We assume the time is initially t_0.

```
  $\left\{ \begin{array}{l} (\mathit{speed} \geq n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = n-y-1 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho \leq y : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array} \right\}$ ▷ precondition
  $\Rightarrow$
  $\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n-y-1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + y) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}) \end{array} \right\}$
1 move N $y$
  $\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = n-1 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y-1 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + y + 1) \operatorname{div} \mathit{cycle} = t_0 + y \operatorname{div} \mathit{cycle}) \end{array} \right\}$
2 move E 1
  $\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + y + 1) \wedge (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y-1 \leq \varrho < n\} \cup \{(n-1, c+1)\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + 2y + 2) \operatorname{div} \mathit{cycle} = t_0 + y + 1 \operatorname{div} \mathit{cycle}) \end{array} \right\}$
3 move S $y+1$
  $\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + 2y + 2) \wedge (0 \leq \mathit{col} = c+1 < m) \wedge (0 \leq \mathit{row} = n-y-2 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y-1 \leq \varrho < n\} \cup \{(\varrho, c+1) | n-y-2 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ ((t_0 + 2y + 3) \operatorname{div} \mathit{cycle} = t_0 + 2y + 2 \operatorname{div} \mathit{cycle}) \end{array} \right\}$
4 move W 1
  $\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (c < m-1) \wedge (y < \lfloor \frac{n}{2} \rfloor) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + 2y + 3) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n-y-2 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \leq y\} \cup \{(\varrho, c) | n-y-2 \leq \varrho < n\} \cup \{(\varrho, c+1) | n-y-2 \leq \varrho < n\} \subseteq \mathit{Clear}) \end{array} \right\}$ ▷ $(\mathcal{A})$
  ▷ need a more elegant mechanism there, for referencing later
5 if $n > 2y+3$
6     then
  ▷ $(y < \lfloor \frac{n}{2} \rfloor) \wedge (n > 2y+3) \Rightarrow \frac{n}{2} > y + \frac{3}{2} \Rightarrow \lfloor \frac{n}{2} \rfloor > y + 1$
```

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y < \left\lfloor \frac{n}{2} \right\rfloor - 1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (time = t_0 + 2y + 3) \wedge (0 \le col = c < m) \wedge (0 \le row = n - y - 2 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \le y\} \cup \{(\varrho,\kappa) | n - y - 2 \le \varrho < n, c \le \kappa \le c+1\} \subseteq Clear) \wedge \\ ((t_0 + n) \operatorname{div} cycle = t_0 + 2y + 3 \operatorname{div} cycle) \end{array}\right\} \triangleright (\mathcal{A}) \wedge (n > 2y+3)$$

7 **move** S $n - 2y - 3$

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y < \left\lfloor \frac{n}{2} \right\rfloor - 1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (time = t_0 + n) \wedge (0 \le col = c < m) \wedge (0 \le row = y + 1 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \le y\} \cup \{(\varrho, c) | y + 1 \le \varrho < n\} \cup \{(\varrho, c+1) | n - y - 2 \le \varrho < n\} \subseteq Clear) \end{array}\right\}$$

$\triangleright$ $(0 \le row = y + 1 < n) \equiv (0 \le row = y + 1 < n) \wedge \text{TRUE}$

$\triangleright$ $\text{TRUE} \equiv (\text{FALSE} \Rightarrow row = y)$

$\triangleright$ $(0 \le row = y + 1 < n) \equiv (\text{TRUE} \Rightarrow row = y + 1)$

$\Rightarrow$

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y \le \left\lceil \frac{n}{2} \right\rceil - 1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (t_0 < time < t_0 + n + 1 < t_0 + cycle) \wedge (0 \le col = c < m) \wedge (y < \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y + 1) \wedge \\ (y = \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \le c\} \cup \{(\varrho, c+1) | n - y - 2 \le \varrho < n\} \subseteq Clear) \end{array}\right\} \triangleright (\mathcal{B})$$

else

$\triangleright$ $\left((y < \left\lfloor \frac{n}{2} \right\rfloor) \wedge \neg(n > 2y + 3) \Rightarrow \frac{n}{2} \le y + \frac{3}{2} \Rightarrow \left\lfloor \frac{n}{2} \right\rfloor \le y + 1\right) \Rightarrow y = \left\lfloor \frac{n}{2} \right\rfloor - 1$

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y = \left\lfloor \frac{n}{2} \right\rfloor - 1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (time = t_0 + 2y + 3) \wedge (0 \le col = c < m) \wedge (0 \le row = n - y - 2 < n) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | \varrho \le y\} \cup \{(\varrho,\kappa) | n - y - 2 \le \varrho < n, c \le \kappa \le c+1\} \subseteq Clear) \end{array}\right\} \triangleright (\mathcal{A}) \wedge \neg(n > 2y+3)$$

$\triangleright$ is-even$(n) \Rightarrow (y = \left\lceil \frac{n}{2} \right\rceil - 1) \wedge (n - y - 2 = y)$

$\triangleright$ is-odd$(n) \Rightarrow (y = \left\lceil \frac{n}{2} \right\rceil - 2 < \left\lceil \frac{n}{2} \right\rceil - 1) \wedge (n - y - 2 = y + 1)$

no-op

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y \le \left\lceil \frac{n}{2} \right\rceil - 1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (t_0 < time = t_0 + 2\left\lfloor \frac{n}{2} \right\rfloor + 1 \le t_0 + n + 1 < t_0 + cycle) \wedge \\ (0 \le col = c < m) \wedge (y < \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y + 1) \wedge (y = \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa \le c\} \cup \{(\varrho, c+1) | n - y - 2 \le \varrho < n\} \subseteq Clear) \end{array}\right\} \triangleright (\mathcal{B})$$

8 **end if**

$(\mathcal{B}) \Rightarrow$

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y \le \left\lceil \frac{n}{2} \right\rceil - 1) \wedge (t_0 \bmod cycle = 0) \wedge (t_0 < time < t_0 + cycle) \wedge (0 \le col = c < m) \wedge \\ (y < \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y + 1) \wedge (y = \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y) \wedge ((t_0 + cycle) \operatorname{div} cycle = time \operatorname{div} cycle + 1) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa \le c\} \cup \{(\varrho, c+1) | n - y - 2 \le \varrho < n\} \subseteq Clear) \wedge (\forall \varrho : \varrho < n - y - 2 : \text{e-adjacent}((\varrho, c+1), (\varrho, c))) \wedge \\ (\forall \varrho : n - y - 2 \le \varrho < n : \text{e-adjacent}((\varrho, c+2), (\varrho, c+1))) \wedge (\text{e-adjacent}((c+1, n-y-2), (c, n-y-3))) \end{array}\right.$$

9 **evader-move**

$$\left\{\begin{array}{l} (cycle > n+1) \wedge (c < m-1) \wedge (y \le \left\lceil \frac{n}{2} \right\rceil - 1) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle) \wedge \\ (0 \le col = c < m) \wedge (y < \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y + 1) \wedge (y = \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho, c) | n - y - 1 \le \varrho < n\} \subseteq Clear) \end{array}\right\}$$

$\Rightarrow$

$$\left\{\begin{array}{l} (time \bmod cycle = 0) \wedge (0 \le col = c < m) \wedge (y < \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y + 1) \wedge (y = \left\lceil \frac{n}{2} \right\rceil - 1 \Rightarrow row = y) \wedge \\ (\forall \varrho < n, \kappa < col : (\varrho,\kappa) \in Clear) \wedge (\forall \varrho : n - y - 1 \le \varrho < n : (\varrho, col) \in Clear) \end{array}\right\} \triangleright \text{postcondi}$$

L. *Derivation of* TRANSITION *in Figure 29*

We assume the time is initially t_0.

$$\left\{\begin{array}{l} (\mathit{speed} \geq n+1) \wedge (0 < c < m) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c-1 < m) \wedge \left(0 \leq \mathit{row} = \left\lceil \frac{n}{2} \right\rceil - 1 < n\right) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho : \mathit{row} < \varrho < n : (\varrho, \mathit{col}) \in \mathit{Clear}) \end{array}\right\} \triangleright \text{prec}$$

$$\Rightarrow$$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c-1 < m) \wedge \\ \left(0 \leq \mathit{row} = \left\lceil \frac{n}{2} \right\rceil - 1 < n\right) \wedge \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \left\{(\varrho, c-1) | \left\lceil \frac{n}{2} \right\rceil \leq \varrho < n\right\} \subseteq \mathit{Clear}\right) \wedge \\ \left((t_0 + \left\lceil \frac{n}{2} \right\rceil - 1) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}\right) \end{array}\right\}$$

1 **move** S $\left\lceil \frac{n}{2} \right\rceil - 1$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + \left\lceil \frac{n}{2} \right\rceil - 1\right) \wedge \\ (0 \leq \mathit{col} = c-1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho, c-1) | 0 \leq \varrho < n\} \subseteq \mathit{Clear}) \wedge \\ \left((t_0 + \left\lceil \frac{n}{2} \right\rceil) \operatorname{div} \mathit{cycle} = (t_0 + \left\lceil \frac{n}{2} \right\rceil - 1) \operatorname{div} \mathit{cycle}\right) \end{array}\right\}$$

2 **move** E 1

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + \left\lceil \frac{n}{2} \right\rceil\right) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa \leq c-1\} \cup \{(0, c)\} \subseteq \mathit{Clear}) \wedge \\ \left((t_0 + \left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor) \operatorname{div} \mathit{cycle} = (t_0 + \left\lceil \frac{n}{2} \right\rceil) \operatorname{div} \mathit{cycle}\right) \end{array}\right\}$$

3 **move** N $\left\lfloor \frac{n}{2} \right\rfloor$

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ \left(0 \leq \mathit{row} = \left\lfloor \frac{n}{2} \right\rfloor < n\right) \wedge \left(\{(\varrho,\kappa) | \varrho < n, \kappa \leq c-1\} \cup \left\{(\varrho, c) | \varrho \leq \left\lfloor \frac{n}{2} \right\rfloor\right\} \subseteq \mathit{Clear}\right) \wedge \\ ((t_0 + n + 1) \operatorname{div} \mathit{cycle} = (t_0 + n) \operatorname{div} \mathit{cycle}) \end{array}\right\}$$

4 **move** S 1

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + n + 1) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ \left(0 \leq \mathit{row} = \left\lfloor \frac{n}{2} \right\rfloor - 1 < n\right) \wedge \left(\{(\varrho,\kappa) | \varrho < n, \kappa \leq c-1\} \cup \left\{(\varrho, c) | \varrho \leq \left\lfloor \frac{n}{2} \right\rfloor\right\} \subseteq \mathit{Clear}\right) \wedge \\ \left(\forall \varrho : \left\lfloor \frac{n}{2} \right\rfloor + 1 \leq \varrho < n : \text{e-adjacent}((\varrho, c), (\varrho, c-1))\right) \wedge \left(\forall \varrho : \varrho \leq \left\lfloor \frac{n}{2} \right\rfloor : \text{e-adjacent}((\varrho, c+1), (\varrho, c))\right) \wedge \\ \left(\text{e-adjacent}((c, \left\lfloor \frac{n}{2} \right\rfloor + 1), (c-1, \left\lfloor \frac{n}{2} \right\rfloor))\right) \wedge ((t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle} = \mathit{time} \operatorname{div} \mathit{cycle} + 1) \end{array}\right\}$$

5 **evader-move**

$$\left\{\begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge ((t_0 + \mathit{cycle}) \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + \mathit{cycle}) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ \left(0 \leq \mathit{row} = \left\lfloor \frac{n}{2} \right\rfloor - 1 < n\right) \wedge \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \left\{(\varrho, c-1) | \varrho \leq \left\lfloor \frac{n}{2} \right\rfloor - 1\right\} \cup \left\{(\left\lfloor \frac{n}{2} \right\rfloor - 1, c)\right\} \subseteq \mathit{Clear}\right) \end{array}\right\}$$

$$\Rightarrow$$

$$\left\{\begin{array}{l} (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge \left(0 \leq \mathit{row} = \left\lfloor \frac{n}{2} \right\rfloor - 1 < n\right) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} - 1 : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho < \mathit{row} : (\varrho, \mathit{col} - 1) \in \mathit{Clear}) \end{array}\right\} \triangleright \text{postcondition}$$

M. Derivation of GROW-TOP2 *in Figure 30*

We assume the time is initially t_0.

$$\left\{ \begin{array}{l} (\mathit{speed} \geq n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ \left(0 \leq \mathit{row} = 2\lfloor \frac{n}{2} \rfloor - y - 1 < n\right) \wedge (\forall \varrho < n, \kappa < \mathit{col} - 1 : (\varrho, \kappa) \in \mathit{Clear}) \wedge (\forall \varrho < y : (\varrho, \mathit{col} - 1) \in \mathit{Clear}) \end{array} \right\} \triangleright \text{precondition}$$

$\Rightarrow$

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge \left(0 \leq \mathit{row} = 2\lfloor \frac{n}{2} \rfloor - y - 1 < n\right) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho, c-1) | \varrho < y\} \subseteq \mathit{Clear}) \wedge \\ \left((t_0 + 2\left(y - \lfloor \frac{n}{2} \rfloor\right) + 1) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}\right) \end{array} \right\}$$

1 **move** N $2\left(y - \lfloor \frac{n}{2} \rfloor\right) + 1$

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + 2y - 2\lfloor \frac{n}{2} \rfloor + 1\right) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho, c-1) | \varrho < y\} \cup \left\{(\varrho, c) | 2\lfloor \frac{n}{2} \rfloor - y - 1 \leq \varrho \leq y\right\} \subseteq \mathit{Clear}\right) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y < n) \wedge \left((t_0 + 2y - 2\lfloor \frac{n}{2} \rfloor + 2) \operatorname{div} \mathit{cycle} = (t_0 + 2y - 2\lfloor \frac{n}{2} \rfloor + 1) \operatorname{div} \mathit{cycle}\right) \end{array} \right\}$$

2 **move** W 1

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + 2y - 2\lfloor \frac{n}{2} \rfloor + 2\right) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho, c-1) | \varrho \leq y\} \cup \left\{(\varrho, c) | 2\lfloor \frac{n}{2} \rfloor - y - 1 \leq \varrho \leq y\right\} \subseteq \mathit{Clear}\right) \wedge \\ (0 \leq \mathit{col} = c-1 < m) \wedge (0 \leq \mathit{row} = y < n) \wedge \left((t_0 + y + n - 2\lfloor \frac{n}{2} \rfloor + 1) \operatorname{div} \mathit{cycle} = (t_0 + 2y - 2\lfloor \frac{n}{2} \rfloor + 2) \operatorname{div} \mathit{cycle}\right) \end{array} \right\}$$

3 **move** N $n - y - 1$

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + y + n - 2\lfloor \frac{n}{2} \rfloor + 1\right) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho, c-1) | \varrho \leq n-1\} \cup \left\{(\varrho, c) | 2\lfloor \frac{n}{2} \rfloor - y - 1 \leq \varrho \leq y\right\} \subseteq \mathit{Clear}\right) \wedge \\ (0 \leq \mathit{col} = c-1 < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge \left((t_0 + y + n - 2\lfloor \frac{n}{2} \rfloor + 2) \operatorname{div} \mathit{cycle} = (t_0 + y + n - 2\lfloor \frac{n}{2} \rfloor + 1) \operatorname{div} \mathit{cyc} \end{array} \right.$$

4 **move** E 1

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + y + n - 2\lfloor \frac{n}{2} \rfloor + 2\right) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \left\{(\varrho, c) | 2\lfloor \frac{n}{2} \rfloor - y - 1 \leq \varrho \leq y\right\} \cup \{(n-1, c)\} \subseteq \mathit{Clear}\right) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = n-1 < n) \wedge \left((t_0 + 2n - 2\lfloor \frac{n}{2} \rfloor) \operatorname{div} \mathit{cycle} = (t_0 + y + n - 2\lfloor \frac{n}{2} \rfloor + 2) \operatorname{div} \mathit{cycle}\right) \end{array} \right\}$$

5 **move** S $n - y - 2$

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \left(\mathit{time} = t_0 + 2\lceil \frac{n}{2} \rceil\right) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \left\{(\varrho, c) | 2\lfloor \frac{n}{2} \rfloor - y - 1 \leq \varrho \leq y\right\} \cup \{(\varrho, c) | y+1 \leq \varrho < n\} \subseteq \mathit{Clear}\right) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y+1 < n) \wedge \left((t_0 + \mathit{cycle}) \operatorname{div} \mathit{cycle} = (t_0 + 2\lceil \frac{n}{2} \rceil) \operatorname{div} \mathit{cycle} + 1\right) \wedge \\ \left(\forall \varrho : 0 \leq \varrho < 2\lfloor \frac{n}{2} \rfloor - y - 1 : \text{e-adjacent}((\varrho, c), (\varrho, c-1))\right) \wedge \\ \left(\forall \varrho : 2\lfloor \frac{n}{2} \rfloor - y - 1 \leq \varrho < n : \text{e-adjacent}((\varrho, c+1), (\varrho, c))\right) \wedge \left(\text{e-adjacent}((c, 2\lfloor \frac{n}{2} \rfloor - y - 2), (c-1, 2\lfloor \frac{n}{2} \rfloor - y - 1))\right) \end{array} \right\}$$

6 **evader-move**

$$\left\{ \begin{array}{l} (\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\lfloor \frac{n}{2} \rfloor \leq y < n-1) \wedge ((t_0 + \mathit{cycle} \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + \mathit{cycle}) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y+1 < n) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \left\{(\varrho, c-1) | 2\lfloor \frac{n}{2} \rfloor - y \leq \varrho < n\right\} \cup \{(y+1, c)\} \subseteq \mathit{Clear}\right) \end{array} \right\}$$

$\Rightarrow$

$$\left\{ \begin{array}{l} (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = y+1 < n) \wedge \\ (\forall \varrho < n, \kappa < \mathit{col} - 1 : (\varrho, \kappa) \in \mathit{Clear}) \wedge \left(\forall \varrho : 2\lfloor \frac{n}{2} \rfloor - y \leq \varrho < n : (\varrho, \mathit{col} - 1) \in \mathit{Clear}\right) \end{array} \right\} \triangleright \text{postcondition}$$

N. Derivation of GROW-BOTTOM2 *in Figure 31*

We make use of the observation in Appendix A.2.c that Rule (11) can be treated as Rule (10) in which the **else** block consists only of a **no-op**; we use this to simplify the derivation by expanding the algorithm's **if** construct to include a **no-op**-only **else** block.

We assume the time is initially t_0.

$$\left\{\begin{array}{l}(\mathit{speed} \geq n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (\mathit{time} \bmod \mathit{cycle} = 0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = y+1 < n) \wedge (\forall \varrho < n, \kappa < \mathit{col} - 1 : (\varrho, \kappa) \in \mathit{Clear}) \wedge \\ \big(\forall \varrho : 2\left\lfloor \frac{n}{2} \right\rfloor - y \leq \varrho < n : (\varrho, \mathit{col} - 1) \in \mathit{Clear}\big)\end{array}\right\} \rhd \text{precondition}$$

$\Rightarrow$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0) \wedge (0 \leq \mathit{col} = c < m) \wedge \\ (0 \leq \mathit{row} = y+1 < n) \wedge \big(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \big\{(\varrho, c-1) | 2\left\lfloor \frac{n}{2} \right\rfloor - y \leq \varrho < n\big\} \subseteq \mathit{Clear}\big) \wedge \\ \big((t_0 + 2\big(y - \left\lfloor \frac{n}{2} \right\rfloor + 1\big)\big) \operatorname{div} \mathit{cycle} = t_0 \operatorname{div} \mathit{cycle}\big)\end{array}\right\}$$

1 **move** S $2\big(y - \left\lfloor \frac{n}{2} \right\rfloor + 1\big)$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \big(\mathit{time} = t_0 + 2y - 2\left\lfloor \frac{n}{2} \right\rfloor + 2\big) \wedge \\ \big(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \big\{(\varrho, c-1) | 2\left\lfloor \frac{n}{2} \right\rfloor - y \leq \varrho < n\big\} \cup \big\{(\varrho, c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\big\} \subseteq \mathit{Clear}\big) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge \big(0 \leq \mathit{row} = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 < n\big) \wedge \big((t_0 + 2y - 2\left\lfloor \frac{n}{2} \right\rfloor + 3) \operatorname{div} \mathit{cycle} = (t_0 + 2y - 2\left\lfloor \frac{n}{2} \right\rfloor + 2) \operatorname{div} \mathit{cycle}\end{array}\right.$$

2 **move** W 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \big(\mathit{time} = t_0 + 2y - 2\left\lfloor \frac{n}{2} \right\rfloor + 3\big) \wedge \\ \big(\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \big\{(\varrho, c-1) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho < n\big\} \cup \big\{(\varrho, c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\big\} \subseteq \mathit{Clear}\big) \wedge \\ (0 \leq \mathit{col} = c - 1 < m) \wedge \big(0 \leq \mathit{row} = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 < n\big) \wedge \big((t_0 + y + 2) \operatorname{div} \mathit{cycle} = (t_0 + 2y - 2\left\lfloor \frac{n}{2} \right\rfloor + 3) \operatorname{div} \mathit{cycle}\big)\end{array}\right\}$$

3 **move** S $2\left\lfloor \frac{n}{2} \right\rfloor - y - 1$

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y + 2) \wedge \\ \big(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \big\{(\varrho, c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\big\} \subseteq \mathit{Clear}\big) \wedge \\ (0 \leq \mathit{col} = c - 1 < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge ((t_0 + y + 3) \operatorname{div} \mathit{cycle} = (t_0 + y + 2) \operatorname{div} \mathit{cycle})\end{array}\right\}$$

4 **move** E 1

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge (\mathit{time} = t_0 + y + 3) \wedge \\ (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \big(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(0, c)\} \cup \big\{(\varrho, c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\big\} \subseteq \mathit{Clear}\end{array}\right.$$

5 **if** $y < 2(\left\lfloor \frac{n}{2} \right\rfloor - 1)$

6 **then**

$$\left\{\begin{array}{l}(\mathit{cycle} > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < 2\left\lfloor \frac{n}{2} \right\rfloor - 2) \wedge (t_0 \bmod \mathit{cycle} = 0) \wedge \\ (\mathit{time} = t_0 + y + 3) \wedge (0 \leq \mathit{col} = c < m) \wedge (0 \leq \mathit{row} = 0 < n) \wedge \\ \big(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(0, c)\} \cup \big\{(\varrho, c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\big\} \subseteq \mathit{Clear}\big) \wedge \\ \big((t_0 + 2\left\lfloor \frac{n}{2} \right\rfloor + 1) \operatorname{div} \mathit{cycle} = (t_0 + y + 3) \operatorname{div} \mathit{cycle}\big)\end{array}\right\} \rhd (\mathcal{D}) \wedge (y < 2(\left\lfloor \frac{n}{2} \right\rfloor - 1))$$

```
7           **move** $\textsc{n}\ 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2$
    $\left\{\begin{array}{l} (cycle > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < 2\left\lfloor \frac{n}{2} \right\rfloor - 2) \wedge (t_0 \bmod cycle = 0) \wedge \\ (t_0 < t_0 + n \leq time \leq t_0 + n + 1 < t_0 + cycle) \wedge (0 \leq col = c < m) \wedge \left(0 \leq row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2 < n\right) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \left\{(\varrho,c) | 0 \leq \varrho \leq 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2\right\} \cup \left\{(\varrho,c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\right\} \subseteq Clear\right) \end{array}\right\}$
    $\triangleright$ is-even$(n) \wedge (y = n-2) \Rightarrow 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2 = 0$
    $\triangleright$ is-odd$(n) \wedge (y < 2(\left\lfloor \frac{n}{2} \right\rfloor - 1) \Rightarrow y < n - 3 < n - 2$
    $\Rightarrow$
    $\left\{\begin{array}{l} (cycle > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (t_0 < t_0 + n \leq time \leq t_0 + n + 1 < t_0 + cycle) \wedge (0 \leq col = c < m) \wedge (y < n-2 \Rightarrow row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2) \wedge \\ (y = n-2 \Rightarrow row = 0) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho,c) | 0 \leq \varrho \leq y+1\} \subseteq Clear) \end{array}\right\} \triangleright (\mathcal{E})$
        **else**
    $\left\{\begin{array}{l} (cycle > n+1) \wedge (0 < c < m) \wedge (2\left\lfloor \frac{n}{2} \right\rfloor - 2 \leq y < n-1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (time = t_0 + y + 3) \wedge (0 \leq col = c < m) \wedge (0 \leq row = 0 < n) \wedge \\ \left(\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(0,c)\} \cup \left\{(\varrho,c) | 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \leq \varrho \leq y+1\right\} \subseteq Clear\right) \end{array}\right\} \triangleright (\mathcal{D}) \wedge \neg(y < 2(\left\lfloor \frac{n}{2} \right\rfloor - 1))$
    $\triangleright$ is-even$(n) \Rightarrow 2\left\lfloor \frac{n}{2} \right\rfloor - 2 = n - 2 \Rightarrow y = n - 2 \Rightarrow 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 = 1$
    $\triangleright$ is-odd$(n) \Rightarrow 2\left\lfloor \frac{n}{2} \right\rfloor - 2 = n - 3 \Rightarrow n - 3 \leq y \leq n - 2 \Rightarrow 2\left\lfloor \frac{n}{2} \right\rfloor - y - 1 \in \{0,1\}$
    $\triangleright$ is-odd$(n) \wedge y = n - 3 \Rightarrow 0 = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2$
            **no-op**
    $\left\{\begin{array}{l} (cycle > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod cycle = 0) \wedge \\ (t_0 < t_0 + n \leq time \leq t_0 + n + 1 < t_0 + cycle) \wedge (0 \leq col = c < m) \wedge (y < n-2 \Rightarrow row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2) \wedge \\ (y = n-2 \Rightarrow row = 0) \wedge (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho,c) | 0 \leq \varrho \leq y+1\} \subseteq Clear) \end{array}\right\} \triangleright (\mathcal{E})$
8       **end if**
    $(\mathcal{E}) \Rightarrow$
    $\left\{\begin{array}{l} (cycle > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge (t_0 \bmod cycle = 0) \wedge (t_0 < time < t_0 + cycle) \wedge (0 \leq col = c < m \\ (y < n-2 \Rightarrow row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2) \wedge (y = n-2 \Rightarrow row = 0) \wedge ((t_0 + cycle) \operatorname{div} cycle = time \operatorname{div} cycle + 1) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c\} \cup \{(\varrho,c) | 0 \leq \varrho \leq y+1\} \subseteq Clear) \wedge (\forall \varrho : y+1 < \varrho < n : \text{e-adjacent}((\varrho,c),(\varrho,c-1))) \wedge \\ (\forall \varrho : 0 \leq \varrho \leq y+1 : \text{e-adjacent}((\varrho,c+1),(\varrho,c))) \wedge (\text{e-adjacent}((c,y+2),(c-1,y+1))) \end{array}\right.$
9   **evader-move**
    $\left\{\begin{array}{l} (cycle > n+1) \wedge (0 < c < m) \wedge (\left\lfloor \frac{n}{2} \right\rfloor \leq y < n-1) \wedge ((t_0 + cycle) \bmod cycle = 0) \wedge (time = t_0 + cycle) \wedge \\ (0 \leq col = c < m) \wedge (y < n-2 \Rightarrow row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2) \wedge (y = n-2 \Rightarrow row = 0) \wedge \\ (\{(\varrho,\kappa) | \varrho < n, \kappa < c-1\} \cup \{(\varrho,c) | 0 \leq \varrho < y+1\} \subseteq Clear) \end{array}\right\}$
    $\Rightarrow$
    $\left\{\begin{array}{l} (time) \bmod cycle = 0) \wedge (0 \leq col = c < m) \wedge (y < n-2 \Rightarrow row = 2\left\lfloor \frac{n}{2} \right\rfloor - y - 2) \wedge \\ (y = n-2 \Rightarrow row = 0) \wedge (\forall \varrho < n, \kappa < col - 1 : (\varrho,\kappa) \in Clear) \wedge \\ (\forall \varrho \leq y : (\varrho, col - 1) \in Clear) \wedge (y = n-2 \Rightarrow \forall \varrho < n, \kappa < col : (\varrho,\kappa) \in Clear) \end{array}\right\} \triangleright$ postcondition
```