Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 12-005

Lion and Man with Visibility in Monotone Polygons

Narges Noori and Volkan Isler

First submitted: February 24, 2012

Revised: June 27, 2013

Lion and Man with Visibility in Monotone Polygons

Narges Noori and Volkan Isler *

June 27, 2013

Abstract

In the original version of the lion and man game, a lion tries to capture a man who is trying to escape in a circular arena. The players have equal speeds. They can observe each other at all times. We study a new variant of the game in which the lion has only line-of-sight visibility. That is it can observe the man's position only if the line segment connecting them does not intersect the boundary. We show that despite this limitation, the lion can capture the man in any monotone polygon in finite time.

1 Introduction

In recent years, solving pursuit-evasion games in complex environments has been receiving increasing attention. Such games model robotics applications such as surveillance and search-and-rescue. An overview of recent results on pursuit-evasion in robotics can be found in [3].

A fundamental pursuit-evasion game with immediate applications in robotics is the lion and man game. In this game, a lion (the pursuer) tries to capture a man (the evader) by moving onto the man's location (or getting close to it). There are some known cases where the pursuer wins the game if it can see the man at all times. For example, in the original setting, which takes place in a circular arena, the pursuer has a winning strategy [10, 11] when the players take turns in moving. In the continuous time setting, i.e. when the players move simultaneously, the pursuer can get arbitrarily close to the evader [1]. In the turn-based game, a single pursuer with global visibility can capture the man in any simply-connected polygon [8] and three lions suffice in arbitrary polygons with holes [2, 9].

In this paper, we focus on a variant of the turn-based lion-and-man game in which the pursuer has only line of sight vision. That is, the pursuer can see the evader only if the line segment connecting them is free of obstacles. We study the game when the goal of the pursuer is to capture the evader. This variant models robotics applications where the pursuer is a robot equipped with a camera or a laser scanner. With the limited vision power, the pursuer has to first find the evader when it disappears and then move toward the evader to capture it. When the goal of the pursuer is to just find the evader the problem is called the visibility-based pursuit evasion problem [6]. The necessary and sufficient conditions for a simple polygon to be searchable by a pursuer with various degrees of visibility power is presented in [12]. Guibas et al. show that for an arbitrarily fast evader, the minimum number of pursuers required in the worst case is $\Theta(\log n)$ for simply-connected polygons and $\Theta(\sqrt{m} + \log n)$ for a polygon with m holes [6].

A simple polygon is called monotone with respect to a line l if for any line l' perpendicular to l the intersection of the polygon with l' is connected [4]. In this work, without loss of generality we consider x-monotone polygons. For the lion and man game with visibility in monotone polygons, we present a pursuit strategy which successfully combines search and capture and guarantees that the evader will be captured after a finite number of steps.

In monotone polygons, merely searching for the evader is straightforward: the evader can be found for example by moving along the shortest path that connects the leftmost vertex to the rightmost vertex. This is because every point inside the polygon is visible from a location on this path and the evader cannot move into a "cleared" region without revealing itself.

*The authors are with the Department of Computer Science & Engineering, University of Minnesota. Emails:{noori, isler}@cs.umn.edu. This work was supported in part by NSF Awards #0916209, and #0917676.

Similarly, the capture strategy is simple when the pursuer knows the location of the evader at all times: it can capture the evader by starting from the leftmost vertex O_L and performing *the lion's move* that is to move toward the evader along the shortest path that connects O_L to the evader's current location. The distance of the pursuer from O_L, defined as the length of the shortest path from O_L to the pursuer's location, provides a natural notion of "progress" which is monotonically increasing in this full visibility setting[1].

What is not obvious is whether such progress can be maintained when the pursuer has to search for the evader when it disappears. If the pursuer ends up retreating after a search, the evader might have a strategy in which the pursuer oscillates between search and progress and the game can last forever. We show that this cannot happen. In particular, we show that the pursuer can successfully combine search and making progress toward capture in monotone polygons. Further, we show that search without risking progress can be achieved with a deterministic strategy. We are not aware of any other results which combine these two objectives for a single pursuer while providing guarantees about the outcome of the game. The randomized strategy proposed for the general simply connected polygons [8], where the pursuer guesses the hiding vertex of the evader, provides exponential capture time[2]. In this work, however, we present a deterministic pursuit strategy which reduces the capture time to a quantity that is polynomial in the number of vertices and the diameter of the polygon. An interesting feature of our strategy is that the lion's distance from O_L is not increasing monotonically. Nevertheless, we show that the pursuer can push the evader further to the right after a finite number of steps. To do this, we introduce a new measure of progress for the pursuer which is more sophisticated than its distance from O_L.

Our results provide a step toward understanding the lion and man game with visibility constraints in general polygons. An important question is to find the class of polygons in which a single pursuer suffices to capture the evader[3]. We show that monotone polygons are included in this class.

In Section 2 we present the definitions used throughout the text. Section 3 provides an overview of the pursuit strategy. In this section, we also present the tools used to show that the strategy guarantees capture. The details of the pursuit strategy is presented in Section 4 and Section 5. For each part of the strategy, we also present the complete algorithm in the form of pseudocode. In particular, we provide the input configuration which describes the conditions that must be satisfied before the pursuer starts the corresponding sub-strategy as well as the exit configuration that the pursuer guarantees as it switches to the next sub-strategy. We present the analysis of the capture time in Section 6. The concluding remarks are discussed in Section 7. For clear presentation of the main points of the strategy we present the detailed proof of some of the technical lemmas as well as the properties of monotone polygons in the Appendix (Section 9).

2 Preliminaries

We now formally define the game. We refer to the pursuer's and the evader's location at time-step t as $\mathcal{P}(t)$ and $\mathcal{E}(t)$ respectively. When the time is obvious from the context, we use $\mathcal{P}$ and $\mathcal{E}$. Our *Game Model* is as follows: (1) The players move alternately in turns. (2) Each turn takes a unit time step. (3) In each turn the players can move along a line segment of length at most one to a point visible to themselves. (4) The evader has global visibility i.e. it knows the location of $\mathcal{P}$ at all time steps. However, the pursuer sees the evader only if the line segment joining the two is not blocked by the boundary of the polygon. Note that since $\mathcal{P}$ has a deterministic strategy, the evader can simulate the pursuer's moves and hence it knows the location of $\mathcal{P}$ at all time steps. (5) The pursuer captures $\mathcal{E}$ if at any time, the distance between them is less than or equal to one (the step-size) while $\mathcal{P}$ can see $\mathcal{E}$, see Fig. 1(c).

Without loss of generality, we assume that the game takes place in an x-monotone polygon Q. Recall that for an x-monotone polygon, the intersection of all vertical lines with the polygon is a connected segment. The leftmost vertex and the rightmost vertex are denoted by O_L and O_R respectively. The boundary of the polygon connects these vertices by two x-monotone chains denoted by $Chain_L$ and $Chain_U$, see Fig. 1(a).

[1]This argument works in any simply connected polygon [8].

[2]It is an open problem whether this bound is tight or not.

[3]It should also be noted that capture using only deterministic strategies remains an open question.

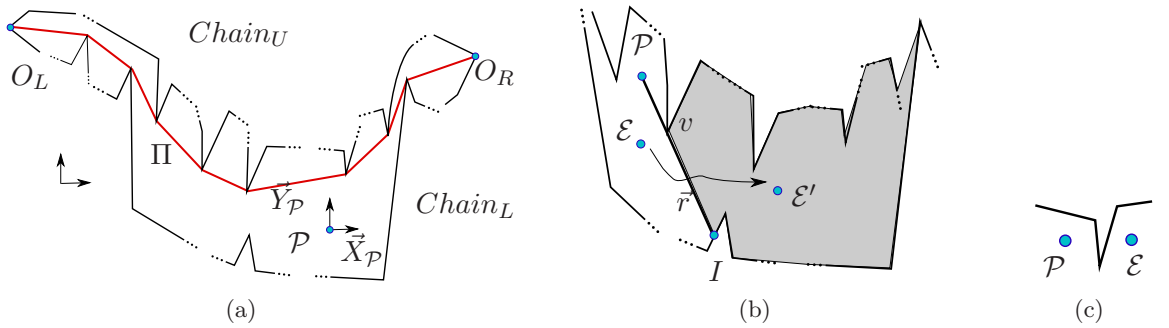

Figure 1: (a) An x-monotone polygon. (b) The pocket $pocket(v, \vec{r})$ is the shaded sub-polygon. (c) Since $\mathcal{E}$ is not visible, capture condition is not satisfied.

We refer to the segment between points u and v as uv. Whenever direction is also important we refer to the ray pointing from u to v as $\vec{uv}$. We define a local reference frame whose origin coincides with $\mathcal{P}$. Its axes $\vec{X}_{\mathcal{P}}$ and $\vec{Y}_{\mathcal{P}}$ are parallel to the axes of the reference frame, see Fig. 1(a). We refer to the boundary of Q as ∂Q, and the number of vertices in Q as n. The shortest path between the two points u and v is denoted by $\pi(u, v)$, and the length of $\pi(u, v)$ is denoted by $d(u, v)$. The diameter of the polygon is $D = max_{\ u,v \in Q}\ d(u, v)$. The shortest path tree rooted at the point o in Q is defined as $\cup_{v \in V} \pi(o, v)$ where V denotes vertices of Q. For a point p inside the polygon, the parent of p, denoted by $parent(p)$, is the first vertex on the shortest path $\pi(o, p)$ from p to o. In the rest of the paper, we consider the shortest path tree rooted at $o = O_L$. We denote $\pi(O_L, O_R)$ by Π (Fig. 1(a)). For simplicity we denote $d(O_L, p)$ by $R(p)$ for a point $p \in Q$.

Throughout the paper we refer to the x coordinate of a point p as $x(p)$. Similarly, the y coordinate is denoted by $y(p)$.

Suppose that it is the evader's turn to move. See Fig. 1(b). Imagine that the pursuer and the evader can see each other before the evader's move but the evader disappears behind a vertex v after moving to $\mathcal{E}'$. Let $\vec{r}$ be a ray originating from $\mathcal{P}$ and passing through v. Let I be the intersection of this ray with the polygon. The sub-polygon which contains $\mathcal{E}$ and is bounded by the ray $\vec{r}$ plus the boundary of the polygon from v to I is called a *pocket* [8]. The ray $\vec{r}$ is called the entrance of the pocket and the pocket is referred to as $pocket(v, \vec{r})$.

Remark 1. In the rest of the paper, we refer to the pocket that the evader is hidden inside of as $pocket(v, \vec{r})$ (also the *contaminated region*) where $\vec{r} = \vec{pv}$ and p is the location of the pursuer at the time that the evader has disappeared behind the vertex v. See Fig. 1(b).

Lion's Move Progress: In our pursuit strategy, we use the *lion's move* strategy [10,11] proposed for capturing $\mathcal{E}$ in circular arena. This strategy is performed with respect to a center C. Initially, the pursuer has to lie on the line segment which connects C to $\mathcal{E}$ in order to be able to perform the lion's move strategy. In each turn, the pursuer moves so that it remains on the line segment which connects C and $\mathcal{E}$. Therefore, the invariant that the lion maintains is that it stays in between C and $\mathcal{E}$. The lion's progress is that after each time step $|L'C|^2 \geq 1 + |LC|^2$ where L and L' are the previous and the current position of the lion respectively. This invariant and notion of progress guarantee that the evader is squeezed between the boundary and the lion in a smaller and smaller region and finally is captured.

Extended Lion's Move Progress: In [8] the lion's move is generalized to the *extended lion's move* for capturing the evader in simply-connected polygons (when the pursuer can always see the evader). Suppose that $\mathcal{E}$ is visible to $\mathcal{P}$ and $\mathcal{P}$ is on the edge which connects $parent(\mathcal{E})$ and $\mathcal{E}$ in the shortest path tree rooted at O_L. Then the pursuer can follow $\mathcal{E}$ by lion's move with respect to the center $C = parent(\mathcal{E})$. Note that as the evader moves, its parent in the shortest path tree also changes. The invariant throughout the game is that the lion stays on the shortest path from O_L to $\mathcal{E}$. The lion's progress is that after each step, its squared distance from O_L increases at least by $\frac{1}{n}$ i.e. $d(O_L, \mathcal{P}(t+1))^2 \geq d(O_L, \mathcal{P}(t))^2 + \frac{1}{n}$. Thus, the evader is squeezed between $\mathcal{P}$ and the polygon's boundary, and it will be captured after $O(nD^2)$ steps where n is the number of vertices in the polygon and D is the diameter of the polygon.

3 Monotone Polygon Capture Strategy

In this section we start with a high level description of the pursuer's strategy. Then we present the definitions used in the strategy and explain the pursuer's notion of pursuer. Finally, we demonstrate the strategy in an example. The details of the strategy is provided in the following sections.

3.1 Overview

We start with an example which demonstrates the difficulty of designing capture strategy for a pursuer that has limited vision power. Fig. 2 provides some intuition. In this example, we also show the importance of pursuer's notion of progress which is required to prove that the proposed pursuer strategy will guarantee capture in finite time. Suppose that the pursuer's notion of progress is its x coordinate and moreover let $x(\mathcal{P}) \leq x(\mathcal{E})$ be the invariant that the pursuer tries to maintain. Therefore, if the pursuer's strategy guarantees that the x coordinate of $\mathcal{P}$ is increased after finite time, the evader will be captured in finite time. Now, suppose that $\mathcal{P}$ is following $\mathcal{E}$ by lion's move with respect to O_L and $\mathcal{E}$ disappears behind v in the shaded pocket. Hence $\mathcal{P}$ has to search for $\mathcal{E}$. A careless search strategy may result in losing the progress that $\mathcal{P}$ has made so far as follows. If the pursuer first visits v_2, then the evader hidden at v_1 will escape to v and re-enter the previously "cleared" region i.e. the set of points p such that $x(p) \leq x(\mathcal{P})$ defined by pursuer's notion of progress. Likewise, if the pursuer first visits v_1, the evader hidden at v_2 can re-contaminate the cleared region. Consequently, the evader can hide in the same portion of the polygon infinitely many times and hence avoid capture (against this naive pursuit strategy).

We will present a pursuit strategy called *Monotone Polygon Capture (MPC)* strategy which guarantees capture. In this strategy, we partition the monotone polygon into sub-polygons called the *critical sub-polygons*. The pursuer clears these sub-polygons from the left to the right. That is $\mathcal{P}$ ensures that the evader cannot re-contaminate the cleared portion and hence it will be captured.

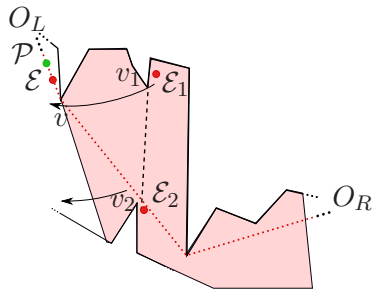

Figure 2: A difficult situation for $\mathcal{P}$: $\mathcal{E}$ can re-contaminate the cleared region depending on how $\mathcal{P}$ enters the pocket.

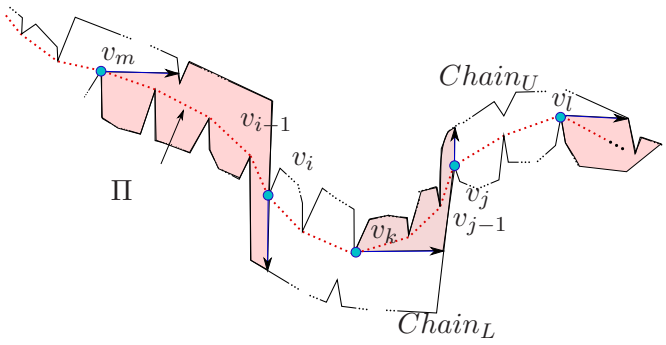

Figure 3: The path Π is shown in dots. The critical sub-polygon defined by v_m and v_i is type 4, v_i and v_k is type 1, v_k and v_j is type 2, and v_j and v_l is type 3.

The state diagram of the MPC strategy is given in Fig. 4, and a high level description is presented in Algorithm 1. The strategy consists of three states: *Search, Guard* and *Extended Lion's Move* denoted by S, G, and L respectively. Initially, $\mathcal{P}$ is at O_L. It starts by the S state if $\mathcal{E}$ is invisible, and the L state otherwise. Whenever $\mathcal{E}$ disappears, $\mathcal{P}$ switches to the search state to find it.

When $\mathcal{E}$ is found as a result of the S strategy, the pursuer performs the *guard* strategy in order to establish the extended lion's move with respect to O_L. Recall that $\mathcal{P}$ has to be on $\pi(O_L, \mathcal{E})$ in order to be able to perform the extended lion's move. Therefore in the guard state $\mathcal{P}$ tries to catch up with $\pi(O_L, \mathcal{E})$ since $\mathcal{P}$ might not be on $\pi(O_L, \mathcal{E})$ at the time that $\mathcal{E}$ is found. After $\mathcal{P}$ catches up with $\pi(O_L, \mathcal{E})$, the pursuer follows it by extended lion's move. During the lion's move state or the guard state, $\mathcal{E}$ might disappear. At this time $\mathcal{P}$ switches to the search strategy. The sequence of state transitions is $((SG)^*L)^*$ and the loops $(SG)^*$ and $L(SG)^*L$ are possible. In the following, we refer to the S state and its following G state as the *combined Search/Guard* state i.e. (SG).

Algorithm 1: Monotone Polygon Capture Strategy

```
repeat
    if E is invisible then
        do Search strategy until E is found;
    else if E is visible and P is not on π(O_L, E) then
        do Guard strategy until P is on π(O_L, E), or E disappears;
    else if E is visible and P is on π(O_L, E) then
        do Extended Lion's Move strategy until E is captured, or E disappears;
    end
until E is captured ;
```

To prove that our proposed strategy guarantees capture, it is necessary to show that these loops terminate after finite time. To do so, we define a *reference vertex*, denoted by p_{ref}, which is a vertex of the polygon. We show that $\mathcal{P}$ maintains the invariant that $\mathcal{E}$ is to the "right" of p_{ref} at the beginning of a combined search/guard state (SG) or an L state. The pursuer makes progress by updating p_{ref} to the "right" after finite number of time steps. Consequently, the evader is confined in a smaller region and hence it will be captured.

We define "right" of a vertex v as the half-plane to the right, below or above v based on the structure of the monotone polygon. We denote the half-plane associated with the vertex v by $h(v)$. Fig. 5 illustrates examples of these half-planes. Then, the invariant is that the evader is forced to remain inside $h(p_{\text{ref}})$ at the beginning of a combined search/guard state (SG) or an L state (otherwise it will be captured), and the progress is to update p_{ref} to a new point p'_{ref} such that $p'_{\text{ref}} \in h(p_{\text{ref}})$.

It is worth emphasizing that the aforementioned invariant is guaranteed only at the beginning of a (SG) state or an L state. During the guard state, the evader can exit $h(p_{\text{ref}})$ and re-contaminate the region before p_{ref}. At this time, the pursuer switches to the *Vertical Guard* or the *Horizontal Guard* sub-state in order to push $\mathcal{E}$ back to $h(p_{\text{ref}})$ and recover its progress. See the state diagram in Fig. 4.

For ease of reference, we now list the terminology and variables that are crucial in this paper. We will present the formal definition of these variables in the following sections.

- The reference vertex p_{ref} which is used to track the progress. The evader is guaranteed to be to the right of p_{ref}.
- Half-planes associated to a vertex v denoted by $h(v)$ which denotes right of a vertex v used to track progress.
- The auxiliary vertex denoted by p_{aux}, which we introduce in Section 5.2, is a local variable used to track the progress.

Let us next present the details of the invariants and the pursuer's notion of progress.

3.2 Definitions, Invariants and the Notion of Progress

The critical sub-polygons that partition the monotone polygon are defined as follows.

Definition 3.1 (Critical Sub-polygons)**.** *Let Π be the shortest path from O_L to O_R and denote the vertices on Π by $\{v_0, ..., v_i, ...\}$. Let s_{i-1} be the slope of the edge $v_{i-1}v_i$ and $\delta s_i = s_i - s_{i-1}$. Then, the* critical vertices *are those vertices on Π on which either s or δs changes sign. For example, in Fig. 3, the vertices v_i, v_k and v_j are the critical vertices. For a critical vertex $p = v_i$, we assign $\vec{Y}_p$ if in $p = v_i$, the values s_i and δs_i have different signs and $\vec{X}_p$ if they have the same signs. Then, each two consecutive critical vertices, say v_i and v_k, define a* critical sub-polygon *given by the sub-polygon formed by ∂Q and the rays assigned to v_i and v_k. See Fig. 3. Depending on the sign of s and δs in the critical sub-polygons, we get four types of critical sub-polygons: Type (1) when $s < 0$ and $0 < \delta s$, Type (2) when $0 < s$ and $0 < \delta s$, Type (3) when $0 < s$ and $\delta s < 0$, Type (4) when $s < 0$ and $\delta s < 0$.*

The critical sub-polygons allow us to enumerate all possible configurations between $\mathcal{P}$, $\mathcal{E}$ and the structure of P since our proposed strategy is symmetric in different types of these critical sub-polygons.

Remark 2. Throughout the paper, we present the strategy for the case that v, the vertex that defines the hiding pocket $pocket(v, \vec{r})$, is inside the 1^{st} type critical sub-polygons. The strategies for the other types are symmetric which we specify them in the form of Remarks.

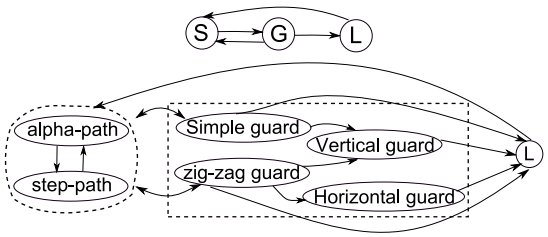

Figure 4: The state diagram for the MPC strategy. The sub-states in S and G are shown at the bottom.

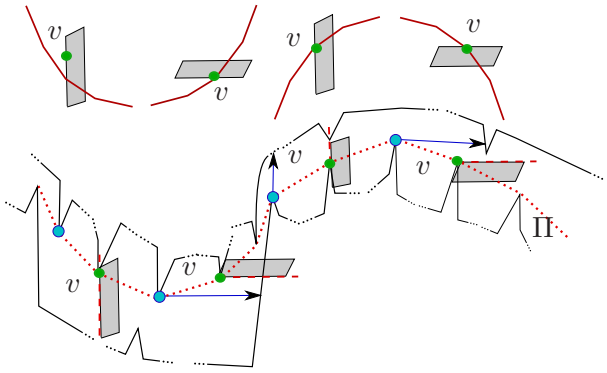

Figure 5: Bottom) The dot path is Π. The half-planes associated to a vertex v. Top) From left to right are types 1 to 4. The path Π and $h(v)$ are shown.

Without loss of generality we assume that at the beginning of the search state the pursuer is at v. This is possible because, after $\mathcal{E}$ disappears, the pursuer moves toward v along the straight line pv until it reaches v. If in the meantime $\mathcal{E}$ appears, the pursuer resumes the last state's strategy, i.e. the G state or the L state. Note that all preconditions of this state are still satisfied.

We now present an important property of Q, and then explain how the pursuer makes progress. Let us first define the *half-plane* of a vertex.

Definition 3.2 (The half-plane of a vertex)**.** *For a vertex $v \in Q$, the* open half-plane*, denoted by $h(v)$, is defined as the set of points to the right of v if v is inside the 1^{st} or the 3^{rd} type critical sub-polygons. For the 2^{nd} type, $h(v)$ is the half-plane above v i.e. the points p with $y(v) < y(p)$. Finally for the 4^{th} type, $h(v)$ is the half-plane below v i.e. the points p with $y(v) > y(p)$. The corresponding closed half-planes are denoted by $h[v]$. See Fig. 5.*

The following property of monotone polygons is crucial in this paper which we prove in Section 9.2.

Property 3.3. *Consider the critical sub-polygons defined in Definition 3.1. Let v be any vertex in Q and refer to Fig. 5.*

- *Suppose that v is inside a critical sub-polygon of the 1^{st} type, $v \in Chain_U$, and the slope of the edge between $parent(v)$ and v is negative. Then for all points $p \in h(v)$, we have $R(v) < R(p)$.*
- *Suppose that v is inside a critical sub-polygon of the 3^{rd} type, $v \in Chain_L$, and the slope of the edge between $parent(v)$ and v is positive. Then for all points $p \in h(v)$, we have $R(v) < R(p)$.*
- *Suppose that v is inside a critical sub-polygon of the 2^{nd} type, $v \in Chain_U$, and the slope of the edge between $parent(v)$ and v is positive. Then for all points $p \in h(v)$, we have $R(v) < R(p)$.*
- *Suppose that v is inside a critical sub-polygon of the 4^{th} type, $v \in Chain_L$, and the slope of the edge between $parent(v)$ and v is negative. Then for all points $p \in h(v)$, we have $R(v) < R(p)$.*

The reference vertex p_{ref}: The invariant that the pursuer maintains and its notion of progress are defined based on the vertex p_{ref}. Let v be the vertex that defines the hiding pocket $pocket(v, \vec{r})$. Initially $p_{\text{ref}} = O_L$. The vertex p_{ref} is defined such that $v \in h[p_{\text{ref}}]$ and $p_{\text{ref}} \in Chain_U$. The pursuer updates p_{ref} at the beginning of an S state which is after a G state (Section 5). Specifically, when v belongs to $Chain_U$ we have $p_{\text{ref}} = v$. In case that $v \in Chain_L$ the vertex p_{ref} is set to another vertex from the upper chain as explained in Section 5.1 and Section 5.2.

Remark 3. When $\mathcal{P}$ is sweeping the 3^{rd} or the 4^{th} type critical sub-polygons $p_{\text{ref}} \in Chain_L$. Also in the 2^{nd} type critical sub-polygon $p_{\text{ref}} \in Chain_U$.

Next we present the invariants that $\mathcal{P}$ maintains during the game. At the beginning of a combined search/guard state (SG) or an L state we have:

- **Invariant (I1)** $\mathcal{E} \in h(p_{\text{ref}})$ and $\mathcal{P} \in h[p_{\text{ref}}]$. Consequently $R(p_{\text{ref}}) \leq R(\mathcal{P})$ and $R(p_{\text{ref}}) < R(\mathcal{E})$ (Property 3.3).

- **Invariant (I2)** whenever $\mathcal{E}$ is invisible, it is inside $pocket(v, \vec{r})$ where v is the leftmost vertex of $pocket(v, \vec{r})$ and $v \in h[p_{\text{ref}}]$.

The pursuer achieves one of the following notions of progress after a finite number of time-steps. Consider two consecutive combined (SG) states, and let t and t' be the time-steps that $\mathcal{P}$ starts them correspondingly. Suppose that p_{ref} and p'_{ref} are the old and the new reference vertices at t and t' respectively. Also let v and v' be the old and new vertices which define the pockets $pocket(v, \vec{r})$ and $pocket(v', \vec{r'})$ respectively. Then:

- **Progress (P1)** either the pursuer updates p_{ref} to a new vertex p'_{ref} so that $p'_{\text{ref}} \in h(p_{\text{ref}})$ and consequently $R(p_{\text{ref}}) < R(p'_{\text{ref}})$ (Property 3.3),
- **Progress (P2)** or p_{ref} remains the same and the pursuer updates the contaminated region $pocket(v, \vec{r})$ to $pocket(v', \vec{r'})$ such that $v' \in h(v)$.

Our main result is the following theorem which we prove in Section 6:

Theorem 3.4. (Progress) *Suppose that Q is a monotone polygon. Then the pursuer by following the MPC strategy can capture the evader in $O(D^{13}n^7)$ steps where n is the number of vertices of Q and D is the diameter of Q.*

3.3 An Illustrative Example

Let us present an example which illustrates the MPC strategy (Fig. 6). Initially $\mathcal{P}$ is at O_L, $\mathcal{E}$ is visible and $p_{\text{ref}} = O_L$ (Fig. 6(a)). Therefore, $\mathcal{P}$ can follow $\mathcal{E}$ by extended lion's move (Fig. 6(b)). This continues until $\mathcal{E}$ disappears behind the vertex v. The pursuer moves toward v but in the meantime the evader also moves to an unknown location inside $pocket(v, \vec{r})$ say $\mathcal{E}_5$ (Fig. 6(c)). The pursuer updates p_{ref} to v since v, the pocket vertex, is on the upper chain. It also switches to the search state and moves along the dotted line in order to find $\mathcal{E}$ (Fig. 6(c)). By moving along this path the pursuer finally finds $\mathcal{E}$ at $\mathcal{P}_6$ (Fig. 6(d)). Next, the pursuer switches to the guard state in order to catch up with $\pi(O_L, \mathcal{E})$ and re-establish the extended lion's move state. To do so, it defines a vertex called p_{aux} which is inside $h[p_{\text{ref}}]$[4] (Fig. 6(d)). Then, $\mathcal{P}$ moves toward p_{aux} until $\mathcal{E}$ crosses the ray shot from $\mathcal{P}$ in direction of p_{aux} to $\mathcal{P}$ e.g. in $\mathcal{P}_7$ and $\mathcal{E}_7$ (Fig. 6(e)). At this time, the pursuer follows $\mathcal{E}$ by lion's move with respect to p_{aux} (Fig. 6(e)). During this lion's move the players cross the vertical line passing through p_{aux} to the left and enter the region $(Q - h[p_{\text{aux}}])$ (Fig. 6(e)). In other words, $\mathcal{E}$ re-contaminates the region to the left of p_{aux}[5]. At this time, the pursuer switches to the vertical guard sub-state in order to push the evader back to $h(p_{\text{aux}})$. The strategy in this state ensures that the evader cannot enter the shaded region in Fig. 6(e). The pursuer does this by performing the lion's move with respect to c which is the center of the circle that passes through $\mathcal{P}_9$ and I. The result is that $\mathcal{P}$ catches up with $\pi(O_L, \mathcal{E})$ inside $h(p_{\text{aux}}) \subseteq h(p_{\text{ref}})$ after finite time (Fig. 6(f) at $\mathcal{P}_{10}$). Hence $\mathcal{P}$ can again follow $\mathcal{E}$ by the extended lion's move while it has pushed p_{ref} and $\mathcal{E}$ to the right. Similarly, the pursuer keeps making progress by updating p_{ref} to the right and ensuring that $\mathcal{E} \in h(p_{\text{ref}})$.

We now present the details of each state.

4 Search State

When $\mathcal{E}$ disappears, the pursuer performs the search strategy in order to find $\mathcal{E}$. Suppose that $\mathcal{E}$ is hidden inside $pocket(v, \vec{r})$. At the time that $\mathcal{E}$ disappears behind v, the pursuer walks toward the blocking vertex v. Hence without loss of generality we can assume that at the beginning of the S state $\mathcal{P}$ is at v, the pocket vertex.

In order to find $\mathcal{E}$, the pursuer moves along a path. The following observation ensures that as long as this path is monotone in the x-axis direction, $\mathcal{P}$ finds $\mathcal{E}$ after finite time.

Observation 4.1. *Let p_1 and p_2 be two points inside the polygon where $x(p_1) < x(p_2)$ and suppose that $x(p_1) < x(\mathcal{E})$. Suppose that the pursuer moves from p_1 to p_2 along any (continuous) arbitrary path. Then, if the pursuer reaches p_2 and the evader is still invisible, it must be that $x(p_2) < x(\mathcal{E})$.*

[4]Note that p_{aux} can be the same as p_{ref}, but in this example they are different.

[5]When $p_{\text{aux}} = p_{\text{ref}}$, this is equivalent to violating the invariant (I1).

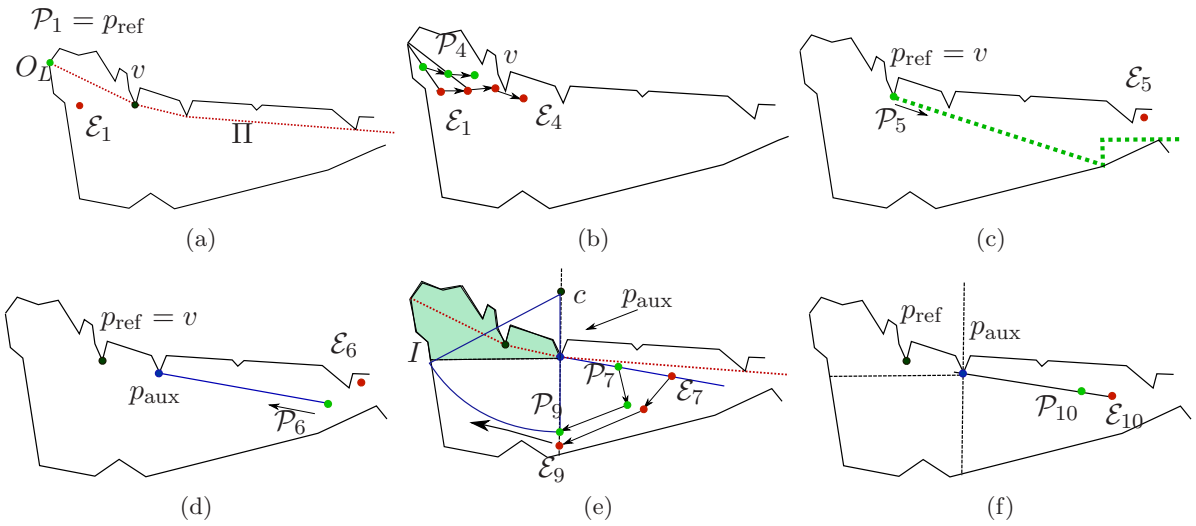

Figure 6: An instance of the game. The path Π is shown in dots. (a) Beginning of the game. (b) $\mathcal{P}$ follows $\mathcal{E}$ by extended lion's move. In the fourth time step the evader hides behind v. (c) By the time that $\mathcal{P}$ arrives at v the evader has moved to $\mathcal{E}_5$. The pursuer updates p_{ref} to v, and searches for $\mathcal{E}$ by moving along the dotted path. (d) The pursuer finds $\mathcal{E}$ at $\mathcal{P}_6$. It defines p_{aux} and moves toward it. (e) As $\mathcal{E}$ crosses the line from p_{aux} to $\mathcal{P}$, the pursuer follows it by lion's move with respect to the center p_{aux} ($\mathcal{P}_7$ and $\mathcal{E}_7$). When $\mathcal{E}$ moves to the left of p_{aux} at $\mathcal{E}_9$, $\mathcal{P}$ follows it by lion's move with respect to c. (f) Finally, $\mathcal{P}$ reaches $\pi(O_L, \mathcal{E})$ and switches to extended lion's move inside $h(p_{\text{ref}})$.

Otherwise if $\mathcal{E}$ becomes visible before the pursuer reaches p_2, then at the time that $\mathcal{E}$ is found it must be that $x(\mathcal{P}) < x(\mathcal{E})$.

Let us refer to the path that the pursuer moves along in order to find the evader as the *search path*. The search path consists of two types of paths, the *α-path* and the *step-path*. Intuitively, the α-path periodically touches the upper chain while the step-path touches the lower chain[6]. The pursuer uses these touching points as landmarks that it has to prevent the evader from re-contaminating the region to the left of those landmarks. In fact, the reference vertex p_{ref} is set to these landmarks in some cases. As we will see shortly, the search path is composed of horizontal lines, vertical lines and lines with negative slope. An example is depicted in Fig. 6(c). The pursuer exploits this slope in order to guarantee progress in the situations that the evader forces the pursuer to retreat to the left of the landmarks. For example, in Fig. 6(e), if this slope was zero, i.e. $\mathcal{P}_7$ was at the same y-coordinate as the landmark p_{aux}, and the evader has disappeared below the pursuer e.g. at $\mathcal{E}_7$, then the pursuer had to retreat along the horizontal line all the way back to I without making any progress. However, the slope provides a lower bound on the distance between the pursuer and the landmark p_{aux} at $\mathcal{P}_9$ which translates into guaranteed progress.

In the following, we will first present the definition of the α-path and the step-path, and afterwards we explain how the search path is built from them.

We define the *α-segments* as follows. The segments that make angle $-\alpha$ with the x-axis are called the *α-segments*. As we will see in Section 5.3, the angle α is used to bound the time spent in the G state.

Definition 4.2. *The angle α is chosen as the minimum of the two angles $\psi_1 = \arcsin \frac{1}{D}$ and $\psi_2 = (\frac{\pi}{2} - 2 \arctan \frac{1}{2})$ where D is the diameter of the polygon.*

The α-steps and the α-path: The α-path is composed of a number of *α-steps*. A single α-step is composed of an α-segment followed by a vertical segment and then another α-segment. For example, in Fig. 7(a) the portion of the search path from v to e_2 is an α-step. More specifically, let $e = e_1 e_2$ be

[6]This is when $\mathcal{P}$ is inside the first type critical sub-polygons. We explain the modifications required for other types in Remark 4.

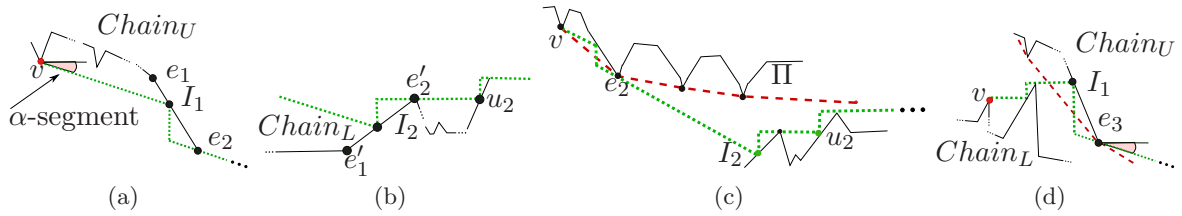

Figure 7: The search path is shown in dots. The path Π is shown in dashed line. (a) A single α-step. The portion of the search path from v to e_2 is one α-step. (b) A single step. The portion of the search path from I_2 to u_2 is one step. Note that I_2 is the floor point. (c) Two α-steps and two steps are shown. (d) When v, the pocket vertex, is from the lower chain, the search path starts by the step-path.

the edge on ∂P that the first α-segment intersects and let I_1 be the point of intersection. The edge e can be either on $Chain_U$ or $Chain_L$.

If $e \in Chain_U$, then the α-step continues along the vertical segment passing through I_1 until this segment intersects the α-segment passing through e_2. The α-step then continues along this α-segment until it reaches e_2. We refer to this part of the search path from v to e_2 as a single α-step (Fig. 7(a)).

If $e \in Chain_L$ then the α-step will be followed by the step-path described below. See Fig. 7(b). In summary, the search path continues along a number of α-steps, which together are called the α-path, until it hits the lower chain in which case it continues along the step-path. See Fig. 7(c).

The steps and the step-path: The step-path can be divided into a number of *steps*. A single step is composed of a vertical segment followed by a horizontal segment. For example, in Fig. 7(b), the portion of the search path from I_2 to u_2 is one step. Detailed definition of an step is the following. Let e_2I_2 be the α-segment from the α-path that intersects the lower chain (Fig. 7(c)). Also, let $e' = e'_1e'_2$ be the corresponding edge on $Chain_L$ (Fig. 7(b)). Then e_2 is called the *ceiling* point and I_2 is called the *floor* point. The step-path starts at the floor point I_2, continues along the vertical line passing through I_2 until this vertical line intersects the horizontal line passing through e'_2 and then continues along this horizontal line until it hits ∂P at u_2. This portion of the search path from I_2 to u_2 is referred to as a single *step* (Fig. 7(c)).

The search path: Finally, the search path is composed of the α-path and the step-path as follows. If $v \in Chain_U$, the search path starts by the α-path, otherwise if $v \in Chain_L$, it starts by the step-path. See Fig. 7 parts (c) and (d) respectively. Suppose that $v \in Chain_U$. Then, the search path continues along the α-path until it hits the lower chain. As it hits the lower chain it continues along the step-path until it hits the upper chain in which case it continues again along the α-path, and so on. This switch between the step-path and the α-path is depicted in the state diagram of Fig. 4. As an example, note that in Fig. 7(d) from v to I_1 we have the step-path and after e_3 we have the α-path.

Remark 4. The general rule for the search path for other types of critical sub-polygons is as follows. The 1^{st} and the 3^{rd} types include both the step-path and the α-path with angles $-\alpha$ and $+\alpha$ respectively. The 2^{nd} and the 4^{th} types only have the step-path. The direction traveled parallel to the y-axis during the step-path and the α-path, is the same as the sign of δs and the sign of s respectively (i.e. positive sign means upward, negative sign means downward).

Lemma 4.3. *The pursuer finds $\mathcal{E}$ after at most $O(nD)$ steps where n is the number of vertices of the polygon and D is the diameter of the polygon. Moreover, at the time that the evader is found we have $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ where v is the pocket vertex. Therefore, at the end of the search state we have $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$.*

Proof. We first bound the length of the search path that will be used to search the whole polygon. Consider the smallest bounding box that encompasses the polygon. Let H and W denote the height and the width of this bounding box respectively. According to the triangle inequality theorem, the length of the search path is less than the displacement of $\mathcal{P}$ along the x-axis and the y-axis as the pursuer moves along the search path. First, the total displacement along the x-axis is less than W since the x coordinate of the points on the search path never decreases. Second, associated with each vertex, the search path traverses in the direction of the y-axis at most once upward and at most once

downward (Fig. 7(d)). Therefore, the total displacement along the y-axis is $2nH$. Thus, the length of the search path is less than $W + 2nH = O(nD)$.

According to Observation 4.1, at the time that $\mathcal{E}$ is found $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ since the x coordinate of $\mathcal{P}$ is increasing as it moves along the search path and at the beginning of the search state $\mathcal{P}$ is at v. Consequently, we have $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$ since $v \in h[p_{\text{ref}}]$ (invariant (I2)). □

5 Guard State

After $\mathcal{P}$ finds $\mathcal{E}$, it starts the *Guard* state. The purpose of the guard strategy is to establish the extended lion's move. The extended lion's move is possible only when $\mathcal{P}$ is on $\pi(O_L, \mathcal{E})$. Therefore, in the guard state the pursuer has to reach $\pi(O_L, \mathcal{E})$. In the meantime, th evader is also moving and thus the pursuer has to preserve the progress it made so far.

At the beginning of the guard state we have $x(\mathcal{P}) < x(\mathcal{E})$ (Lemma 4.3,). At this time, based on the quadrant that $\mathcal{E}$ has appeared in, the pursuer starts different sub-states, either the *zig-zag guard* strategy or the *simple guard* strategy as shown in Algorithm 2 and the state diagram of Fig. 4.

1. If at the beginning of the G state, $\mathcal{E}$ is inside the fourth quadrant of $\mathcal{P}$ (lines 1-5 in Algorithm 2): the pursuer performs the zig-zag guard strategy.
2. Otherwise, if $\mathcal{E}$ is inside the first quadrant of $\mathcal{P}$ (lines 6-11 in Algorithm 2): the pursuer starts by simple guard sub-state.

Let p_{ref} be the current reference vertex used for tracking progress. The pursuer's ultimate goal is to maintain the invariant and the notion of progress. That is to ensure that $\mathcal{E}$ is to the right of p_{ref} i.e. inside $h(p_{\text{ref}})$. However, during zig-zag guard sub-state and simple guard sub-state, the evader can re-contaminate the region to the left of p_{ref} i.e. the region $(Q - h[p_{\text{ref}}])$. The evader does this by crossing the vertical line passing through p_{ref} to the left, and entering $(Q - h[p_{\text{ref}}])$. See line 5 and line 11 in Algorithm 2. This violates the invariant (I1). At this time, the pursuer switches to the *vertical guard* sub-state in order to push the evader back to the right of p_{ref} (line 24 in Algorithm 2). Hence at the end of the vertical guard state the invariant (I1) will be re-established.

At the end of the guard state, as the pursuer switches to the next state, it guarantees progress (P1) or (P2). Specifically, if progress (P1) is achieved the pursuer updates the reference vertex p_{ref}. See lines 15- 21. We show that the new reference vertex p'_{ref} is inside $h(p_{\text{ref}})$ and so the evader is pushed further to the right since the invariant $\mathcal{E} \in h(p'_{\text{ref}})$ is also valid (Lemma 5.1 and Lemma 5.2). We proceed by presenting the details of each sub-state, the zig-zag guard, the simple guard and the vertical guard.

Remark 5. If at the beginning of the G state, the players are inside a critical sub-polygon of the 2^{nd} type, the 3^{rd} type or the 4^{th} type, then:

1. if they are inside the 2^{nd} type (or the 4^{th} type) critical sub-polygon: $\mathcal{P}$ always starts by zig-zag guard sub-state. During zig-zag guard, the pursuer might retreat beyond p_{ref} in which case the pursuer switches to the *horizontal guard* sub-state, similar to the vertical guard, in order to recover its progress.
2. If they are inside the 3^{rd} type critical sub-polygon: the pursuer starts by zig-zag guard if $\mathcal{E}$ has appeared inside the first quadrant of $\mathcal{P}$. Otherwise, if $\mathcal{E}$ has appeared inside the fourth quadrant, the pursuer starts by simple guard. Similar to the previous case, during zig-zag guard or simple guard, the pursuer might retreat beyond p_{ref} in which case it switches to the vertical guard sub-state.

5.1 Zig-Zag Guard

After finding the evader, $\mathcal{P}$ switches to the zig-zag guard sub-state if $\mathcal{E}$ is inside the fourth quadrant of $\mathcal{P}$. The goal of this state is to establish the extended lion's move state while the invariants and the notions of progress are maintained. The *Zig-Zag Guard* strategy is shown in Algorithm 3.

Algorithm 2: Guard Strategy

Input Configuration: The evader is visible while $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$. Here, v is the pocket vertex.
Exit Configuration : One of the following two configurations: (1) The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$, or (2) The evader disappears inside $h(p_{\text{ref}})$.
The sub-states : The *Zig-zag Guard* sub-state, the *Simple Guard* sub-state, and the *Vertical Guard* sub-state. Also the *horizontal guard* sub-state when the pursuer is inside the 2^{nd} or the 4^{th} type critical sub-polygon.

if $y(\mathcal{P}) > y(\mathcal{E})$ **then**
  $state \leftarrow ZigZagGuard$;
  $l_v = \vec{Y}_{p_{\text{ref}}}$;
  do *Zig-zag Guard* strategy ;
  /* The zigzag guard ends up in one of the following configurations: (1) The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$, (2) The evader disappears inside $h(p_{\text{ref}})$, or (3) $x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{ref}})$, $y(\mathcal{E}) < y(\mathcal{P}) \leq y(p_{\text{ref}})$, and $\mathcal{E}$ is crossing l_v to the left. */
else
  $state \leftarrow SimpleGuard$;
  define $p_{\text{aux}} \in h[p_{\text{ref}}]$ as explained in Section 5.2;
  $l_v = \vec{Y}_{p_{\text{aux}}}$;
  do *Simple Guard* strategy ;
  /* The simple guard ends up in one of the following configurations: (1) $\mathcal{P}$ is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{aux}}]$ and $\mathcal{E} \in h(p_{\text{aux}})$, or (2) $\mathcal{E}$ disappears inside $h(p_{\text{aux}})$, or (3) $x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{aux}})$, $y(\mathcal{E}) < y(\mathcal{P}) \leq y(p_{\text{aux}})$, and $\mathcal{E}$ is crossing l_v to the left. */
end
/* The next state (after $\mathcal{P}$ exits the zig-zag guard state or the simple guard state). */
if $\mathcal{P}$ *is on* $\pi(O_L, \mathcal{E})$ **then**
  do *Extended Lion's Move* strategy
else if $\mathcal{E}$ *has disappeared* **then**
  /* update p_{ref} */
  Let v' be the vertex that $\mathcal{E}$ has disappeared behind;
  if $state = ZigZagGuard$ *and* $v' \in Chain_U$ **then**
    $p_{\text{ref}} \leftarrow v'$;
  else if $state = SimpleGuard$ *and* $v' \in Chain_U$ **then**
    $p_{\text{ref}} \leftarrow v'$;
  else if $state = SimpleGuard$ *and* $v' \in Chain_L$ **then**
    $p_{\text{ref}} \leftarrow p_{\text{aux}}$;
  end
  do *Search* strategy
else if $x(\mathcal{P}) = x(\mathcal{E})$ *and* $\mathcal{E}$ *is crossing* l_v *to the left* **then**
  do Vertical Guard strategy;
  /* The vertical guard ends up in one of the following two configurations: (1) The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$, (2) The evader disappears inside $h(p_{\text{ref}})$. */
end

Algorithm 3: Zig-zag Guard Strategy

Input Configuration: The evader is visible while $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ and $y(\mathcal{P}) > y(\mathcal{E})$. Here, v is the pocket vertex.

Exit Configuration : One of the following three configurations: (1) The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$, (2) The evader disappears inside $h(p_{\text{ref}})$, or (3) $x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{ref}})$, $y(\mathcal{E}) < y(\mathcal{P}) \leq y(p_{\text{ref}})$, and $\mathcal{E}$ is crossing $l_v = \vec{Y}_{p_{\text{ref}}}$ to the left.

```
repeat
    if x(P) < x(E) then
        if P is below π(O_L, E) then move in the positive ȳ direction;
        else if P is above π(O_L, E) then move in the negative ȳ direction;
    else
        move in the negative x̄ direction;
    end
until (1) The pursuer catches up with π(O_L, E), (2) The evader disappears, or (3)
x(P) = x(E) = x(p_ref), y(E) < y(P) ≤ y(p_ref), and E is crossing l_v = Ȳ_{p_ref} to the left ;
```

In order to establish the extended lion's move state, the pursuer moves toward $\pi(O_L, \mathcal{E})$ along the x-axis or the y-axis: if $x(\mathcal{P}) < x(\mathcal{E})$, the pursuer moves parallel to the $\vec{y}$-axis. Otherwise, if $\mathcal{E}$ moves to a point which is to the left of $\mathcal{P}$, then $\mathcal{P}$ moves in the negative $\vec{x}$ direction. We show that by following these zig-zag moves the evader will remain inside the fourth quadrant of $\mathcal{P}$ (Lemma 5.1). See Fig. 8 and Fig. 9(a).

Let p_{ref} be the current reference vertex used for tracking progress. According to the invariants, the pursuer must ensure that $\mathcal{E}$ is to the right of p_{ref}. However, in the strategy described above, $\mathcal{E}$ can force $\mathcal{P}$ to move along the negative x-axis direction. Therefore, the evader can re-contaminate the region to the left of p_{ref}. That is it crosses the vertical line passing through p_{ref} to the left, and enters the region $(Q - h[p_{\text{ref}}])$. This violates the invariant (I1). See Fig. 9(a). Also note that this is the third exit configuration in Algorithm 3. At this time, the pursuer switches to the *vertical guard* strategy. The vertical guard strategy (presented in section 5.3) ensures that the evader will be pushed back to the right of p_{ref} and hence the invariant and the progress are recovered. In the following lemma we show that the invariants are maintained as $\mathcal{P}$ exits the zig-zag guard state. This lemma also presents the progress that $\mathcal{P}$ gains at the end of zig-zag guard state.

Lemma 5.1 (Zig-zag guard progress)**.** *When the pursuer exits the zig-zag guard sub-state, the players are in one of the following three configurations:*

- *The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$.*
- *The evader disappears inside $h(p_{\text{ref}})$.*
- *$x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{ref}})$, $y(\mathcal{E}) < y(\mathcal{P}) \leq y(p_{\text{ref}})$, and $\mathcal{E}$ is crossing $l_v = \vec{Y}_{p_{\text{ref}}}$ to the left.*

For each of these configurations, the pursuer achieves the following notions of progress correspondingly:

- *the pursuer switches to the L state while $R(p_{\text{ref}}) \leq R(\mathcal{P}) < R(\mathcal{E})$*
- *the pursuer switches to the S state while progress (P1) or (P2) is guaranteed.*
- *the pursuer switches to the* vertical guard *sub-state.*

Proof. Let $\mathcal{P}_0$ and $\mathcal{E}_0$ be the positions of $\mathcal{P}$ and $\mathcal{E}$ at the beginning of the zig-zag guard. Observe that $parent(\mathcal{P}_0)$ is in the second quadrant of $\mathcal{P}_0$ (Lemma 9.4). Now consider the funnel [5] formed by $\pi(O_L, \mathcal{E})$ and $\pi(O_L, \mathcal{P})$. Let d be the deepest common vertex between these two paths. The shortest path to all points inside this funnel starts by $\pi(O_L, d)$ and then continues inside the funnel. Observe that as $\mathcal{E}$ moves, $\pi(O_L, \mathcal{E})$ changes continuously. See Fig. 8.

Suppose that $\mathcal{P}$ is below $\pi(O_L, \mathcal{E})$ (Fig. 8(c)). Then $\mathcal{P}$ is getting closer to $\pi(O_L, \mathcal{E})$ *just* by moving in the positive $\vec{y}$ direction. Note that the slope of the edge between $\mathcal{P}$ and $parent(\mathcal{P})$ is negative (Lemma 9.4). Hence, if the evader tries to cross $\vec{Y}_{\mathcal{P}}$ to the left, $\mathcal{P}$ will be on $\pi(O_L, \mathcal{E})$ and thus it can switch to the L state. Therefore, $\mathcal{E}$ has to remain inside the fourth quadrant of $\mathcal{P}$ until one of the

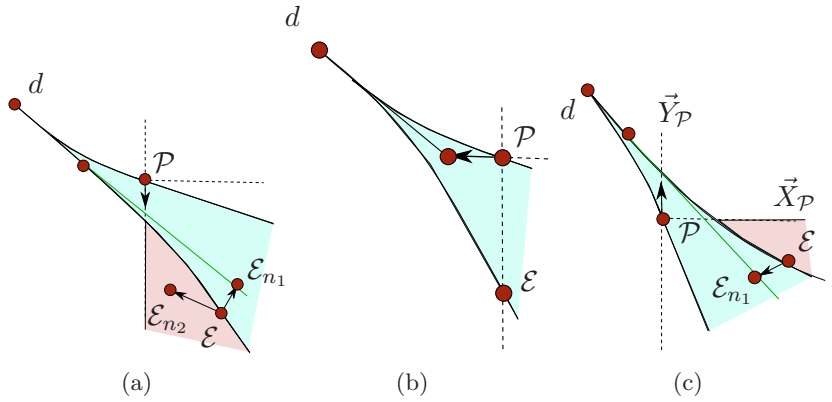

Figure 8: The zig-zag guard strategy. (a) When $x(\mathcal{P}) < x(\mathcal{E})$ and $\mathcal{P}$ is above $\pi(O_L, \mathcal{E})$. (b) When $x(\mathcal{P}) = x(\mathcal{E})$ and $\mathcal{P}$ is above $\pi(O_L, \mathcal{E})$. (c) When $x(\mathcal{P}) < x(\mathcal{E})$ and $\mathcal{P}$ is below $\pi(O_L, \mathcal{E})$.

following happens: (1) $\mathcal{P}$ is on $\pi(O_L, \mathcal{E})$, (2) the evader disappears. Now, let v be the pocket vertex that the pursuer searched right before this zig-zag guard state. According to Lemma 4.3, $x(v) \leq x(\mathcal{P})$. Also, according to the invariant (I2) we have $v \in h[p_{\text{ref}}]$. Thus, $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$. Hence, invariant (I1) is valid at the end of zig-zag guard state.

Similarly, if the pursuer is above $\pi(O_L, \mathcal{E})$ it moves in the negative $\vec{y}$ direction when $x(\mathcal{P}) < x(\mathcal{E})$, and then to the left, i.e. in the negative $\vec{x}$ direction, at the moment that $x(\mathcal{P}) = x(\mathcal{E})$. See Fig. 8 parts (a) and (b), also Fig. 9(a). These zig-zag moves continue until (1) the pursuer is on $\pi(O_L, \mathcal{E})$, (2) the evader disappears, or (3) the players cross $\vec{Y}_{p_{\text{ref}}}$ to the left (Fig. 9(b)). At the time that each of these happen, the players are inside $h(p_{\text{ref}})$ and thus invariant (I1) is valid.

Next let us consider invariant (I2) that the pocket vertex is inside $h[p_{\text{ref}}]$. Initially, $p_{\text{ref}} = O_L$. Thus, the invariant holds. According to the above argument, as the game proceeds to zig-zag guard state the invariant is still valid since when $\mathcal{E}$ disappears it is inside $h(p_{\text{ref}})$. Therefore, using induction on time we can prove that the pocket vertex is inside $h[p_{\text{ref}}]$.

So far we have shown that the invariants (I1) and (I2) are maintained. Next, let us consider the pursuer's progress. As discussed above, the pursuer finishes the zig-zag guard state when it reaches $\pi(O_L, \mathcal{E})$, or when the evader disappears, or when the players move to the left of p_{ref}. In the first case, since $\mathcal{P}$ is on $\pi(O_L, \mathcal{E})$ we have $R(\mathcal{P}) < R(\mathcal{E})$. Also, since $\mathcal{P} \in h[p_{\text{ref}}]$ we have $R(p_{\text{ref}}) \leq R(\mathcal{P}) < R(\mathcal{E})$.

Now consider the case that the evader disappears. Let v' be the new pocket vertex that $\mathcal{E}$ disappears behind, and v be the pocket vertex that was used right before this zig-zag guard state. Also, let p_{ref} be the reference vertex at the beginning of this zig-zag guard state, and p'_{ref} be the new reference vertex at the end of this state.

1. $v \in Chain_L$: suppose that at the beginning of the zig-zag guard the pursuer was below $\pi(O_L, \mathcal{E})$. Then, $v' \in h(v)$ since $\mathcal{P}$ is only moving upward. If $v' \in Chain_L$ we do not update p_{ref} but we have $v' \in h(v)$ (Progress (P2)). Otherwise, if $v' \in Chain_U$ we set p'_{ref} to v'. Since $p'_{\text{ref}} = v' \in h(v)$ and $v \in h[p_{\text{ref}}]$ (Invariant (I2)), we have $R(p_{\text{ref}}) < R(p'_{\text{ref}})$ (Property 3.3) and hence we have Progress (P1).

   Next, suppose that at the beginning of zig-zag guard the pursuer was above $\pi(O_L, \mathcal{E})$. Since $\mathcal{E}$ remains inside the fourth quadrant of $\mathcal{P}$, $v \in Chain_L$, and $\mathcal{P}$ is moving downward and to the left, $\mathcal{E}$ cannot cross v to the left. Hence, v' is also in the fourth quadrant of v. If $v' \in Chain_L$ we do not update p_{ref} but we have $v' \in h(v)$ (Progress (P2)). Otherwise, if $v' \in Chain_U$ we set p'_{ref} to v'. Since $p'_{\text{ref}} \in h(v) \subseteq h(p_{\text{ref}})$ we have Progress (P1).
2. $v \in Chain_U$: then $p_{\text{ref}} = v$. As we showed above, whether $\mathcal{P}$ is initially below or above $\pi(O_L, \mathcal{E})$, the evader disappears inside $h(p_{\text{ref}})$. Similar to the previous case, if $v' \in Chain_L$ we have $v' \in h(p_{\text{ref}} = v)$ (Progress (P2)), and if $v' \in Chain_U$ we set p'_{ref} to v' and we have Progress (P1).

In summary, if $\mathcal{E}$ disappears during zig-zag guard state, the pursuer updates p_{ref} to v' when $v' \in Chain_U$ and achieves progress (P1) (line 17 in Algorithm 2). Otherwise, if $v' \in Chain_L$ the pursuer achieves progress (P2).

□

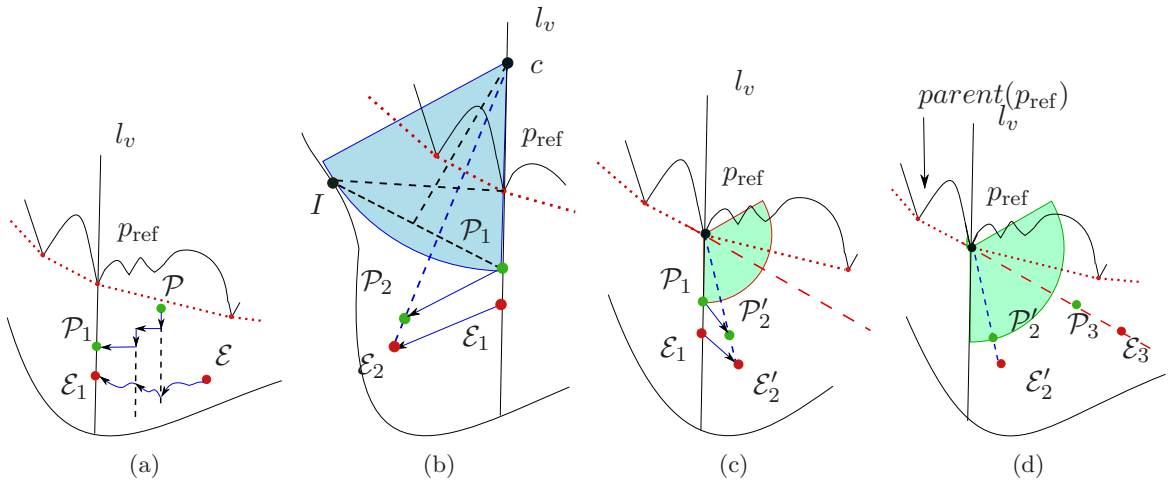

Figure 9: The zig-zag guard followed by the vertical guard . The path Π is shown in dots. (a) The zig-zag moves. (b) If $\mathcal{E}$ moves to the left of p_{ref}, the pursuer follows it by lion's move with respect to c. (c) Afterwards, if $\mathcal{E}$ moves to the right of p_{ref}, the pursuer performs the lion's move with respect to p_{ref}. (d) Finally, $\mathcal{P}$ will be on $\pi(O_L, \mathcal{E})$ inside $h(p_{\text{ref}})$.

5.2 Simple Guard

After finding the evader, $\mathcal{P}$ switches to the simple guard sub-state if $\mathcal{E}$ is inside the first quadrant of $\mathcal{P}$. The main goal of the pursuer is to reach $\pi(O_L, \mathcal{E})$ so it can start the extended lion's move state. Since the evader can disappear in the meantime, this translates to establishing the next state which could be the search state or the extended lion's move state while the invariants and the notions of progress are maintained. The *Simple Guard* strategy is presented in Algorithm 4.

The general idea for the pursuer's strategy is the following. See Fig. 10, and let p_{ref} be the current reference vertex used for tracking progress. Let v be the vertex that defines the contaminated pocket right before this guard state. When the pursuer starts the simple guard state we have $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ (the exit configuration of the search state). Therefore at this time, $\mathcal{P}$ is to the right of p_{ref} i.e. inside $h[p_{\text{ref}}]$ since $v \in h[p_{\text{ref}}]$ (Invariant (I2)). Now let p_0 be the location of the pursuer at the beginning of the simple guard state. The pursuer moves back toward the vertex p_{ref} along the line segment that connects p_0 to p_{ref} ($\mathcal{P}_1$ in Fig. 10(a)). It continues moving toward p_{ref} until it reaches p_{ref}, or $\mathcal{E}$ crosses the ray shot from $\mathcal{P}$ in the direction of p_{ref} to $\mathcal{P}$ ($\mathcal{P}_2$ and $\mathcal{E}_2$ in Fig. 10(b)). In both of these cases, $\mathcal{P}$ follows $\mathcal{E}$ by lion's move with respect to the center p_{ref}. The pursuer continues with this lion's move until one of the following configurations hold: 1) either $\mathcal{P}$ reaches $\pi(O_L, \mathcal{E})$ while it is inside $h(p_{\text{ref}})$, 2) or $\mathcal{E}$ moves to the region which is to the left of p_{ref} i.e. it crosses the vertical line which passes through p_{ref} to the left and re-contaminates $h(p_{\text{ref}})$. See $\mathcal{P}_3$ and $\mathcal{E}_3$ in Fig. 10(b).

In the former case, when the pursuer catches up with $\pi(O_L, \mathcal{P})$, it switches to the extended lion's move state, and since it is inside $h(p_{\text{ref}})$ the invariant (I1) is maintained. In the latter case, when $\mathcal{E}$ moves to the left of p_{ref}, invariant (I1) is violated. At this time, the pursuer switches to the next state which is called the vertical guard state in order to push the evader back to the right of p_{ref}, and re-establish invariant (I1). The vertical guard strategy, presented in section 5.3, guarantees that the pursuer reaches $\pi(O_L, \mathcal{E})$ inside $h(p_{\text{ref}})$ and therefore is ready for the extended lion's move state while invariant (I1) holds.

The simple guard strategy described above has two subtle points. First, p_{ref} must be visible to p_0 so that $\mathcal{P}$ can move along the segment that connects p_{ref} to p_0. However, it might be the case that p_{ref} is not visible to p_0 since p_0 is the location of the pursuer at the time that $\mathcal{P}$ has found $\mathcal{E}$. Therefore, we define an auxiliary vertex, referred to as p_{aux}, so that it is visible to p_0 and moreover $p_{\text{aux}} \in h[p_{\text{ref}}]$. The pursuer performs the aforementioned strategy with respect to p_{aux} i.e. it moves toward p_{aux} and the rest of the strategy. Since the simple guard strategy ensures that in the next state the players are inside $h(p_{\text{aux}})$, and because $p_{\text{aux}} \in h[p_{\text{ref}}]$ the invariant (I1) is maintained. The second important part of the strategy is that p_0 must be closer to all points on the segment between p_0 and p_{aux} than

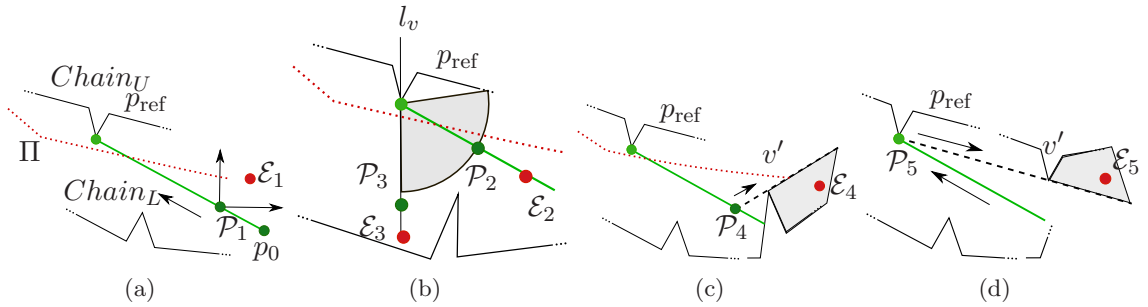

Figure 10: The Simple Guard. (a) $\mathcal{P}$ moves toward p_{ref}. (b) As $\mathcal{E}$ crosses the ray from $\mathcal{P}$ to p_0, $\mathcal{P}$ performs lion's move w.r.t. p_{ref} ($\mathcal{P}_2$ and $\mathcal{E}_2$). If during this lion's move $\mathcal{E}$ enters the region to the left of p_{ref}, the pursuer switches to the vertical guard sub-state e.g. $\mathcal{P}_3$ and $\mathcal{E}_3$. (c) If $\mathcal{E}$ hides inside $pocket(v', \vec{r'})$, $\mathcal{P}$ moves toward v' if $v' \in Chain_L$, (d) otherwise if $v' \in Chain_U$ it continues moving toward p_{ref} and then from there it moves toward v'. Note that here, as the pursuer keeps moving back to p_{ref}, the new pocket formed by the ray connecting $\mathcal{P}$ to v', includes the initial hiding pocket $pocket(v', \vec{r'})$. Also in this example $p_{\text{aux}} = p_{\text{ref}}$.

the evader so that $\mathcal{P}$ can prevent $\mathcal{E}$ from crossing this segment and thus escaping to the cleared region $(Q - h[p_{\text{ref}}])$. Notice that if the pursuer does not prevent this type of re-contamination the evader will be above the pursuer i.e. $y(\mathcal{P}) < y(\mathcal{E})$. Therefore $\mathcal{P}$ cannot force $\mathcal{E}$ back to $h(p_{\text{ref}})$ by performing the vertical guard strategy as it does when $y(\mathcal{P}) > y(\mathcal{E})$ (we will see in section 5.3 that one of the conditions that $\mathcal{P}$ is allowed to perform the vertical guard sub-state is $y(\mathcal{P}) > y(\mathcal{E})$). Instead the pursuer prevents this situation by guaranteeing that it is closer to all points on the segment between p_0 and p_{aux}. This, in addition to the capture condition that $\mathcal{E}$ will be captured if its distance to $\mathcal{P}$ is less than one unit, ensures that $\mathcal{E}$ will be captured if it tries to cross the segment between p_{aux} and $\mathcal{P}$. Finally, the pursuer ensures that it is closer to p_0 by guaranteeing that the angle between the aforementioned segment and the x-axis is less than or equal to α (Lemma 9.7).

An illustrative example of the auxiliary vertex p_{aux} is shown in Fig. 11(a). The interested reader is referred to the Appendix, Section 9.4, for the definition of the auxiliary vertex p_{aux} based on the structure of the polygon and the location of the pursuer.

Lemma 5.2 (Simple guard progress)**.** *When the pursuer exits the simple guard sub-state, the players are in one of the following three configurations:*

- *The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{aux}}]$ and $\mathcal{E} \in h(p_{\text{aux}})$.*
- *The evader disappears inside $h(p_{\text{aux}})$.*
- *$x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{aux}})$, $y(\mathcal{E}) < y(\mathcal{P}) \leq y(p_{\text{aux}})$, and $\mathcal{E}$ is crossing $l_v = \vec{Y}_{p_{\text{aux}}}$ to the left.*

For each of these configurations, the pursuer achieves the following notions of progress correspondingly:

- *the pursuer switches to the L state while $R(p_{\text{ref}}) \leq R(p_{\text{aux}}) \leq R(\mathcal{P}) < R(\mathcal{E})$*
- *the pursuer switches to the S state while progress (P1) or (P2) is guaranteed.*
- *the pursuer switches to the* vertical guard *sub-state.*

Proof. From the description above, the exit configuration of simple guard is one of the following: (1) the L state inside $h(p_{\text{aux}})$, (2) the S state inside $h(p_{\text{aux}})$, or (3) vertical guard while the player are crossing l_v to the left. In case of the L state, since $p_{\text{aux}} \in h[p_{\text{ref}}]$ we would have $R(p_{\text{ref}}) \leq R(p_{\text{aux}}) \leq R(\mathcal{P}) < R(\mathcal{E})$.

In case of the S state, let v' be the new pocket vertex. Then $v' \in h(p_{\text{aux}}) \subseteq h(p_{\text{ref}})$. Therefore, the invariant (I2) is valid.

Next, let us consider the progress. Let p'_{ref} denote the new reference vertex that is updated in this guard state. If $v' \in Chain_U$ we set p'_{ref} to v'. Since $p'_{\text{ref}} = v' \in h(p_{\text{ref}})$ we have $R(p_{\text{ref}}) < R(p'_{\text{ref}})$ (Property 3.3) and hence we have Progress (P1).

If $v' \in Chain_L$ we update p_{ref} to p_{aux}. Note that p_{aux} can be the same as p_{ref}. However, we show that $v' \in h(v)$. Suppose that $v \in Chain_L$ and p_{aux} is to the left of v. The remaining situations are similar. Now if $v' \notin h(v)$, the pocket $pocket(v', \vec{r'})$ defined by v' will be a simple pocket and thus by performing the simple pocket strategy (section 9.1) the pursuer can force the evader to exit $pocket(v', \vec{r'})$ and continue the simple pocket strategy. See Fig. 13(a) for an illustration. □

The complete description of simple guard strategy is given in Algorithm 4. Notice that when the evader disappears while $\mathcal{P}$ is moving back toward p_{ref}, the pursuer's reaction depends on whether the hiding vertex v' is from $Chain_L$ or $Chain_U$ (lines 12-16 in Algorithm 4). An example is shown in Fig. 10(c) parts (c) and (d). When $v' \in Chain_L$, the pursuer moves toward v' until it reaches v' at which time it switches to the S state ($\mathcal{P}_4$ and $\mathcal{E}_4$ in Fig. 10(c)). When $v' \in Chain_U$, the pursuer continues moving back toward p_{ref} and if in the meantime $\mathcal{E}$ crosses $\overrightarrow{ray}$ (line 3) the pursuer performs the same strategy from line 4. In the following, we briefly explain the reason that $\mathcal{P}$ must make distinction between $v' \in Chain_L$ and $v' \in Chain_U$:

- $v' \in Chain_L$: in this case, if the pursuer keeps moving back toward p_{ref}, it cannot keep track of the hiding vertex v'. For example, in Fig. 11(c), at the time that $\mathcal{E}$ disappears, $\mathcal{P}$ defines the hiding pocket with respect to $v' = v_1$. If $\mathcal{P}$ continues moving back toward p_{ref}, at $\mathcal{P}_2$ the hiding pocket with respect to v_1 doesn't include $\mathcal{E}$ (Fig. 11(d)). In other words, $\mathcal{P}$ cannot keep track of the hiding vertex v'.
- $v' \in Chain_U$: in this case $x(v')$ can be less than $x(\mathcal{P})$. Therefore, if the pursuer moves toward v' the evader *can cross* the segment between p_{ref} and $\mathcal{P}$ (Fig. 11(b)) and thus it escapes to the previously cleared region.

Algorithm 4: Simple Guard Strategy

Input Configuration: The evader is visible while $x(v) \le x(\mathcal{P}) < x(\mathcal{E})$ and $y(\mathcal{P}) < y(\mathcal{E})$. Here, v is the pocket vertex.

Exit Configuration : One of the following three configurations: (1) The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{aux}}]$ and $\mathcal{E} \in h(p_{\text{aux}})$, (2) The evader disappears inside $h(p_{\text{aux}})$, or (3) $x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{aux}})$, $y(\mathcal{E}) < y(\mathcal{P}) \le y(p_{\text{aux}})$, and $\mathcal{E}$ is crossing $l_v = \vec{Y}_{p_{\text{aux}}}$ to the left.

```
1  define p_aux as explained in section 5.2;
2  let p_0 be the location of E at the beginning of the simple guard;
3  let ray be the ray shot from P in the direction of p_aux to p_0;
4   repeat
5   |  move toward p_aux along the segment p_0 p_aux;
   until E crosses ray, or E disappears, or P reaches p_aux ;
6  if E has crossed ray, or P has reached p_aux then
7   |  if E has crossed ray, or P has reached p_aux and E is visible then
8   |  |  repeat
9   |  |  |  do lion's move with respect to the center p_aux;
   |  |  until P is on π(O_L, E), or E disappears, or x(P) = x(E) = x(p_aux) and E is crossing
   |  |  l_v = Y_{p_aux} to the left ;
10  |  else if P has reached p_aux and E is hidden behind v' inside pocket(v', r') then
11  |  |  move toward v';
   |  end
12  else if E has disappeared behind v' inside pocket(v', r') then
   |  /* P is not at p_aux yet.                                              */
13  |  if v' ∈ Chain_L then
14  |  |  move toward v' until P is at v';
15  |  else if v' ∈ Chain_U then
16  |  |  continue moving toward p_aux and the rest of the strategy at line 4;
17  |  |  in the meantime keep track of the hiding vertex v';
   |  end
   end
```

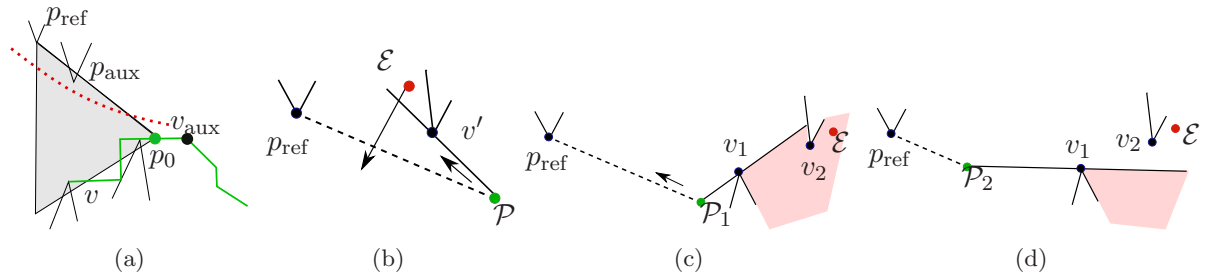

Figure 11: (a) An example of p_{aux}. (b), (c) and (d) Examples when $\mathcal{E}$ disappears behind a vertex, namely v', during simple guard. The pursuer has to make distinction between $v' \in Chain_U$ and $v' \in Chain_L$, otherwise $\mathcal{E}$ can escape. Refer to the text.

5.3 Vertical Guard

The vertical guard strategy is presented in Algorithm 5. The pursuer switches to the vertical guard sub-state from either the zig-zag guard sub-state or the simple guard sub-state. See Fig. 4. Let p_{ref} be the current reference vertex used for tracking the progress. The condition for state transition to the vertical guard sub-state is when the evader re-contaminates the region to the left of p_{ref} i.e. it enters the region $P - h[p_{\text{ref}}]$ and violates the invariant (I1). Since $h(p_{\text{ref}})$ is defined as the set of points to the right of p_{ref}[7], this condition is in fact when $x(\mathcal{P}) = x(\mathcal{E}) = x(p_{\text{ref}})$, $y(\mathcal{E}) < y(\mathcal{P}) \leq y(p_{\text{ref}})$ and $\mathcal{E}$ is moving to the left of p_{ref}. Let l_v denote the line $\vec{Y}_{p_{\text{ref}}}$. Then, the pursuer switches to the vertical guard state when $\mathcal{E}$ crosses l_v to the left. The goal of the vertical guard strategy is to push the evader back to the right of p_{ref} and hence re-establish the invariant (I1).

The vertical guard strategy is composed of two parts: lion's move with respect to a center c (which we define soon) and lion's move with respect to the center p_{ref}. The pursuer uses c as the center for the lion's move if $\mathcal{E}$ crosses l_v to the left. It also uses p_{ref} as the center for lion's move if $\mathcal{E}$ crosses l_v to the right. The role of the circle centered at c is to push $\mathcal{E}$ back to the right of p_{ref} (inside $h(p_{\text{ref}})$) and re-establish the invariant (I1). The role of the other circle centered at p_{ref} is to force $\mathcal{E}$ to cross the ray connecting $parent(p_{\text{ref}})$ to p_{ref} and hence to establish the extended lion's move state. See Fig. 9. The vertical guard strategy is as follows, see Fig. 9:

1. As the evader crosses the vertical line l_v to the left, $\mathcal{P}$ follows him by lion's move with respect to c, see Fig. 9-(b) $\mathcal{P}_1$ and $\mathcal{E}_1$ to $\mathcal{P}_2$ and $\mathcal{E}_2$ respectively.

   If the evader disappears behind the lower chain vertices which are to the left of l_v, the pocket would be of a special form that we call it *simple pocket*[8]. In simple pockets the pursuer has a relatively simple strategy so that the evader *has to* exit the pocket to prevent capture. See section 9.1 for definition of simple pocket and the corresponding pursuit strategy. Therefore, when $\mathcal{E}$ disappears somewhere to the left of l_v, the pursuer can repel him outside the hiding pocket by simple pocket strategy.

   Consequently, as long as $\mathcal{E}$ is on the left side of l_v, the distance $c\mathcal{P}$ increases while $\mathcal{P}$ lies on $c\mathcal{E}$. As a result, $\mathcal{E}$ will be pushed to the right of l_v after finite time (Fig. 9-(c) where $\mathcal{E}$ is at $\mathcal{E}_2'$).
2. As the evader crosses the vertical line l_v to the right, $\mathcal{P}$ switches to lion's move with respect to the center p_{ref} (Fig. 9-(c) where $\mathcal{P}$ moves to $\mathcal{P}_2'$).
3. This back and forth switch between lion's move with respect to centers c and p_{ref} continues until one of the following two configurations hold: 1) the evader disappears behind a vertex to the right of p_{ref}, 2) or the extended lion's move is established ($\mathcal{P}_3$ and $\mathcal{E}_3$ in Fig. 9-(d)).

Next, let us proceed with the definition of the center c. Let I be the intersection between the horizontal line passing through p_{ref} and ∂P, see Fig. 12(a). Then c is the intersection between bisector of $\mathcal{P}I$ and the line l_v.

[7] Inside the first type critical sub-polygon.
[8] See Lemma 9.12.

Algorithm 5: Vertical Guard Strategy

Input Configuration: The evader and the pursuer are both on the line l_v and $\mathcal{E}$ is crossing l_v to the left. In other words, $x(\mathcal{P}) = x(\mathcal{E}) = x(c_{\text{aux}})$ and $y(\mathcal{E}) < y(\mathcal{P}) \leq y(c_{\text{aux}})$, and $\mathcal{E}$ is moving to the left of the vertex c_{aux}. Refer to the lines 1- 6 for definition of the vertex c_{aux} and the line l_v.

Exit Configuration : One of the following two configurations: (1) The pursuer is on $\pi(O_L, \mathcal{E})$ while $\mathcal{P} \in h[p_{\text{ref}}]$ and $\mathcal{E} \in h(p_{\text{ref}})$, (2) The evader disappears inside $h(p_{\text{ref}})$.

1 **if** *the previous state is zig-zag guard* **then**
2 $l_v = \vec{Y}_{p_{\text{ref}}}$;
3 $c_{\text{aux}} \leftarrow p_{\text{ref}}$;
4 **else if** *the previous state is simple guard* **then**
5 $l_v = \vec{Y}_{p_{\text{aux}}}$;
6 $c_{\text{aux}} \leftarrow p_{\text{aux}}$;
end
7 $I \leftarrow \vec{X}_{c_{\text{aux}}} \cap \partial Q$;
8 $c =$ bisector of $\mathcal{P}I \cap l_v$;
9
10 **repeat**
11 **if** $\mathcal{E}$ *is to the left of* c_{aux} **then**
12 **repeat**
13 do lion's move with respect to the center c;
until $\mathcal{E}$ *disappears somewhere to the left of* c_{aux}*, or* $\mathcal{E}$ *moves to the right of* c_{aux} ;
14 **if** $\mathcal{E}$ *has disappeared somewhere to the left of* c_{aux} **then**
15 perform the *simple pocket* strategy presented in section 9.1;
`/* as a result of the simple pocket strategy the evader is forced to exit the hiding pocket while` $\mathcal{P}$ `is on the entrance of the pocket and their distance has been increased. */`
16 continue from line 12 ;
17 **else if** $\mathcal{E}$ *has moved to the right of* c_{aux} **then**
18 continue from line 19
end
end
19 **if** $\mathcal{E}$ *is to the right of* c_{aux} **then**
20 **repeat**
21 do lion's move with respect to the center c_{aux};
until $\mathcal{P}$ *is on* $\pi(O_L, \mathcal{E})$ *to the right of* c_{aux}*, or* $\mathcal{E}$ *disappears somewhere to the right of* c_{aux}*, or* $\mathcal{E}$ *moves to the left of* c_{aux} ;
22 **if** $\mathcal{E}$ *has moved to the left of* c_{aux} **then**
23 continue from line 11;
24 **else**
25 exit the vertical guard sub-state;
end
end
until *The pursuer catches up with* $\pi(O_L, \mathcal{E})$ *inside* $h(c_{\text{aux}})$*, or (2) The evader disappears inside* $h(c_{\text{aux}})$;

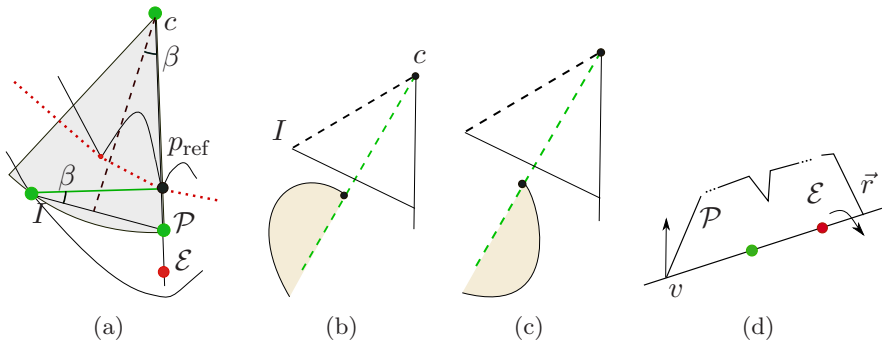

Figure 12: (a) the vertical guard (Π is shown in dots). (b) These pockets are impossible. (c) The possible pockets before l_v are simple pockets. (d) The simple pockets.

Remark 6. When the state before the vertical guard state is the simple guard state, the pursuer defines the center c based on p_{aux} instead of p_{ref}. In Algorithm 5, we use the notation c_{aux} to refer to p_{ref} or p_{aux}. See lines 1-6 in Algorithm 5.

The lion's move circle centered at c has the following important properties. First, all upper chain vertices before p_{ref} are above $p_{\text{ref}}I$ and thus the lion's move with respect to c is feasible (Lemma 9.14).

Second, the lion's move circle centered at c prevents $\mathcal{E}$ from hiding behind the upper chain vertices which are to the left of p_{ref} without being captured (Lemma 9.11). Also see Fig. 6-(e). These vertices are the ones on $Chain_U$ with their x-coordinate less than p_{ref}. In addition, the circle is defined such that if $\mathcal{E}$ disappears behind lower chain vertices which are to the left of p_{ref}, the resulting pocket would be a simple pocket (Lemma 9.12), pockets that $\mathcal{P}$ has a relatively simple strategy to push $\mathcal{E}$ out of the pocket (section 9.1).

Third, the initial radius of this circle is upper bounded which is necessary for the lion's move with respect to center c to result in progress in finite number of steps [11]. In the following, we show that the angle α plays an important role in bounding the radius. Let r be the radius of the lion's circle centered at c i.e. $r = c\mathcal{P}$.

Lemma 5.3. *The initial radius (r) of the circle defined above is upper bounded by $r \leq \frac{2l^2}{h_{min}}$ where h_{min} is a lower bound for $h = y(p_{\text{ref}}) - y(\mathcal{P})$ at the beginning of vertical guard ($h_{min} \leq h$).*

Proof. Let β be the angle between $\mathcal{P}I$ and the horizontal line passing from c_{aux}, see Fig. 12(a). Also let $2l = \mathcal{P}I$. We have $\sin\beta = \frac{l}{r} = \frac{h}{2l}$. Hence $r = \frac{2l^2}{h}$. Therefore $r \leq \frac{2l^2}{h_{min}}$. □

In Lemma 9.5 and Lemma 9.9, we provide $h_{min} = \sin\alpha$ as the lower bound for h at the beginning of the vertical guard strategy.

Corollary 5.4. *The initial radius of the vertical guard circle centered at c is upper bounded by $r \leq \frac{D^2}{2\sin\alpha}$. According to Definition 4.2, we have r is $O(D^3)$.*

Lemma 5.5 (Vertical guard progress)**.** *At the end of vertical guard the pursuer achieves Progress (P1) or (P2).*

Proof. Let $pocket(v, \vec{r})$ be the pocket searched in the previous search state. Recall that if the vertical guard is invoked from zig-zag guard then $c_{\text{aux}} = p_{\text{ref}}$ and if it is invoked from simple guard then $c_{\text{aux}} = p_{\text{aux}}$ (Remark 6).

1. from zig-zag guard ($c_{\text{aux}} = p_{\text{ref}}$):
   (a) $v \in Chain_U$: hence $p_{\text{ref}} = v$. The next state after the vertical guard is either L or S. As the evader exits the vertical guard state the players are inside $h(p_{\text{ref}}) = h(v)$. If the next state is S, let $pocket(v', \vec{r'})$ be the corresponding pocket. Then, $v' \in h(v)$. If $v' \in Chain_U$ we set $p_{\text{ref}} = v'$ and we have Progress (P1). If $v' \in Chain_L$ we have Progress (P2).

(b) $v \in Chain_L$: this case is not possible. This is because: the evader is inside the fourth quadrant of $\mathcal{P}$ (zig-zag guard state), $v \in h[p_{\text{ref}}]$ according to invariant (I2), and the search path used for searching $pocket(v, \vec{r})$ ensures that the evader cannot cross p_{ref} to the left. See Fig. 13-(b).

2. from simple guard ($c_{\text{aux}} = p_{\text{aux}}$):
   (a) $v \in Chain_U$: then $p_{\text{ref}} = v$ and $p_{\text{aux}} \in h[p_{\text{ref}}]$. At the time that $\mathcal{P}$ exits the vertical guard state the players are inside $h(p_{\text{aux}})$ which is a subset of $h(p_{\text{ref}})$. The pursuer achieves Progress (P1) or (P2) similar to the above case.
   (b) $v \in Chain_L$: refer to the definition of p_{aux} in section 5.2 and note that p_{aux} can be to the left or to the right of v. The case that p_{aux} is to the right of v is similar to the above cases. Suppose that p_{aux} is to the left of v. Referring to the definition of p_{aux} in section 5.2 this is the case only when p_0 is in between v and v_{aux} (the first definition in section 5.2). Suppose that during vertical guard (lion's move w.r.t. p_{aux}) the evader disappears behind v'. When $v' \in Chain_U$ we have $v' \in h(p_{\text{aux}}) \subseteq h(p_{\text{ref}})$. We set p_{ref} to v' and achieve progress (P1). When $v' \in Chain_L$ there are two cases. The first is when $x(v) < x(v')$ which we have progress (P2). The second is when $x(v') \leq x(v)$ in which case the resulting pocket $pocket(v', \vec{r'})$ would be a simple pocket and the pursuer performs the simple pocket strategy in section 9.1 in order to resume the lion's move with respect to p_{aux}. See Fig. 13-(a).

□

Lemma 5.6 (The time spent in the guard state)**.** *The pursuer exits the guard state and switches to the next state in $O(n^4D^{11})$ steps where n is the number of vertices of Q and D is the diameter of Q.*

Proof. Let T_1 be the time spent in simple guard, T_2 be the time spent in vertical guard, and T_3 be the time spent in the zig-zag guard state. Then the guard time is at most $T_1T_2 + T_3T_2$.

The vertical guard strategy is composed of simple pocket strategy (to the left of l_v) and lion's move with respect to c_{aux} or c. In the worst case, every single step of the lion's move can be followed by a round of simple pocket strategy. Therefore, the total time in vertical guard would be the product of the time spent in lion's move and the simple pocket strategy. The time spent in simple pocket strategy is $O(n^2D^3)$ (Lemma 9.1). The initial radius of the circle centered at c used during vertical guard is $O(D^3)$ (Corollary 5.4). Hence the lion's move with respect to c during vertical guard takes $O(nD^6)$ steps [8]. Therefore, the vertical guard state takes $T_2 = O((nD^6) \cdot (n^2D^3)) = O(n^3D^9)$.

The simple guard is lion's move with respect to p_{aux} which can take at most $T_1 = O(nD^2)$. The zig-zag guard is composed of zig-zag moves which are of time $T_3 = O(D)$.

Thus, the guard time is $O((nD^2) \cdot (n^3D^9) + D \cdot (n^3D^9)) = O(n^4D^{11})$. □

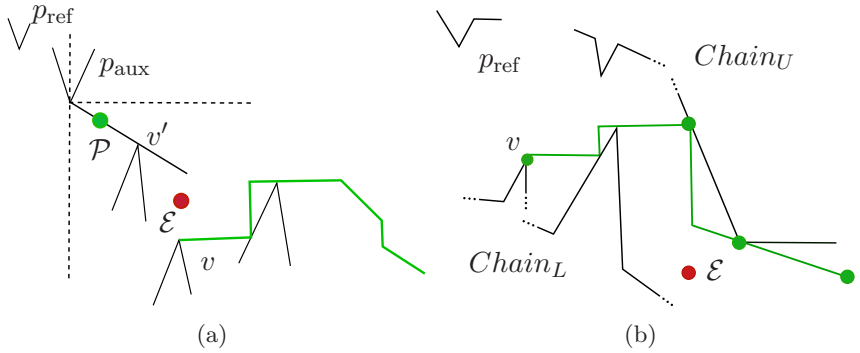

Figure 13: (a) When $v \in Chain_L$ and $\mathcal{E}$ disappears behind $v' \in Chain_L$ so that $x(v') < x(v)$, the resulting pocket is a simple pocket. (b) When $v \in Chain_L$ and $\mathcal{E}$ appears in the fourth quadrant, $\mathcal{E}$ cannot cross p_{ref} to the left since it is confined with ∂Q.

6 Analysis of Capture Time

We are now ready to present the proof of Theorem 3.4 which gives the worst capture time of the proposed pursuit strategy.

Proof of Theorem 3.4. Suppose $\mathcal{P}$ is currently in a combined (SG) state. In Lemma 5.1, Lemma 5.2, and Lemma 5.5, we showed that after finite time this combined state will terminate to another combined (SG) state or an L state.

In the latter case, the aforementioned lemmas ensure that $R(p_{\text{ref}}) \leq R(\mathcal{P}) < R(\mathcal{E})$. Moreover $\mathcal{P}$ is on $\pi(O_L, \mathcal{E})$ and $d(O_L, \mathcal{P})$ is increasing after each step of the L state [8]. Hence either $\mathcal{P}$ captures $\mathcal{E}$ in the L state or it switches to another (SG) state.

Now consider two consecutive (SG) states and suppose that $\mathcal{E}$ is not captured yet. According to the aforementioned lemmas, $\mathcal{P}$ achieves progress (P1) or (P2). Since in (P2), v and v' are vertices of Q, after at most n progress updates of type (P2), there would be one progress update of type (P1). Also since p_{ref} is a vertex, at some point $D \leq R(p_{\text{ref}})$. Since D is the diameter of the polygon, at some point $D = R(p_{\text{ref}})$. According to invariant (I1), we must have $D = R(p_{\text{ref}}) < R(\mathcal{E})$. This is a contradiction since $R(\mathcal{E})$ cannot be greater than the diameter.

Next let us provide an upper bound for the number of time-steps required for capture. Let T_1 be the time spent in the guard state plus the time spent in the search state (the time spent in the combined (SG) state). Also let T_2 be the number of steps for a pursuer, which is performing the extended lion's move, to travel the diameter of the polygon. Thus, the number of time-steps between two consecutive combined states (SG) would be T_1T_2. Since $p_{\text{ref}} \in Q$ and $v \in Q$, and we achieve (P1) or (P2) after each (SG) combined state, the total capture time T would be $T = n \cdot nT_1T_2$. According to [8] we have $T_2 = nD^2$. Next, the search time is $O(nD)$ (Lemma 4.3), and the guard time is bounded by $O(n^4D^{11})$ (Lemma 5.6). Hence $T_1 = n^4D^{11}$ and $T = O(n^2 \cdot (n^4D^{11}) \cdot (nD^2)) = O(n^7D^{13})$. □

7 Concluding Remarks

In this paper, we showed that a single deterministic pursuer with line-of-sight visibility can capture an evader whose speed is equal to the pursuer's in any monotone polygon. A general question regarding the lion-and-man game with visibility is the class of environments in which a single pursuer can capture the evader. Our result provides a step toward characterizing this class by showing that it includes monotone polygons. It turns out that if we slightly relax the monotonicity constraint and consider the class of weakly monotone polygons[9], capture is no longer guaranteed. Fig. 14 shows a weakly monotone polygon in which the evader can escape forever.

When a single pursuer is not enough, a natural question is to find the sufficient number of deterministic pursuers with line-of-sight visibility for a given polygon. In general, it is easy to see that if k pursuers can locate the evader, then $k + 1$ pursuers can capture it in any simply-connected polygon. This suggests studying a new version of the visibility based pursuit evasion game in which the evader is not arbitrarily fast but only as fast as the pursuers. Preliminary results on this problem were presented in [13].

Our result is also applicable to the continuous setting, when both the players move at the same time, as follows. The pursuer considers the continuous movement of the evader at discrete time steps with its specific time unit Δt. Then it plays the same turn-based pursuit strategy with respect to the location of evader at $t - \Delta t$. Recall that our capture condition in turn-based version is whether the distance between the players becomes less than the step-size. With the aforementioned modification to the continuous setting, the capture guarantee is that the pursuer will decrease its distance to the evader to at most twice the step-size. In our turn based strategy the time unit Δt can be chosen arbitrarily small. Consequently, the step-size can be arbitrarily small since the players' speed is fixed. Therefore, as long as the pursuer can change its step-size, our pursuit strategy guarantees that it can get arbitrarily close to the evader.

[9] A simply connected polygon is weakly monotone with respect to vertices s and t if the following hold. Consider a particle that walks from s to t along the boundary in clockwise and counterclockwise directions. If in each of these walks, the range of the directions that the particle sweep does not include the negative x-axis, the polygon is weakly monotone with respect to s, t and x [7].

Currently the only known lower bound on capture time for the lion and man game is $\Omega(D \log D)$ for the case of a circular environment and full visibility [1], and $O(nD)$ for the case of line-of-sight and general polygons [8]. One interesting question is whether the capture time for monotone polygons can be improved (perhaps using a randomized strategy).

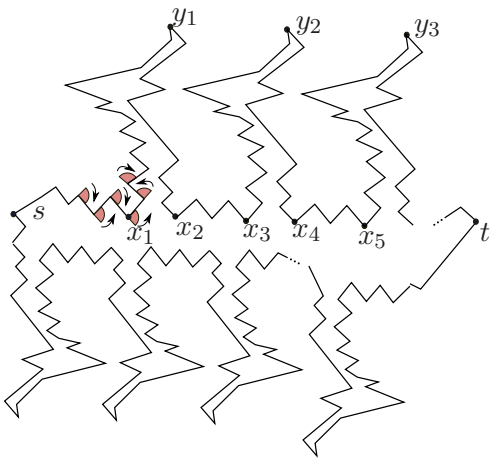

Figure 14: A weakly monotone polygon with respect to s and t. The upper chain that connects s to t is a repetition of the chain from s to y_4. The number of repetitions can be arbitrarily large. The chains from s to x_1, from x_2 to x_3, and from x_4 to x_5 are x-monotone chains. Also, the chains from x_1 to y_1, from y_1 to x_2, from x_3 to y_2 to x_4, from x_5 to y_3 are y-monotone chains. After disappearing from the pursuer's sight, the evader can hide in an upper or lower y-monotone polygon (whichever will be visited by the pursuer last) and escape when the evader is searching the other one.

8 Acknowledgements

The authors would like to thank Dr. Andrew Beveridge for helpful comments.

References

[1] L. Alonso, A. S. Goldstein, and E. M. Reingold. Lion and Man: Upper and lower bounds. *INFORMS Journal on Computing*, 4(4):447, 1992.

[2] D. Bhadauria and V. Isler. Capturing an evader in a polygonal environment with obstacles. In *Proc. International Joint Conference on Artificial Intelligence*, 2011.

[3] T. Chung, G. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics. *Autonomous Robots*, (3), 2011.

[4] M. De Berg, O. Cheong, and M. van Kreveld. *Computational geometry: algorithms and applications*. Springer, 2008.

[5] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[6] L. J. Guibas, J.-C. Latombe, S. M. Lavalle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *Internat. J. Comput. Geom. Appl.*, 9(4-5):471–493, 1999.

[7] P. J. Heffernan. Linear-time algorithms for weakly-monotone polygons. *Computational Geometry*, 3(3):121 – 137, 1993.

[8] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, 21(5):875–884, 2005.

[9] K. Klein and S. Suri. Complete information pursuit evasion in polygonal environments. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI)*, pages 1120–1125, 2011.

[10] J. E. Littlewood. *A mathematician's miscellany / J. E. Littlewood.* Methuen, London :, 1953.

[11] J. Sgall. Solution of david gale's lion and man problem. *Theor. Comput. Sci.*, 259:663–670, May 2001.

[12] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21(5):863–888, 1992.

[13] B. Tovar and S. M. LaValle. Visibility-based pursuit-evasion with bounded speed. *The International Journal of Robotics Research*, 27(11-12):1350–1360, November/December 2008.

9 Appendix

9.1 The simple pockets

The pocket $pocket(v, \vec{r})$ is called a *simple pocket* if its boundary except the entrance $\vec{r}$, is a single x-monotone chain and the angle between $\vec{r}$ and the y axis is smaller than $\frac{\pi}{2}$.

The pursuer by following the MPC strategy described in this paper, can force the evader to exit the pocket in order to prevent capture. The only difference is that during search it is sufficient to move along $\vec{r}$ in order to find $\mathcal{E}$. As the evader is found the pursuer starts the simple guard strategy presented in Sect. 5.2 i.e. it moves toward v along $\vec{r}$. Note that $\mathcal{E}$ cannot cross the segment between v and $\mathcal{P}$ because of the slope.

Lemma 9.1 (Simple Pockets)**.** *By following the MPC pursuit strategy on a simple pocket $pocket(v, \vec{r})$, after at most $O(n^2D^3)$ time-steps, $\mathcal{E}$ is forced to cross the entrance and exit the pocket in order to prevent being captured. At the crossing time, $\mathcal{P}$ and $\mathcal{E}$ both lie on $\vec{r}$ and $\mathcal{P}$ is in between v and $\mathcal{E}$, see Fig. 12-(d). The only difference is that during search the pursuer moves along $\vec{r}$.*

Proof. First observe that $\mathcal{P}$ will eventually stop performing the current state, which can be one of: *Search, Guard* or *Lion*, and switch to the next state. Suppose that the current state is *Guard*. Then the next state can be either *Lion* or *Search*.

In the former case, the pursuer gains lion's progress with respect to v until $\mathcal{E}$ is captured or $\mathcal{E}$ disappears behind a vertex v' in the new pocket $pocket(v', \overrightarrow{\mathcal{P}v'})$ or $\mathcal{E}$ exits the initial pocket.

In the latter case, the vertex v' which defines the new pocket $pocket(v', \overrightarrow{vv'})$ has the property that $x(v) < x(v')$. Hence the new pocket is a smaller one contained in the original pocket.

To complete the proof, note that once the evader hides behind a vertex, say v, it cannot disappear behind the vertex for the second time.

Since there are n vertices and the entrance is traversed twice, once during search and once during guard, the total time spent in the guard and search states would be $O(2nD)$. Together with $O(nD^2)$ required for extended lion's move progress, the total capture time would be $O((nD^2) \cdot (2nD)) = O(n^2D^3)$.

It remains to show that the entrance of all possible new pockets is positive and hence we can recursively use the simple pocket strategy. Clearly, all possible new pockets after the *Guard* state has positive slope since v' is inside $pocket(v, \vec{r})$.

Next, we show that all possible pockets after the lions move state must have positive slope. Since during lion's move, $\mathcal{P}$ lies on the edge $parent(\mathcal{E})\mathcal{E}$, the entrance $\overrightarrow{\mathcal{P}v'}$ is in direction of the tree edge $parent(v')v'$. See Fig. 15(a). Therefore, it is enough to show that in the shortest path tree rooted at v all edges have positive slope and hence if the evader disappears during *Lion's* state, the new pocket $pocket(v', \overrightarrow{\mathcal{P}v'})$ will have positive slope entrance (the entrance is $\overrightarrow{\mathcal{P}v'}$).

Suppose that there is an edge uw on the shortest path tree rooted at v with negative slope i.e. u is the parent of w in $\pi(v, w)$. We will show that there exists a path from v to w which is shorter than $\pi(v, w)$ which is a contradiction. See Fig. 15(c). Since the original pocket had a positive slope, there exists a vertex u' below vw such that w is visible to u'. Note that u' can be v itself. Also since w is visible to u, it must be that $x(u') < x(u)$ because otherwise u' would block the edge uw (note that slope of uw is negative and the slope of $u'w$ is positive). Then according to the triangulation property the shortest path from v to u' followed by the edge $u'w$ yields a shorter path than $\pi(v, w) = \pi(v, u) + uw$. A contradiction.

□

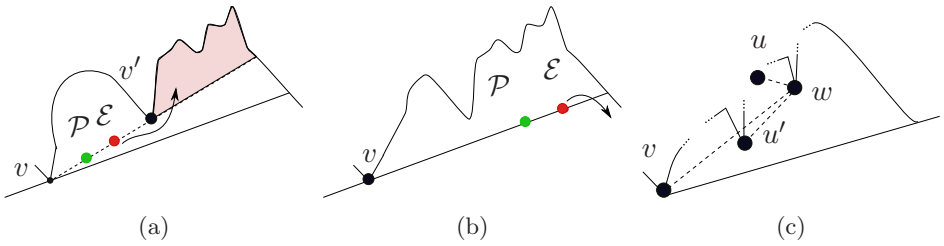

Figure 15: (a) The evader disappears into the shaded pocket. (b) Result of the simple pocket strategy. (c) Proof of Lemma 9.1.

9.2 Monotone Polygon Properties

Proof of Property 3.3. We present the proof for the 4^{th} type critical sub-polygons. The proof for other types is similar. Let $v = v_{i-1}$, see Fig. 5 and Fig. 16. For other possible v the proof is similar.

According to Lemma 9.2 the shortest path to all points to the right of $\overrightarrow{v_{i-2}v_{i-1}}$ passes from v_{i-1}. Hence the length of their shortest path is greater than $d(O_L, v_{i-1})$. Next consider the region in between $\vec{X}_{v_{i-1}}$ and $\overrightarrow{v_{i-2}v_{i-1}}$ and let p be a point in this region. As a corollary of Lemma 9.2 we observe that there is a vertex $v_c \in \Pi(v_{m-1}, (v_{i-1})$ that p is the region defined by two rays $\overrightarrow{v_cv_{c+1}}$ and $\overrightarrow{v_{c-1}v_c}$. Moreover p is a descendant of v_c. Next consider the line v_cp and its intersection point with $\vec{X}_{v_{i-1}}$ namely I_p. In the following we will show that $d(v_c, v_{i-1}) < v_cI_p$. Since $v_cI_p < v_cp \leq d(v_c, p)$ we would have $d(v_c, v_{i-1}) < d(v_c, p)$ and thus $d(O_L, v_{i-1}) < d(O_L, p)$.

Let us now present our proof for $d(v_c, v_{i-1}) < v_cI_p$, see Fig. 16 top-right. Consider the rays shot in direction of $v_{c'}v_{c'+1}$ where $c \leq c' \leq (i-3)$ and denote their intersection point with $\vec{X}_{v_{i-1}}$ by $I_{c'}$. By induction we will show that $v_{c'}I_{c'} + d(v_c, v_{c'}) = d(v_c, I_{c'}) < v_cI_p$.

For the base case consider $c' = c$. Since the angle $v_cI_cI_p$ is greater than 90 degrees we would have $v_cI_c < v_cI_p$.

For the inductive hypothesis let us suppose that the statement is true for c' and prove it for $(c'+1)$. According to our hypothesis we have $v_{c'}I_{c'} + d(v_c, v_{c'}) = d(v_c, I_{c'}) < v_cI_p$. Since the angle $v_{c'+1}I_{c'+1}I_{c'}$ is greater than 90 degrees we would have $v_{c'+1}I_{c'+1} < v_{c'+1}I_{c'}$. We also have $v_{c'}I_{c'} = v_{c'}v_{c'+1} + v_{c'+1}I_{c'}$.

Hence $d(v_c, I_{c'}) = v_{c'}I_{c'} + d(v_c, v_{c'}) = v_{c'}v_{c'+1} + v_{c'+1}I_{c'} + d(v_c, v_{c'}) = d(v_c, v_{c'+1}) + v_{c'+1}I_{c'}$. Since $v_{c'+1}I_{c'+1} < v_{c'+1}I_{c'}$ we would have $d(v_c, v_{c'+1}) + v_{c'+1}I_{c'+1} < d(v_c, I_{c'})$. Recall that $d(v_c, I_{c'}) < v_cI_p$. Thus $d(v_c, I_{c'+1}) < v_cI_p$.

□

Lemma 9.2. *Let (v_{e-1}, v_e) be an edge on Π. Consider the ray shot in direction of $\vec{r} = \overrightarrow{v_{e-1}v_e}$. Then the shortest path to all vertices to the right of $\vec{r}$ passes through the vertex v_e. See Fig. 16 left and middle.*

Proof. For the sake of contradiction suppose that there are some edges on Π that this property does not hold for them. Among these edges let (v_{e-1}, v_e) be the first one i.e. with smallest v_e. Thus there is a point to the right of $\vec{r} = \overrightarrow{v_{e-1}v_e}$ that the shortest path to that point does not pass from v_e. Then the direct parent of one of its ancestors should be a vertex to the left of $\vec{r}$. Let v be this ancestor and let $v' = parent(v)$. Let $(v_{e'-1}, v_{e'})$ be the edge on Π that the shortest path to v' passes from $v_{e'}$. Since v' is to the left of $\overrightarrow{v_{e-1}v_e}$ and since the property holds for all edges before (v_{e-1}, v_e) we would have $e' \leq (e-1)$. Next observe that $\overrightarrow{v_{e-1}v_e}$ intersects with $\pi(e', v')$. This can be seen by enumeration over all possible situations arisen depending on the type of the critical sub-polygon that (v_{e-1}, v_e) belongs to. For example the case where this edge is in the second type critical sub-polygon and before the summit vertex is depicted in Fig. 16 left and middle. In this case since slope of v_{e-1}, v_e is negative and slope of Π we observe that $\overrightarrow{v_{e-1}v_e}$ intersects with $\pi(e', v')$. Next let I_1 and I_2 be the intersection of $\overrightarrow{v_{e-1}v_e}$ with $v'v$ and $\pi(e', v')$ respectively. Then according to triangle inequality we have: $I_1I_2 < v'I_1 + d(v', I_2)$. Hence $vI_1, I_1I_2, \pi(e'I_2), \pi(O_L, e')$ is a shorter path than $\pi(O_L, v) = vI_1, v'I_1, v'I_2, I_2e', \pi(O_L, e')$ which is a contradiction.

□

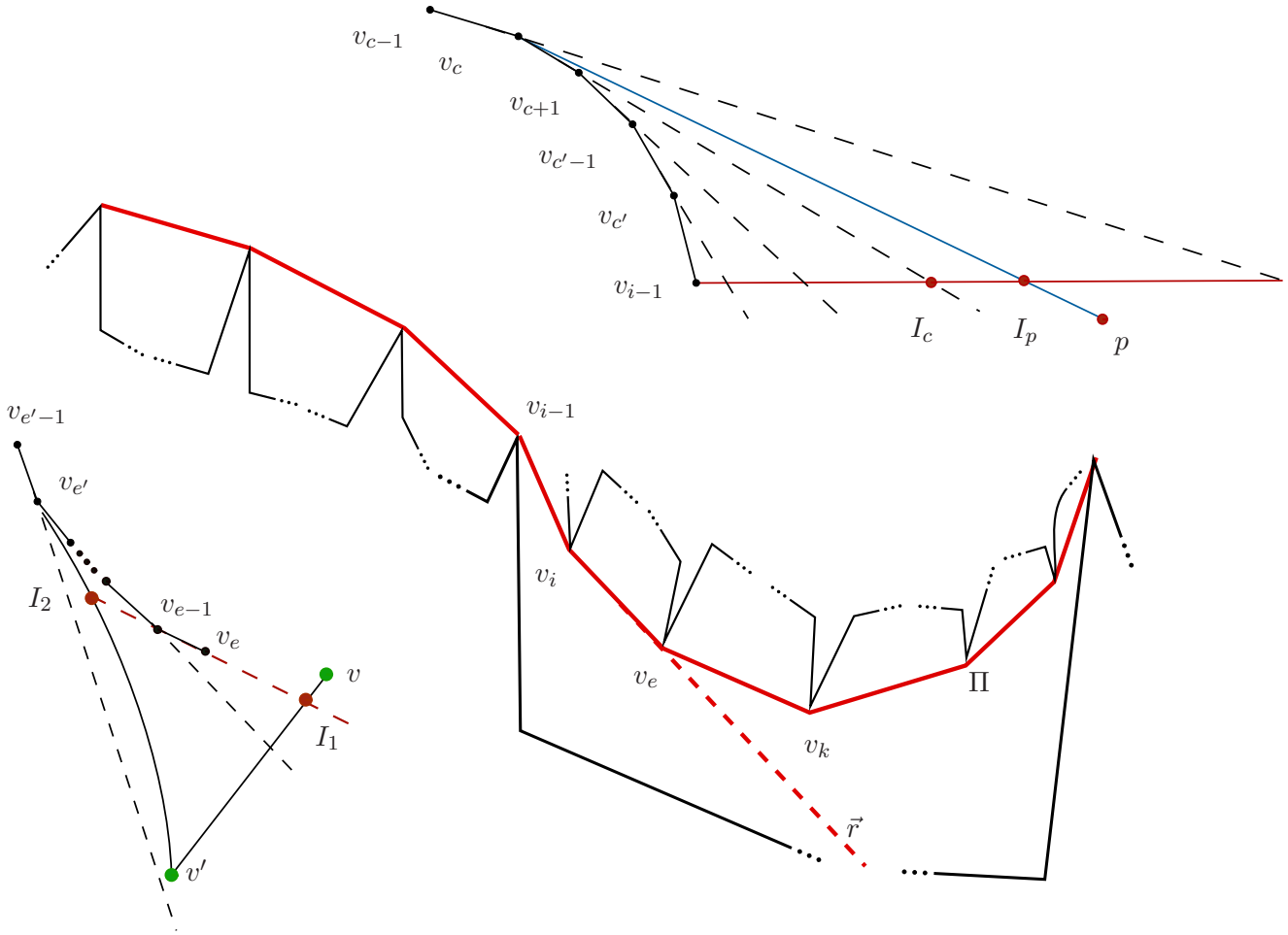

Figure 16: proof of Lemma 9.2 and Property 3.3.

Lemma 9.3. *Suppose $v_{i-1}v_i$ and $v_{j-1}v_j$ are two consecutive critical edges and consider $\pi(v_{i-1}, v_{j-1})$ which is also part of Π. Then (see Fig. 5-left):*

(i) If $v_i \in Chain_U$, then the slope of edges on $\pi(v_{i-1}, v_j)$ is monotonically increasing.

(ii) If $v_i \in Chain_L$, then the slope of edges on $\pi(v_{i-1}, v_j)$ is monotonically decreasing.

Proof. First, consider a segment wz and a point k with $x(z) < x(k)$, see Fig. 5:

1. k is below the ray $\overrightarrow{wz}$: then $slope(zk) < slope(wz)$.
2. k is above the ray $\overrightarrow{wz}$: then $slope(zk) > slope(wz)$.

Let w, z and k be three consecutive vertices on $\pi(v_{i-1}, v_j)$. Now, consider the case that $v_i \in Chain_U$. In this case, k *must* be above the ray wz because otherwise Π *cannot* be a shortest path [5]. Similarly, when $v_i \in Chain_L$, k *must* be below the ray wz. Therefore:

(i) when $v_i \in Chain_U$ the slope of edges is monotonically increasing. (ii) and, when $v_i \in Chain_L$ the slope of edges is monotonically decreasing.

□

Lemma 9.4. *Consider the search path inside the first type critical sub-polygon. Let p be a point on this part of the search path. Then the slope of the edge that connects p to $parent(p)$ is negative.*

Next consider the 2^{nd} type critical sub-polygons and suppose that p is in this portion of the search path. Then the slope of the edge that connects p to $parent(p)$ is positive.

Proof. First observe for all points p, $x(parent(p)) < x(p)$.

- p is in the first type critical sub-polygon: Note that all points in this part are descendants of v_{i-1}, see Lemma 9.2. For the sake of contradiction let us assume that $parent(p)$ is in the third quadrant of p, see Fig. 17. In the following we will show that there exist a shorter path than $\pi(O_L, p) = \pi(O_L, parent(p)) + parent(p)p$ which is a contradiction.

  Note that p is on the search path.

  1. p is on the α-path (Fig. 17(a)): Let A be the intersection of $\vec{X}_p$ with $\pi(O_L, parent(p))$. Observe that A is visible to p since all upper chain vertices are above the search path. Because of the triangulation inequality, pA followed by $\pi(O_L, A)$ is a shorter path than $\pi(O_L, p) = \pi(O_L, parent(p)) + parent(p)p$ which is a contradiction.
  2. p is on the step-path (Fig. 17(b)): Similar to the previous case.

- p is in the second type critical sub-polygon: The same as the previous case. See Fig. 17(c). Note that all points in this part are descendants of the summit vertex s, see Lemma 9.2. Also, A is visible to p since all lower chain vertices are below the search path. The rest of the proof is the same as above.

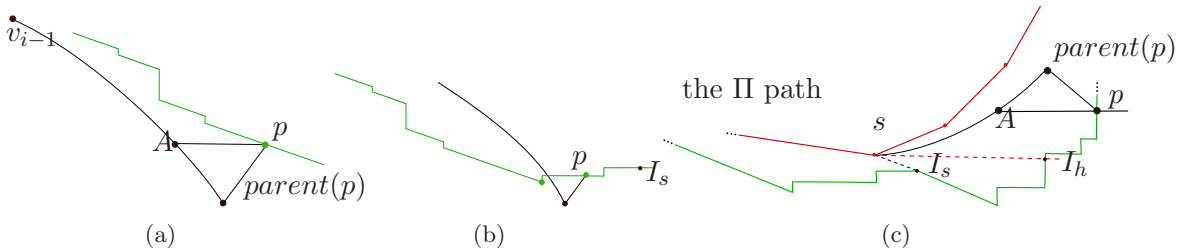

Figure 17: Proof of Lemma 9.4. The path Π is shown in dots.

□

9.3 Zig Zag Guard Correctness Proof

Lemma 9.5. *At the beginning of* Vertical Guard *invoked from* Zig-Zag Guard*, we have* $\sin\alpha \leq h = y(c_{\text{aux}}) - y(\mathcal{P})$.

Proof. Recall that the pursuer performs vertical guard while it is in the 1^{st} or the 3^{rd} type critical sub-polygons (see Section 5). In the following, we present the proof for the 1^{st} type.

Consider the preceding *Search*. Recall that during the search state $\mathcal{P}$ moves along the search path from v. The pursuer performs the zig-zag guard when $\mathcal{E}$ appears in its fourth quadrant (see Section 5). Recall that $c_{\text{aux}} = v$ (Section 5.3).

Suppose that $v \in Chain_U$.

First, suppose that $\mathcal{P}$ is on the step-path. Then observe that $\mathcal{E}$ cannot force $\mathcal{P}$ to retreat beyond v because the step that $\mathcal{P}$ lies on is after the floor point and moreover the evader is confined in the corresponding step. Hence only by following the zig-zag moves the pursuer will catch up to $\pi(O_L, \mathcal{E})$.

Next suppose that $\mathcal{P}$ is on the α-path. Let l_1 be the distance $\mathcal{P}$ has traveled from v. Thus $l_1 \sin\alpha$ is the minimum height $\mathcal{P}$ obtains during search. Moreover $1 - l_1$ is the residual move which $\mathcal{P}$ travels downward during the *Zig Zag Guard*. Thus the total height $\mathcal{P}$ obtains during search phase is at least $h_{min} = l_1 \sin\alpha + 1 - l_1 = l_1(\sin\alpha - 1) + 1$ which is at least $\sin\alpha$. Recall that vertical guard is invoked in the case that initially $\mathcal{P}$ is above $\pi(O_L, \mathcal{E})$ and the zig-zag strategy is to move downward or to the left. In other words $y(c_{\text{aux}}) - y(\mathcal{P})$ increases afterward and hence the lower bound $\sin\alpha$ remains valid.

The above argument is valid when $v \in Chain_L$. □

Lemma 9.6. *Suppose that the pursuer starts Search state (on pocket$(v, \vec{r})$) in the 1^{st} or the 3^{rd} type critical sub-polygons but it finds $\mathcal{E}$ while it is inside the 2^{nd} or the 4^{th} type critical sub-polygons. Recall that here independent of the quadrant that $\mathcal{E}$ is inside, the pursuer invokes the zig-zag guard (Section 5). Then the evader* cannot *force $\mathcal{P}$ to retreat beyond v. In other words, only by following the zig-zag moves, $\mathcal{P}$ will start the next state S or L.*

Proof. First, note that the portion of the polygon formed by two consecutive α lines or horizontal lines, ∂Q and the vertical lines is a triangle, see Section 4.

Note that we omitted presenting the zig-zag moves for the 2^{nd} or the 4^{th} types. Let us start by presenting the detailed description of the zig-zag moves that $\mathcal{P}$ takes toward $\pi(O_L, \mathcal{E})$. According to Lemma 9.4, the slope of $parent(\mathcal{P})\mathcal{P}$ is positive ($0 \leq slope$). Also, at the moment that $\mathcal{E}$ becomes visible $x(\mathcal{P}) < x(\mathcal{E})$ (Observation 4.1). Recall that the search path in these types is composed of only the step-path (section 4). We present the strategy by dividing each step into two parts. Also we only present the argument for the 2^{nd} type. See Fig. 18 and consider the step from A to D:

1. The segment uD: As a corollary of Lemma 9.4, all points on this segment are direct children of u. Therefore, we would have $parent(\mathcal{P}) = u$.
   (a) If $\mathcal{E}$ is inside the fourth quadrant of $\mathcal{P}$: See Fig. 18(a). Observe that the pocket formed by the segment uD and ∂Q from u to D is a simple pocket. By performing the simple pocket strategy presented in section 9.1, the evader is forced to cross uD while $\mathcal{P}$ is also on this segment. Note that uD is an edge of the shortest path tree. Note that all points on this segment are direct children of u. In other words, the next state will be L while $\mathcal{P}$ is in $h(v)$.
   (b) If $\mathcal{E}$ is inside the first quadrant of $\mathcal{P}$: See Fig. 18(b). Note that $\pi(O_L, \mathcal{E})$ has to be above $\mathcal{P}$. Then $\mathcal{P}$ moves along $-\vec{X}_{\mathcal{P}}$ toward $\pi(O_L, \mathcal{E})$. The L state or the S state will be established while $\mathcal{P}$ is in $hf(v)$.
2. From A to u: Note that $\mathcal{E}$ has to be in the first quadrant of $\mathcal{P}$ because otherwise $\mathcal{P}$ must have seen him sooner. The configuration that $\pi(O_L, \mathcal{E})$ is above $\mathcal{P}$, shown in Fig. 18(b), is similar to the case (b) above. The configuration that $\pi(O_L, \mathcal{E})$ is below $\mathcal{P}$, shown in Fig. 18(c), is as follows. The pursuer moves toward $\pi(O_L, \mathcal{E})$ along $\vec{X}_{\mathcal{P}}$. This ensures that $\mathcal{E}$ is in the first quadrant of $\mathcal{P}$ until $\mathcal{E}$ crosses $\vec{X}_{\mathcal{P}}$. At this time, $\mathcal{P}$ moves toward $\pi(O_L, \mathcal{E})$ along $-\vec{Y}_{\mathcal{P}}$, see Fig. 18(d). Note that $\mathcal{P}$ is becoming closer and closer to $\pi(O_L, \mathcal{E})$ while $\pi(O_L, \mathcal{E})$ is confined in the triangular region $\triangle ABu$. Hence the next state (L or S) will be established while $\mathcal{P}$ is in $h(v)$.

□

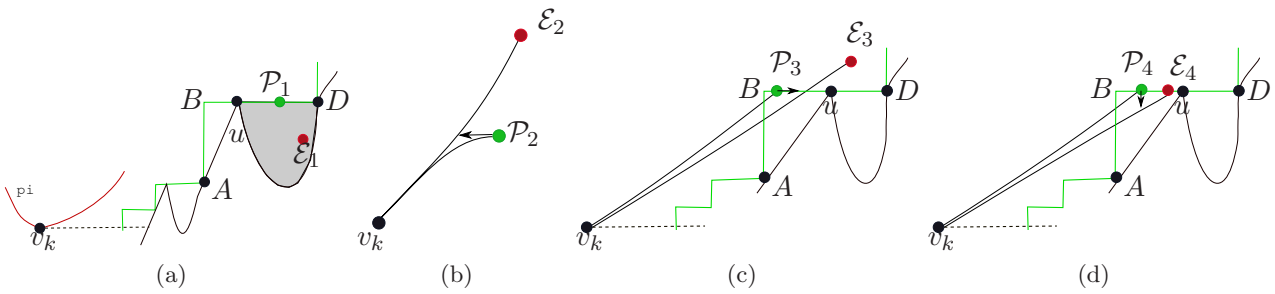

Figure 18: The zig-zag moves when $\mathcal{P}$ invokes zig-zag guard inside the 2^{nd} type critical sub-polygons while v in the preceding S state was in the 1^{st} type. See Lemma 9.6.

9.4 The Auxiliary Vertex in Simple Guard

In simple guard strategy, we define a local variable called the auxiliary vertex p_{aux} which is used as a landmark to guarantee progress. In simple guard state, the pursuer's goal is to prevent the evader from contaminating the region to the left of p_{aux}. In other words, the pursuer guarantees that the evader is inside $h(p_{\text{aux}})$. We define p_{aux} such that it is inside $h[p_{\text{ref}}]$. Therefore, at the end of this state the evader is inside $h(p_{\text{ref}})$.

Next let us present the selection of the vertex p_{aux}. Let $pocket(v, \vec{r})$ be the pocket which has been searched in the previous S state. Suppose that v_{aux} is the vertex that the α-path starts from (if $v \in Chain_U$, then $v_{\text{aux}} = v$). Moreover, let p_{ceil} be the ceiling point (refer to Section 4). Then:

- if p_0 is in the portion of the search path, from v to v_{aux}: See Fig. 19(a). Here p_{aux} is the bottommost vertex from the upper chain which is in the region in $h[p_{\text{ref}}]$ and to the left of the

segment p_0p_{ref}. Note that only when $v \in Chain_L$, we have $v_{\text{aux}} \neq v$ e.g. in Fig. 7(d) we have $v_{\text{aux}} = e_3$.

- If p_0 is in the portion of the search path from v_{aux} to the floor point: then p_{aux} is the first endpoint of the α-step that p_0 lies on it. For example, in Fig. 7(c), if p_0 is on the α-step defined from e_2 to I_2 then $p_{\text{aux}} = e_2$.
- If p_0 is in the portion of the search path after the floor point: See Fig. 19 parts (b) and (c). Let a be the intersection point between the α line passing through p_0 and Π. Suppose that w_1w_2 and $w_1'w_2'$ are the edges on Π such that $x(w_1) \leq x(p_{\text{ceil}}) < x(w_2)$ and $x(w_1') \leq x(a) < x(w_2')$ respectively. Then, if a is inside the next critical sub-polygon, p_{aux} is the second critical endpoint that defines the current critical sub-polygon. If $w_1 \neq w_1'$, i.e. a and p_{ceil} are not in between the endpoints of the same edge on Π, then $p_{\text{aux}} = w_1'$, see Fig. 19(b). Otherwise, p_{aux} is the bottommost vertex from the upper chain which is inside $h[p_{\text{ceil}}]$ and to the left of the line that connects p_{ceil} to p_0, see Fig. 19(c).

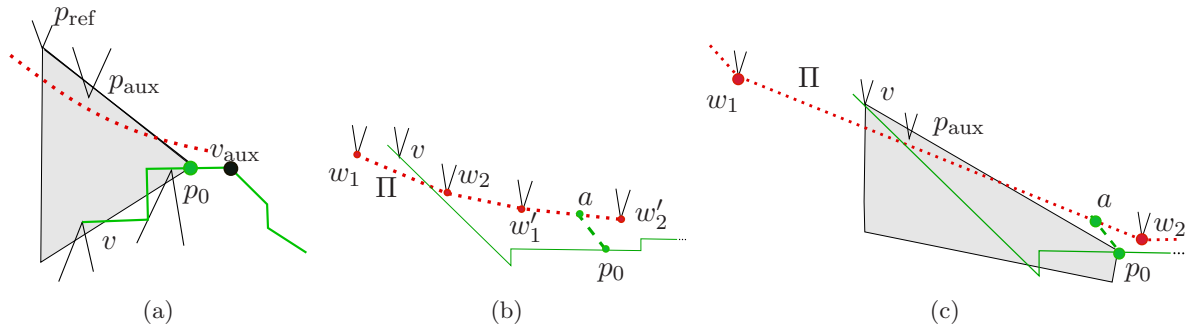

Figure 19: The path Π is shown in dots. Note that all upper chain vertices are above Π. (a) here p_{aux} is the bottommost vertex from $Chain_U$ in the shaded region. (b) here $p_{\text{aux}} = w_1'$ ($p_{\text{ceil}} = v$). (c) p_{aux} is the bottommost vertex from $Chain_U$ in the shaded region ($p_{\text{ceil}} = v$).

9.5 Simple Guard Correctness Proof

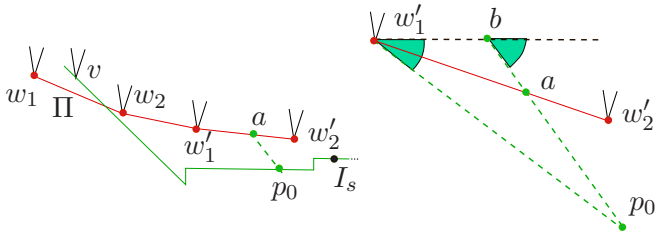

Figure 20: The point p_{aux} when $w_1 \neq w_1'$. Here $p_{\text{aux}} = w_1'$.

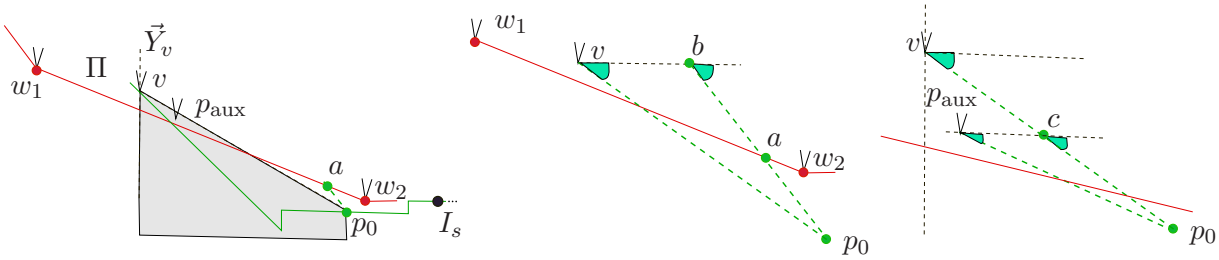

Figure 21: The point p_{aux} when $w_1 = w_1'$.

Lemma 9.7. *Suppose that the pursuer is in simple guard state and p_0 be the location of the pursuer at the beginning of the state. Then p_{aux} defined in section 5.2 is visible to p_0 and moreover, the angle built by p_0p_{aux} and the x-axis is less than α.*

Proof. Let $pocket(v, \vec{r})$ be the pocket being searched in the previous search state.

1. $v \in Chain_U$: here p_{aux} are defined based on the second and the third definition in section 5.2.
   - If p_0 is in the portion of the search path from v_{aux} to the floor point: since both p_0 and p_{aux} are on the α line of the corresponding α-step, the slope of p_0p_{aux} is equal to $-\alpha$.
   - If p_0 is in the portion of the search path after the floor point: first suppose that a and p_{ceil} are not in between the endpoints of the same edge on Π. Then $p_{\text{aux}} = w'_1$. See Fig. 20-right. Note that all upper chain vertices are above Π and all lower chain vertices are below the search path. Hence p_{aux} is visible to p_0. Next, let b be the intersection between p_0a and the horizontal line passing through w'_1. Note that the angle between this horizontal line and p_0b is equal to α. Also, observe that w'_1 is the left of p_0b. Hence, considering the triangle $\triangle p_0bw'_1$, it can be concluded that the angle between p_0p_{aux} and the x-axis must be less than α.
     Next, suppose that a and p_{ceil} are in between the endpoints of the same edge on Π. Observe that p_{aux} is visible to p_0 as follows. This is because p_{aux} is the bottommost upper chain vertex in the shaded region, and moreover the slope of the edge $w'_1w'_2$ is negative (Lemma 9.3), and all upper chain vertices are above Π and all lower chain vertices are below the search path. See Fig. 21. Next consider the angle between p_0p_{ceil} and the x-axis. Considering the triangle $\triangle p_0p_{\text{ceil}}b$ and the fact that the angle between p_0a and the x-axis is equal to α and p_{ceil} is to the left of p_0a, we can conclude that the angle between p_0p_{ceil} and the x-axis is smaller than α, see Fig. 21-middle. Now observe that p_{aux} is in the triangular region formed by p_0p_{ceil}, the edge $w'_1w'_2$, and $\vec{Y}_{p_{\text{ceil}}}$. Let c be the intersection between p_0p_{ceil} and the horizontal line passing through p_{aux}, see Fig. 21-right. Then, since the angle between this horizontal line and p_0c is smaller than α and considering the triangle $\triangle p_0p_{\text{aux}}c$, we conclude that the angle between p_0p_{aux} and the x-axis is smaller than α.
2. $v \in Chain_L$:
   (a) The slope of $\vec{r}$ is negative: recall that when $\mathcal{E}$ appears inside the first quadrant of $\mathcal{P}$, the pursuer performs the simple guard (section 5.2). Also, recall that when $v \in Chain_L$ the search path starts by the step-path (section 4). Since the slope of $\vec{r}$ is negative, the evader can appear in the first quadrant of $\mathcal{P}$ only when p_0 is after v_{aux}. Similar to the above case where $v \in Chain_U$, it can be shown that the angle between p_0p_{aux} and the x-axis is smaller than α.
   (b) The slope of $\vec{r}$ is positive: let us refer to the current simple guard state as G_2 and its corresponding search state which is on $pocket(v, \vec{r})$ as S_2. Let us refer to the state before S_2 as $state_{prev}$. Then $state_{prev}$ must be a simple guard. This is because during zig-zag guard, the evader remains inside the fourth quadrant of $\mathcal{P}$ and hence the resulting pocket ($pocket(v, \vec{r})$) would have negative slope. Also, if the previous state was L the resulting pocket would have negative slope (a corollary of Lemma 9.2). Let $state_{prev} = G_1$ which is a simple guard. Also let p'_{aux} be the auxiliary point defined in G_1, and p' be the location of $\mathcal{P}$ at the beginning of G_1. Therefore, the sequence of states is $G_1S_2G_2$, the pursuer is moving toward p'_{aux} along $p'p'_{\text{aux}}$ during G_1, and v is to the right of the line $p'p'_{\text{aux}}$. Recall that at the beginning of S_2, we set $p_{\text{ref}} = p'_{\text{aux}}$ (Lemma 5.2).
   Now consider p_{aux} defined in G_2 (section 5.2). If the second or the third definition applies, the proof is similar to the above cases. Hence suppose that p_{aux} is defined according to the first definition (i.e. p_0 is in between v and v_{aux}).
   We continue by an inductive argument as follows. Suppose that the angle between $p'p'_{\text{aux}}$ and the x-axis (the absolute value) is equal to or less than α. Since $p_{\text{ref}} = p'_{\text{aux}}$, and v is to the right of $p'p'_{\text{aux}}$ the slope of vp_{ref} is less than α. Recall that the search state in between v and v_{aux} is increasing in the x-coordinate and decreasing in the y coordinate, see section 4. Therefore, the slope of p_0p_{ref} is also less than α. See Fig. 22(a).
   Since p_{aux} is defined as the bottommost vertex in $h[p_{\text{ref}}]$ which is also to the left of p_0p_{ref}, it can be shown that the slope of p_0p_{aux} is less than α (Fig.21-right).

It remains to show that p_{aux} is visible to p_0. According to our inductive argument, $p_{\text{ref}} = p'_{\text{aux}}$ is visible to p'. For the sake of contradiction, let us assume that p_{aux} is not visible to p_0 and hence is blocked by a vertex namely v_b. We must have $v_b \in Chain_L$. Also it must that $x(p') < x(v_b) < x(v)$. See Fig. 22(b). Let p_g be the position of $\mathcal{P}$ during G_1 at which the evader the evader disappears behind v. Note that p_g is on the segment $p'p'_{\text{aux}}$ and moreover v is visible to p_g. Since all lower chain vertices are below the path formed by $p'_{\text{aux}}pg$, p_gv and the search path between v and p_0, the vertex v_b cannot block p_0p_{aux}. Contradiction.

□

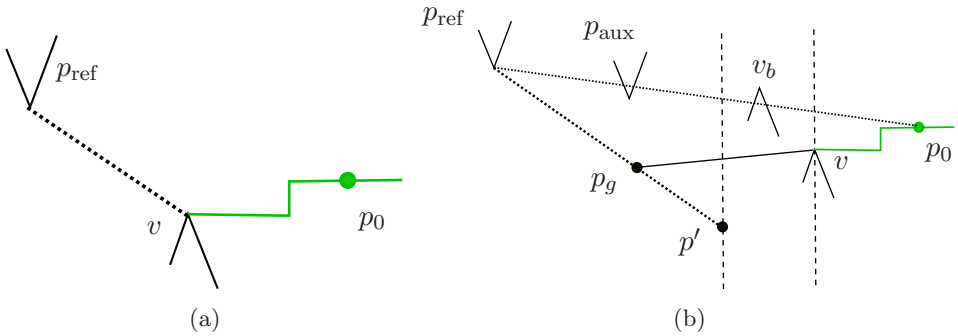

Figure 22: Proof of Lemma 9.7. (a) the angle p_0p_{aux} is less than α. (b) the vertex v_b has to be below the line p_gv.

Lemma 9.8. *Suppose that $\mathcal{P}$ is in the simple guard sub-state, see section 5.2. While the pursuer is moving back to p_{aux} the evader cannot cross the segment p_0p_{aux}.*

Proof. Refer to Fig. 23-(a) let p_0 and $\mathcal{E}_0$ be the position of the players at the beginning of the *Simple Guard*. Hence $x(p_0) < x(\mathcal{E}_0)$ Observation 4.1. We will show that for all points A on the line segment between p_{aux} to p_0, the length of p_0A is smaller than the length of the shortest path from $\mathcal{E}_0$ to A minus *one*. Hence if $\mathcal{E}$ tries to cross the p_0p_{aux} at A is will be captured by $\mathcal{P}$. Specifically we show that $Ap_0 - A\mathcal{E}_0 \leq 1$.

First observe that the angle $p_{\text{aux}}p_0$ is equal to or smaller than α, see Lemma 9.7.

Let H be the point on Y_{p_0} where AH is perpendicular to Y_{p_0}.

1. $\alpha = \psi_1$: we have:

$$\begin{aligned}
&\cos\alpha = (1 - 1/D^2)^{0.5} \\
&Ap_0 \leq D, \quad 1 \leq D \\
&Ap_0(1 - \cos\alpha) \leq D(1 - (1 - 1/D^2)^{0.5}) = \\
&D - (D^2 - 1)^{0.5} \leq 1
\end{aligned}$$

2. $\alpha = \psi_2 < \psi_1$: we have:

$$\begin{aligned}
&\cos\psi_1 < \cos\psi_2, \quad -\cos\psi_2 = -\cos\alpha < -\cos\psi_1, \\
&Ap_0(1 - \cos\alpha) < Ap_0(1 - \cos\psi_1) \leq 1
\end{aligned} \tag{1}$$

□

Lemma 9.9. *At the beginning of* Vertical Guard *invoked from* Simple Guard *we would have* $\sin\alpha \leq h = y(c_{\_aux}) - y(\mathcal{P})$.

Proof. Recall that here $c\_aux = p_{\text{aux}}$. In the following, the key observation is that the angle between $p_{\text{aux}}p$ and the x-axis is smaller than α, see Lemma 9.7.

Note that $\mathcal{E}$ has to cross the line $p_{\text{aux}}p$ before crossing l_v. From $p_{\text{aux}}p$ to l_v the pursuer follows $\mathcal{E}$ by lion's move with respect to p_{aux}. Now consider the time-step that $\mathcal{P}$ and $\mathcal{E}$ are on $p_{\text{aux}}p$. Let $\mathcal{P}'$ and $\mathcal{E}'$ be the position of the pursuer and the evader after one step of the lion's move with respect to p_{aux}. According to Lemma 9.10 the distance from all points on $p_{\text{aux}}p$ to all points on l_v is equal or greater than one. Therefore $\mathcal{E}'$ is a point in between $p_{\text{aux}}p$ and l_v see 23-(b).

1. $1 \leq p_{\text{aux}}\mathcal{P}$: Because of the lion's progress $p_{\text{aux}}\mathcal{P} < p_{\text{aux}}\mathcal{P}'$. Hence $1 \leq p_{\text{aux}}\mathcal{P}'$. Moreover $\mathcal{P}'$ is in between the $p_{\text{aux}}p$ and l_v. Thus $\sin\alpha \leq h = y(p_{\text{aux}}) - y(\mathcal{P}')$ see Fig. 23-(c).
2. $p_{\text{aux}}\mathcal{P} \leq 1$: Let $A = \vec{X}_{\mathcal{P}} \cap \vec{Y}_{\mathcal{P}'}$.
   (a) $\mathcal{P}'$ is outside the unit circle: similar to the first case we have $\sin\alpha \leq h$
   (b) $x(\mathcal{P}') < x(\mathcal{P})$, see Fig. 23-($e$) and ($f$) for $\mathcal{P}'_2$:

$$\begin{aligned} &h = A\mathcal{P}'_2, \qquad \frac{A\mathcal{P}}{p_{\text{aux}}\mathcal{P}} < \cos\alpha, \\ &p_{\text{aux}}\mathcal{P} < 1, \qquad A\mathcal{P} \leq p_{\text{aux}}\mathcal{P} \cdot \cos\alpha < \cos\alpha \\ &A\mathcal{P} < \cos\alpha, \qquad h^2 = 1 - A\mathcal{P}^2 \\ &\sin\alpha < h \end{aligned} \tag{2}$$

   (c) $x(\mathcal{P}) < x(\mathcal{P}')$, see Fig. 23-($d$) and ($g$) for $\mathcal{P}'_1$:

$$\begin{aligned} &\beta \leq \frac{\pi}{2} - \alpha, \quad \sin\alpha \leq \cos\beta, \quad \mathcal{P}\mathcal{P}'_1 = 1, \quad h = \cos\beta, \\ &\sin\alpha < h \end{aligned} \tag{3}$$

□

Lemma 9.10. *Let p_{aux} be a point inside Q. Consider the α-line passing through p_{aux}, the unit circle centered at p_{aux}, and the vertical line $l_v = \vec{Y}_{p_{\text{aux}}}$. For all points e above the α-line and all points e' on l_v in which e and e' are outside the unit circle we would have $1 \leq ee'$.*

Proof. Refer to Fig. 23-(b) note that $ee' = \tan\theta_1 + \tan\theta_2$ where $\theta_1 + \theta_2 = \pi - \alpha$. The function $\tan\theta_1 + \tan\theta_2 - 1$ is positive for α angles equal or smaller than $\psi_2 = (\frac{\pi}{2} - 2\arctan\frac{1}{2})$. Hence $1 \leq ee'$ for α which is equal or smaller than ψ_2. See Definition 4.2. □

9.6 Horizontal Guard

In this section, we present the *horizontal guard* strategy which is the counterpart of vertical guard (section 5.3) in the second type critical sub-polygon. Suppose that the horizontal guard has been invoked in the 2^{nd} type. Recall that at this time $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ and $y(\mathcal{P}) = y(\mathcal{E}) = y(v)$ (Lemma 5.1). The center c for the horizontal guard is found as follows: Let I be the intersection between ∂Q and $\vec{Y}_v$. Then c is the intersection point between bisector of $\mathcal{P}I$ and $\vec{X}_v$. See Fig. 24(b). Symmetric to what we saw in vertical guard, the pursuer performs lion's move with respect to c or v as the evader moves below $l_h = \vec{X}_v$ or above $l_h = \vec{X}_v$. This continues until the next state is established in $h(v)$.

9.7 The Vertical Guard and the Horizontal Guard Correctness Proof

Lemma 9.11. *The vertical guard circle centered at c prevents $\mathcal{E}$ from escaping to upper chain vertices which are to the left of p_{ref}.*

Proof. See Lemma 9.14. □

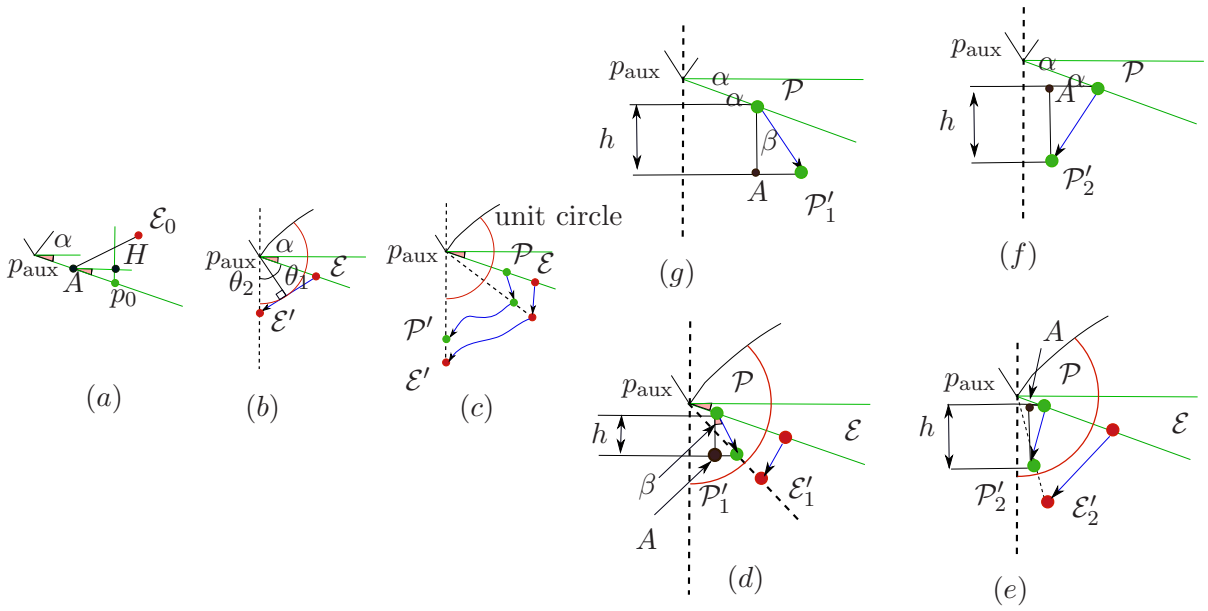

Figure 23: $\mathcal{E}$ cannot cross $p_{\text{aux}}\mathcal{P}$ during simple guard

Lemma 9.12. *If during vertical guard strategy $\mathcal{E}$ disappears behind a vertex to the left of l_v, then the resulting pocket would be a simple pocket. Also refer to Fig. 12 parts (b) and (c).*

Proof. The vertex that defines the pocket must be from the lower chain (Lemma 9.11). In this figure, because of monotonicity the pocket in the left is impossible and hence the pocket must be an simple pocket (middle part of the figure). □

Lemma 9.13. Feasibility of the Horizontal Guard. *Suppose that $\mathcal{P}$ invokes the horizontal guard sub-state. Then, the radius of the circle centered at c is finite (i.e. upper bounded) and the pursuer can perform lion's move with respect to c.*

Proof. Recall that the pursuer performs the horizontal guard strategy when: (1) the previous search state was on $pocket(v, \vec{r})$ where v is inside the 2^{nd} or the 4^{th} type critical sub-polygon, and (2) $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ and $y(\mathcal{P}) = y(\mathcal{E}) = y(v)$ (Lemma 5.1).

Note that $x(v) \leq x(\mathcal{P}) < x(\mathcal{E})$ and $y(\mathcal{P}) = y(\mathcal{E}) = y(v)$ can occur only when the evader appears while $\mathcal{P}$ is on the horizontal line passing through v (i.e. the first step) (see Fig. 24). If $\mathcal{P}$ is on other steps, similar to Lemma 9.6 we can show that the zig-zag moves are sufficient.

Also note that since the slope of the entrance $\vec{r}$ is positive, at the beginning of G state (end of the S state) the evader would be in the first quadrant of $\mathcal{P}$.

Similar to Lemma 5.3, it can shown that the radius of the circle centered at c is upper bound if we could provide a lower bound for $x(\mathcal{P}) - x(v)$. Note that $1 \leq x(\mathcal{P}) - x(v)$. This is because during search $\mathcal{P}$ is moving in the direction of the x-axis (the horizontal line passing through v which is on the first step). Even if the evader appears as $\mathcal{P}$ moves for $\epsilon < 1$ (during the one time unit of the S state), the pursuer immediately switches to the G state and moves for the residual move toward $\pi(O_L, \mathcal{E})$ (the residual move is $(1 - \epsilon)$). Therefore at the time that $\mathcal{E}$ crosses $\vec{X}_v$, we would have $1 \leq x(\mathcal{P}) - x(v)$. □

Lemma 9.14. Feasibility of the Vertical Guard. *Consider the vertical guard state, section 5.3. Then all upper chain vertices before c_{aux} are above $c_{\text{aux}}I$. Therefore, $\mathcal{P}$ can perform the lion's move with respect to c_{aux} and c. In other words, the next location that $\mathcal{P}$ must move to according to the lion's move is in the free space.*

Proof. Note that the vertical guard state can be invoked from zig-zag guard, simple guard. Therefore, $c_{\text{aux}} = v$ (in case of zig-zag guard) or $c_{\text{aux}} = p_{\text{aux}}$ (in case of simple guard).

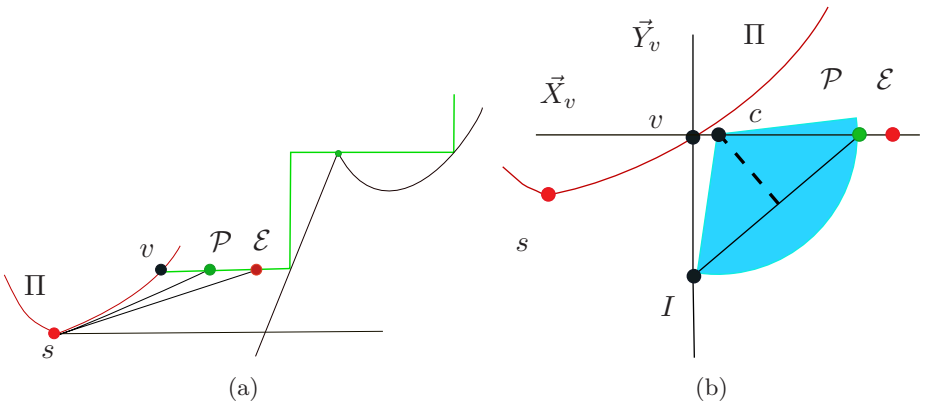

Figure 24: (a) The configuration that $\mathcal{P}$ performs the horizontal guard strategy. (b) The horizontal guard circle centered at c.

In the following, we present the proof by arguing the 1^{st} and the 2^{nd} type critical sub-polygons. The other two types are symmetric.

1. $c_{\text{aux}} = v$ (i.e. the vertical guard is invoked from zig-zag guard): Here, v must be a vertex inside the 1^{st} type and moreover at the beginning of the G state $\mathcal{P}$ must be inside the same critical sub-polygon (see section 5 and Lemma 9.6). Also note that $v \in Chain_U$. To see this suppose that $v \in Chain_L$. Recall that the search path on $pocket(v, \vec{r})$ starts by the step-path, then continues along the α-path and then the step-path (section 4). Let I_1 be the intersection of the first step-path with ∂Q. Recall that $\mathcal{P}$ performs the zig-zag guard when $\mathcal{E}$ appears inside its fourth quadrant (section 5.1). Suppose that $\mathcal{P}$ is in between v and I_1. According to zig-zag guard, the pursuer moves downward and to the left. Hence the evader cannot force the pursuer to retreat beyond v (the players will hit ∂Q). When $\mathcal{P}$ is after I_1 the similar result is valid.

   Therefore only when $v \in Chain_U$, the pursuer invokes the vertical guard during zig-zag guard. In the following we prove that all upper chain vertices before v are above vI. Let us refer to the current G state as G_2 and the S state before it (which is on $pocket(v, \vec{r})$) as S_2. Also, let $state_{prev}$ be the state before S_2. Hence the sequence of states is $state_{prev}S_2G_2$.

   (a) $state_{prev} = L$: Let w_1w_2 be the edge on Π so that $x(w_1) \leq x(v) < x(w_2)$. Then v must be in the fourth quadrant of w_1. Because otherwise $pocket(v, \vec{r})$ would be a simple pocket and $\mathcal{P}$ can recover the L state by following the simple pocket strategy presented in section 9.1. Since the slope of edges on Π before w_1 is negative (Lemma 9.3) it must be that all upper chain vertices before v are above vI.

   (b) $state_{prev} = G$: Let us denote this G state as G_1 and its previous S state as S_1. Hence the sequence of states is $S_1G_1S_2G_2$. Also suppose that S_1 is on $pocket(v_1, \vec{r_1})$.

      i. G_1 is zig-zag guard and $v_1 \in Chain_U$: We continue by an inductive argument as follows. Suppose that for all invokes to the S state before S_2 all upper chain vertices are above the corresponding horizontal line. In other words, all upper chain vertices before v_1 (in S_1) which are before v_1 are above the horizontal line passing through v_1. Note that during S_1, the pursuer moves downward and to the right. During G_1, which is a zig-zag guard, the pursuer moves downward and to the left and moreover the evader remains inside the fourth quadrant of the pursuer. See Fig. 25(a). Hence all upper chain vertices are above the horizontal line passing through v.
      Now we prove the property for the first invoke to G. Therefore, the sequence of states is $S_2G_2...$ where S_2 is the first search state which is done on $pocket(v, \vec{r})$. Observe that the first time that the pursuer invokes S in a critical sub-polygon, the vertex $v \in \Pi$. Even if the sequence of states is $LS_2G_2...$, either $pocket(v, \vec{r})$ is a simple pocket which $\mathcal{P}$ can recover the L state by following the simple pocket strategy, or the property must hold for v.

      ii. G_1 is zig-zag guard and $v_1 \in Chain_L$: Likewise, since all upper chain vertices are above the search path and the blocking vertex v is inside the fourth quadrant of $\mathcal{P}$'s location

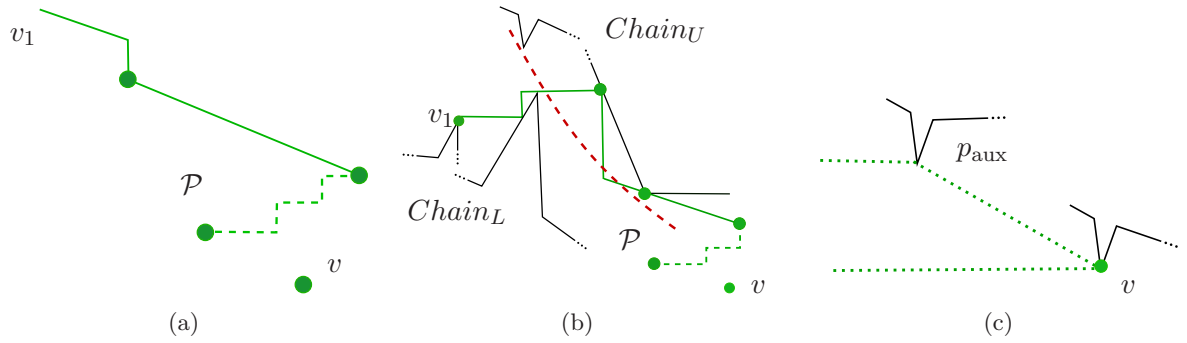

Figure 25: All upper chain vertices are above vI. The search path on v_1 is shown in green and the path traversed during G_1 is shown dash green lines. (a) $v_1 \in Chain_U$. Here G_1 is a zig-zag guard. (b) $v_1 \in Chain_L$. Here G_1 is a zig-zag guard. (c) Here G_1 is a simple guard.

during G_1. See Fig. 25(b).

iii. G_1 is simple guard: Recall that $v \in Chain_U$. According to the simple guard the entrance of $pocket(v, \vec{r})$ is $\vec{r} = p_{\text{aux}}v$ where p_{aux} is the auxiliary reference point defined in G_1. Note that v must be inside the fourth quadrant of p_{aux} because otherwise $pocket(v, \vec{r})$ would be a simple pocket and by following the simple pocket strategy (section 9.1) the pursuer will recover the simple guard strategy.
In the following we show that in simple guard all upper chain vertices before p_{aux} are above the horizontal line passing through p_{aux}. Therefore all upper chain vertices before v are above the horizontal line passing through v. See Fig. 25(c).

(c) $c_{\text{aux}} = p_{\text{aux}}$ (i.e. the vertical guard is invoked from simple guard): By an inductive argument we show that all upper chain vertices before p_{ref} are above the horizontal line passing through p_{ref}. Using this we show that all upper chain vertices before p_{aux} are above its corresponding horizontal line.
In our inductive argument we also use the result obtained above: all upper chain vertices are above v if $v \in Chain_U$ and v is defining $pocket(v, \vec{r})$ being searched in an S state.
Suppose that the property holds for the current p_{ref}. Then referring to section 5.2, the point p_{aux} is defined based on the location of $\mathcal{P}$ at the beginning of the simple guard. Let us denote the current guard state as G_2 and the previous search state as S_2 which is performed on $pocket(v, \vec{r})$. Also recall that p_0 is the location of $\mathcal{P}$ at the beginning of G_2 (section 5.2).

- If p_0 is in the portion of the search path, from v to v_{aux}: Recall that here $v \in Chain_L$ and p_{aux} is the bottommost vertex from the upper chain which is in the region in $h[p_{\text{ref}}]$ and to the left of the ray p_0p_{ref}. See Fig. 26(a). Since p_{aux} is the bottommost vertex in this region and all upper chain vertices are above the horizontal line passing through p_{ref}, the property also holds for p_{aux}. Also recall that we update p_{ref} to p_{aux} at the end of the simple guard (Lemma 5.2). Therefore, the property is valid for the next p_{ref} point.
- If p_0 is in the portion of the search path from v_{aux} to the floor point: First, suppose that $v \in Chain_L$. Since all upper chain vertices are above the search path, the property holds for p_{aux}. Next suppose that $v \in Chain_U$. Since the y coordinate of the points on the search path is decreasing and all upper chain vertices are above the search path and the property holds for v, we conclude that the property also holds for p_{aux}.
- If p_0 is in the portion of the search path after the floor point: Recall that if $w_1 \neq w_1'$, i.e. a and p_{ceil} are not in between the endpoints of the same edge on Π, then $p_{\text{aux}} = w_1'$, see Fig. 26(b). Note that since $w_1' \in \Pi$, the property holds for $p_{\text{aux}} = w_1'$.
If $w_1 = w_1'$, i.e. a and p_{ceil} are in between the endpoints of the same edge on Π, then p_{aux} is the bottommost vertex from the upper chain which is inside $h[p_{\text{ceil}}]$ and to the left of the line connecting p_{ceil} to p_0, see Fig. 26(c). Here, v could be from the lower chain or the upper chain. In the case that $v \in Chain_L$, since all upper chain vertices are above the search path the property will be concluded. When $v \in Chain_U$, since the property is valid for v and p_{aux} is the bottommost vertex, the property is concluded.

□

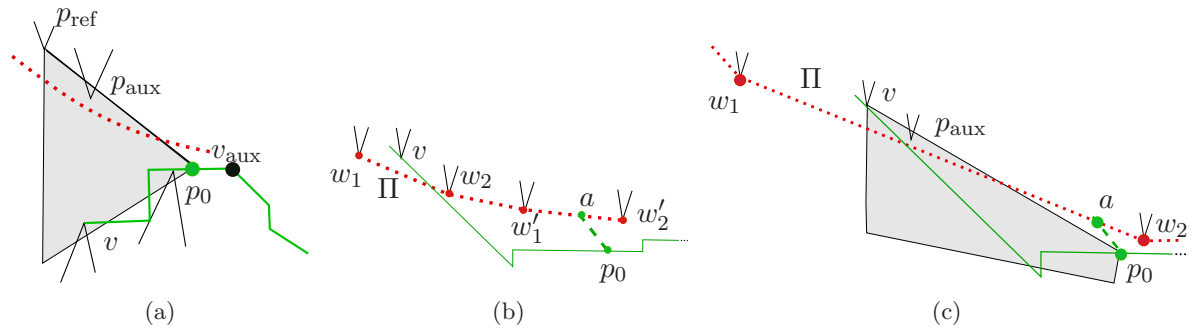

Figure 26: Fig. 19 shown for convenience. The path Π is shown in dots. Note that all upper chain vertices are above Π. (a) Here p_{aux} is the bottommost vertex from $Chain_U$ in the shaded region. (b) Here $p_{\text{aux}} = w_1'$ ($p_{\text{ceil}} = v$). (c) p_{aux} is the bottommost vertex from $Chain_U$ in the shaded region ($p_{\text{ceil}} = v$).