

■ UNIVERSITÄT DUISBURG-ESSEN  
FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN  
ABTEILUNG INFORMATIK UND ANGEWANDTE KOGNITIONSWISSENSCHAFT

Master's Thesis

**Infinite Games: Algorithms and Reductions**

Maxime Nederkorn  
Matrikelnummer: 3004376



Department of Computer Science and  
Applied Cognitive Science  
Faculty of Engineering  
University of Duisburg-Essen

31. January 2022

**Examiners:**  
Prof. Dr. Barbara König  
Prof. Dr. Janis Voigtländer  
  
**Advisor:**  
Richard Eggert

# Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Definitions</b>	<b>3</b>
2.1. Directed Graphs . . . . .	3
2.2. Arenas . . . . .	3
2.3. Positions, Moves . . . . .	3
2.4. Strategies . . . . .	4
2.5. Plays . . . . .	4
2.6. Infinite Games . . . . .	4
2.6.1. Parity Games . . . . .	4
2.6.2. Mean Payoff Games . . . . .	4
2.6.3. Energy Games . . . . .	5
2.6.4. Discounted Payoff Games . . . . .	5
2.7. Simple Stochastic Games . . . . .	5
2.8. Reductions . . . . .	6
<b>3. Reductions and Solutions in Theory</b>	<b>7</b>
3.1. Reductions in Theory . . . . .	7
3.1.1. PGs to MPG <sub>s</sub> . . . . .	7
3.1.2. MPG <sub>s</sub> to DPG <sub>s</sub> . . . . .	7
3.1.3. MPG <sub>s</sub> to EG <sub>s</sub> . . . . .	7
3.1.4. DPG <sub>s</sub> to SSG <sub>s</sub> . . . . .	7
3.2. Solutions in Theory . . . . .	7
3.2.1. Value Iteration . . . . .	7
3.2.2. Strategy Iteration . . . . .	7
3.2.3. PG <sub>s</sub> . . . . .	7
3.2.4. MPG <sub>s</sub> . . . . .	7
3.2.5. DPG <sub>s</sub> . . . . .	7
3.2.6. EG <sub>s</sub> . . . . .	7
3.2.7. SSG <sub>s</sub> . . . . .	7
<b>4. Reductions and Solutions in Practise</b>	<b>7</b>
4.1. Reductions in Practise . . . . .	7
4.2. Solutions in Practise . . . . .	7
<b>5. Implementation</b>	<b>7</b>
<b>6. Evaluation</b>	<b>7</b>
<b>7. Conclusion and further approaches</b>	<b>7</b>
<b>A. Versicherung an Eides Statt</b>	<b>8</b>

## 1. Introduction

## 2. Definitions

Foundational to our task are the different kinds of *Infinite Games* and how to determine the outcome of each such game. To do so, we also want a notion of *strategies* on such Infinite Games. To later reduce the games to one another, we furthermore want a definition of a *Reduction* in the computational complexity sense.

### 2.1. Directed Graphs

Let  $V$  be a finite set of Vertices, let  $E \subseteq V \times V$  be a set of Edges, then  $G = (V, E)$  is a *Directed Graph*. We also define  $\text{src}: E \rightarrow V$  as  $\text{src}((u, v)) = u$  and  $\text{tgt}: E \rightarrow V$  as  $\text{tgt}((u, v)) = v$ .

### 2.2. Arenas

An arena is an extension of Directed Graphs, where the set of Vertices,  $V$ , is partitioned into two disjunct subsets  $V_0$  and  $V_1$ , respectively denoting the regions where player 0, also represented by  $\square$ , and player 1, also represented by  $\circlearrowleft$ , are to play. We also require that the out-degree of every vertex is at least one, so that any play on the Arena can always be prolonged.

Formally, let  $(V, E)$  be a non-trivial Directed Graph,  $V_0 \cup V_1 = V$ ,  $V_0 \cap V_1 = \emptyset$  be a partition of  $V$  and  $\forall v \in V: \exists e \in E: \text{src}(e) = v$ , then  $A = (V, (V_0, V_1), E)$  is an Arena.

#### Example 1:

In Fig. 1: An Arena  $A = ((v_0, v_1, v_2, v_3, v_4), ((v_0, v_1), (v_2, v_3, v_4)), ((v_0, v_1), (v_0, v_3), (v_1, v_2), (v_2, v_1), (v_2, v_4), (v_4, v_4), (v_1, v_3), (v_3, v_3)))$

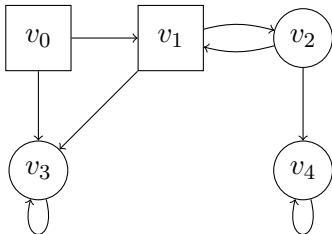


Figure 1: test

### 2.3. Positions, Moves

A *position*,  $\pi_i = (v_0, v_1, \dots, v_i)$ , in a Game describes a finite path on the underlying graph, i.e. a position is an element of  $V^+$ . E.g. in Fig. 1 starting at  $v_0$ ,  $(v_0, v_1, v_2, v_4)$  could be a play.

A *move* is an extension of a position by one more step in the graph. E.g. in Fig. 1  $(v_0, v_1, v_2) \mapsto (v_0, v_1, v_2, v_4)$  could be move. Player P as in  $v_i \in V_P \in (V_0, V_1)$  chooses the move.

## 2.4. Strategies

A *strategy*,  $V^* \times V_P \rightarrow V$ , is a function by which player  $P$  chooses the next move for any given position.

We call a strategy *memoryless* if for any given position the next move only depends on the last vertex of the position, i.e.  $V_P \rightarrow V$ . We will refer to memoryless strategies of player 0 as  $\sigma$  and of player 1 as  $\tau$ .

## 2.5. Plays

A *play* describes the path of arbitrary length  $\langle v_0, v_1, \dots \rangle$  the player go through in the process of playing the game. We refer to the play generated by applying strategies  $\sigma, \tau$  to game  $G$  starting at  $v_0 \in V$  as  $\pi_{\sigma, \tau}(G, v_0) = \langle v_0, v_1, \dots \rangle$

## 2.6. Infinite Games

Infinite Games are a category of games played by two players on a finite, directed graph. They are infinite in the sense that we require the out-degree of every vertex to be at least one. As such, regardless of the strategies chosen by the players, they never terminate.

### 2.6.1. Parity Games

Parity Games are played by two players, *Even* or player 0 ( $\square$ ) and *Odd* or player 1 ( $\circlearrowleft$ ). A Parity Game,  $PG = (A, p)$ , is played on an Arena  $A$  with a priority function  $p: V \rightarrow \{0, 1, \dots, |V|\}$ .

Let  $\pi_{\sigma, \tau}(PG, v_0) = \langle v_0, v_1, \dots \rangle$  be the *play* resulting from applying the strategies  $\sigma$  and  $\tau$  to Parity Game  $PG$ .

Let  $\#_\infty(\pi_{\sigma, \tau}(PG, v_0)) = \{i \in \langle p(v_0), p(v_1), \dots \rangle \mid \forall j \in \mathbb{N}: j < |\langle v \in \langle v_0, v_1, \dots \rangle \mid p(v) = i \rangle|\}$  be the set of priorities that appear arbitrarily often in the play. If  $\max(\#_\infty(\pi_{\sigma, \tau}(PG, v_0)))$  is Even, then *Even* wins and vice versa.

### 2.6.2. Mean Payoff Games

Mean Payoff Games are played by two players, *Max* or player 0 ( $\square$ ) and *Min* or player 1 ( $\circlearrowright$ ).

A Mean Payoff Game,  $MPG = (A, \nu, d, w)$ , is played on an Arena  $A$ , with threshold  $\nu \in \mathbb{Z}$  and an edge-weight function  $w: E \rightarrow \{-d, \dots, -1, 0, 1, \dots, d\}$ ,  $d \in \mathbb{N}_0$ .

*Max* wins play  $\pi_{\sigma, \tau}(MPG, v_0) = \langle v_0, v_1, \dots \rangle$  if

$$\liminf_{n \rightarrow \infty} \left( \frac{1}{n} \sum_{i=0}^{n-1} w((v_i, v_{i+1})) \right) \geq \nu.$$

### 2.6.3. Energy Games

Energy Games are played by two players, *Charging* or player 0 ( $\square$ ) and *Depleting* or player 1 ( $\circlearrowleft$ ).

An Energy Game,  $EG = (A, c, d, w)$ , is played on an Arena  $A$ , with credit  $c \in \mathbb{N}_0$  and an edge-weight function  $w: E \rightarrow \{-d, \dots, -1, 0, 1, \dots, d\}$ ,  $d \in \mathbb{N}_0$ .

*Charging* wins play  $\pi_{\sigma,\tau}(EG, v_0) = \langle v_0, v_1, \dots \rangle$  if

$$\forall k \in \mathbb{N}_0: \left( \sum_{i=0}^k w((v_i, v_{i+1})) \right) + c \geq 0.$$

For any given position  $\langle v_0, v_1, \dots, v_k \rangle$  in a play we call  $\left( \sum_{i=0}^{k-1} w((v_i, v_{i+1})) \right) + c$  the *energy level* at that position. Keep in mind that *Charging* doesn't necessarily aim to maximise their energy level at *any* specific position but rather to maintain a non-negative energy level at *every* position. In a sense, they aim to minimise the  $c$  necessary to maintain the winning condition for any given position of a play.

### 2.6.4. Discounted Payoff Games

Discounted Payoff Games are played by two players, *Max* or player 0 ( $\square$ ) and *Min* or player 1 ( $\circlearrowleft$ ).

A Discounted Payoff Game,  $DPG = (A, \nu, d, w, \lambda)$ , is played on an Arena  $A$ , with threshold  $\nu \in \mathbb{Z}$ , an edge-weight function  $w: E \rightarrow \{-d, \dots, -1, 0, 1, \dots, d\}$ ,  $d \in \mathbb{N}_0$  and a discount factor  $0 < \lambda < 1$ .

*Max* wins play  $\pi_{\sigma,\tau}(DPG, v_0) = \langle v_0, v_1, \dots \rangle$  if

$$(1 - \lambda) \left( \sum_{i=0}^{\infty} \lambda^i \cdot w((v_i, v_{i+1})) \right) \geq \nu.$$

## 2.7. Simple Stochastic Games

Simple Stochastic Games are played by two players, *Max* or player 0 ( $\square$ ) and *Min* or player 1 ( $\circlearrowleft$ ).

A Simple Stochastic Game,  $SSG = (G, (V_0, V_1, V_2), V_-, V_+, p)$ , is played on a Directed Graph  $G$ , with a partition  $(V_0, V_1, V_2)$ , two sink vertices  $V_-$ ,  $V_+$  and a probability function  $p: E \ni (V_2 \times V) \rightarrow (0, 1]$ . We require that the probabilities of all outgoing edges of each  $V_2$  vertex sum to 1:

$$\forall v \in V_2: \left( \sum_{e \in E: \text{src}(e)=v} p(e) \right) = 1$$

We also require that the out-degree of every vertex is at least one, with the exception of  $V_0, V_1$ , for which it is zero. We say that a SSG is *stopping* if for every possible position in a play there is a path to one of the sink vertices.

*Max* wins if the  $V_+$  sink is reached, *Min* wins if the  $V_-$  sink is reached or the game doesn't terminate. Since the result of a play  $\pi_{\sigma,\tau}(SSG, v_0)$  can be probabilistic, it is

assigned a probability to reach the  $V_+$  sink rather than a fixed value. Since SSGs can terminate, they are not Infinite Games. For simplicity, we henceforth still colloquially include SSG under the label *Infinite Games*. Note that plays of non-stopping SSGs may still be arbitrarily long.

## 2.8. Reductions

A *reduction* describes an algorithm with which we can transform (reduce) one class of problems to another. For our purposes, we choose target problem classes that are already solved. Aside from reducing to an already solved problem class and therefore, per definition, also solving the original problem class, reducing and subsequent solving of instances of the original problem may be faster than any attempt to directly solve the original problem without an intermediary reduction.

Formally, we express our problem classes as formal languages  $A$  and  $B$  over the alphabets  $\Sigma^*$  and  $\Gamma^*$ . If  $f$  is totally computable and

$$f: \Sigma^* \rightarrow \Gamma^*, A \subseteq \Sigma^*, B \subseteq \Gamma^* \\ \forall w \in \Sigma^*: w \in A \iff f(w) \in B$$

then  $f$  is a reduction from  $\Sigma^*$  to  $\Gamma^*$ .

In practice our formal languages  $A$  and  $B$  will be some instances of types of infinite games together with statements about such games, such as the values under perfect play or perfect strategies. The reduction will then draw an equivalence to the other infinite game, e.g. if  $w = "v(v_u) = 5 \text{ in } G"$  is a word in  $A$  then  $f(w) = "v(v_v) = 16 \text{ in } H"$  is a word in  $B$ .

### **3. Reductions and Solutions in Theory**

#### **3.1. Reductions in Theory**

**3.1.1. PGs to MPG<sub>s</sub>**

**3.1.2. MPG<sub>s</sub> to DPG<sub>s</sub>**

**3.1.3. MPG<sub>s</sub> to EG<sub>s</sub>**

**3.1.4. DPG<sub>s</sub> to SSG<sub>s</sub>**

#### **3.2. Solutions in Theory**

**3.2.1. Value Iteration**

**3.2.2. Strategy Iteration**

**3.2.3. PG<sub>s</sub>**

**3.2.4. MPG<sub>s</sub>**

**3.2.5. DPG<sub>s</sub>**

**3.2.6. EG<sub>s</sub>**

**3.2.7. SSG<sub>s</sub>**

### **4. Reductions and Solutions in Practise**

#### **4.1. Reductions in Practise**

#### **4.2. Solutions in Practise**

### **5. Implementation**

### **6. Evaluation**

### **7. Conclusion and further approaches**

## **A. Versicherung an Eides Statt**

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit - mit Ausnahme der Anleitung durch die Betreuer - selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides Statt nach § 156 und nach § 163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

---

Ort, Datum

Unterschrift