

Church Synthesis Problem for Noisy Input

Yaron Velner and Alexander Rabinovich

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract. We study two variants of infinite games with imperfect information. In the first variant, in each round player-1 may decide to hide his move from player-2. This captures situations where the input signal is subject to fluctuations (noises), and every error in the input signal can be detected by the controller. In the second variant, all of player-1 moves are visible to player-2; however, after the game ends, player-1 may change some of his moves. This captures situations where the input signal is subject to fluctuations; however, the controller cannot detect errors in the input signal.

We consider several cases, according to the amount of errors allowed in the input signal: a fixed number of errors, finitely many errors and the case where the rate of errors is bounded by a threshold. For each of these cases we consider games with regular and mean-payoff winning conditions. We investigate the decidability of these games.

There is a natural reduction for some of these games to (perfect information) multidimensional mean-payoff games recently considered in [6]. However, the decidability of the winner of multidimensional mean-payoff games was stated as an open question. We prove its decidability and provide tight complexity bounds.

1 Introduction

The algorithmic theory of infinite games is a powerful and flexible framework for the design of reactive systems (see e.g., [11]). It is well known for instance, that the construction of a controller acting indefinitely within its environment amounts to the computation of a winning strategy in an infinite game. For the case of regular games, algorithmic solutions of the synthesis problem have been developed, providing methods for the automatic construction of controllers. The basis of this approach is the Büchi-Landweber Theorem, which states that in a regular infinite game, i.e., a game over a finite game graph with a winning condition given by an ω -regular language, a finite state winning strategy for the winner can be constructed [3]. Much work has been devoted to the generalizations and extensions of this fundamental result. One well known extension are mean-payoff games where the winning condition is an ω language recognized by an automaton with a *mean-payoff acceptance condition*. These games have been studied since the end of the seventies [10, 15] and still attract a large interest. Another well known extension are games with imperfect information. In most of the previous work, the setting of an imperfect information game is given by a game graph with a coloring of the state space that defines equivalence

classes of indistinguishable states called observations [8, 13], and the strategies are observation-based (i.e., they rely on the past sequence of observations rather than on states).

In the present paper we investigate *games with errors* which are different kind of imperfect information games.

To present games with errors it is convenient to refer to the simplest format of infinite games, also called Gale-Stewart games. In such game we abstract from graphs but just let the two players choose letters from a finite alphabets Σ_1, Σ_2 in turn. A play is built up a sequence $(\begin{smallmatrix} b_0 \\ a_0 \end{smallmatrix})(\begin{smallmatrix} b_1 \\ a_1 \end{smallmatrix})(\begin{smallmatrix} b_2 \\ a_2 \end{smallmatrix}) \dots \in (\Sigma_1 \times \Sigma_2)^\omega$. A natural view is to consider the sequence $a = a_0a_1\dots$ as the input stream and the sequence $b = b_0b_1\dots$ as the output stream. In the Gale-Stewart game, the play is won by player-2 if the ω word $(\begin{smallmatrix} b_0 \\ a_0 \end{smallmatrix})(\begin{smallmatrix} b_1 \\ a_1 \end{smallmatrix})(\begin{smallmatrix} b_2 \\ a_2 \end{smallmatrix})\dots$ satisfies the winning condition, i.e., if it belongs to a given *specification* language $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$.

We consider a *game with detected errors* where in each round of the game, player-1 has the possibility to hide his move from the opponent; however, player-2 can detect whether player-1 hides a move. Player-1 needs to decide on the value of the hidden moves, only at the end of the game. Hence, at the end of the game he replaces his hidden moves in the produced play ρ by letters from Σ_1 and this defines an *interpretation* $Int(\rho) \in (\Sigma_1 \times \Sigma_2)^\omega$ for the play ρ . Round $i \in \mathbb{N}$ has an *error* if player-1 decides to hide his move in round i . This game captures the cases where the input signal is noisy and the controller can detect errors in the input signal. We also consider a *game with undetected errors* where in each round player-2 is fully aware of player-1 moves; however, at the end of the game player-1 can change some of his moves, i.e., player-1 provides an *interpretation* $Int(\rho) \in (\Sigma_1 \times \Sigma_2)^\omega$ for the play ρ . An *error* is made in round $i \in \mathbb{N}$ if at the end of the game player-1 decides to change his move in round i . Note that in this game, an error is determined according to the interpretation of the play. This game captures the cases where the input signal is noisy and the controller cannot detect errors in the input signal.

We measure the amount of errors in a play according to two scales. The *error count* scale counts the number of errors in a play (the result is in $\mathbb{N} \cup \{\infty\}$). The *error rate* of a play ρ is $\limsup_{n \rightarrow \infty} \frac{1}{n} \cdot (\text{number of errors in first } n \text{ rounds})$.

In both games a limitation on the amount of errors allowed for player-1 is given by one of the following conditions types. The first type of conditions is a bound $n \in \mathbb{N}$ on the error count. The second type of conditions requires that the error count of a play is finite. The third type of conditions is a bound $\delta \in \mathbb{Q}$ on the error rate. The last type seems to be the most interesting for real life applications.

For a specification language L : Player-1 is the winner of a play with detected errors if the play satisfies the amount of errors limitation and there exists an interpretation $Int \notin L$. Player-1 is the winner of a play with undetected errors if there exists an interpretation $Int \notin L$ which satisfies the amount of errors limitation.

We consider the cases where the specification language is either an ω regular language or when the language is recognizable by a mean-payoff automaton.

We investigate the decidability of who is the winner, and the computability of the winning strategies. In addition, we investigate the *bounded number of errors* problem which asks if there exists $n \in \mathbb{N}$ such that player-1 is the winner of a game when he allowed to do at most n errors.

Table 1 summarizes our decidability results for games with detected and undetected errors with regular and mean-payoff winning condition.

	Bounded ¹		Finite ²		Rate ³ $\delta = 0$		Rate ³ $\delta \in (0, 1)$		Rate ³ $\delta = 1$	
	Regular	MP	Regular	MP	Regular	MP	Regular	MP	Regular	MP
Detected	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Undetected	?	?	✓	?	?	?	✗	✗	✓	✗

Table 1. ✓ - Decidable. ✗ - Undecidable. ? - Open.

We reduced games with detected errors and a mean-payoff winning condition to *multidimensional mean-payoff games*.

A multidimensional mean-payoff game, introduced in [6], is played on a finite weighted graph by two players. The edges of the graphs are labeled by k -dimensional vectors $w \in \mathbb{Z}^k$. The game is played as follows. A pebble is placed on a designated initial vertex of the game graph $G = (V, E)$. The game is played in rounds in which the player owning the position where the pebble lies moves the pebble to an adjacent position of the graph using an outgoing edge. An infinite play results an infinite path $\pi = e_0 e_1 \dots \in E^\omega$ through the game graph. The energy level vector of the finite path $e_0 e_1 \dots e_{n-1}$ is $EL(e_0 e_1 \dots e_{n-1}) = \sum_{i=1}^{n-1} w(e_i)$. The infinite path π produces two mean-payoff vectors. $\underline{MP}(\pi) = \liminf_{n \rightarrow \infty} \frac{1}{n} \cdot EL(e_0 e_1 \dots e_{n-1})$ and $\overline{MP}(\pi) = \limsup_{n \rightarrow \infty} \frac{1}{n} \cdot EL(e_0 e_1 \dots e_{n-1})$. Hence, the vector associated to a play ρ which induces an infinite path π_ρ is the $2k$ dimensional vector $\overrightarrow{MP}(\rho) = (\underline{MP}(\pi_\rho), \overline{MP}(\pi_\rho))$. The winning condition for player-2 is given by a threshold $\nu = \langle \nu_1, \dots, \nu_{2k} \rangle \in \mathbb{Q}^{2k}$, which induces $2k$ boolean variables $x_i^\nu \doteq \overrightarrow{MP}(\rho)_i \geq \nu_i$, and by a boolean formula on x_i^ν (for $i = 1, \dots, 2k$).

In [6] the players were restricted to use only finite state strategies. The play π produced by finite state strategies is always quasi periodic and $\overrightarrow{MP}(\pi) = \underline{MP}(\pi)$.

It was proved in [6] that when the players are restricted to use the finite state strategies and the winning condition is a conjunction of the form $\bigwedge_{i=1}^{2k} x_i^\nu$, then it is decidable who is the winner. The decidability of who is the winner when players are allowed to use arbitrary strategies was stated as an open question. We prove that this problem is decidable and provide tight bounds on its complexity. The lower complexity bounds are easily derived from [6, 2].

We investigate the case where the winning condition is given by the formula $\varphi_{\bigwedge \text{MeanPayoffInf}^\geq(\nu)} \doteq \bigwedge_{i=1}^k x_i^\nu$ or by the formula $\varphi_{\bigwedge \text{MeanPayoffSup}^\geq(\nu)} \doteq \bigwedge_{i=k+1}^{2k} x_i^\nu$. We show that the problem who wins $\varphi_{\bigwedge \text{MeanPayoffInf}^\geq(\nu)}$ games is Strongly coNP complete, while there exists a pseudo-polynomial algorithm which determines the winner for a condition of the form $\varphi_{\bigwedge \text{MeanPayoffSup}^\geq(\nu)}$.

¹ Decidability of the bounded number of errors problem

² Games with finitely many errors

³ Games with the errors rate limitation δ

We also consider the case where the winning condition is given by a positive boolean formula which depends only on the $\underline{MP}(\pi_\rho)$ vector or on the $\overline{MP}(\pi_\rho)$ vector, and prove decidability for this case.

This paper is organized as follows. In the next section we introduce notations and well known definitions. In section 3, we provide formal definitions of games with errors. In sections 4-5, we investigate regular games with errors. In section 6, we investigate mean-payoff games with errors. In section 7, we investigate multidimensional mean-payoff games and complete the proof of the main theorem of section 6. Due to lack of space, only the outlines of the proofs are presented.

2 Definitions

Words and languages. Let Σ be a finite *alphabet*. A *finite word* over Σ is $w = \sigma_0\sigma_1\dots\sigma_n$, for $n \in \mathbb{N}$ and $\sigma_i \in \Sigma$. An *infinite word* over Σ is $w = \sigma_0\sigma_1\dots\sigma_n\dots$. Let $w = \sigma_0\sigma_1\dots\sigma_n\dots$, we denote σ_i by $w(i)$. For $j > i \geq 0$, $w[i, j] = w(i)w(i+1)\dots w(j-1)w(j)$, $w[0, n-1]$ is a *prefix* of length n of w . $w[i, \infty]$ is a *suffix* of w starting from position i . For every $i \geq 0$, $w[i, i-1]$ is defined to be the *empty word* ϵ . We denote by Σ^* the set of all finite words, and by Σ^ω the set of all infinite words. A *language* is a subset of Σ^* , and an ω *language* is a subset of Σ^ω . In the sequel, when it is clear from the context, we shall also refer to an ω language $L \subseteq \Sigma^\omega$ as a language. A word over $\Sigma = \Sigma_1 \times \Sigma_2$ is $w = (\sigma_2(0))(\sigma_2(1))\dots(\sigma_2(n))\dots$. We denote $(\sigma_2(i))$ by $w(i)$, $\sigma_1(i)$ by $w(i)_1$ and $\sigma_2(i)$ by $w(i)_2$, for $j = 1, 2$ we denote $\sigma_j(0)\sigma_j(1)\dots$ by w_j .

Gale-Stewart game. A *Gale-Stewart game* is a two-player game of perfect information. The game is defined using an alphabet $\Sigma = \Sigma_1 \times \Sigma_2$ and a language $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$, and is denoted by G_L . The two players alternate turns, and each player is aware of all moves before making the next one. A *play* of game has ω rounds. At round $i \in \mathbb{N}$: first player-1 chooses $\sigma_1 \in \Sigma_1$, then player-2 chooses $\sigma_2 \in \Sigma_2$. At the end of the play, an ω word $w = (\sigma_2(0))(\sigma_2(1))\dots$ is formed. Player-2 wins the play if $w \in L$.

Strategies. A player-1 *strategy* is $\tau : (\Sigma_1 \times \Sigma_2)^* \rightarrow \Sigma_1$. A play $\rho = (\sigma_2(0))(\sigma_2(1))\dots$ is consistent with player-1 strategy τ if $\tau((\sigma_2(0))(\sigma_2(1))\dots(\sigma_2(i))) = \sigma_1(i+1)$ for every $i \in \mathbb{N}$. A player-2 strategy is $\tau : (\Sigma_1 \times \Sigma_2)^* \times \Sigma_1 \rightarrow \Sigma_2$. The consistency of a play with player-2 strategy is defined similarly. Player- j (for $j = 1, 2$) is the *winner* of game G_L if it has a strategy τ such that in every play consistent with τ , player- j wins.

Automaton. An *automaton* over Σ is a tuple $\mathcal{A} = (\Sigma, Q, Q_0, E)$ Where Σ is a finite alphabet, Q is a finite set of *states*, $Q_0 \subseteq Q$ is a set of *initial states* and $E \subseteq Q \times Q \times \Sigma$ is a *transition relation*. Automaton \mathcal{A} is *deterministic* if $|Q_0| = 1$ and $\forall q \in Q, \forall \sigma \in \Sigma, \forall q_1, q_2 \in Q$ if $(q, q_1, \sigma) \in E$ and $(q, q_2, \sigma) \in E$ then $q_1 = q_2$. For infinite word $\rho = \sigma_1\sigma_2\dots$ a *run* of ρ is an infinite sequence $\pi = q_0e_0q_1e_1\dots$ with $q_0 \in Q_0$ and $e_i = (q_i, q_{i+1}, \rho(i)) \in E$ for every $i \in \mathbb{N}$. The set of all states reachable by a finite word ρ is $E^*(\rho) = \{q \in Q \mid \rho \text{ has a (finite) run that ends in } q\}$.

Language recognized by automaton. Let $\mathcal{A} = (\Sigma, Q, Q_0, E)$. An *acceptance condition* for \mathcal{A} is a set $\phi \subseteq (Q \times E)^\omega$, note that we are interested only in the cases where ϕ has a finite description. An infinite word is *accepted* by automaton \mathcal{A} and condition ϕ if it has a run π such that $\pi \in \phi$. The set of all ω words accepted by automaton \mathcal{A} and condition ϕ is denoted by $L_{\mathcal{A},\phi}$. In the sequel, when the acceptance condition ϕ it is clear from the context, we shall omit it and denote the language by $L_{\mathcal{A}}$.

Acceptance conditions (objectives). Let $p : Q \rightarrow \mathbb{N}$ be a *priority function* and $w : E \rightarrow \mathbb{Z}$ be a *weight function*. The *energy level* of a finite path $\gamma = q_0e_0q_1e_1\dots e_{n-1}q_n$ is $EL(w, \gamma) = \sum_{i=0}^{n-1} w(e_i)$, and the *mean-payoff value* of an infinite path $\pi = q_0e_0q_1e_1\dots$ is defined by either $\overline{MP}(w, \pi) = \limsup_{n \rightarrow \infty} \frac{1}{n} \cdot EL(w, \pi[0, n-1])$ or by $\underline{MP}(w, \pi) = \liminf_{n \rightarrow \infty} \frac{1}{n} \cdot EL(w, \pi[0, n-1])$. In the sequel, when the weight function w is clear from the context we will omit it and simply write $EL(\gamma)$ and $\overline{MP}(\pi)$ or $\underline{MP}(\pi)$. We denote by $Inf(\pi)$ the set of states that occur infinitely often in π . We consider the following conditions:

- *Parity condition.* The *parity condition* $Parity_{\mathcal{A}}(p) = \{\pi \in (Q \times E)^\omega \mid \min\{p(q) \mid q \in Inf(\pi)\} \text{ is even}\}$ requires that the minimum priority visited infinitely often is even.
- *Mean-payoff condition.* Given a threshold $\nu \in \mathbb{Q}$, and $\sim \in \{>, \geq\}$ the *mean-payoff condition* is either
 - $MeanPayoffSup^\sim(\nu) = \{\pi \in (Q \times E)^\omega \mid \overline{MP}(\pi) \sim \nu\}$
 - $MeanPayoffInf^\sim(\nu) = \{\pi \in (Q \times E)^\omega \mid \underline{MP}(\pi) \sim \nu\}$
 Mean-payoff conditions defined with $\sim \in \{<, \leq\}$ are obtained by duality since $\limsup_{n \rightarrow \infty} x_n = -\liminf_{n \rightarrow \infty} -x_n$.
- *Tail objective.* Informally the class of tail objectives is the class of objectives that are independent of all finite prefixes. Formally $\phi \subseteq (Q \times E)^\omega$ is a *tail objective* if for every $\pi \in (Q \times E)^\omega$, $\pi \in \phi \Leftrightarrow$ every suffix of π is in ϕ .

Language defined by a MSO formula. Let $\varphi(X_1, X_2)$ be a formula of Monadic Second-order logic over the signature $\{<\}$, where X_1 and X_2 are set (second-order) variables. We define the language $L_\varphi \subseteq (\{0, 1\} \times \{0, 1\})^\omega$ as usual. Let $w = (\sigma_1(0))(\sigma_1(1))\dots$ be an ω word. Let P_1, P_2 be monadic predicates on \mathbb{N} defined as: $P_1(t) \Leftrightarrow w(t)_1 = 1$ and $P_2(t) \Leftrightarrow w(t)_2 = 1$ for every $t \in \mathbb{N}$. Then $w \in L_\varphi$ iff $(\mathbb{N}, <) \models \varphi(P_1, P_2)$. A language $L \subseteq (\{0, 1\} \times \{0, 1\})^\omega$ is *MSO definable* if there exists a MSO formula $\varphi(X_1, X_2)$ such that $L = L_\varphi$. The MSO definable languages over an alphabet $\Sigma_1 \times \Sigma_2$ are defined similarly [14].

ω regular languages. It is well-known that the class of ω language definable by MSO formulas is the same as the class recognizable by parity automata. An ω language is *regular* if it is definable by a MSO formula.

3 Games with Errors

We define two versions of games with errors. Informally we want to distinguish the case where player-2 can detect errors made by player-1, from the case where errors cannot be detected during the play.

The first case is captured by *games with detected errors*, and the second case is captured by *games with undetected errors*.

3.1 Games with Detected Errors

Let z be a symbol not in Σ_1 . We use z to represent a (detected) error.

Play with detected errors. A *play with detected errors* has ω rounds. In round $i \in \mathbb{N}$: First player-1 chooses $\sigma_1 \in \Sigma_1 \cup \{z\}$, then player-2 responds with $\sigma_2 \in \Sigma_2$.

A play with detected errors $\rho \in ((\Sigma_1 \cup \{z\}) \times \Sigma_2)^\omega$ is formed. In the sequel if it is clear from the context, we shall refer a to play with detected errors simply as a play.

Interpretation of a play with detected errors. Let ρ be a play with detected errors. $\rho' \in (\Sigma_1 \times \Sigma_2)^\omega$ is an *interpretation* of ρ if for every $i \in \mathbb{N}$, $\rho(i)_2 = \rho'(i)_2$ and if $\rho(i)_1 \neq z$ then $\rho(i)_1 = \rho'(i)_1$. The *set of interpretations* of a play ρ is denoted by $\text{Interp}(\rho)$. We define the interpretation of a play prefix similarly.

Error count and error rate. Let ρ be a play with detected errors.

The *error count* of a play prefix $\rho[0, n]$ is denoted by $EC(\rho[0, n])$ and it is the number of occurrences of z in $\rho[0, n]$. The *error rate* of a play ρ is denoted by $ER(\rho)$ and is defined as $\limsup_{n \rightarrow \infty} \frac{1}{n} \cdot EC(\rho[0, n - 1])$.

Winning conditions for plays with detected errors. A *winning condition* is a tuple (L, χ) , where $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$ is the target objective, and $\chi \subseteq ((\Sigma_1 \cup \{z\}) \times \Sigma_2)^\omega$ is the *quantitative threshold* objective. For a play with detected errors ρ , we consider three types of quantitative thresholds.

- *Fixed number of errors.* For $m \in \mathbb{N}$, $\rho \in \chi \Leftrightarrow EC(\rho[0, n - 1]) \leq m$ for every $n \geq 0$.
- *Finite number of errors.* $\rho \in \chi$ if z does not occur infinitely often in ρ .
- *Error rate.* For $\delta \in \mathbb{Q}$, $\rho \in \chi \Leftrightarrow ER(\rho) \leq \delta$.

Let ρ be a play with detected errors, player-2 wins in ρ if $\text{Interp}(\rho) \subseteq L$ or if $\rho \notin \chi$.

We name a class of games according to the target objective and the quantitative threshold. For example, the class of games with ω regular language target objective and any quantitative threshold χ is named: Regular games with detected errors.

In the sequel, if the target objective is clear from the context, we omit the name of the target objective. In addition instead of defining χ explicitly we simply state the threshold value (i.e., n , fin or δ).

Definition 1 Let $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$ be the target objective.

1. $DE_n(L)$ is a game with detected errors and a winning condition (L, n) .
2. $DE_{fin}(L)$ is a game with detected errors and a winning condition (L, fin) .
3. $DE_\delta(L)$ is a game with detected errors and a winning condition (L, δ) .

Strategies for plays with detected errors. A *strategy* determines player next move according to the history of moves (of both players) in the play. Formally, a player-1 strategy is a function $\tau_1 : ((\Sigma_1 \cup \{z\}) \times \Sigma_2)^* \rightarrow (\Sigma_1 \cup \{z\})$, a player-2 strategy is a function $\tau_2 : ((\Sigma_1 \cup \{z\}) \times \Sigma_2)^* \times (\Sigma_1 \cup \{z\}) \rightarrow \Sigma_2$.

A play with detected errors ρ is played according to player-1 strategy τ_1 if $\rho(i)_1 = \tau_1(\rho[0, i-1])$ for $i \geq 0$. A play according to player-2 strategy is defined similarly.

Finite-memory strategies. A strategy of player-1 is a *finite-memory* strategy if it can be encoded by a deterministic transducer $(M, m_0, \alpha_u, \alpha_n)$ where M is a finite set (the memory of the strategy), $m_0 \in M$ is the initial memory value, $\alpha_u : M \times (\Sigma_1 \cup \{z\}) \times \Sigma_2 \rightarrow M$ is an update function, and $\alpha_n : M \rightarrow \Sigma_1 \cup \{z\}$ is the next move function. The *size* of the strategy is the cardinality of M . In every round, let m be the current memory value, then the strategy chooses $\sigma_1 = \alpha_n(m)$ as player-1 next move and the memory is updated to $\alpha_u(m, \sigma_1, \sigma_2)$, where σ_2 is player-2 move. Finite-memory strategy for player-2 is defined similarly.

Winning strategy. Player- i strategy τ_i (for $i = 1, 2$) is a *winning strategy* if for every play played according to τ_i , player- i wins. If player- i has a winning strategy, player- i is said to be the *winner* of the game.

3.2 Games with Undetected Errors

In this game, player-1 errors are not detected by player-2, thus the error count and error rate cannot be measured by the amount of z s in the play. In this subsection we give appropriate definitions for plays, play interpretation, error count, error rate, and winning conditions.

Play with undetected errors. A play with undetected error is defined exactly as a play (without errors) in section 2. Thus every play with undetected errors forms $\rho \in (\Sigma_1 \times \Sigma_2)^\omega$.

Interpretation of play with undetected errors. Let $\rho \in (\Sigma_1 \times \Sigma_2)^\omega$ be a play with undetected errors. $\rho' \in (\Sigma_1 \times \Sigma_2)^\omega$ is an *interpretation* of ρ if $\rho(i)_2 = \rho'(i)_2$ for every $i \geq 0$. The *set of interpretations* of a play ρ is denoted by $\text{Interp}(\rho)$. The interpretations of a play prefix are defined similarly.

Error count and error rate. Since errors are not detected, the existence of an error in a certain position of the play is relative to a specific interpretation of the play. i.e., for play ρ , different interpretations ρ_1, ρ_2 may define different error positions.

Let ρ be a play with undetected errors, and $\rho' \in \text{Interp}(\rho)$. ρ' is said to have an *error* in position i if $\rho(i)_1 \neq \rho'(i)_1$. The *Error count* of the finite play prefix ρ and $\rho' \in \text{Interp}(\rho)$, denoted by $EC(\rho, \rho')$, is the number of positions in ρ with errors relatively to ρ' .

The *error rate* of a play ρ and $\rho' \in \text{Interp}(\rho)$ is denoted by $ER(\rho, \rho') = \limsup_{n \rightarrow \infty} \frac{1}{n} \cdot EC(\rho[0, n-1], \rho'[0, n-1])$

Winning conditions for plays with undetected errors. For $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$, the *winning condition* is a tuple (L, χ) . We refer to L as the *target objective*, and to $\chi \subseteq ((\Sigma_1 \times \Sigma_2)^\omega \times (\Sigma_1 \times \Sigma_2)^\omega)$ as the *quantitative threshold* objective. For a play with undetected errors ρ , we consider three types of quantitative thresholds.

- *Fixed number of errors.* For $m \in \mathbb{N}$, $(\rho, \rho') \in \chi \Leftrightarrow EC(\rho[0, n-1], \rho'[0, n-1]) \leq m$ for every $n \geq 0$.
- *Finite number of errors.* $(\rho, \rho') \in \chi \Leftrightarrow \rho' \in \text{Interp}(\rho)$ and ρ' differs from ρ in finite number of positions.
- *Error rate.* For $\delta \in \mathbb{Q}$, $(\rho, \rho') \in \chi \Leftrightarrow ER(\rho, \rho') \leq \delta$.

Let ρ be a play with undetected errors, player-2 wins in play ρ if for every $\rho' \in \text{Interp}(\rho)$, either $\rho' \in L$ or $(\rho, \rho') \notin \chi$.

We name a class of games according to the target objective and the quantitative threshold. For example a class of games with languages recognized by a mean-payoff automaton as a target objective, and any quantitative threshold χ is named: mean-payoff games with undetected errors.

In the sequel, if the target objective is clear from the context, we omit the name of the target objective. In addition instead of defining χ explicitly we simply state the threshold value (i.e., n , fin or δ).

Definition 2 Let L be the target objective.

1. $UDE_n(L)$ is a game with undetected errors and a winning condition (L, n) .
2. $UDE_{\text{fin}}(L)$ is a game with undetected errors and a winning condition (L, fin) .
3. $UDE_\delta(L)$ is a game with undetected errors and a winning condition (L, δ) .

4 Regular Games with Detected Errors

Regular games with detected errors consists of ω regular language L and an error threshold objective χ .

Decision problem. *Deciding who is the winner:*

- Input: An ω regular language $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$ (given either as a MSO formula or as a parity automaton), numbers $n \in \mathbb{N}$ and $\delta \in \mathbb{Q}$.
- Output: decide who is the winner of $DE_n(L)$, $DE_{\text{fin}}(L)$ and $DE_\delta(L)$

Computation problems.

- The *computability of winning parameters* problem is:
 - Input: ω regular language $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$
 - Output: Find the minimal $n \in \mathbb{N}$ such that player-1 is the winner of $DE_n(L)$ and the minimal $\delta \in \mathbb{Q}$ such that player-1 is the winner of $DE_\delta(L)$.
- The *computability of winning strategy* problem is:
 - Input: An ω regular language $L \subseteq \Sigma_1 \times \Sigma_2)^\omega$, numbers $n \in \mathbb{N}$ and $\delta \in \mathbb{Q}$.
 - Output: A finite description of the winning strategies for the winning player of $DE_n(L)$, $DE_{\text{fin}}(L)$ and $DE_\delta(L)$

The main result of this section is

Theorem 1 *Let L be an ω regular language.*

1. *For every $n \in \mathbb{N}$ and $\delta \in \mathbb{Q}$ It is decidable who is the winner of $DE_n(L)$, $DE_{fin}(L)$ and $DE_\delta(L)$.*
2. *The winning parameters problem is computable.*
3. *The winning strategy problem is computable.*

A proof outline for Theorem 1 We present an immediate reduction from $DE_n(L)$ and $DE_{fin}(L)$ games, where L is an ω regular language to regular games without errors.

In addition we present a reduction from $DE_\delta(L)$ games, where L is an ω regular language to mean-payoff parity games (studied in [4]). This reduction together with results of [4] imply the decidability of who is the winner of $DE_\delta(L)$ game and the computability of the minimal δ such that player-1 is the winner of $DE_\delta(L)$.

The computability of the minimal $m \in \mathbb{N}$ such that player-1 is the winner of $DE_m(L)$ for an ω regular language L is obtained by the following lemma.

Lemma 1 *Let $\mathcal{A} = (\Sigma_1 \times \Sigma_2, Q, Q_0, E)$ be a an automaton, and let ϕ be a tail objective. Then player-2 is the winner of $DE_{fin}(L_{\mathcal{A}, \phi}) \Leftrightarrow$ player-2 is the winner of $DE_m(L_{\mathcal{A}, \phi})$ for $m = 2^{|Q|}$.*

Due to Lemma 1 and Theorem 1(1) and since every parity objective is a tail objective, one can decide the winner of $DE_n(L)$ for every $n = 0, 1, \dots, 2^{|Q|}$ and return the minimal n such that player-1 is the winner of $DE_n(L)$.

Need of infinite memory strategies. The following proposition asserts the need of infinite memory strategies for player-1 for regular games with detected errors and error rate threshold.

Proposition 1 *There exists an ω regular language $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$ and a threshold $\delta \in \mathbb{Q}$ such that player-1 is the winner of $DE_\delta(L)$, however player-1 does not have a finite memory winning strategy.*

Proof (of Proposition 1). Fix $\Sigma_1 = \Sigma_2 = \{a, b\}$. Consider the following ω language $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$. A tuple $(X_1, X_2) \in (\Sigma_1 \times \Sigma_2)^\omega$ is in the language L if the following conditions are hold.

1. $(\exists^\omega t X_1(t) = b) \rightarrow (\exists t (\forall t' > t (X_2(t') = a)))$
2. $(\forall t X_1(t) = a) \rightarrow (\exists^\omega t (X_2(t) = b))$

First, let us show that for $\delta = 0$, player-1 is the winner of $DE_\delta(L)$. The following strategy τ is a winning strategy for player-1: In round $i \in \mathbb{N}$, if there exists $n \in \mathbb{N}$ such that $i = 2^n$, play z , otherwise play a .

Clearly the error rate of a play played according to τ is 0. Player-1 violates the target objective in the following way. If player-2 played infinitely often b , then the play interpretation which replaces every occurrence of z with b violates the first condition. If player-2 played finitely often b , then the play interpretation which replaces every occurrence of z with a violates the second condition.

Thus, player-1 is the winner of $DE_\delta(L)$.

Second, the reader can verify that for every $n \in \mathbb{N}$, every player-1 finite memory winning strategy for $DE_{\frac{1}{n}}(L)$ requires memory size of at least n . Hence, player-1 does not have a finite memory winning strategy for $DE_\delta(L)$ for $\delta = 0$, which concludes the proof of proposition 1. \square

5 Regular Games with Undetected Errors

Analyzing games with errors is much more difficult when the errors are not detected. As in section 4 we are interesting in the decidability of who is the winner and in the computation of winning parameters and winning strategies. However, we are able to give only partial answer to these questions.

The main result of this section is:

Theorem 2 *The following problem is undecidable.*

- *Input: $\delta \in (0, 1) \cap \mathbb{Q}$ and an ω regular language L .*
- *Question: Decide who is the winner of $UDE_\delta(L)$.*

A proof outline for Theorem 2 In [9, 7] it was proved that the universality problem of a non-deterministic mean-payoff automaton is undecidable. We provide a reduction from this problem to the problem of who is the winner of $UDE_\delta(L)$ game.

The next remark deals with the simple cases.

Remark 1 *Let L be an ω regular language.*

1. *For $n \in \mathbb{N}$ it is decidable who is the winner of $UDE_n(L)$, and a winning strategy is computable.*
2. *It is decidable who is the winner of $UDE_{fin}(L)$, and the winning strategy is computable.*
3. *For $\delta = 1$, it is decidable who is the winner of $UDE_\delta(L)$, and the winning strategy is computable.*

Proof (of Remark 1). All items in Remark 1 are immediately proved by encoding the winning condition as a MSO formula. \square

The following two decision problems remain open.

1. The *bounded errors* problem asks if for an ω -regular language L , there exists $n \in \mathbb{N}$ such that player-1 is the winner of $UDE_n(L)$.
2. The *zero error rate* problem, asks for an ω -regular language L and for $\delta = 0$, who is the winner of $DE_\delta(L)$.

At first glance one can hope to answer the bounded number of errors problem with the same technique used in Lemma 1, i.e., try to prove that if for an ω -regular language L player-2 is the winner of $UDE_{fin}(L)$, then there exists $m_L \in \mathbb{N}$ such that player-2 is the winner of $UDE_{m_L}(L)$. The next proposition show that this is not the case.

Proposition 2 *There exists an ω regular language L such that player-1 is the winner of $UDE_{fin}(L)$, however for every $n \in \mathbb{N}$ player-2 is the winner of $UDE_n(L)$.*

6 Mean-Payoff Games with Errors

Quantitative languages do not enjoy the same robustness as ω regular languages. In particular, the class of quantitative languages recognized by the deterministic mean-payoff automata is a strict subset of the class of quantitative languages recognized by the non-deterministic mean-payoff automata [7]. In addition, deterministic mean-payoff automata are not closed under the conjunction operation [7].

In this section we consider only quantitative languages defined by a **deterministic** mean-payoff automaton. We also assume that the mean-payoff condition is an objective of the form of $\text{MeanPayoffInf}^{\geq}(0)$. However unless noted otherwise, the proofs can be easily modified for $\text{MeanPayoffInf}^>(0)$, $\text{MeanPayoffSup}^{\geq}(0)$ and $\text{MeanPayoffSup}^>(0)$ objectives.

The next theorem is the main result of this section.

Theorem 3 *Let $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$ be a quantitative language recognized by a mean-payoff automaton \mathcal{A} .*

1. *For every $\delta > 0$, it is undecidable who is the winner of $\text{DE}_\delta(L)$.*
2. *For every $\delta > 0$, it is undecidable who is the winner of $\text{UDE}_\delta(L)$.*
3. *For every $n \in \mathbb{N}$ it is decidable who is the winner of $\text{DE}_n(L)$ and of $\text{DE}_{fin}(L)$.*
4. *For $\delta = 0$, it is decidable who is the winner of $\text{DE}_\delta(L)$. Moreover, player-2 is the winner of $\text{DE}_\delta(L)$, for $\delta = 0 \Leftrightarrow$ player-2 is the winner $\text{DE}_{fin}(L)$ ¹.*

A proof outline for Theorem 3 The first two items are proved by an immediate reduction from the universality problem of non-deterministic mean-payoff automata. This problem was proved to be undecidable in [9, 7].

For the last two items we prove that for every quantitative language defined by a mean-payoff automaton \mathcal{A} , there exists a computable $m_{\mathcal{A}}$ such that: Player-1 is the winner of $\text{DE}_{m_{\mathcal{A}}}(L) \Leftrightarrow$ Player-1 is the winner of $\text{DE}_{fin}(L) \Leftrightarrow$ Player-1 is the winner of $\text{DE}_\delta(L)$ for $\delta = 0$.

Determining the winner of $\text{DE}_m(L)$ for fixed $m \in \mathbb{N}$ is done by a reduction to a game without errors with winning condition of the form $\bigcap_{i=1}^{f(m)} L_i$, where $f(m)$ is computable from m and L_i is a quantitative language computable from L , m and i . The proof is concluded with the following lemma.

Lemma 2 *Let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be mean-payoff automata. Let $L = \bigcap_{i=1}^K L_{A_i, \text{MeanPayoffInf}^{\geq}(0)}$. Then it is decidable who is the winner of the Gale-Stewart game G_L .*

The proof of Lemma 2 is given in section 7.

Need for infinite memory strategies. The following proposition assert that player-2 has to use infinite memory strategies for DE_n games even for $n = 1$.

Proposition 3 *There exists an ω language L recognized by a mean-payoff automaton such that player-2 is the winner of $\text{DE}_n(L)$ for every $n \in \mathbb{N}$, however player-2 does not have a finite memory winning strategy for $\text{DE}_n(L)$ even for $n = 1$.*

¹ The proof of item 4 holds only for $\text{MeanPayoffInf}^{\geq}(0)$ objective

7 Multidimensional Mean-Payoff Games

Multidimensional mean-payoff games are an extension of mean-payoff games to graphs with multidimensional weights. A multidimensional mean-payoff objective is a boolean combination of one dimensional mean-payoff objectives. For example, let G be a graph with weight function $w : E \rightarrow \mathbb{Z}^2$, and $\nu \in \mathbb{Q}$. Let ϕ_1 be the $\text{MeanPayoffInf}^{\geq}(\nu)$ objective according to the projection of w to the first dimension, and ϕ_2 be the $\text{MeanPayoffInf}^{\geq}(\nu)$ objective according to the projection of w to the second dimension. A possible multidimensional objective can be any boolean combination of ϕ_1 and ϕ_2 .

One interesting form of multidimensional objective is a conjunction of one dimensional mean-payoff objectives. This objective was introduced in [6]. However, the decidability of who is the winner was proved only for the case where player-2 is restricted to finite memory strategies. The general case (i.e., when player-2 strategy is not limited to finite memory) was stated as an open question. In this section we prove decidability and present tight complexity bounds to the question of who is the winner. Thus we answer an open question from [6], and complete the proof of Theorem 3.

Another interesting multidimensional objective is a positive boolean combination of one dimensional mean-payoff objectives. For this case we prove decidability of who is the winner.

Since in this section we consider games without errors, we find it convenient to use the standard definitions of games on graphs presented below.

Game graph. A *game graph* $G = \langle Q, E \rangle$ consists of a finite set Q of states partitioned into player-1 states Q_1 and player-2 states Q_2 . The graph is bipartite, i.e., $E \subseteq (Q_1 \times Q_2) \cup (Q_2 \times Q_1)$. In addition, every state must have an outdegree of at least 1.

Plays and strategies. A game on G starting from an *initial state* $q_0 \in Q$ is played in rounds as follows. If the game is in a player-1 state, then player-1 chooses an outgoing edges; otherwise the game is in a player-2 state, and player-2 chooses an outgoing edges. The game results is a *play* from q_0 , i.e., an infinite path $\rho = q_0 e_0 q_1 e_1 \dots$ such that e_i is an edge from q_i to q_{i+1} for all $i \geq 0$. A strategy for player-1 is a function $\tau : (Q \times E)^* \times Q_1 \rightarrow E$. A strategy for player-2 is a function $\tau : (Q \times E)^* \times Q_2 \rightarrow E$.

Mean-payoff vector. Let G be a game graph, $k \in \mathbb{N}$, and $w : E \rightarrow \mathbb{Z}^k$ be a multi-dimension weight function for the edges. We denote by w_i the projection of w to dimension i . Let $\pi \in (Q \times E)^\omega$ be an infinite path (i.e., play) in G . The *mean-payoff vector* $\overrightarrow{MP}(\pi) = (\underline{MP}(\pi)_1, \dots, \underline{MP}(\pi)_k, \overline{MP}(\pi)_1, \dots, \overline{MP}(\pi)_k)$ has $2k$ dimensions. In the first k dimensions, $\underline{MP}(\pi)_i = \underline{MP}(w_i, \pi)$, for $i = 1, \dots, k$. Similarly, in the last k dimensions, $\overline{MP}(\pi)_i = \overline{MP}(w_i, \pi)$, for $i = 1, \dots, k$.

Multidimensional mean-payoff objectives. For a k -dimensional game and $S \subseteq \{1, \dots, k\}$ and $\nu \in \mathbb{Q}$ we denote by $\bigwedge \text{MeanPayoffSup}_S^{\geq}(\nu)$ the following objectives:

Player-2 wins $\bigwedge \text{MeanPayoffSup}_{\bar{S}}^{\geq}(\nu)$ in a play π if $\bar{MP}(\pi)_i \geq \nu$, for every $i \in S$.

The objectives $\bigwedge \text{MeanPayoffSup}_{\bar{S}}^{\geq}(\nu)$, $\bigwedge \text{MeanPayoffInf}_{\bar{S}}^{\geq}(\nu)$ and $\bigwedge \text{MeanPayoffInf}_{\bar{S}}^{\geq}(\nu)$, are defined similarly. When $S = \{1, \dots, k\}$ we will drop the subscript and write $\bigwedge \text{MeanPayoffSup}^{\geq}(\nu)$ instead of $\bigwedge \text{MeanPayoffSup}_{\{1, \dots, k\}}^{\geq}(\nu)$.

An objective is *conjunctive* if it is a conjunction of conditions of the form $\bar{MP}(\pi)_i \sim \nu$ and $\underline{MP}(\pi)_i \sim \nu$, where $\sim \in \{>, \geq\}$.

$\bigvee \bigwedge \text{MeanPayoffInf}^{\geq, >}(\nu)$ is the class of objectives of the form $\bigvee_{i \in \{1, \dots, m\}} \bigwedge_{j \in S_i} \underline{MP}(\pi)_j \sim \nu$, where $\sim \in \{>, \geq\}$ and $S_1, \dots, S_m \subseteq \{1, \dots, k\}$.

$\bigwedge \bigvee \text{MeanPayoffSup}^{\geq, >}(\nu)$ is the class of objectives of the form $\bigwedge_{i \in \{1, \dots, m\}} \bigvee_{j \in S_i} \bar{MP}(\pi)_j \sim \nu$, where $\sim \in \{>, \geq\}$ and $S_1, \dots, S_m \subseteq \{1, \dots, k\}$.

Let w be a weight function and $b, c \in \mathbb{Q}$ such that $b > 0$, and π be a play. Then $\bar{MP}(w, \pi)_i \geq \nu$ iff $\bar{MP}(w', \pi)_i \geq b\nu + c$, where w' is a weight function equal to w in all dimensions except i and $w'_i = bw_i + c$. Similar equivalences hold for \underline{MP} .

For $\vec{\nu} \in \mathbb{Q}^{2k}$ an objective $\text{MultiDimensionMeanPayoff}(\vec{\nu})$ is defined as $\{\pi \in (Q \times E)^\omega \mid \bar{MP}(\pi) \geq \vec{\nu}\}$. Clearly determining the winner of a game with $\text{MultiDimensionMeanPayoff}(\vec{\nu})$ objective is log-space reducible to determining the winner of a game with conjunctive objective.

The next two theorems are the main result of this section.

Theorem 4 For every finite game graph $G = \langle Q, E \rangle$ with a weight function $w : E \rightarrow \mathbb{Z}^k$, and a conjunctive objective ϕ : player-1 has a winning strategy iff he has a memoryless winning strategy.

Theorem 5 For input: game graph $G = \langle Q, E, w : E \rightarrow \mathbb{Z}^k \rangle$, initial state $q \in Q$ and $\nu \in \mathbb{Q}$. The problem of deciding whether player-2 is the winner for a multidimensional mean-payoff objective ϕ is

1. In $\text{coNP} \cap \text{NP}$ when $\phi \in \{\bigwedge \text{MeanPayoffSup}^{\geq}(\nu), \bigwedge \text{MeanPayoffSup}^{>}(\nu)\}$. Moreover the problem of determining the winner is in \tilde{P} (i.e., it has a pseudo-polynomial-time algorithm) and there is a polynomial time Cook reduction to the problem of determining the winner of a (one dimensional) mean-payoff game.
2. (Strongly) coNP complete when $\phi \in \{\bigwedge \text{MeanPayoffInf}^{\geq}(\nu), \bigwedge \text{MeanPayoffInf}^{>}(\nu)\}$.
3. (Strongly) coNP complete when ϕ is an arbitrary conjunctive objective.
4. (Strongly) coNP complete when $\phi \in \bigvee \bigwedge \text{MeanPayoffInf}^{\geq, >}(\nu)$ and (Strongly) NP complete when $\phi \in \bigwedge \bigvee \text{MeanPayoffSup}^{\geq, >}(\nu)$.

We recall that an algorithm runs in pseudo-polynomial time, for input $G = \langle Q, E, w : E \rightarrow \mathbb{Z}^k \rangle$, if its running time is polynomial in the size of G and the description of w , when the values of the weight function w are represented in unary. A (co)NP-complete problem is called strongly (co)NP-complete if it is proven that it cannot be solved by a pseudo-polynomial time algorithm unless $\text{P}=\text{NP}$.

We would like to note that the lower bounds of Theorems 5(2)-(3) are easily obtained from proofs of lower bounds in [6]. Our proof of Theorems 5(2) relies on the corresponding result of [6] for the case when the players are restricted to use only finite state memory strategies.

We are now ready to prove Lemma 2.

Proof of Lemma 2. We begin with the following remark.

Remark 2 *Let $\mathcal{A}_1, \mathcal{A}_2$ be automata with a $\bigwedge \text{MeanPayoffInf}^{\geq}(0)$ objective. Then there exists \mathcal{A}_3 with a $\bigwedge \text{MeanPayoffInf}^{\geq}(0)$ objective such that $L_{\mathcal{A}_3} = L_{\mathcal{A}_1} \cap L_{\mathcal{A}_2}$. Moreover \mathcal{A}_3 is computable from \mathcal{A}_1 and \mathcal{A}_2 , and $|\mathcal{A}_3| = |\mathcal{A}_1| \times |\mathcal{A}_2|$.*

By Remark 2 we conclude the proof of Lemma 2 in the following way. Let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be one dimension mean-payoff automata. Let $L = \bigcap_{i=1}^k L_{\mathcal{A}_i, \text{MeanPayoffInf}^{\geq}(0)}$. Then one can compute a k dimensional mean-payoff automaton \mathcal{A} such that $L_{\mathcal{A}, \bigwedge \text{MeanPayoffInf}^{\geq}(0)} = L$. By Theorem 5 deciding the winner in the Gale-Stewart game defined by L is decidable.

Moreover, Remark 2 implies that Theorem 3 can be extended also to quantitative languages defined by a multidimensional mean-payoff automaton with $\bigwedge \text{MeanPayoffInf}^{\geq}(0)$ objective.

Note that by the same arguments we can prove Lemma 2 for $\bigwedge \text{MeanPayoffInf}^>(0)$, $\bigwedge \text{MeanPayoffSup}^{\geq}(0)$ and $\bigwedge \text{MeanPayoffSup}^>(0)$ objectives.

8 Conclusion and Further Work

In this work we investigated games with errors and obtained decidability results described in Table 1. Our proofs immediately imply a 2-EXPTIME upper bound for the complexity of determining the winner of a game with errors (for the decidable fragments). Further work is required to give tighter complexity bounds. Further work may also consider additional classes of specification languages. While Table 1 contains five open problems, we believe that the following two open problems may have applications outside the framework of games with errors.

1. Decidability of the bounded number of errors problem for regular games with undetected errors.
2. Decidability of who is the winner of $\text{UDE}_\delta(L)$ game for ω regular language L and $\delta = 0$.

We also investigated multidimensional mean-payoff games and provided complexity bounds for interesting fragments of these games. The following two interesting problems are open.

1. Decidability of who is the winner for arbitrary multidimensional mean-payoff objective (as defined in section 7). For example, the decidability of who is the winner for the objective $\{\pi \in (Q \times E)^\omega | (\underline{MP}(\pi)_1 \geq 0) \wedge (\overline{MP}(\pi)_2 \geq 0) \vee (\underline{MP}(\pi)_3 \geq 0)\}$ was not determined in this paper.
2. Complexity of solving multidimensional mean-payoff games with fixed number of dimensions and a $\bigwedge \text{MeanPayoffInf}^{\geq}(0)$ objective. Specifically whether the problem of determining the winner is in $\text{NP} \cap \text{coNP}$, and/or in \tilde{P} ?

References

1. R. Alur, A. Degorre, O. Maler, and G. Weiss. On omega-languages defined by mean-payoff conditions. In Proc. of FOSSACS, pages 333-347. Springer, 2009.
2. T. Brázdil, P. Jancar, and A. Kucera. Reachability games on extended vector addition systems with states. In Proc. of ICALP, pages 478-489. Springer, 2010.
3. J. R. Büchi , L. H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the AMS* 138 (1969) 295-311
4. K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Mean payoff parity games. In Proc. of LICS, pages 178-187. IEEE Computer Society, 2005.
5. K. Chatterjee and L. Doyen. Energy Parity Games. In Proc. of ICALP, pages 599-610. Springer, 2010.
6. K. Chatterjee, L. Doyen, T. A. Henzinger, J.F. Raskin Generalized mean payoff and Energy Games, To appear in Proc. of FSTTCS 2010
7. K. Chatterjee, L. Doyen, H. Edelsbrunner, T. A. Henzinger and P. Rannou mean payoff Automaton Expressions. In Proc. of CONCUR, pages 269-283. Springer 2010.
8. K. Chatterjee, L. Doyen, T. A. Henzinger, and J. -F. Raskin. Algorithms for omega-regular games of incomplete information. In Proc. of CSL, pages 287-302. Springer 2006.
9. A. Degorre, L. Doyen, R. Gentilini, J. -F. Raskin, and S. Torunczyk. Energy and Mean-Payoff Games with Imperfect Information. In Proc. of CSL, pages 260-274. Springer, 2010.
10. A. Ehrenfeucht and J. Mycielski. International journal of game theory. Positional Strategies for Mean-Payoff Games, 8:109-113, 1979.
11. E. Grädel, W. Thomas, T. Wilke. Automata, Logics and Infinite Games. Volume 2500 of LNCS. Springer, 2002
12. S. R. Kosaraju G. and Sullivan. Detecting cycles in dynamic graphs in polynomial time. In Proceedings of the Twentieth Annual ACM Symposium on theory of Computing 1988, pages: 398 - 406.
13. J. H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274-301, 1984.
14. W. Thomas, Automata on infinite objects. *Handbook of Theoretical Computer Science* , volume B, pages 133–191. Elsevier Science Pub., 1990.
15. U. Zwick and M. Paterson. The Complexity of Mean Payoff Games on Graphs. *Theoretical Computer Science* 1996, volume 158, pages 343-359