Dependency Matrices for Multiplayer Strategic Dependencies

Dylan Bellier
Univ Rennes, IRISA, CNRS, France

Sophie Pinchinat
Univ Rennes, IRISA, CNRS, France

François Schwarzentruber
Univ Rennes, IRISA, CNRS, France

Abstract

In multi-player games, players take their decisions on the basis of their knowledge about what other players have done, or currently do, or even, in some cases, will do. An ability to reason in games with temporal dependencies between players' decisions is a challenging topic, in particular because it involves imperfect information. In this work, we propose a theoretical framework based on *dependency matrices* that includes many instances of strategic dependencies in multi-player imperfect information games. For our framework to be well-defined, we get inspiration from quantified linear-time logic where each player has to label the timeline with truth values of the propositional variable she owns. We study the problem of the existence of a winning strategy for a coalition of players, show it is undecidable in general, and exhibit an interesting subclass of dependency matrices that makes the problem decidable: the class of perfect-information dependency matrices.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics; Theory of computation → Logic and verification; Theory of computation → Automata over infinite objects

Keywords and phrases Temporal dependency, Delay games, Strategic reasoning, Temporal logic

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2022.31

Supplementary Material
InteractiveResource: `https://francoisschwarzentruber.github.io/fsttcs2022/`
Software (Source Code): `https://github.com/francoisschwarzentruber/fsttcs2022`
archived at `swh:1:dir:eed0f57a83b2979abe84fd80bd804a6d730539b3`

1 Introduction

In perfect-information multi-player games, decisions of players depend on their knowledge about what other players have done so far. This setting is adopted in various logics for strategic reasoning such as Alternating-time Temporal Logics [2] and Strategy Logic [4]. A *strategy* for a player assigns to each history the next action to play. In imperfect-information games, a player may have partial knowledge about what other players did, enforcing her strategy to depend only on her knowledge about the current history.

However, there are situations where dependencies involve the knowledge about the future of a play. For instance, Grove and Clarkson in [7] developed an online algorithm for the bin packing problem with a *look-ahead* that takes advantage of information on some future items. Look-ahead also exists in parsing: for LL(1) (*for Left to right, Leftmost derivation*) grammars, the production rule to be chosen depends on the next symbol in the read word [1]. The extreme case occurs when the decisions depend on the *entire* future of the play. This is the case for *offline* algorithms, where the future is finite (for bin packing it is given by the sequence of all incoming items). It also appears for an infinite future, such as in Quantified Propositional Linear-time Temporal Logic (QPTL) [14]: for a formula $\forall a \exists b \varphi$, the choice of the truth values of proposition b depends on all those of proposition a.

42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022).
Editors: Anuj Dawar and Venkatesan Guruswami; Article No. 31; pp. 31:1–31:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In addition, players may experience delays to receive information. For instance, a proponent may have to play without having received yet information about the last three moves of her opponent. Klein et al. in [8] formalized *delay games* as an extension of Gale-Stewart games [6], i.e. two player infinite games with perfect information.

Our paper proposes a unifying theoretical framework to specify dependencies. To this aim, we define the notion of a *dependency matrix* D. The entry $D[a, b]$ is a generalized integer (i.e. in $\mathbb{Z} \cup \{-\infty, +\infty\}$) so that

> "Player a's decision at time-step $t_{current}$ depends on Player b's decisions up to time-step $t_{current} + D[a, b]$".

The semantics of a dependency matrix relies on an involved machinery based on an imperfect-information multi-player game, called the *meta game*. The positions of the game, called *configurations*, are all the possible partial labelings of the timeline by the players – and are thus in infinite number. The dynamics of the meta game is involved because the dependencies may desynchronize players in their choices for labeling a time point.

Moreover, some matrices encode circular dependencies between players, leading to deadlocked situations where none of the players can progress anymore in the meta game, preventing them from completing their labeling of the timeline. Upon the study of this phenomenon, we introduce the class of *progressing* matrices, that guarantee that any play in the meta arena provides a full labeling of the timeline for each propositional variable. We establish an effective property that characterizes these matrices. Thus, plays can be qualified as winning or losing according to some linear-time formula, here an LTL formula.

We then study the problem called EWS (Existence of Winning Strategies) of deciding, given a dependency matrix, a coalition (subset of players) and an LTL formula, the existence of a joint strategy for the coalition such that any play brought about by this strategy satisfies the LTL formula. Importantly, the imperfect-information feature of the meta game addresses two issues: first, this game is not determined in general, and second, winning strategies for the coalition need being *uniform*, a non-trivial notion in our rich setting since players may be desynchronized.

Although we prove that EWS is unsurprisingly undecidable, we exhibit the subclass of so-called *perfect information* dependency matrices for which EWS turns to a decidable problem. We first consider the perfect information property for matrices whose values range over $\mathbb{Z}$ and show how EWS can be reduced to solving a two-player perfect-information parity game, yielding a 2-EXPTIME-complete complexity. We then generalize the perfect-information property to arbitrary matrices and provide a decision procedure for EWS that generalizes the one for QPTL [14], thus a non-elementary complexity.

To our knowledge, our proposal offers the first framework amenable for merging many, and yet remote, game settings such as concurrent or turn-based games [2], (two-player) delay games [8, 9, 16], logic QPTL [14], and Church Synthesis Problem (see the survey [5]) – it can be shown that our framework also subsumes DQBF (Dependency Quantified Boolean Formulae) [10].

Outline. In Section 2, we define dependency matrices, and show that they embed several settings of games. Section 3 contains the necessary background. In Section 4, we present the formal machinery to define an arena specified by a dependency matrix. In Section 5, we address the problem EWS of the existence of winning strategies and show its undecidability in the general case. Next, in Section 6, we design the decision procedure for EWS when restricted to a perfect-information matrix input. We conclude our contribution in Section 7.

2 Dependency Matrices and Examples

In this section, we propose the generic notion of dependency matrix and show that it subsumes several classic settings in games. We denote players by lowercase letters such as a, b, c, etc. A dependency matrix specifies the mutual dependencies between players' decisions in a game where each player owns an atomic proposition and aims at filling the whole timeline with a valuation for it at each time point. Formally,

▶ **Definition 1.** *A* dependency matrix *(or simply a* matrix*) over a finite set $\mathcal{P}$ of at least two players is a matrix $D = (D[a,b])_{a \neq b \in \mathcal{P}}$, and whose values range over $\mathbb{Z} \cup \{-\infty, +\infty\}$.*

In a dependency matrix $(D[a,b])_{a \neq b \in \mathcal{P}}$ over $\mathcal{P}$, the value $D[a,b]$ describes how Player a's decisions depends on Player b's: Player a's decision for choosing the valuation at time point t depends on the ones chosen by Player b at all time points in the interval $[0, t + D[a,b]]$. As such, whenever $D[a,b] < 0$, Player a's decision at time point t is independent of the decisions made by Player b up to some time point before t. In particular, if $D[a,b] = -\infty$, Player a's decisions is independent of any of Player b's.

On the contrary, when $D[a,b] \geq 0$, Player a's decision at t depends on some Player b's decisions up to some time point after t, so that Player a is not able to make her decision without this required information. In particular, if $D[a,b] = +\infty$, Player a's decisions do depend on the decisions of Player b's over all the timeline.

Because it is natural to consider that a player is aware of her own decisions so far, values on the diagonal $D[a,a]$ are irrelevant and are left undefined. As such, the matrix line $D[a,.]$ specifies the dependencies of Player a with respect to all other players, and her ability to make a decision is constrained by all these dependencies. Unsurprisingly, some matrices may yield blocking situations for some players, an issue that we address in the sequel.

Beforehand, we illustrate how several settings in games can be captured with matrices.

▶ **Example 2** (Concurrent Game). *In a standard* concurrent game *(as in logics ATL, ATL*), players have to concurrently choose a move. Thus the move of one player can only depend on the* strict past *of the history of moves. The corresponding matrix for 3 players is D_1, where strict past is reflected by the value -1.*

$$D_1 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} \cdot & -1 & -1 \\ -1 & \cdot & -1 \\ -1 & -1 & \cdot \end{pmatrix} \end{array}$$

▶ **Example 3** (Round Robin Game). *In a* Round Robin *game, players play in turn: first Player a, then b, then c. The matrix is D_2: decisions of Player a depend on the strict past, hence the values -1 in the a-row; decisions of b depend on the non-strict past of a (hence the value 0), and the strict past of c (hence the value -1); decisions of c depend on the non-strict past of the other players.*

$$D_2 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} \cdot & -1 & -1 \\ 0 & \cdot & -1 \\ 0 & 0 & \cdot \end{pmatrix} \end{array}$$

▶ **Example 4** (QPTL). *In QPTL, dependencies stem from the order of the quantifiers: in the formula $\exists a \forall b \exists c \varphi$ where φ is an LTL-formula, Player a plays first on the* full timeline *and independently of the others. Then b only depends on what Player a did. Finally, c depends on what both Players a and b did. All this is reflected by matrix D_3.*

$$D_3 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} \cdot & -\infty & -\infty \\ +\infty & \cdot & -\infty \\ +\infty & +\infty & \cdot \end{pmatrix} \end{array}$$

▶ **Example 5** (Church Synthesis). *The* Church Synthesis *problem (see the survey [5]) consists in responding to a stream of inputs by a stream of outputs, so that a given property holds. If Player a and Player b are in charge of the output, Player c and Player d are in charge of the input, the players dependencies are captured by matrix D_4: output Players a and b only depend on the past values of the input players, and that input Players c and d see all values and have to respond on the spot.*

$$D_4 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} \cdot & -1 & -1 & -1 \\ -1 & \cdot & -1 & -1 \\ 0 & 0 & \cdot & -1 \\ 0 & 0 & -1 & \cdot \end{pmatrix} \end{array}$$

▶ **Example 6** (Fixed Delay games). *A delay game is a two-player game, say between Player a and Player b. Player a must make a given finite number k of moves beforehand and then, Players a and b play in a turn based manner, hence maintaining the delay between them (see Klein et al. [8]).* [1] *This setting is represented by the matrix D_5.*

$$D_5 = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{pmatrix} \cdot & -k \\ k-1 & \cdot \end{pmatrix} \end{array}$$

In the next section, we fix some notations and recall some useful definitions to develop our theory around matrices.

3 Background

Given a finite alphabet Σ, we use the standard notations $\Sigma^\star$, Σ^ω and Σ^∞ for the set of finite words, infinite words and their union respectively, and ε to denote the empty word. Given two words $u, w \in \Sigma^\star$, we write $u \cdot w$ for their concatenation. Given a non-empty word $u = u_0 u_1 \cdots u_n$, we let $|u| := n + 1$ be its length and set $|\varepsilon| = 0$. For $k \leq n$, we write $u[k]$ for the letter u_k, $u[: k]$ for the k-th prefix $u_0 \cdots u_k$ and $u[k :]$ for the k-th suffix $u_k \cdots u_n$.

We now recall the basics of *Linear-time Temporal Logic* (LTL) [13]. An LTL formula φ (over a set AP of propositions) is evaluated on a labeling of the (discrete) timeline $\mathbb{N}$, called an *LTL assignment*[2] $\lambda \in (\{\top, \bot\}^\omega)^{\mathrm{AP}}$. We classically write $\lambda \models \varphi$ whenever λ satisfies

[1] In their setting, the delay is defined with a delay function that gives at each round the number of moves the input Player has to make. However, Klein et al. mainly studied the fixed delay setting where the function is set to 1 after the first round.

[2] also named *trace* in the model-checking setting.

φ (we omit the semantics here and refer to [13]). Alternatively, an LTL assignment can be seen as an infinite word over the alphabet $\{\top, \bot\}^{\text{AP}}$, which makes a tight connection between logic and automata: given an LTL formula φ, one can build – via the Vardi-Wolper construction [17] together with the Safra-like translation from Büchi to parity acceptance condition [11] – a deterministic *infinite-word parity automaton* $\mathcal{A}_\varphi$ whose language $\mathcal{L}(\mathcal{A}_\varphi)$ is composed of all LTL assignments that satisfy φ.

Noticeably, LTL can be extended with propositional quantifications to *Quantified Propositional Temporal Logic* (QPTL), which enjoys the prenex normal form [15]. Therefore, we can assume that QPTL formulas are all of the form $\vec{Q}\ \varphi$, where $\vec{Q}$ is a finite sequence of propositional quantifications ($\exists a$ or $\forall a$, where $a \in \text{AP}$) and φ is an LTL formula.

In this paper, we consider vectors of words over the alphabet $\{\top, \bot\}$, indexed by a finite subset $\mathcal{P}$ of AP. Given a vector $U = (U_a)_{a\in\mathcal{P}}$ of words over $\{\top, \bot\}$ and a player $b \in \mathcal{P}$, we denote by $U(b)$ the b's component of U. We extend all notations for words to word vectors with their meaning component-wise. In addition, we introduce the notation $U +_a u$ for the word vector obtain by concatenating the word u to $U(a)$.

As we will see, in our framework, the set AP of propositions for logics LTL and QPTL will be the breeding ground to pick the finite set $\mathcal{P}$ of players for our matrices.

4 The Formal Setting of Dependency Matrices

From a matrix D over $\mathcal{P}$, a coalition $\Gamma \subseteq \mathcal{P}$ and an LTL formula φ, we derive the *meta game* $\langle D, \Gamma, \varphi\rangle$ that specifies how players can progress to label the timeline with truth values for thier proposition, what the coalition is, and what the winning condition is. Precisely, the goal of the coalition is to make φ true.

The arena for the meta game, fully determined by the matrix D, is a modified multiplayer Gale-Stewart arena [6] taking into account the information flow between players according to D. We distinguish *progressing* matrices that yield arenas with only infinite plays where no player is blocked, thus resulting in an LTL assignment. We also provide a graph-based polynomial-time algorithm to characterize progressing matrices.

Sticking to progressing matrices, we develop the proper notions of (player and coalition) strategies, along with the property of *uniformity* of a strategy that takes into account the imperfect information feature of our games. Intuitively, a strategy in the meta game is uniform if it depends only on the informations prescribed by the matrix. For pedagogical purposes, we start with bounded-value matrices, then consider arbitrary ones.

4.1 The Meta Arena of a Matrix

We fix a matrix D with values in $\mathbb{Z}$ and we describe the *meta arena* of D. A position in the meta arena is called a *configuration*, that is a word vector C that reflects the labeling over $\{\top, \bot\}$ chosen so far by each player of $\mathcal{P}$. The set of configurations is denoted by $\mathcal{C} := (\{\top, \bot\}^\star)^{\mathcal{P}}$ and the initial configuration C^0 is the empty vector, namely $\varepsilon^{\mathcal{P}}$.

▶ **Example 7.** Figure 1 shows the configuration C in which Player a played the word $C(a) = \top\top\bot\top\bot$, Player b played $C(b) = \bot\bot\top$, and Player c played $C(c) = \bot\bot\top\top$.

We define the dynamics of the game, namely which player can play/*progress*, in a given configuration and which moves are available to her. Here is an intuitive example of those dynamics with a Round Robin matrix.

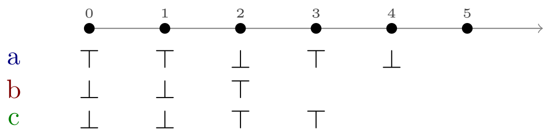
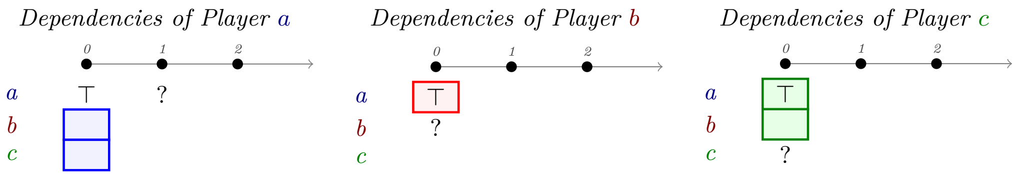


Figure 1 A configuration C where Player a has chosen her labeling up to time point 4, Player b up to time point 2 and Player c up to 3.

▶ **Example 8.** *For the matrix D_2 of Example 3 and for the first round, only Player a can make a move of length 1. Then, assuming Player a chooses $\top$, the dependencies for each player are depicted separately below: in each picture, the squares identify expected information for the considered player to make her decision about the question mark (?).*



Observe that, here, neither Player a, nor Player c can make a move as the labeling of Player b at time point 0 is not set. On the contrary, Player b is able to progress.

A *move* of a Player a is a word $u_a \in \{\top, \bot\}^\star$ that is to extend her labeling along the timeline. For Player a to progress in configuration C, all her dependencies must be fulfilled. This is formalized as follows: in order to make a move (necessarily starting at the time point $t = |C(a)|$), Player a needs to access the labeling of Player b up to time point $t + D[a, b]$, included. Therefore, the value $\alpha^C_{a,b} := |C(b)| - (D[a, b] + |C(a)|)$ characterizes the length of a move available to Player a with regard to her dependency on Player b only. Thus, the overall *progress value* of Player a, written α^C_a, takes into account all quantities $\alpha^C_{a,b}$ for $b \neq a$ in a conjunctive manner, leading to consider the most restrictive one. Formally:

$$\alpha^C_a \stackrel{\text{def}}{=} \max(0, \min_{b \neq a}(\alpha^C_{a,b})) \tag{1}$$

As such, α^C_a is the maximum number of steps that Player a can perform in configuration C. Note that, if some $\alpha^C_{a,b}$ is negative (including 0), then Player a is stuck in C because Player b has not yet provided the expected information.

Based on the progress value, a *legitimate move* for Player a in a configuration C is a word in $\{\top, \bot\}^{\alpha^C_a}$. Then, a *legitimate joint move* in C is a vector of words $u \in (\{\top, \bot\}^\star)^{\mathcal{P}}$ such that, for every Player a, $u(a)$ is a legitimate move for a. We can now define the *move function* between configurations:

$$\begin{array}{rcl} \Delta : \mathcal{C} \times (\{\top, \bot\}^\star)^{\mathcal{P}} & \rightharpoonup & \mathcal{C} \\ (C, u) & \mapsto & \begin{cases} C \cdot u & \text{when } u \text{ is a legitimate joint move in } C \\ \text{undefined} & \text{otherwise} \end{cases} \end{array}$$

with u a legitimate joint move. Remark that all players that can move have to play concurrently and greedily (i.e. their maximum number α^C_a of actions). A configuration C is said *reachable* if there is a finite sequence of legitimate joint moves that leads to C from the initial configuration C^0.

▶ **Proposition 9.** *Every reachable configuration, has a unique predecessor by Δ.*

Proof. Suppose that there are two reachable configurations C^1 and C^2 and two legitimate joint moves u^1 and u^2 such that $C^1 \cdot u^1 = C^2 \cdot u^2$. Then, consider a reachable configuration C' that is a prefix of both C^1 and C^2. Suppose toward contradiction that there are legitimate joint moves $u^{1'} \neq u^{2'}$ with $C^{1'} = C' \cdot u^{1'}$ prefix of C^1 and $C^{2'} = C' \cdot u^{2'}$ prefix of C^2. Since $u^{1'} \neq u^{2'}$, there is $t \in \mathbb{N}$ and $a \in \mathcal{P}$ such that $C^{1'}(a)[t] \neq C^{2'}(a)[t]$. By definition, $C^{i'}$ is a prefix of C^i for $i \in \{1,2\}$. Hence $C^1(a)[t] \neq C^2(a)[t]$ which is in contradiction with $C^1 \cdot u^1 = C^2 \cdot u^2$. In conclusion, if $C^1 \cdot u^1 = C^2 \cdot u^2$, then $C^1 = C^2$ and $u^1 = u^2$. ◀

By Proposition 9, the meta arena of reachable configurations is a tree and every reachable configuration contains all moves since the start of the game.

Observe that some meta arenas have reachable configurations C where for every Player a, we have $\alpha_a^C = 0$. This happens because of cyclic dependencies. Here is an example.

▶ **Example 10.** *Consider the matrix D_6 on the right. Observe that Player b cannot make any move because she needs Player a's labeling at time point 1, but for Player a to label time point 1, she needs the label of Player b at time point 0. This deadlocked situation is depicted in Figure 2. However, observe that Player c can progress independently up to time point 3.*

$$D_6 = \begin{matrix} & a & b & c \\ a \\ b \\ c \end{matrix}\! \begin{pmatrix} \cdot & -1 & -1 \\ +1 & \cdot & -1 \\ -4 & -4 & \cdot \end{pmatrix}$$

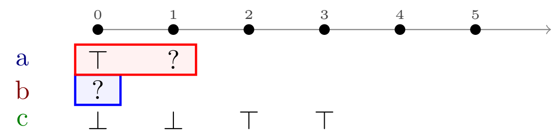
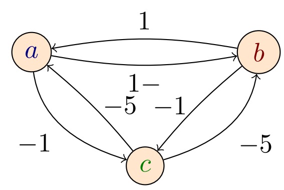


Figure 2 Player a wants to know the moves of Player b and reciprocally.

Situations where some players eventually get stuck can be characterized by analyzing some graph: for the case of Example 10, the graph is depicted below, and interestingly, it contains the cycle (a, b, a) whose weight is 0, a positive value. We will see in the next section, where the graph is formally defined, that such a cycle provides evidence that Players a and b eventually get stuck.

G_{D_6} :



4.2 Progressing Matrix

We aim at characterizing matrices where no player gets stuck because of cyclic dependencies, so that each play yields an assignment of the timeline in order to interpret the LTL winning condition. Such matrices are called *progressing* and can be identified by means of their adjacency weighted graph, here called the *dependency graph.*

▶ **Definition 11.** *Given a matrix $D = (D[a, b])_{a \neq b \in \mathcal{P}}$, the dependency graph of D is the weighted directed graph $G_D = (V, E, r)$ where:*

- *$V = P$ is the set of vertices,*
- *$E = \{(a, b) | a \neq b\}$ is the set of edges,*
- *$r(a, b) = D[a, b]$ is the weight of the edge (a, b).*

The following proposition gives a characterization of progressing matrices.

▶ **Proposition 12.** *A matrix D is progressing if, and only if, its dependency graph G_D has no non-negative-weighted cycle.*

The graph G_{D_6} of Example 10 has a non-negative 0-weight cycle (a, b, a), so, matrix D_6 is not progressing. As a corollary, we have the following:

▶ **Theorem 13.** *Deciding if a matrix is progressing is in PTIME.*

Proof. The size of the dependency graph is linear in the size of the matrix and finding a non-negative cycle is polynomial in the size of the graph. ◀

From now on, unless stated otherwise, we only consider progressing matrices, that we keep calling "matrices" for simplicity. On the basis of such matrices, only infinite-horizon *plays* take place that consist in consecutive applications of the move function Δ in the meta arena (see page 6):

▶ **Definition 14.** *A* play *in the meta arena associated to a matrix is an infinite sequence of configurations $(C^n)_{n\in\mathbb{N}}$ where C^0 is the empty configuration, and for every $n \in \mathbb{N}$, C^{n+1} is the successor of C^n.*

The interested reader can explore the dynamics of plays at:

`https://francoisschwarzentruber.github.io/fsttcs2022`

Notice that along a play $(C^n)_{n\in\mathbb{N}}$, C^{n+1} extends C^n, so that to the limit, the play naturally yields a temporal assignment λ of the timeline: for every $t \in \mathbb{N}$, and every Player a, we let $\lambda(t)(a) \overset{\text{def}}{=} C^n(a)(t)$, for a sufficiently large integer n so that $C^n(a)(t)$ is defined.

The next section focuses on strategies and winning strategies in the meta arena, where we discuss how a strategy complies with a matrix.

4.3 Strategies in the Meta Arena

In this section, we fix a matrix D over $\mathcal{P}$, a coalition $\Gamma \subseteq \mathcal{P}$ and an LTL formula φ. Classically, a strategy maps histories to moves. However, since in our setting, a configuration fully characterizes a history (Proposition 9), we can equivalently define strategies as mappings from configurations to moves. A *strategy* for Player $a \in \Gamma$ is a function $f : \mathcal{C} \to \{\top, \bot\}^\star$ where $f(C)$ prescribes a legitimate move for Player a.

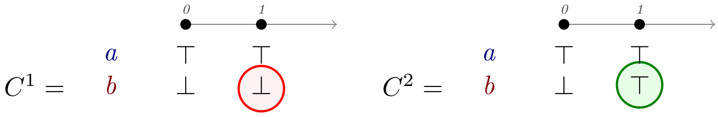
Furthermore, we define a *joint strategy* for the Γ as a function $F : \mathcal{C} \to (\{\top, \bot\}^\star)^\Gamma$ such that $F(C)(a)$ is a legitimate move for Player $a \in \Gamma$. A joint strategy F provides a strategy F_a for each Player $a \in \Gamma$ defined by: $F_a(C) \overset{\text{def}}{=} F(C)(a)$.

Given a joint strategy F for a coalition Γ, a play $(C^n)_{n\in\mathbb{N}}$ is an *outcome* of F if for any $n \in \mathbb{N}$ and any Player $a \in \Gamma$, we have $C^{n+1}(a) = C^n(a) \cdot F_a(C^n)$. We denote by $out(F)$ the set of outcomes of F. A joint strategy F is *winning* φ whenever all assignments associated to the plays in $out(F)$ satisfy φ.

However, in our dependency-based setting, a strategy is relevant only if it is *uniform*, in the sense that they only rely on the information available to the player. We illustrate this important feature in Example 15.

▶ **Example 15.** *Consider matrix D_7 and configurations C^1 and C^2 below.*

$$D_7 = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} a \quad\; b \\ \begin{pmatrix} \cdot & -2 \\ -2 & \cdot \end{pmatrix} \end{array}$$



When Player a in C^1 comes to label time point 2, and since at time point 2, she can only access Player b's labeling up to time point 0, she cannot distinguish it from C^2. However, once she labels time point 2, she is able to access Player b's label at time point 1, and is allowed to take this information into account before choosing her label at time point 3. Now, according to the matrix D_7, we have $\alpha_a^{C^1} = \alpha_a^{C^2} = 2$. Although Player a cannot distinguish between C^1 and C^2, she is allowed to choose different 2-length moves that only differ in their second letters. Indeed, her choice at time point 2 has to be uniform (and therefore the same in both configurations C^1 and C^2). On the contrary, she may play differently for her choice at time point 3. For instance, the overall move in C^1 can be $\top\bot$ while it is $\top\top$ in C^2.

We formalize the phenomenon described in Example 15 with equivalence relations between configurations. In the example, Player a has to choose a move u_0u_1 but C^1 and C^2 are indistinguishable for her when it comes to choosing the first letter u_0 . However, they can be distinguished for the choice of the second letter u_1. Then, we need multiple relations, one for each letter of a move.

Formally, we introduce an equivalence relation between configurations parameterized by a scope k: two configurations C^1 and C^2 are *k-indistinguishable for Player a*, denoted $C^1 \overset{a}{\sim}{}_D^k C^2$, whenever $|C^1(a)| = |C^2(a)|$, and for every Player $b \neq a$ and every $t \leq |C^1(a)| + D[a,b] + k$, we have:

- either both $C^1(b)(t)$ and $C^2(b)(t)$ are undefined (meaning k is greater than the progress of b in both configurations), or
- $C^1(b)(t) = C^2(b)(t)$.

We resort to relations $\overset{a}{\sim}{}_D^k$ to formalize the notion of uniform strategies in our framework, where the parameter k is meant to range over $[0, \min(\alpha_a^{C^1}, \alpha_a^{C^2})[$.

▶ **Definition 16.** *A strategy f for Player a is D-*uniform *whenever for any two configurations C^1 and C^2, and any natural number $k \leq \min(\alpha_a^{C^1}, \alpha_a^{C^2})$,*

$$C^1 \overset{a}{\sim}{}_D^k C^2 \text{ implies } f(C^1)[:k] = f(C^2)[:k].$$

Observe that $C^1 \overset{a}{\sim}{}_D^{k+1} C^2$ implies $C^1 \overset{a}{\sim}{}_D^k C^2$. We generalize Definition 16 to joint strategies in a natural way by requiring the uniform property for every individual strategy of the joint strategy. For the rest of the paper, all strategies are implicitly D-uniform.

Before addressing the central decision problem of the existence of a winning joint strategy, we extend our setting to allow matrices with infinite values.

4.4 Arbitrary Matrices

Recall, by Definition 1, that a value $D[a,b] = -\infty$ indicates that Player a's decision are independent from Player b's. In particular, if the matrix line $D[a,.]$ is all filled with value $-\infty$, Player a fills the whole timeline in the first round. On the contrary, $D[a,b] = +\infty$ forces Player a to wait until Player b has entirely filled the timeline.

A typical example of a matrix with infinite values is provided by the matrix D_3 of Example 4, and is reminiscent of what is expressed in the setting of the logic QPTL: in this example, a play takes place as follows. First, Player a chooses an a-assignment of the timeline. Second, since Player a is done and Player b is independent from Player c, Player b has all the required information for choosing the b-assignment over the timeline. Third and finally, Player c can proceed for the c-assignment, and the play ends.

For the cases that mix finite and infinite value, we consider first Example 17

▶ **Example 17.** *Consider matrix D_8 below. In a play, Player b and Player c's mutual dependencies enforce them to proceed in turn for choosing their respective labeling, while Player a cannot play until the other two have labeled the whole timeline.*

$$D_8 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} \cdot & +\infty & +\infty \\ -\infty & \cdot & -1 \\ -\infty & 0 & \cdot \end{pmatrix} \end{array}$$

As observed in Example 17, $(\mathbb{Z} \cup \{-\infty, +\infty\})$-valued matrices may yield to "transfinite" configurations (i.e. with possibly components in $\{\top, \bot\}^\omega$ instead of $\{\top, \bot\}^\star$). As a result, a play may now be a finite sequence of (possibly infinite) sequences of configurations. We can adapt the notion of reachable configuration accordingly – since this is routine, we omit the precise definition here.

From now on, unless stated otherwise, we consider arbitrary matrices. In the next section, we address the decision problem EWS of the existence of a winning strategy.

5 Undecidability of EWS

We consider the following central decision problem EWS of deciding the Existence of a Winning (uniform) Strategy for a coalition of players (EWS for short):

▶ **Theorem 18.** *Let EWS be the following problem:*
Input: *A matrix D, a coalition Γ and an LTL-formula φ.*
Output: *"Yes" if and only if there is a D-uniform joint strategy for Γ that is winning for φ.*
EWS is undecidable.

This reslut is unsurprising, given the imperfect-information nature of our multi-player game.

Theorem 18 is proved by reducing the Tiling problem [3] to EWS. Recall that the Tiling problem takes in input a finite set of square tiles and two binary connectivity relations over the tiles, that specify which pairs of tiles may be adjacent (resp. horizontally and vertically). The output is the answer to the question whether there exists a tiling of the plane, that is a mapping from $\mathbb{N}^2$ to the set of tiles such that any two of adjacent tiles respect the connectivity constraints. In a nutshell, our reduction involves four players τ_1, τ_2 (tilers) and c_1, c_2 (challengers): in a round, Challenger c_1 chooses a place (x, y) in $\mathbb{N}^2$ and privately communicates it to his tiler companion τ_1 by playing $\top^x \bot \top^y \bot^\omega$. Tiler τ_1 then responds by choosing a tile t independently of the choice of Challenger c_2 and plays $\top^t \bot^\omega$, and symmetrically for Players τ_2 and c_2. This way, the two tilers play independently. The two different Challengers are used to test the binary relations by choosing adjacent places. The following matrix encodes this situation.

$$D_{\text{Tiling}} := \begin{array}{c} \\ c_1 \\ c_2 \\ \tau_1 \\ \tau_2 \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \tau_1 & \tau_2 \end{array} \\ \begin{pmatrix} \cdot & -\infty & -\infty & -\infty \\ -\infty & \cdot & -\infty & -\infty \\ +\infty & -\infty & \cdot & -\infty \\ -\infty & +\infty & -\infty & \cdot \end{pmatrix} \end{array}.$$

In the next section, we present a decidable sub-case.

6 Decidability of EWS for perfect-information Matrices

A close inspection of the proof of Theorem 18 reveals that not being able to circumvent the amount of information hidden to players is a matter. We introduce the subclass of *perfect-information* matrices where every player always has full information about the current configuration before proceeding, yielding a meta game that is turn-based with perfect information.

6.1 Definition and Properties

A meta game is perfect-information as long as two reachable configurations are not k-indistinguishable, for every k. Actually, not being 0-indistinguishable is sufficient, since k-indistinguishability are nested (see Section 4.3). Furthermore, for the meta arena to be turn-based, we must guarantee that in each round, only one player can progress. This yields the following definition.

▶ **Definition 19.** *A matrix D is* perfect-information *if for every reachable configuration C:*

- *for every player a and reachable configuration $C' \neq C$, we have $C \not\sim^{a\,0}_{D} C'$ and*
- *there is exactly one player a such that $\alpha^C_a \geq 1$.*

Remark that, by Definition 19, every move of a single player, from a reachable configuration that is not the initial one, is necessarily of length 1. Indeed, if a player could make a move of length strictly greater than 1, we could create another reachable configuration that would be 0-indistinguishable from the first one, a contradiction.

▶ **Lemma 20.** *Let D be a perfect-information matrix, then $\alpha^C_a \leq 1$ for any non-initial reachable configuration C and every Player a.*

Proof. By contradiction, suppose that there is a non-initial reachable configuration C^1 with $\alpha^{C^1}_a \geq 2$ for some Player a. Because D is progressing and perfect-information, there is a unique Player b that can progress in $\Delta^{-1}(C^1)$. If $b = a$, we would not have $\alpha^{C^1}_a \geq 2$ because players play greedily. Then, we have $a \neq\ b$.

We now exhibit a reachable configuration $C^2 \neq C^1$ such that $C^1 \sim^{a\,0}_{D} C^2$. Let u^1 be the joint move leading to C^1, that is $\Delta(\Delta^{-1}(C^1), u^1) = C^1$. As exactly one player moves, $u^1(c) = \varepsilon$ for every Player $c \neq b$. We define the joint move u^2 as follows: for every Player $c \neq b$, let $u^2(c) = \varepsilon$ and $u^2(b) = \mathsf{fliplast}(u^1(b))$ where $\mathsf{fliplast}$ flips the last letter of the word (mapping $\top$ to $\bot$, and $\bot$ to $\top$). For $C^2 = \Delta(\Delta^{-1}(C^1), u^2)$, we have $|C^1(a)| = |C^2(a)|$, and $C^1(c) = C^2(c)$ for *Player* $c \neq b$. We have $\alpha^{C^i}_{a,b} \geq 2$ for $i \in\ \{1, 2\}$. Then, by definition, $|C^i(b)| - (D[a, b] + |C^i(a)|) \geq 2$, whence $(D[a, b] + |C^i(a)|) \leq |C^i(b)| - 2$. Now, let $t \leq |C^1| + D[a, b]$. By transitivity, $t\ \leq |C^i(b)| - 2$. However, $C^1(b)[t] = C^2(b)[t]$ for $t\ \leq |C^1(b)| - 2$ since only the last letter of $C^1(b)$ differs from $C^1(b)$. Therefore, $C^1 \sim^{a\,0}_{D} C^2$ which is a contradiction. ◀

▶ **Corollary 21.** *Let D be a perfect-information matrix, and C a non-initial reachable configuration, there is Player a with $\alpha^C_a = 1$ and for every other Player b, we have $\alpha^C_b = 0$.*

We can use this result to establish a characterization of perfect-information matrices. Let C be a reachable non-initial configuration and let Player a be the player that can progress in C. By Corollary 21, we have $\alpha^C_a = 1$ and $\alpha^C_b = 0$ for every Player $b \neq a$. We first make a claim:

▷ Claim 22. $\alpha^C_{a,b} = 1$ and $\alpha^C_{a,b} \leq 0$

Using this claim, we obtain $|C(b)|-(D[a,b]+|C(a)|) = 1$ and $|C(a)|-(D[b,a]+|C(b)|) \leq 0$. Then, $D[a,b] + D[b,a] \geq -1$. Since the matrix is progressing, we have $D[a,b] + D[b,a] \leq -1$ and then $D[a,b] + D[b,a] = -1$. In fact, we show that this necessary condition is a precise characterization of the perfect-information matrices.

▶ **Theorem 23.** *A matrix D is perfect-information if and only if for all players a and b, with $a \neq b$ we have $D[a,b] + D[b,a] = -1$.*

We now show the reciprocal, namely, if D satisfies $D[a,b] + D[b,a] = -1$, then D is perfect-information. The first step is to prove that, the associated meta arena is turn based.

▶ **Lemma 24.** *For D with $D[a,b] + D[b,a] = -1$ whenever $a \neq b$, there is at most one Player a with $\alpha_a^C \geq 1$ in every configuration C.*

Proof. By contradiction, suppose $\alpha_{a,b}^C \geq 1$ and $\alpha_{b,a}^C \geq 1$. Then, $\alpha_{a,b}^C + \alpha_{b,a}^C \geq 2$. Since $\alpha_{a,b}^C = |C(b)|-(D[a,b]+|C(a)|)$ and $\alpha_{b,a}^C = |C(a)|-(D[b,a]+|C(b)|)$, we obtain $D[a,b]+D[b,a] \leq -2$ which contradicts the assumption on D. ◀

It is left to prove that any two different reachable configurations are not $\overset{a}{\sim}{}_D^0$-equivalent for any a. We here just give an intuition of the proof by contradiction. Suppose that there are two different reachable configurations C^1 and C^2 such that $C^1 \overset{a}{\sim}{}_D^0 C^2$. We can assume without loss of generality that they are immediate successors of the same reachable configuration C. We compare the progress values of Player a with the one of the only player that can progress in C, and prove that the configurations C^1 and C^2 are equal.

6.2 A Parity Game to solve the Existence of a Winning Strategy

For perfect-information matrices, we establish a reduction from EWS to solving a parity game, thus attaining decidability (Theorem 30). Consider a perfect-information matrix D, a coalition Γ and a formula φ. We define the parity game $G(D,\Gamma,\varphi)$ where the coalition Γ has a winning D-uniform strategy if, and only if, Player 0 has a winning strategy in $G(D,\Gamma,\varphi)$ against Player 1.

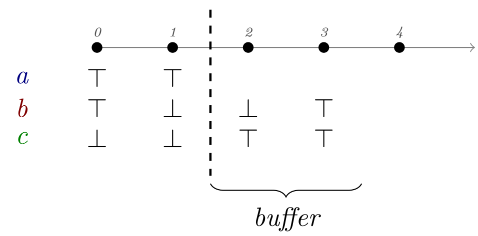
The parity game $G(D,\Gamma,\varphi)$ is built up from the deterministic parity automaton $\mathcal{A}_\varphi$ for φ (see Section 3). Its plays simulate runs of automaton $\mathcal{A}_\varphi$ on the sequence of growing configurations along a play in the meta arena. Positions in the parity games are pairs composed of states of $\mathcal{A}_\varphi$ and *buffers*: a *buffer* β is a word vector $(\beta_a)_{a\in\mathcal{P}}$ with at least one empty component. Formally, the set of buffers is:

$$\{(\beta_a)_{a\in\mathcal{P}} \in (\{\top,\bot\}^\star)^{\mathcal{P}} \mid \beta_b = \varepsilon \text{ for some } b \in \mathcal{P}\}.$$

The *buffer of a configuration* is the "pending part" of the configuration, namely its greatest suffix that is a buffer.

▶ **Example 25.** *Consider the perfect-information matrix D_9 below where a reachable configuration C and its buffer are depicted.*

$$D_9 = \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} \cdot & 2 & 3 \\ -3 & \cdot & -1 \\ -4 & 0 & \cdot \end{pmatrix} \end{array}$$

$C =$ 

We say that a buffer is *reachable* if it is the buffer of some reachable configuration. We can show that in a reachable configuration, the single player that can progress only depends on the buffer β of this configuration, and we write p_β this player.

We denote by B the set of reachable buffers, by $B_\exists$ the set of buffers β in B where $p_\beta \in \Gamma$, and we let $B \stackrel{\text{def}}{=} B \backslash B_\exists$. Although our matrix is perfect-information, we remark that, by Lemma 20, for the particular case of the empty buffer β^0 (of the initial configuration), Player p_{β^0} might be playing a long move. Moreover, it can be shown that the reachable buffers are finitely many[1] and that their number is exponential in the values of the matrix (this is because the longest component of a reachable buffer is given by the biggest absolute value in the matrix).

We now informally describe the parity game with its two players Player 0 and Player 1. As said, a position in the parity arena is pair (q, β) composed of a state q of the automaton and a buffer β. Position (q, β) belongs to Player 0 whenever $\beta \in B_\exists$, otherwise $\beta \in B_\forall$ and it belongs to Player 1.

In a given position (q, β), only p_β progresses by choosing a move $u \in \{\top, \bot\}$. We consider the word vector obtained by catenating u to p_β's component in buffer β, that we write $\beta +_{p_\beta} u$ in the following.

If $\beta +_{p_\beta} u$ is still a buffer we update the position to $(q, \beta +_{p_\beta} u)$. Note that this is always the case for the initial buffer β^0. Otherwise, the first letter of word vector $\beta +_{p_\beta} u$ is all filled with labels on every component, and thus can be read by automaton $\mathcal{A}_\varphi$. We then update the position to the new current state of $\mathcal{A}_\varphi$ and to the buffer obtained by removing the first letter of $\beta +_{p_\beta} u$ (which is a buffer as β is not initial).

Formally, the parity game is the following.

▶ **Definition 26.** *Given a perfect-information matrix D, a coalition of players Γ and a deterministic parity automaton $\mathcal{A}_\varphi = (Q, q_0, \Sigma, \delta, par)$ with $\Sigma = \{\top, \bot\}^{\mathcal{P}}$, we define the parity game $G(D, \Gamma, \varphi) = \langle P_0, P_1, s_0, \to, par_G \rangle$ where:*

- *$P_0 = Q \times B_\exists$ is the set of positions for Player 0,*
- *$P_1 = Q \times B_\forall$ is the set of positions for Player 1,*
- *$s_0 = (q_0, \beta^0)$ is the initial position,*
- *$(q, \beta) \to (q', \beta')$ when there is a legitimate move u for p_β in the meta arena such that:*
  1. *either $\beta +_{p_\beta} u$ is a buffer and $q = q'$ and $\beta' = \beta +_{p_\beta} u$.*
  2. *or $q' = \delta(q, (\beta +_{p_\beta} u)[0])$ and $\beta' = (\beta +_{p_\beta} u)[1\colon]$ and p_β is the only player s.t. $\beta_{p_\beta} = \varepsilon$;*
- *$par_G(q, \beta) = par(q)$, that is the priority of a position (q, β) is the priority of the state q in the automaton $\mathcal{A}_\varphi$.*

Note that the number of positions in the parity game is the product of the number of states in the automaton and the number of buffers, and that the game has the same priorities as the automaton.

▶ **Proposition 27** (For a perfect-information matrix D)**.** *$\langle D, \Gamma, \varphi \rangle$ is a positive instance of EWS if, and only if, Player 0 has a winning strategy in $G(D, \Gamma, \varphi)$.*

Proposition 27 gives us an upper bound complexity of EWS by the following algorithm.

1. Compute the deterministic parity automaton $\mathcal{A}_\varphi$ (accepting the models of φ);
2. Compute the parity game $G(D, \Gamma, \varphi)$;
3. Solve $G(D, \Gamma, \varphi)$.

---

[1] Actually the set of reachable configurations is a regular language that can be recognized by a word automaton with buffers as states.

In the following, the *size* of the matrix D is the quantity $|D| = \sum_{a \neq b} |D[a,b]|$. Observe that we can build $\mathcal{A}_\varphi$ by using the Vardi-Wolper construction [17] with the Safra-like translation from Büchi to parity acceptance condition [11], so that parity game of Step 2 has $O(2^{2^{|\varphi|}} \times 2^{2^{|D|}})$ positions and $O(2^{|\varphi|})$ priorities, hence a 2-EXPTIME decision procedure for EWS.

The next subsection, we show that this algorithm is essentially optimal by a reduction of the Church Synthesis problem.

6.3 Reduction from the Church Synthesis problem

For the lower bound, we reduce the Church Synthesis for LTL properties [5, 12]. Our Example 5 (page 4) illustrates the reduction.

▶ **Definition 28.** *Given a coalition* Γ*, a* Church matrix *is a matrix* D *where for any two players* $a \neq b$*, we have:*

$$D[a,b] = \begin{cases} 0 & \text{if } a \notin \Gamma \text{ and } b \in \Gamma \\ -1 & \text{otherwise} \end{cases}$$

In essence, for Church matrices, players have the same knowledge about the current configuration, allowing them to foresee their allies moves.

Observe that Church matrices may not be perfect-information, since the moves of every player in a team (coalition or opponents) are concurrent. Nonetheless, we can "transform" any Church matrix D into a linear sized perfect-information Round Robin matrix D' (see Example 3 and Definition 29) such that $\langle D, \Gamma, \varphi \rangle$ is a positive instance of EWS if, and only if, $\langle D', \Gamma, \varphi \rangle$ is a positive instance of EWS.

▶ **Definition 29.** *A* Round Robin matrix *is a matrix* D *such that there exists a total order* $\prec$ *over* $\mathcal{P}$*, where*

$$D[a,b] = \begin{cases} -1 & \text{if } a \prec b \\ 0 & \text{otherwise} \end{cases}$$

The total order $\prec$ describes the order in which players will play (the player that is minimal for $\prec$ plays first). Remark that $D[a,b] + D[b,a] = -1$ for any two players $a \neq b$ then every Round Robin matrix is perfect-information (by Theorem 23). Given a Church matrix, we can choose order $\prec$ so that all players in the coalition play before their opponents.

By summing up, we polynomially reduce[2] a Church synthesis problem to a EWS problem for a Church matrix and that is in turn linearly reduced to a EWS problem for a Round Robin matrix. From this latter reductions, we can state the following.

▶ **Theorem 30.** *EWS for perfect-information matrices is* 2-EXPTIME*-complete in the size of the LTL formula.*

In the next section, we extend the class of perfect-information matrices to allow some matrices with infinite values, while keeping the decidability of EWS for the resulting superclass. In particular, QPTL matrices (see Example 4 on page 4) falls into this class.

[2] The size of the Church matrix is quadratic in the number of propositions of the Church synthesis problem

6.4 Perfect-Information Matrices with Possibly Infinite Values

The proof of Proposition 27 can be extended to QPTL formulas instead of LTL formulas. The following example illustrates a procedure for matrices with infinite values that yields a generalization of the perfect-information property (see Definition 32).

▶ **Example 31.** *Consider the following matrices where the latter is perfect-information:*

$$D_8 = \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} \cdot & +\infty & +\infty \\ -\infty & \cdot & -1 \\ -\infty & 0 & \cdot \end{pmatrix}\end{array} \quad \text{and} \quad D'_8 = \begin{array}{c} \\ b \\ c \end{array}\begin{array}{c} \begin{array}{cc} b & c \end{array} \\ \begin{pmatrix} \cdot & -1 \\ 0 & \cdot \end{pmatrix}\end{array}$$

According to D_8, Player a depends on the whole labeling of Player b and Player c. Given an LTL formula φ and say coalition $\{a, b\}$, we can answer the EWS on instance $\langle D_8, \{b\}, \varphi\rangle$ as follows: we can first answer EWS for $\langle D'_8, \{b\}, \exists a\ \varphi\rangle$ (since D'_8 is perfect-information). If no, then return no for $\langle D_8, \{b\}, \varphi\rangle$. Otherwise, each outcome of the winning strategy for Player 0 in $\langle D'_8, \{b\}, \exists a\ \varphi\rangle$ reflects a play ρ in $\langle D'_8, \{b\}, \exists a\ \varphi\rangle$. From play ρ, exhibit a unique accepting run in $\mathcal{A}_{\exists a \varphi}$. By tracing back this run inside $\mathcal{A}_\varphi$, reconstruct Player a's response to the play ρ.

The procedure employed in Example 31 applies to arbitrary matrices as long as they fulfill the Definition 32.

▶ **Definition 32.** *An arbitrary matrix D is* perfect-information *if for any $a \neq b$:*
1. $D[a, b] \in \mathbb{Z}$ *implies* $D[a, b] + D[b, a] = -1$*;*
2. $D[a, b] \in \{-\infty, +\infty\}$ *implies* $D[a, b] = -D[b, a]$*;*
3. $D[a, b] = +\infty$ *implies* $D[a, c] \in \{-\infty, +\infty\}$*, for all* $c \neq a$*.*

Observe that the procedure is in fact non-elementary in the number of players with $+\infty$ dependencies. Moreover, since the validity problem for QPTL [14] reduces to EWS for arbitrary perfect-information matrices, we have the following.

▶ **Theorem 33.** *EWS is non-elementary for arbitrary perfect-information matrices.*

7 Conclusion

We presented the expressive framework of dependency matrices that can capture several game settings such as concurrent and turn-based games [2], (two-player) delay games [8,9,16], logic QPTL [14], and Church Synthesis Problem [5].

We proved that the existence of a winning strategy for a coalition to achieve an LTL formula (EWS) is undecidable for arbitrary matrices.

We then exhibited the subclass of perfect-information bounded-value matrices for which the problem EWS is 2-EXPTIME-complete in the size of the formula.

Finally, we extended the class of perfect-information matrices with a narrow use of infinite dependencies allowing to re-use known techniques of automata projection for QPTL. For these matrices, EWS becomes non-elementary. Still our complexity analysis of EWS needs beeing refine regarding the matrix parameter: we do not know yet the lower bound complexity when the LTL formula is fixed.

A first track to continue this work concerns EWS for the whole class of bounded-value matrices. We conjecture it is decidable, since, for a bounded-value matrix, each k-indistinguishable equivalence class of a reachable configurations has a bounded size.

A second track regards our transformation of Church matrices into perfect-information Round Robin ones. We believe that our approach can generalize to a class of bounded-values matrices enlarging the one of Church matrices.

References

1. Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley series in computer science / World student series edition. Addison-Wesley, 1986. URL: `https://www.worldcat.org/oclc/12285707`.
2. Rajeev Alur, Thomas A Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5):672–713, 2002.
3. Robert Berger. *The undecidability of the domino problem*. American Mathematical Soc., 1966.
4. Krishnendu Chatterjee, Thomas A Henzinger, and Nir Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.
5. Bernd Finkbeiner. Synthesis of reactive systems. *Dependable Software Systems Engineering*, 45:72–98, 2016.
6. David Gale and Frank M Stewart. Infinite games with perfect information. *Contributions to the Theory of Games*, 2(245-266):2–16, 1953.
7. Edward F. Grove. Online bin packing with lookahead. In Kenneth L. Clarkson, editor, *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1995. San Francisco, California, USA*, pages 430–436. ACM/SIAM, 1995. URL: `http://dl.acm.org/citation.cfm?id=313651.313781`.
8. Felix Klein and Martin Zimmermann. What are strategies in delay games? borel determinacy for games with lookahead. *arXiv preprint*, 2015. `arXiv:1504.02627`.
9. Felix Klein and Martin Zimmermann. How much lookahead is needed to win infinite games? *Logical Methods in Computer Science*, 12, 2017.
10. Gary Peterson, John Reif, and Salman Azhar. Lower bounds for multiplayer noncooperative games of incomplete information. *Computers & Mathematics with Applications*, 41(7-8):957–992, 2001.
11. Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Logical Methods in Computer Science*, 3, 2007.
12. A Pnueli and R Rosner. On the synthesis of reactive systems. *POPL, Austin, Texas*, pages 179–190, 1989.
13. Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57. IEEE, 1977.
14. A Prasad Sistla, Moshe Y Vardi, and Pierre Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49(2-3):217–237, 1987.
15. Aravinda Prasad Sistla. *Theoretical issues in the design and verification of distributed systems*. Harvard University, 1983.
16. Wolfgang Thomas, Lukasz Kaiser, and Michael Holtmann. Degrees of lookahead in regular infinite games. *Logical Methods in Computer Science*, 8, 2012.
17. Moshe Y Vardi and Pierre Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32(2):183–221, 1986.

A Proofs of Section 4

Given a matrix D, we say that a player a is *eventually blocked* if there is a natural $k \in \mathbb{N}$ such that for all reachable configurations C, we have $|C(a)| \leq k$. First we prove a lemma on non-negative cycles. Intuitively, this lemma helps to find a player that can never progress in a non-negative cycle.

Given two vertices c_i and c_j of a non-negative cycle $c = (c_0, \dots, c_{|c|})$ (with $c_0 = c_{|c|}$), we denote by w_i the label $r(c_i, c_{i+1})$ and $W_{i,j}$ the circular sum of the labels between c_i and c_j:

- If $i < j$, then $W_{i,j} = w_i + \dots + w_{j-1}$;
- else, $W_{i,j} = W_{i,|c|} + W_{0,j}$.

Remark that for every index i, $W_{i,i}$ is the sum of all labels of the cycle. So, for a non-negative cycle, $W_{i,i} \geq 0$. Furthermore, we denote by $W_i^\star$ the minimal sum $W_i^\star = \min_j(W_{i,j})$.

▶ **Lemma 34.** *Given a matrix D with a non-negative cycle $c = (c_0, \dots, c_{|c|})$, for all player c_i in the cycle, for all reachable configuration C, we have the following.*

$$|C(c_i)| \leq \max(0, -W_i^\star)$$

Proof. We do the proof by induction on reachable configuration.

(base case) In the initial configuration C^0, the property is obvious.

(inductive case) Consider a configuration $C' = \Delta(C, (u_a)_{a \in \mathcal{P}})$ for some joint move $(u_a)_{a \in \mathcal{P}}$ and a reachable configuration C such that $|C(c_i)| \leq \max(0, W_i^\star)$ for every $c_i \in c$. Consider $i \in [\![0, \dots, |c|]\!]$.

$$\begin{aligned}
|C'(c_i)| = &\ |C(c_i)| + \alpha^C_{c_i} \\
= &\ |C(c_i)| + \max(0, \min_{a \neq c_i}(\alpha^C_{c_i,a})) \\
\leq &\ |C(c_i)| + \max(0, \alpha^C_{c_i,c_{i+1}}) \\
\leq &\ |C(c_i)| + \max(0, |C(c_{i+1})| - D[c_i, c_{i+1}] - |C(c_i)|) \\
\leq &\ \max(|C(c_i)|, |C(c_{i+1})| - w_i) \\
\leq &\ \max(\max(0, -W_i^\star), \max(-w_i, -W_{i+1}^\star - w_i))
\end{aligned}$$

The last inequality is obtained thanks to the inductive hypothesis. We now prove that $-W_i^\star \geq \max(-w_i, -W_{i+1}^\star - w_i)$. Since $w_i = W_{i,i+1}$, we have $-W_i^\star \geq -w_i$. Let $j_0 \in [\![0, \dots, |c|]\!]$ such that $W_{i+1}^\star = W_{i+1,j_0}$. We now do a case study on j_0.

- if $j_0 = i + 1$, then, because c is non-negative, $-W_{i+1}^\star - w_i \leq -w_i$.
- if $j_0 \neq i + 1$, then, $-W_{i+1}^\star - w_i = -W_{i,j_0} \leq -W_i^\star$

Then we have $|C'(c_i)| \leq \max(0, -W_i^\star)$.

We have proven the property by induction. ◀

We now state a lemma to prove the other way around: that non-negative cycles are necessary for a matrix not to be progressing.

▶ **Lemma 35.** *Given a bounded matrix D, if there is a player that is eventually blocked, then every player is eventually blocked.*

Proof. Consider a bounded matrix D where there is a player a and a natural number $k \in \mathbb{Z}$ such that for every reachable configuration C, we have $|C(a)| \leq k$. We prove by induction on reachable configurations that for each Player b, we have $|C(a)| \leq \max(0, k - D[b, a])$.

(base case) In the initial configuration C^0, the property is obvious.

(inductive case) Consider a configuration $C' = \Delta(C, (u_a)_{a \in \mathcal{P}})$ for some joint move $(u_a)_{a \in \mathcal{P}}$ and a configuration C such that $|C(a)| \leq k - D[b, a]$ for every b. Then, for every Player b we have the following.

$$\begin{array}{rcl} |C'(b)| &=& |C(b)| + \alpha^C_b \\ &=& |C(b)| + \max(0, \min_{c \neq b}(\alpha^C_{b,c})) \\ &\leq& |C(b)| + \max(0, \alpha^C_{b,a}) \\ &\leq& |C(b)| + \max(0, |C(a)| - D[b,a] - |C(b)|) \\ &\leq& \max(|C(b)|, |C(a)| - D[b,a]) \\ &\leq& \max(\max(0, k - D[b,a]), k - D[b,a]) \\ &\leq& \max(0, k - D[b,a]) \end{array}$$

We have proven the property by induction. ◀

Now, when all players are blocked, we use the next well known result of graph theory to find our non-negative cycle.

▶ **Lemma 36.** *Given a directed graph G with no self loop, if every vertex is the source of an edge, then, there is a cycle in the graph.*

Now, we can give a proof for Proposition 12.

▶ **Proposition 12.** *A matrix D is progressing if, and only if, its dependency graph G_D has no non-negative-weighted cycle.*

Proof. Given a matrix D with a non-negative cycle, then, by Lemma 34, we immediately have that every player in the cycle is eventually blocked.

In a second time, consider a matrix D that is not progressing. Then, by Lemma 35, for every player a, there is an integer k_a such that for every reachable configuration C, we have $|C(a)| \leq k_a$. Consider a configuration C such that every player is blocked. Then, by immediate contradiction, for every player a, there is a player $q_a \neq a$ such that $\alpha^C_{a,q_a} \leq 0$ (otherwise, there would be a player that can progress). The graph $G = \langle V, E \rangle$ with the vertices $V = \mathcal{P}$ are the players of the matrix and the edges are defined as $E = \{(a, q_a) \mid a \in \mathcal{P}\}$. Since, G is a directed graph with no self loop, by Lemma 36, there is a cycle $c = (c_0, \ldots, c_{|c|})$ in the graph thus, for every $i \in [\![0, \ldots, |c| - 1]\!]$, we have $c_{i+1} = q_{c_i}$ and $c_{|c|} = c_0$. We have the following.

$$\begin{array}{rl} \sum_{i=0}^{|c|-1} \alpha^C_{c_i,c_{i+1}} &= \sum_{i=0}^{|c|-1} |C(c_{i+1})| - D[c_i; c_{i+1}] - |C(c_i)| \\ &= -\sum_{i=0}^{|c|-1} D[c_i; c_{i+1}] \end{array}$$

Since $\alpha^C_{c_i,c_{i+1}} \leq 0$ for every $i \in [\![0, \ldots, |c|]\!]$, we have proven that c is a non-negative cycle. ◀

B Proofs of Section 6

We now address the proof of Theorem 23. The first direction states that a perfect-information matrix D satisfies that for every different Players a and b, we have $D[a,b] + D[b,a] = -1$ and is presented in Section 6.1. We just need to prove Claim 22. Recall that C is a reachable non-initial configuration and Player a is the player that can progress in C. By Corollary 21, we have $\alpha^C_a = 1$ and $\alpha^C_b = 0$ for every player $b \neq a$. We consider a player $b \neq a$.

▷ Claim 22. $\alpha^C_{a,b} = 1$ and $\alpha^C_{a,b} \leq 0$

Proof. First we prove that $\alpha^C_{a,b} \leq 0$. Remark that, in C, Player a has two legitimate moves: $\top$ and $\bot$. Let $C^1 = \Delta(C, (\top)_a)$ and $C^1 = \Delta(C, (\bot)_a)$. Note that C^1 and C^2 are reachable. Toward contradiction, assume that $\alpha^C_{a,b} > 0$. We prove the contradiction $C^1 \overset{b}{\sim}^0_D C^2$. We have $\alpha^C_{a,b} = |C(a)| - (D[b,a] + |C(b)|) > 0$. Then $|C(a)| > D[b,a] + |C(b)|$. Therefore, for all $t \leq D[b,a] + |C(b)|$, we have $t < |C(a)|$. And because $C^1(a)(t) = C(a)(t) = C^2(a)(t)$, we conclude that $C^1 \overset{b}{\sim}{}^0_D C^2$.

Then we prove that $\alpha^C_{a,b} = 1$. By definition, $\alpha^C_{a,b} \geq 1$. Toward contradiction, suppose $\alpha^C_{a,b} > 1$. Let C' be the configuration defined by $C'(c) = C(c)$ for all $c \neq b$ and $C'(b) = \mathsf{fliplast}(C(b))$. We have that C' is reachable and, by the same kind of reasoning than previous point, we have $C \overset{a}{\sim}{}^0_D C'$, which is in contradiction with the assumption on D. ◁

Let us prove the other direction, namely that a matrix D is perfect-information if $D[a,b] + D[b,a] = -1$ for every pair of different Players a and b. Lemma 24 states that the meta game of such a matrix is turn based. Now, we need to prove that two different reachable configurations are not 0-indistinguishable. To do so, we first show two results on 0-indistinguishable relations. These results allow us to consider the "first time" at which two configurations diverge while being 0-indistinguishable.

▶ **Lemma 37.** *Given a matrix D and two reachable configurations C and C', with $(C^k)_{k\leq n}$ and $(C'^k)_{k\leq m}$ such that $C^n = C$ and $C'^m = C'$. If $n \geq m$ then, for every Player a we have $|C(a)| \geq |C'(a)|$.*

Proof. The proof is done by induction on $n - m$.

(base case) If $n = m$, we do an induction on n.

- **(base case)** If $n = m = 0$, then $C = C' = C^0$, the property is immediate.
- **(induction)** Suppose the property holds for some $n, m \in \mathbb{N}$ with $n = m$. Consider C and C' reachable. There are $(C^k)_{k\leq n+1}$ and $(C'^k)_{k\leq m+1}$ with $C^{n+1} = C$ and $C'^{m+1} = C'$. By inductive hypothesis, we have $|C^n(a)| = |C'^m(a)|$ for every Player a. Then, $\alpha^{C^n}_a = \alpha^{C'^n}_a$ and so, $|C(a)| = |C'(a)|$.

(induction) Suppose the property holds for some $n, m \in \mathbb{N}$ with $n \geq m$. Consider C and C' reachable: there are $(C^k)_{k\leq n+1}$ and $(C'^k)_{k\leq m}$ with $C^{n+1} = C$ and $C'^m = C'$. By inductive hypothesis, for every Player a $|C^n(a)| \geq |C'^m(a)|$ and because $|C^{n+1}(a)| \geq |C^n(a)|$, we have $|C^{n+1}(a)| \geq |C'^m(a)|$.

We have proved the property by induction. ◀

▶ **Lemma 38.** *Given a dependency matrix D, and two reachable configurations C and C' that are non-initial, if $C \overset{a}{\sim}{}^0_D C'$, then, one of the following holds.*

1. $\Delta^{-1}(C) \overset{a}{\sim}{}^0_D C'$
2. $C \overset{a}{\sim}{}^0_D \Delta^{-1}(C')$
3. $\Delta^{-1}(C) \overset{a}{\sim}{}^0_D \Delta^{-1}(C')$

Proof. Consider two reachable configurations C and C' such that $C \overset{a}{\sim}{}^0_D C'$. By definition, $|C(a)| = |C'(a)|$, and for every Player $b \neq a$, every $t \leq |C(a)| + D[a,b]$, we have $C(b)(t) = C'(b)(t)$. Since C and C' are reachable, there are two sequences of successive configurations $(C^k)_{k\leq n}$ and $(C'^k)_{k\leq m}$ such that $C^n = C$ and $C'^m = C'$. By symmetry we can assume $n \geq m$. We do a case study.

If $n = m$, then $\Delta^{-1}(C)$ and $\Delta^{-1}(C')$ are also reachable and their sequences have the same length. By Lemma 37 for every player b, we have that $|\Delta^{-1}(C)(b)| = |\Delta^{-1}(C')(b)|$. Immediately, $|\Delta^{-1}(C)(a)| = |\Delta^{-1}(C')(a)|$ and because $C \overset{a}{\sim}{}^0_D C'$, for every $t \leq |\Delta^{-1}(C)(a)| + D[a,b] \leq |C(a)| + D[a,b]$, we have $\Delta^{-1}(C)(b)(t) = C(b)(t) = C'(b)(t) = \Delta^{-1}(C')(b)(t)$. Hence, $\Delta^{-1}(C) \overset{a}{\sim}{}^0_D \Delta^{-1}(C')$

If $n > m$ then $\Delta^{-1}(C)$ is reachable with a sequence of length $n-1 \geq m$ as $\Delta^{-1}(C) = C^{n-1}$ then, by Lemma 37, for every player b, we have $|\Delta^{-1}(C)(b)| \geq |C'^m(b)|$. And, because $C \overset{a}{\sim}{}^0_D C'$, we have $|C(a)| = |C'(a)|$, then $|\Delta^{-1}(C)(a)| = |C'(a)|$. Furthermore, for every $t \leq |\Delta^{-1}(C)(a)| + D[a,b]$, if both $\Delta^{-1}(C)(b)(t)$ and $C'(b)(t)$ are defined, we have $\Delta^{-1}(C)(b)(t) = C(b)(t) = C'(b)(t)$. Finally, we prove that $C(b)(t)$ is defined if and only if $C'(b)(t)$ is defined. By Lemma 37, we already have that $|\Delta^{-1}(C)(b)| \geq |C'^m(b)|$, and if $\Delta^{-1}(C)(b)(t)$ is defined, so is $C(b)(t)$, and by hypothesis, so is $C'^m(b)(t)$.

We have proved the property. ◀

We now prove the theorem.

▶ **Theorem 23.** *A matrix D is perfect-information if and only if for all players a and b, with $a \neq b$ we have $D[a,b] + D[b,a] = -1$.*

Proof. The first direction is presented in Section 6.1. For the other direction, consider D such that $D[a,b] + D[b,a] = -1$. By Lemma 24, there is only one player that can progress in any reachable configuration. Is left to prove that for any two reachable configurations C and C' with $C \neq C'$, for every player a, we have $C \overset{a}{\not\sim}{}^0_D$. The proof is done by contradiction.

Suppose that there are two different reachable configurations C^1 and C^2 with $C^1 \overset{a}{\sim}{}^0_D C^2$. Then, by *Lemma* 38, we can assume that $\Delta^{-1}(C^1) = \Delta^{-1}(C^2) = C$. Let b the player that can progress in C. Then, for all $c \neq b$, we have $C(c) = C^1(c) = C^2(c)$ and, for all $t < |C(b)|$, we have $C(b)(t) = C^1(b)(t) = C^2(b)(t)$. Because $C^1 \neq C^2$, then we necessarily have the following.

$$C^1(b)(t_0) \neq C^2(b)(t_0) \text{ for some } t_0 \in \{|C(b)| - \alpha^C_b, |C(b)| - 1\} \tag{2}$$

Since $\alpha^C_{b,a} \geq \alpha^C_b$, we have $|C(a)| - (D[b,a] + |C(b)|) \geq \alpha^C_b$. As Player a does not progress in C, we obtain $|C^1(a)| - (D[b,a] + |C(b)|) \geq \alpha^C_b$. By hypothesis $D[b,a] = -1 - D[a,b]$, we have $|C^1(a)| + 1 + D[a,b] - |C(b)| \geq \alpha^C_b$ and because $|\Delta^{-1}(C^1)(b)| + \alpha^C_b = |C^1(b)|$, we have:

$$|C^1(a)| + D[a,b] \geq |C^1(b)| - 1 \tag{3}$$

Finally, since $C^1 \overset{a}{\sim}{}^0_D C^2$, we have that $C^1(b)(t) = C^2(b)(t)$ for every $t \leq |C^1(a)| + D[a,b]$. In particular, thanks to Equation (3), we can take $t = t_0$, and we have $C^1(b)(t_0) = C^2(b)(t_0)$, in contradiction with Equation (2). ◀

C Reduction from Church matrices to Round Robin matrices

We now prove the claim made in Section 6.3 that Church matrices can be reduced to Round Robin matrices.

From a Church matrix D for a coalition Γ, we define a Round Robin matrix $\text{Rob}(D)$ as follows. We take an arbitrary order $\prec$ on the players such that for every Player $a \in \Gamma$ and every Player $b \notin \Gamma$, it holds $a \prec b$. $\text{Rob}(D)$ is the Round Robin matrix for this order.

▶ **Lemma 39.** *Given an LTL formula φ and a Church matrix D, there is a winning joint D-uniform strategy iff there is a winning joint strategy $Rob(D)$.*

Proof. Suppose that there is a winning joint strategy F for Γ that is $\text{Rob}(D)$-uniform. Only the moves of players of the coalition can make F non-D-uniform. But, given the joint strategy for the whole coalition, we cannot reach two configurations that are equal on everything except a labeling of a player of the coalition because our strategies are deterministic. Thus, in practice, F is D-uniform.

Conversely, consider two configurations C^1 and C^2 such that $C^1 \overset{a}{\sim}_{\mathrm{Rob}(D)} C^2$ for some $a \in \Gamma$. We denote by k the length of $C^1(a)$. Then for every $b \in \mathcal{P}$, if $a \prec b$, we have $C^1(b)[: k-1] = C^2(b)[: k-1]$ and otherwise, $C^1(b)[: k] = C^2(b)[: k]$. Because we chose the order so that $a \prec b$ for every $a \in \Gamma$ and $b \notin \Gamma$, we have that $C^1 \overset{a}{\sim}_D C^2$. Thus every strategy D-uniform is $\mathrm{Rob}(D)$-uniform. ◀

D Proof of Theorem 33

▶ **Theorem 33.** *EWS is non-elementary for arbitrary perfect-information matrices.*

Proof. In this proof, we use the notation $F_{\restriction \mathcal{P}\setminus\{a\}}$ to denote the function F restrained to the domain $\mathcal{P}\setminus\{a\}$. Given a dependency matrix D, we decompose the set of Players $\mathcal{P}$ as follows.

$$\begin{aligned} \mathcal{P}_\infty &:= \{a \in \mathcal{P} \mid \text{ There is } b \in \mathcal{P} \text{ such that } D[a,b] = +\infty\} \\ \mathcal{P}_\mathbb{Z} &:= \{a \in \mathcal{P} \mid \text{ For all } b \in \mathcal{P} \text{ it holds } D[a,b] < +\infty\} \end{aligned}$$

We reason by induction on the size of $\mathcal{P}_\infty$. The base case is $|\mathcal{P}_\infty| = 0$. In this case, we can apply Theorem 30.

Suppose that we can decide the EWS problem for matrices with $|\mathcal{P}_\infty| = n$ for some $n \in \mathbb{N}$. We now prove that we can decide the problem for matrices with $|\mathcal{P}_\infty| = n+1$. Consider a matrix D such that $|\mathcal{P}_\infty| = n+1$, a coalition Γ and a QPTL formula ψ. We define an order $\prec$ on $\mathcal{P}_\infty$ that is given as follows. $a \prec b$ iff for all $c \in \mathcal{P}$, if $D[b,c] = +\infty$, then $D[a,c] = +\infty$. Intuitively, $a \prec b$ means that Player a is to play after Player b. Consider Player a, the smallest player for this order. For all players b different than a we have $D[a,b] = +\infty$. We now construct a new instance of the problem by projecting out Player a.

If ($a \in \Gamma$) we state $\Gamma' = \Gamma\setminus\{a\}$ and $\psi' = \exists a.\ \psi$. By inductive hypothesis, we can decide whether there is a joint strategy winning for the entry $\langle D', \Gamma', \psi'\rangle$. Let us prove that $\langle D', \Gamma', \psi'\rangle$ is a positive instance iff $\langle D, \Gamma, \psi\rangle$ is a positive instance. If there is a joint strategy F' for the coalition winning for the entry $\langle D', \Gamma', \psi'\rangle$, then, for every play $(C_1^n)_n, \ldots, (C_k^n)_n$ in the meta game of D', the assignment λ which is the limit of that play satisfy $\langle D', \Gamma', \psi'\rangle$. Therefore, there is an infinite word u_λ such that $\lambda[a \mapsto u_\lambda]$ satisfies ψ'. Then we define the joint strategy F as follows. For every Player $b \neq a$, we set $F_b = F'_b$. For Player a, consider a configuration C such that $\alpha_a^C > 0$. Because of the dependencies of Player a, it holds that $|C(b)| = +\infty$ for every Player $b \neq a$. Let λ be the temporal assignment on $\mathcal{P}\setminus\{a\}$ defined by C. We set $F_a(C) = u_\lambda$. This joint strategy is winning. The converse follows the same idea: if there is F winning for the entry (D, Γ, ψ) then, the joint strategy $F' = F_{\restriction \mathcal{P}\setminus\{a\}}$ is winning for the entry (D', Γ', ψ').

If ($a \notin \Gamma$) we state $\psi' = \forall a.\ \psi$. We then decide the instance $\langle D', \Gamma, \psi'\rangle$. Joint strategies for the coalition translate naturally between the two instances. If F' is winning for $\langle D', \Gamma, \psi'\rangle$, then every play in the outcome of F' satisfies $\forall a \psi$. Then, by defining $F(C) = F'(C_{\restriction \mathcal{P}\setminus a})$ we define a wining strategy for $\langle D, \Gamma, \psi\rangle$. The converse is similar. ◀