

The complexity of game isomorphism^{☆,☆☆}

Joaquim Gabarró, Alina García, Maria Serna*

*ALBCOM Research Group
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona 1-3, Ω Building, E-08034 Barcelona, Spain*

Abstract

We address the question of whether two multiplayer strategic games are equivalent and the computational complexity of deciding such a property. We introduce two notions of isomorphisms, *strong* and *weak*. Each one of those isomorphisms preserves a different structure of the game. Strong isomorphisms are defined to preserve the utility functions and Nash equilibria. Weak isomorphisms preserve only the player's preference relations and thus pure Nash equilibria. We show that the computational complexity of the game isomorphism problem depends on the level of succinctness of the description of the input games but it is independent on which of the two types of isomorphisms is considered. Utilities in games can be given succinctly by Turing machines, boolean circuits or boolean formulas, or explicitly by tables. Actions can be given also explicitly or succinctly. When the games are given in *general form*, we assume an explicit description of actions and a succinct description of utilities. We show that the game isomorphism problem for general form games is equivalent to the circuit isomorphism when utilities are described by TMs and to the boolean formula isomorphism problem when utilities are described by formulas. When the game is given in explicit form, we show that the game isomorphism problem is equivalent to the graph isomorphism problem.

Key words: Game isomorphism, succinct representations, formula games, boolean formulas, computational complexity, circuit isomorphism, boolean formula isomorphism, graph isomorphism

1. Introduction

Game Theory provides the mathematical tools and models to analyze strategic situations in which multiple participants interact or affect each others. In the last years a huge amount of research has been devoted to explore the usefulness of Game Theory in situations arising on the Internet. In those situations many participants interact with competing goals and therefore can be modelled by strategic or cooperative games. Computational issues arising in this framework is one of the main objectives of the Algorithmic Game Theory community [19, 17, 26].

The informal idea of *strategic equivalence* [12] has been widely discussed and explored along the history of Game Theory. Traditionally the notion of equivalence is studied at different levels using different types of isomorphism, depending on the family of games and the structural properties to be preserved. In 1951, J. Nash [16] gave a definition of automorphism between strategic games. J.C Harsanyi and R. Selten have introduced other definitions of isomorphism [11] for strategic games. Equivalence by the way of transformations to a common form have been considered in [7]. More recently, B. Peleg, J. Rosemuller, and

[☆]This work was partially supported by FET pro-active Integrated Project 15964 (AEOLUS), the projects TIN2007-66523 (FORMALISM) and TIN2005-25859-E of "Ministerio de Ciencia e Innovación y el Fondo Europeo de Desarrollo Regional".

^{☆☆}Part of this work was presented as *On the Complexity of Game Isomorphism. MFCS 2007: LNCS 4708:559-571, 2007*

*Corresponding author

Email addresses: gabarro@lsi.upc.edu (Joaquim Gabarró), agarcia@lsi.upc.edu (Alina García), mjserna@lsi.upc.edu (Maria Serna)

P. Sudhölter [20, 29] consider a notion of isomorphisms for strategic and extensive games with incomplete information, another notion of isomorphism for extensive games has been introduced in [8]. Further results for cooperative games can be found in [9].

In this paper we are interested in the computational aspects of game equivalence for the case of strategic games. Our motivation for selecting strategic games is twofold. First, strategic games are used as ingredient of more complicated games, but usually there is a way to transform any game into a strategic game. Furthermore, in [7] equivalence between extensive games is defined in terms of strategic games. Therefore strategic games are the first game structure to start analyzing game equivalence. Second, the combinatorial structure of an strategic game is simple enough to allow such kind of analysis by comparison with isomorphism on other combinatorial structures. In particular, to relate the problems with isomorphisms for well studied structures as graphs [13], boolean formulas or boolean circuits [1, 4, 6].

In defining a concrete equivalence between games we have to pay attention to the structural properties that are preserved in equivalent games. In this paper, we consider two versions of isomorphisms that preserve at different levels the structure of the Nash equilibria. A *strong isomorphism* preserves utilities corresponding to the notion introduced in [16]. A *weak isomorphism* preserves preferences. Each of them requires to preserve less information about the relative structure of profiles while preserving still the structure of the Nash or pure Nash equilibria. More precisely, as we will show later, strong isomorphisms preserve pure and mixed Nash equilibria, while weak isomorphisms only preserve pure Nash equilibria.

In this paper we are interested in the computational complexity of deciding whether two games are equivalent. We consider two problems related to isomorphisms. In the ISISO problem, given two games Γ and Γ' and a mapping ψ we have to decide whether ψ is an isomorphism. In the ISO problem we have to decide whether two games are isomorphic. In order to study the computational aspects of isomorphism problems on strategic games, we need first to determine the way in which games and morphisms are represented as inputs to a program. For the representation of strategic games we adopt the proposal given in [2] and consider the following two representation, each with a different level of succinctness. When a game is given in *general* form the actions are listed explicitly but utilities and mappings are given by deterministic Turing machines. In the *explicit* case utilities are stored in tables. In both cases morphisms are always represented by tables. This is not a restriction as in polynomial time we can transform a morphism representation by Turing machines into a tabular representation by tables, because the actions are given explicitly.

The main contributions of the paper are the following problem classification:

- The ISISO problem is coNP-complete, for games given in general form, and belongs to NC when games are given in explicit form.
- The ISO problem belong to Σ_2^P , for games given in general form, and to NP when games are given in explicit form.
- The ISO problem is equivalent to the boolean circuit isomorphism problem, for games in general form, and to the graph isomorphism problem, for games given in explicit form.

The above results hold independently of the type of isomorphism considered, observe that the boolean circuit isomorphism problem is believed not to be Σ_2^P -hard [1], and that the graph isomorphism problem is conjectured not to be NP-hard [13]. Therefore the same results are valid for the ISO problem.

Besides the above generic forms of representing games we will also consider another particular class of strategic games, that we call *formula games*. Our formula games are (as we will show) equivalent in power of representation to a subfamily of the *weighted boolean formula games* introduced in [14]. We analyze the complexity of the ISO problem when the games correspond to a general form, that is, the number of bits controlled by each player is a constant. For formula games in general form we show that the ISO problem is equivalent to boolean formula isomorphism. Recall that the complexity of the boolean formula isomorphism problem is the same as that of circuit isomorphism, however it is conjectured that both problems are not equivalent.

The paper is organized as follows. In Section 2 we introduce the definitions, problems and representations that will be used through the paper, we also introduce the notion of game mappings and the definition of the

different notions of game isomorphism in which we are interested. In Section 3 we provide the complexity results for the case of strong isomorphism. Section 4 is devoted to the weak isomorphism. Finally, Section 5 is devoted to state further results and open problems related to isomorphism and game classification. The paper concludes with an appendix where some of the most technical details of the proofs are given.

2. Definitions and preliminaries

In this section we provide the definitions and terminology used in the paper. We start with strategic games and their representations. We continue with game mapping and the definition of the two types of isomorphism considered in this work. We finalize this section with the definition of several computational problems.

Strategic games. We start stating the mathematical definition of strategic game as given in [18].

Definition 1. A strategic game Γ is a tuple $(N, (A_i)_{i \in N}, (u_i)_{i \in N})$. The set of players is $N = \{1, \dots, n\}$. Player $i \in N$ has a finite set of actions A_i , we note a_i any action belonging to A_i . The elements $a = (a_1, \dots, a_n) \in A_1 \times \dots \times A_n$ are the strategy profiles. The utility (or payoff) function u_i , for each player $i \in N$, is a mapping from $A_1 \times \dots \times A_n$ to the rationals.

In the context of computational complexity it is very important to fix how games are represented as problem inputs. In all the different types of representations we will always assume that the actions for each player are giving explicitly, by listing all its components. This leads us with two types of representations depending on whether the utilities are given explicitly or succinctly.

Our first representation is the generic representation of strategic games given in [2] where the pay-off functions of a game is described by a deterministic Turing Machine.

Strategic game in general form. The game Γ is given by a tuple

$$\Gamma = \langle 1^n, A_1, \dots, A_n, M, 1^t \rangle.$$

The game has n players, and for each player i , where $1 \leq i \leq n$, their set of actions A_i is given by listing all its elements. Given a strategy profile and a player i , $1 \leq i \leq n$, $u_i(a)$ is the output of M on input $\langle a, i \rangle$ after t steps.

In [2] a more succinct representation of games is obtained by defining implicitly the sets of actions A_i as subsets of $\{0, 1\}^m$, in such a case a game Γ is given by $\langle 1^n, 1^m, M, 1^t \rangle$, which is called *implicit form*. For reasons that we will clarify later, we do not consider strategic games in implicit form.

Our second representation assumes that the pay-off functions are given explicitly by means of a table.

Strategic game in explicit form. A game is given by a tuple

$$\Gamma = \langle 1^n, A_1, \dots, A_n, T \rangle,$$

where T is a table of dimensions $|A_1| \times \dots \times |A_n| \times n$. Given a strategy profile and a player i , $1 \leq i \leq n$, $u_i(a) = T[a][i]$.

In the following we consider strategic games in which the utility functions are described by boolean formulas. In [5], player i has a goal φ_i to fulfill. Goals are usually described by boolean formulas. The utility of the player is binary. It is 1 if the goal is satisfied and 0 otherwise. Along the lines suggested by circuit games [25] we consider the following family of strategic games, whose representation is close to a game given in general form [2].

Formula game in general form. A game is given by a tuple

$$\Gamma = \langle 1^n, A_1, \dots, A_n, 1^\ell, (\varphi_{i,j})_{1 \leq i \leq n, 0 \leq j < \ell} \rangle$$

The set of actions for player i , $1 \leq i \leq n$ is $A_i = \{0, 1\}^{m_i}$. The utility of player i is given by the boolean formulas $\varphi_{i,j}(a_1, \dots, a_n) \in \{0, 1\}$, $0 \leq j < \ell$, by the equation $u_i(a_1, \dots, a_n) = \sum_{0 \leq j < \ell} \varphi_{i,j}(a_1, \dots, a_n) 2^j$.

Another model for strategic games that use boolean formula was introduced in [14], the *weighted boolean formula games*. Whose definition is as the following:

Weighted boolean formula game (WBF) [14]. A game is given by a tuple

$$\Gamma = \langle 1^n, 1^m, 1^r, 1^\ell, (\mathfrak{F}_i)_{1 \leq i \leq n} \rangle$$

where player i has the set of actions $A_i = \{0, 1\}^m$. For each player i , there is a set $\mathfrak{F}_i = \{(f_{i,1}, w_{i,1}), \dots, (f_{i,r}, w_{i,r})\}$ such that $f_{i,j} : A_1 \times \dots \times A_n \rightarrow \{0, 1\}$, $1 \leq j \leq r$, are boolean formulas and $w_{i,j} \in \{0, 1\}^\ell$, $1 \leq j \leq r$. The utility for player i is computed by the formula $u_i(a_1, \dots, a_n) = \sum_{(f,w) \in \mathfrak{F}_i} w \cdot f(a_1, \dots, a_n)$.

In the above definition the set of actions are described implicitly, in the rest of the paper we will restrict to WBF in which the set of actions are described explicitly. Following our previous notation we will refer to such games as *weighted boolean formula games in general form*. Formula games and WBF in general form are equivalent as, given a WBF we can build in polynomial time in the size of Γ a Formula Game Γ' with the same utilities and conversely. The details of the proof are given in the Claim 1 in Appendix A. Thus our results for formula games will apply also to WBF.

In the case that the number of players is constant, with respect to the number of actions, we can obtain an explicit representation in polynomial time from a given general form representation, otherwise the transformation requires exponential time.

Game mappings. We consider game mappings that provide the way to associate players and their actions in one game to players and actions in the other, as usual those mappings are independent of the utilities. We adapt notations and definitions given in [20, 29].

Definition 2. Given $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ and $\Gamma' = (N, (A'_i)_{i \in N}, (u'_i)_{i \in N})$, a game mapping ψ from Γ to Γ' is a tuple $\psi = (\pi, (\varphi_i)_{i \in N})$ where π is a bijection from N to N , the player's bijection, and, for any $i \in N$, φ_i is a bijection from A_i to $A'_{\pi(i)}$, the i -th player actions bijection.

Observe that the player bijection identifies player $i \in N$ with player $\pi(i)$ and the corresponding actions bijection φ_i maps the set of actions of player i to the set of actions of player $\pi(i)$. A game mapping ψ from Γ to Γ' induces, in a natural way, a bijection from $A_1 \times \dots \times A_n$ to $A'_1 \times \dots \times A'_n$ where strategy profile (a_1, \dots, a_n) is mapped into the strategy profile (a'_1, \dots, a'_n) defined as $a'_{\pi(i)} = \varphi_i(a_i)$, for all $1 \leq i \leq n$. We note this mapping as $\psi(a_1, \dots, a_n) = (a'_1, \dots, a'_n)$, overloading the use of ψ . A mixed strategy profile $p = (p_1, \dots, p_i, \dots, p_n)$ is given by probabilities p_i on A_i (such that $\sum_{a_i \in A_i} p_i(a_i) = 1$) for $1 \leq i \leq n$. A game mapping ψ also induces a mapping $\psi(p_1, \dots, p_n) = (p'_1, \dots, p'_n)$ such that $p'_{\pi(i)}$ is a probability on $A'_{\pi(i)}$ defined by $p'_{\pi(i)}(\varphi_i(a_i)) = p_i(a_i)$. Isomorphisms are game mappings fulfilling some additional restrictions on utilities or preferences as we will see later.

In order to describe a game mapping, we consider the less succinct approach. Observe that for the information on each game, we have to keep only the set of actions for each player.

Game mapping in explicit form. All the components of the mapping is given explicitly, action sets are given by listing all its elements and permutations are given by tables, that is

$$\psi = \langle 1^n, A_1, \dots, A_n, A'_1, \dots, A'_n, T_\pi, T_{\varphi_1}, \dots, T_{\varphi_n} \rangle$$

where $T_\pi, T_{\varphi_1}, \dots, T_{\varphi_n}$ are tables such that $T_{\varphi_i}[a_i] = a'_{\pi(i)}$.

We have not considered the description of a mapping by Turing machines,

$$\psi = \langle 1^n, A_1, \dots, A_n, A'_1, \dots, A'_n, M_\pi, M_{\varphi_1}, \dots, M_{\varphi_n}, 1^t \rangle$$

because in such a case we can construct an explicit coding of ψ with size bounded by $2|\psi|$ in time $|\psi|^2$.

Observe that it is not trivial to consider an adequate description of mapping associated to a set with exponentially many actions, in view of that we are not considering implicit form representation for mappings and games.

Game isomorphism. We start defining the stronger version of an isomorphism introduced by J. Nash [16] (see also [20, 29]).

Definition 3. Given two strategic games $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ and $\Gamma' = (N, (A'_i)_{i \in N}, (u'_i)_{i \in N})$, a game mapping $\psi = (\pi, (\varphi_i)_{i \in N})$ is called a strong isomorphism $\psi : \Gamma \rightarrow \Gamma'$ when, for any player $1 \leq i \leq n$ and any strategy profile a , it holds $u_i(a) = u'_{\pi(i)}(\psi(a))$. In the particular case that Γ' is Γ a strong isomorphism is called a strong automorphism.

In Example 1 we provide an example of strong isomorphism.

Example 1. Given the following games Γ and Γ'

$$\begin{array}{ccc}
 & \begin{array}{cc} \text{Player 2} \\ l & r \end{array} & \\
 \begin{array}{c} \text{Player 1} \\ t \\ b \end{array} & \begin{array}{|cc|} \hline 0,0 & 0,1 \\ \hline 1,1 & 1,0 \\ \hline \end{array} & \xrightarrow{\psi} & \begin{array}{c} \text{Player 1} \\ t' \\ b' \end{array} & \begin{array}{|cc|} \hline 1,0 & 0,1 \\ \hline 0,0 & 1,1 \\ \hline \end{array} \\
 & \Gamma & & & \Gamma'
 \end{array}$$

Consider the morphism $\psi : \Gamma \rightarrow \Gamma'$ defined as $\psi = (\pi, \varphi_1, \varphi_2)$ where $\pi = (1 \rightarrow 2, 2 \rightarrow 1)$, $\varphi_1 = (t \rightarrow l', b \rightarrow r')$ and $\varphi_2 = (l \rightarrow b', r \rightarrow t')$. This morphism maps strategy profiles as: $\psi(t, l) = (b', l')$, $\psi(t, r) = (t', l')$, $\psi(b, l) = (b', r')$ and $\psi(b, r) = (t', r')$. Therefore it is a strong isomorphism.

Given a strong isomorphism ψ between Γ and Γ' , observe that a mixed strategy profile p is a Nash equilibrium in Γ iff $\psi(p)$ is a Nash equilibrium in Γ' and, of course, the same holds for pure Nash equilibria. Thus the bijection induced by strong isomorphisms on the set of mixed strategy profiles preserves the structure of the Nash equilibria. Observe that, furthermore, a strong isomorphism induces an isomorphism among the Nash dynamics graphs of the two games.

There are several ways to relax the notion of strong isomorphism while maintaining the structure of Nash equilibria. For instance, Harsanyi and Selten [11] substitute $u_{\pi(i)}(\psi(a)) = u_i(a)$ for $u_{\pi(i)}(\psi(a)) = \alpha_i u_i(a) + \beta_i$. In order to generalize this approach we consider, following [18], game isomorphism in which the preference relations $(\preceq_i)_{i \in N}$ induced by the utility functions are preserved. We note *strict preference* as usual, $a \prec_i a'$ iff $a \preceq_i a'$ but not $a' \preceq_i a$. We note *indifference* by $a \sim_i a'$, as usual indifference occurs when $a \preceq_i a'$ and $a' \preceq_i a$ holds. The definition of isomorphism can be adapted to respect only preference relations instead of utility functions.

Definition 4. A weak isomorphism $\psi : \Gamma \rightarrow \Gamma'$ is a mapping $\psi = (\pi, (\varphi_i)_{i \in N})$ such that any triple a, a' and i verifies: $a \preceq_i a'$ iff $\psi(a) \preceq_{\pi(i)} \psi(a')$.

Example 2. We consider $a \preceq_i a'$ iff $u_i(a) \leq u_i(a')$. Following there is an example of weak isomorphism $\psi = (\pi, \varphi_1, \varphi_2)$.

$$\begin{array}{ccc}
 & \begin{array}{cc} \text{Player 2} \\ l & r \end{array} & \\
 \begin{array}{c} \text{Player 1} \\ t \\ b \end{array} & \begin{array}{|cc|} \hline 0,0 & 0,1 \\ \hline 1,1 & 1,0 \\ \hline \end{array} & \xrightarrow{\psi} & \begin{array}{c} \text{Player 1} \\ t' \\ b' \end{array} & \begin{array}{|cc|} \hline 3,3 & 2,2 \\ \hline 2,3 & 3,2 \\ \hline \end{array} \\
 & \Gamma & & & \Gamma'
 \end{array}$$

where $\pi = (1 \rightarrow 2, 2 \rightarrow 1)$, and $\varphi_1 = (t \rightarrow r', b \rightarrow l')$, $\varphi_2 = (l \rightarrow t', r \rightarrow b')$. Observe that $u_i(a)$ and $u_i(\psi(a))$ are not even related by a linear function.

Weak isomorphisms preserve preferences for any pair of strategy profiles and any player, therefore maintains the structure of pure Nash equilibria.

We consider the following computational problems related to games and morphisms.

Is Game Isomorphism (ISISO). Given two games Γ, Γ' and a game mapping $\psi : \Gamma \rightarrow \Gamma'$, decide whether ψ is a game isomorphism.

Game Isomorphism (ISO). Given two games Γ, Γ' , decide whether there exists a game isomorphism between Γ and Γ' .

The two problems can be formulated for strong and weak isomorphism introduced above and also for games in general form (strategic or boolean formula) or games in explicit form. The game isomorphism problem can also be considered for the case in which $n = 1$. For this particular case, the isomorphism problem is computationally easy.

Theorem 1. *The ISO problem for games with one player is polynomial time solvable, for strong and weak isomorphism, and for general form strategic and formula games and for explicit form games.*

Proof. Consider a 1-player game $\Gamma(\{1\}, A_1, (u_1))$. Consider the vector $x = (x_1, \dots, x_m)$ where $x_i = u_1(i)$ and $m = |A_1|$. Define its characteristic vector $S(\Gamma)$ as the vector obtained after sorting x in increasing order. Then we have that two 1-player games are strongly isomorphic iff their characteristic vectors are identical.

For the case of weak isomorphism the condition is equivalent to the fact that the relative order of two consecutive elements is the same in both characteristic vectors.

The vector can be obtained in polynomial time for any of the considered game representation, and thus the problems can be solved in polynomial time. \square

Assumption In view of the above result we will assume for the rest of the paper that all the games have at least two players.

Other problems. Our coNP-hardness results follow from reductions from the following coNP-complete problem [10]:

Validity problem (VALIDITY): Given a boolean formula F decide whether F is satisfiable by all truth assignments.

We also consider the following problems on boolean circuits. Recall that two circuits $C_1(x_1, \dots, x_n)$ and $C_2(x_1, \dots, x_n)$ are *isomorphic* if there is a permutation π of $\{1, \dots, n\}$ such that, for any truth assignment $x \in \{0, 1\}^n$, $C_1(x) = C_2(\pi(x))$.

Boolean circuit isomorphism problem (CIRCUITISO): Given two boolean circuits C_1 and C_2 decide whether C_1 and C_2 are isomorphic.

A related problem is based on the notion of congruence. A *congruence* between two circuits on n variables, $C_1(x_1, \dots, x_n)$ and $C_2(x_1, \dots, x_n)$ is a mapping $\psi = (\pi, f_1, \dots, f_n)$, where π is a permutation of $\{1, \dots, n\}$ and, for any $1 \leq i \leq n$, f_i is a permutation on $\{0, 1\}$ (either the identity or the negation function). As in the case of game morphism, the image $\psi(x)$ is obtained by permuting the positions of the input bits, according to permutation π , and then applying to any bit i the permutation f_i .

Boolean circuit congruence problem (CIRCUITCONG): Given two circuits C_1 and C_2 decide whether C_1 and C_2 are congruent.

The CIRCUITISO problem has been studied by B. Borchert, D. Ranjan and F. Stephan in [6], among many other results they show that CIRCUITISO and CIRCUITCONG are equivalent. It is known that CIRCUITISO $\in \Sigma_2^P$, but M. Agrawal and T. Thierauf prove that it cannot be Σ_2^P -hard unless the polynomial hierarchy collapses (see Corollary 3.5 in [1]).

We also consider the isomorphism and congruence problems for boolean formulas. Recall that two formulas $\Phi_1(x_1, \dots, x_n)$ and $\Phi_2(x_1, \dots, x_n)$ are *isomorphic* if there is a permutation π of $\{1, \dots, n\}$ such that, for any truth assignment $x \in \{0, 1\}^n$, $C_1(x) = C_2(\pi(x))$. They are *congruent* if there is a mapping $\psi = (\pi, f_1, \dots, f_n)$, where π is a permutation of $\{1, \dots, n\}$ and, for any $1 \leq i \leq n$, f_i is a permutation on $\{0, 1\}$ such that, for any truth assignment $x \in \{0, 1\}^n$, $C_1(x) = C_2(\psi(x))$.

Boolean formula isomorphism problem (FORMULAISO): Given two boolean formulas Φ_1 and Φ_2 decide whether Φ_1 and Φ_2 are isomorphic.

Boolean formula congruence problem (FORMULACONG): Given two boolean formulas Φ_1 and Φ_2 decide whether Φ_1 and Φ_2 are congruent.

B. Borchert, D. Ranjan and F. Stephan in [6] show that FORMULAISO and FORMULACONG are equivalent. It is known that FORMULAISO $\in \Sigma_2^P$. but it cannot be Σ_2^P -hard unless the polynomial hierarchy collapses (see Corollary 3.4 in [1]).

Two graphs are *isomorphic* if there is a one-to-one correspondence between their vertices and there is an edge between two vertices of one graph if and only if there is an edge between the two corresponding vertices in the other graph.

Graph isomorphism (GI): Given two graphs, decide whether they are isomorphic.

It is well known that GI is not expected to be NP-hard [13].

Notation. We finish this section with some additional definitions and notation that will be used in this paper.

A *binary actions game* is a game in which the set of actions for each player is $\{0, 1\}$. A *binary game* is a binary actions game in which the utility functions range is $\{0, 1\}$. We will need to construct binary actions games associated to general games, for doing so we use a *binify* process on the strategies of the original game.

Given a strategic game $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$, assume without loss of generality that $N = \{1, \dots, n\}$ and that, for any $i \in N$, $A_i = \{1, \dots, k_i\}$. We “binify” an action $j \in A_i$, coding it with k_i bits, as $\text{binify}(j) = 0^{j-1}10^{k_i-j}$. The binify process can be used in a strategy profile, given $a = (a_1, \dots, a_n)$, we write $\text{binify}(a) = (\text{binify}(a_1), \dots, \text{binify}(a_n))$. Observe that by setting $k = \sum_{i \in N} k_i$, we have $\text{binify}(a) \in A' = \{0, 1\}^k$. We define $\text{good}(A') = \{\text{binify}(a) | a \in A\}$ and $\text{bad}(A') = A' \setminus \text{good}(A')$. Note that $\text{binify} : A \rightarrow \text{good}(A')$ is a bijection and therefore the inverse function is also a bijection.

Example 3. Given Γ with 3 players $A_1 = A_3 = \{1, 2\}$ and $A_2 = \{1, 2, 3\}$ we have $\text{binify}(1, 2, 2) = (10, 010, 01) = (1, 0, 0, 1, 0, 0, 1)$ and $\text{binify}^{-1}(10, 010, 01) = (1, 2, 2)$.

3. Complexity results for strong isomorphisms

Let us start with the complexity for ISISO problem in the case of strategic games.

Theorem 1. *The ISISO problem for strong isomorphisms is coNP-complete for strategic games in general form and for boolean formula games in general form. The problem belongs to NC whenever the games are given in explicit form. The strong isomorphism is given in both cases in explicit form.*

Proof. Let us first assume that the games are given in general form. In this case the input is formed by two games $\Gamma = \langle 1^n, A_1, \dots, A_n, M_1, 1^{t_1} \rangle$ and $\Gamma' = \langle 1^n, A'_1, \dots, A'_n, M_2, 1^{t_2} \rangle$ and a game mapping between the two games $\psi = \langle 1^n, A_1, \dots, A_n, A'_1, \dots, A'_n, T_\pi, T_{\varphi_1}, \dots, T_{\varphi_n} \rangle$. Then we have $\langle \Gamma, \Gamma', \psi \rangle \in \text{ISISO}$ iff

$$\forall (a_1, \dots, a_n) \in A_1 \times \dots \times A_n \forall i \in N \quad u'_{\pi(i)}(\psi(a_1, \dots, a_n)) = u_i(a_1, \dots, a_n).$$

Therefore ISISO belongs to coNP because it is enough to guess a strategy profile $a = (a_1, \dots, a_n)$ and a player i , using polynomial space, and check $u'_{\pi(i)}(\psi(a)) \neq u_i(a)$ in polynomial time.

To prove hardness we define two games, the first one is associated to a boolean formula, and a mapping between them. Given a boolean formula F with n variables, consider the following game.

WINWHENTRUE(F): This game has n players, $N = \{1, \dots, n\}$, and player i has $A_i = \{0, 1\}$. All the players $1 \leq i \leq n$ have the same utility $u_i(a_1, \dots, a_n) = F(a_1, \dots, a_n)$.

The game is coded in general form as $\langle 1^n, A_1, \dots, A_n, \text{Eval}, 1^{\log n(n+|F|)^2} \rangle$ where **Eval** is a TM that evaluates a formula in time $O((n+|F|)^2)$, we provide some additional time to get rid of the constant. Observe that this codification can be obtained in polynomial time given F .

ALWAYSWIN: This game has n players, $N = \{1, \dots, n\}$, and player i has $A_i = \{0, 1\}$. All the players $1 \leq i \leq n$ have the same utility $u_i(a_1, \dots, a_n) = 1$.

This game can be represented in general form as $\langle 1^n, A_1, \dots, A_n, \text{One}, 1^{n+1} \rangle$ where **One** is a TM that, after reading the input, outputs 1 in time $n + 1$. Furthermore the representation can be computed in $O(n)$ time.

IDENTITY: This mapping combines the identity function on $N = \{1, \dots, n\}$ with the identity function on $\{0, 1\}$.

The mapping is represented by $\langle 1^n, A_1, \dots, A_n, A_1, \dots, A_n, id_\pi, id_1, \dots, id_n \rangle$, where $id_\pi(i) = i$ and $id_i(a_i) = a_i$ for $i \leq i \leq n$. Observe that such a representation can be obtained in time $O(n)$.

We claim $\langle \text{WINWHENTRUE}(F), \text{ALWAYSWIN}, \text{IDENTITY} \rangle \in \text{ISISO}$ iff F is valid. When F is a valid formula both games have the same utility functions, therefore the mapping **IDENTITY** is a strong isomorphism. When F is not valid there exists x_1, \dots, x_n such that $F(x_1, \dots, x_n) = 0$, therefore the utility of this strategy profile for player 1 in **WINWHENTRUE** is 0, while the same player gets utility 1 in the **ALWAYSWIN** game. Therefore, **IDENTITY** is not a strong isomorphism.

When Γ_1, Γ_2 and ψ are formula games in general form, the same arguments shows that the ISISO problem is **coNP**-complete.

When Γ_1, Γ_2 and ψ are given in explicit form the strong isomorphism can be verified in polylogarithmic parallel. We have to compute in parallel for each $a = (a_1, \dots, a_n)$ the corresponding $a' = (a'_1, \dots, a'_n)$ such that $a'_{T_\pi[i]} = T_{\varphi_i}[a_i]$ and test if $T_1[a, i] = T_2[a', T_\pi[i]]$ for each player i . \square

Our next step is to provide upper bounds for the complexity of the ISO problem. Later on we show that the bounds are best possible for the ISO problems.

Theorem 2. *The ISO problem for strong morphism belong to Σ_2^P for strategic and formula games in general form. The problem belong to **NP** when the games are given in explicit form.*

Proof. Let us consider first the membership proofs. We define a nondeterministic algorithm working in polynomial space/time depending on the representation of the input game. Assume that we are given two strategic games $\Gamma_1 = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ and that $\Gamma_2 = (N, (A'_i)_{i \in N}, (u'_i)_{i \in N})$, by definition, there is a strong isomorphism between Γ_1 and Γ_2 iff

$$\exists \psi = (\pi, \varphi_1, \dots, \varphi_n) \forall a \in A_1 \times \dots \times A_n \forall i \in N u_i(a) = u'_{\pi(i)}(\psi(a)),$$

where ψ is a mapping of Γ_1 to Γ_2 . Observe that it is possible to guess, using polynomial space, an isomorphism $\psi = \langle A_1, \dots, A_n, A'_1, \dots, A'_n, T_\pi, T_{\varphi_1}, \dots, T_{\varphi_n} \rangle$. Furthermore, given a strategy profile $a = (a_1, \dots, a_n)$ it is possible to compute $\psi(a)$ in polynomial time just doing $a'_{T_\pi[i]} = T_{\varphi_i}[a_i]$. To check the correctness of the guess, we need to verify that, for every player i and strategy profile a , it holds $u_i(a) = u'_{\pi(i)}(\psi(a))$.

When the games are given in general form the strategy profile can be represented in polynomial space and the test performed in polynomial time, both for utilities given by TM or formulas, therefore the ISO problem belongs to Σ_2^P . When both games are given in explicit form, the number of strategy profiles is polynomial in the size of the input and therefore we can check for all a the condition $u_i(a) = u'_i(\psi(a))$ in polynomial time once the mapping has been guessed. Therefore, the ISO problem belongs to **NP**. \square

We prove that ISO is equivalent to **CIRCUITISO** for games in general form. This is done through a series of reductions transforming the game while preserving the existence of strong isomorphism. First, we show how to construct corresponding isomorphic binary actions games. Second, we show the construction from a binary action game of a binary game preserving isomorphism. Finally, we show the equivalence with the Boolean circuit congruence. All the transformations presented in the paper can be computed in polynomial time, thus we avoid to mention this fact all through the paper. Let us start with the first transformation.

We start defining a construction for the first reduction that makes use of the binify process. Let $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ be a strategic game. In this case we get $k = \sum_{i \in N} k_i$, where $k_i = |A_i|$. The binify process can be used in a strategy profile, given $a = (a_1, \dots, a_n) \in A$, we write $\text{binify}(a) = (\text{binify}(a_1), \dots, \text{binify}(a_n))$. Recall that $\text{good}(A') = \text{binify}(A)$.

$\text{BINARYACT}(\Gamma, \mu) = (N', (A'_i)_{i \in N'}, (u'_i)_{i \in N'})$
where $N' = \{1, \dots, k\}$ and, for any $i \in N'$, $A'_i = \{0, 1\}$ and thus the set of action profiles is $A' = \{0, 1\}^k$.
The players are partitioned into B_1, \dots, B_n blocks. Block i is formed by k_i players. Given $i \in B_j$ we say that i belongs to block j of players and write $\text{block}(i) = j$. The utilities are defined by

$$u'_i(a') = \begin{cases} u_{\text{block}(i)}(\text{binify}^{-1}(a')) & \text{if } a' \in \text{good}(A'), \\ \mu & \text{if } a' \in \text{bad}(A'). \end{cases}$$

Notice that, for $a \in A$, $u'_i(\text{binify}(a)) = u_{\text{block}(i)}(a)$, furthermore, all the players in a given block have the same utility function. Each strategy profile a' in $\text{BINARYACT}(\Gamma, \mu)$ can be factorized giving the actions taken by the k players as $a' = (a'_1, \dots, a'_k)$ or grouping the actions according to the blocks B_1, \dots, B_n as $a' = (b_1, \dots, b_n)$ where $b_i \in \{0, 1\}^{k_i}$. The value μ will be selected to create a gap on the utility that separates the profiles in $\text{BINARYACT}(\Gamma, \mu)$ that codify correctly a profile of Γ , from those that do not.

Example 4. We give an example of the transformation from Γ to $\text{BINARYACT}(\Gamma, \mu)$. We take as Γ a version of BS game with nonzero utilities and setting $\mu = 0$ we have:

		Player 2	
		1	2
Player 1	1	3, 2	1, 1
	2	1, 1	2, 3
BS game			

A'	u_1	u_2	u_3	u_4
1010	3	3	2	2
1001	1	1	1	1
0110	2	2	3	3
0110	1	1	1	1
$a' \in \text{bad}(A')$	0	0	0	0

In the BS game $A_1 = A_2 = \{1, 2\}$ and $\text{binify}(1) = 10$, $\text{binify}(2) = 01$. Therefore $\text{good}(A') = \{1010, 1001, 0110, 0101\}$ and $\text{bad}(A') = \{0, 1\}^4 \setminus \text{good}(A')$. The game $\text{BINARYACT}(BS, 0)$ has $N' = \{1, 2, 3, 4\}$. The partition of players into blocks is given by $B_1 = \{1, 2\}$ and $B_2 = \{3, 4\}$.

Given a good strategy profile a and a player i we compute $u'_i(a)$ as follows. Suppose $a = 1010 = (\text{binify}(1), \text{binify}(1))$ and $i = 4$. As player 4 belongs to B_2 it holds $\text{block}(4) = 2$ and $u'_4(1010) = u_4(\text{binify}(1), \text{binify}(1)) = u_{\text{block}(4)}(1, 1) = u_2(1, 1) = 2$.

Now we provide the reduction from the ISO problem for strong isomorphism to the same problem for binary actions games. We provide here the main arguments of the proof and delay the more technical details to Appendix B.

Lemma 1. Let Γ_1, Γ_2 be two strategic games given in general form and let t be $\max\{t_1, t_2\}$, where t_i , $1 \leq i \leq 2$, is the time allowed to the utility TM of the game Γ_i . There is a strong isomorphism between Γ_1 and Γ_2 iff there is a strong isomorphism between the games $\text{BINARYACT}(\Gamma_1, \mu)$ and $\text{BINARYACT}(\Gamma_2, \mu)$ where $\mu = -2^t$.

Proof. When M is a TM computing the utilities in time t we have $|u_i(a)| \leq t$ and when the output is in binary, $-2^t \leq u_i(a) \leq 2^t$. Moreover, when M is a TM computing the utilities in time t , we can construct a TM M' computing the utilities in time tk . Given Γ and Γ' with utilities computed in times t and t' and taking $t = \max\{t, t'\}$ and $\mu = -2^t$ we can find a TMS for $\text{BINARYACT}(\Gamma, \mu)$ and $\text{BINARYACT}(\Gamma', \mu)$ computing utilities in $O(t)$. Furthermore a description of both machines can be obtained in polynomial time.

Given a strong isomorphism $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ of Γ into Γ' , let us define a mapping $\psi' = (p, f_1, \dots, f_k)$ of $\text{BINARYACT}(\Gamma, \mu)$ into $\text{BINARYACT}(\Gamma', \mu)$. Suppose that in ψ it holds $\pi(i) = j$, then as $\varphi_i : A_i \rightarrow A'_j$ is a bijection, by construction blocks B_i and B'_j in binary games have the same cardinality and we ask p to be a bijection $p : B_i \rightarrow B'_j$. Writing $A_i = A'_j = \{1, \dots, \ell\}$ and $B_i = \{i_1, \dots, i_\ell\}$ and $B'_j = \{j_1, \dots, j_\ell\}$, the action bijection $\varphi_i(p) = q$, $1 \leq p \leq \ell$, induces the bijection $p(i_p) = j_q$ between both blocks. This concludes the definition of p . In ψ' we take all the f_i for $1 \leq i \leq k$ to be identities. It holds that ψ' is a strong isomorphism which is proved as Claim 2 in Appendix B.

For the reverse implication, assume that $\psi' = (p, f_1, \dots, f_k)$ is an strong isomorphism between the games $\text{BINARYACT}(\Gamma_1, \mu)$ and $\text{BINARYACT}(\Gamma_2, \mu)$ having players N'_1 and N'_2 with $N'_1 = N'_2$. The strategy profiles in both binary action games are A'_1 and A'_2 . Now we can define a mapping $(\pi, \varphi_1, \dots, \varphi_n)$ of Γ_1 to Γ_2 . The

permutation of players π mimics the block permutation induced by p , thus if B_i is mapped to $B'_{p(i)}$ we set $\pi(i) = p(i)$. The i action bijection is defined as follows. The action j in A_i corresponds in $\text{BINARYACT}(\Gamma_1, \mu)$ to the profile $\text{binify}(j)$ in block B_i . As this block is mapped into $B'_{p(i)}$, the profile is mapped into another good profile $\text{binify}(j')$ and we define $\varphi_i(j) = j'$. The mapping $(\pi, \varphi_1, \dots, \varphi_n)$ is an strong isomorphism from Γ_1 to Γ_2 . Look at the Claim 3 of Appendix B for a detailed proof of this reverse part. \square

Let us now transform a binary actions game into a binary game. Given a game $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ in which $A_i = \{0, 1\}$, for any $i \in N$, and $N = \{1, \dots, n\}$. Given positive values t and m such that, for any action profile a and any player i , $|u_i(a)| \leq t$ and $m \geq \{n, t\}$. We set $k = n + tn + m + 2$ and consider the following game.

$$\text{BINARY}(\Gamma, t, m) = (N', (A'_i)_{i \in N'}, (u'_i)_{i \in N'})$$

where $N' = \{1, \dots, k\}$ and, for any $i \in N'$, $A'_i = \{0, 1\}$. The set N' is partitioned into $n + 2$ consecutive intervals B_0, \dots, B_n, B_{n+1} so that the interval B_0 has exactly n players, for $1 \leq i \leq n$, the block B_i has t players, finally block B_{n+1} has $m + 2$ players. Inside the blocks we use relative coordinates to identify the players. In all the blocks coordinates start at 1 except for the last block that starts with 0. In this situation a strategy profile a is usually factorized as $a = x b_1 \dots b_n z$ where $x = x_1 \dots x_n$, $b_i = b_{i_1} \dots b_{i_t}$ and $z = z_0 \dots z_{m+1}$. Sometimes, to improve readability, we write $a = x b_1 \dots b_n z$ as $a = (x, b_1, \dots, b_n, z)$. We define the utility function by properties of the strategy profile, assume that $a = x b_1 \dots b_n z$ is a strategy profile of $\text{BINARY}(\Gamma, t, m)$.

- In the case that, for some ℓ , $0 \leq \ell \leq m + 1$, the last ℓ bits of z are 1, all the players except the last ℓ get utility 0. The remaining players get utility 1. Observe, that in the case $\ell = 0$, we have that $z = 0^{m+2}$ and therefore all the players get utility 0.
- In the case that, for some j , $1 \leq j \leq t$, the j -th bit of z is the unique 1 in z , all the players in blocks B_1, \dots, B_n that do not occupy position j in their block get utility 0, all the players in blocks B_0 and B_{n+1} get utility 1, all the remaining players get as utility their action.
- In the case that, the 0-th bit of z is the unique 1 in z , for any i , $1 \leq i \leq n$, player i in block B_0 and all the players in block B_i get utility 1 when $u_i(x) = b_i$ and 0 otherwise. All the players in block B_{n+1} get utility 0.
- In the remaining cases all the players get utility 1.

As in a strategy profile $a = x b_1 \dots b_n z$ the parts $x = x_1 \dots x_n$, $b_i = b_{i_1} \dots b_{i_t}$ and $z = z_0 \dots z_{m+1}$ are binary words, the whole profile a is also a binary string having length $k = n + tn + m + 2$. As the utilities for all the players are either 0 or 1, we all the utilities together as a binary string $u(a) = u_1 \dots u_k$.

Example 5. We continue with the game used in Example 4. Consider the game $\Gamma = \text{BINARYACT}(BS, 0)$ where actions are binary but utilities are not. The values of the utilities are 1, 2 and 3 obtained from the utilities in BS and 0 corresponding the utility of any bad profile. As expressed in binary the utilities are 00, 01, 10 and 11, two bits suffices. The game Γ has $n = 4$. Therefore we can take $t = 2$ and $m = 4$. The game $\text{BINARY}(\Gamma, 2, 4)$ has $k = n + tn + m + 2 = 18$ players.

The set N' is partitioned into 6 blocks. The block B_0 contains 4 players, each B_i , $1 \leq i \leq 4$ has 2 players and B_5 has 6 players. A strategy profile has the format $a = x b_1 \dots b_4 z$ with $x = x_1 \dots x_4$, $b_i = b_{i_1} b_{i_2}$ for $1 \leq i \leq 4$ and $z = z_0 z_1 \dots z_4 z_5$. The utilities are coded $u(a) = u_1 \dots u_{18}$. Let us consider examples of utilities in each of the preceding four cases.

- Take for instance $\ell = 3$, then $a = x b_1 \dots b_4 0^3 1^3$ and $u(a) = 0^{15} 1^3$. We can display the block structure of the preceding utility as

$$u(a) = \underbrace{0 \dots 0}_B \underbrace{0 \dots 0}_{B_1, \dots, B_4} \underbrace{0000111}_B$$

When $\ell = 0$ we have profiles like $a = x b_1 \dots b_4 0^6$. In this case all the players get utility 0.

- When $z = z_0 z_1 z_2 z_3 z_4 z_5 = 001000$, the profile z “looks at” the second bit of each b_i , $1 \leq i \leq 4$. In this case

$$u(a) = \underbrace{1 \dots 1}_B \underbrace{0 b_{1_2} 0 b_{2_2} 0 b_{3_2} 0 b_{4_2}}_{B_1, \dots, B_4} \underbrace{1 \dots 1}_B$$

When $z = 000010$, the profile z points to “out of range” position in blocks b_i , $1 \leq i \leq 4$. In this case

$$u(a) = \underbrace{1\dots 1}_{B_0} \underbrace{0\dots\dots 0}_{B_1, \dots, B_4} \underbrace{1\dots 1}_{B_5}$$

- The connections between strategy profiles and utilities appears when $z = 100000$. Remind that in the game $\Gamma = \text{BINARYACT}(BS, 0)$ it holds

$$u_1(1010) = u_2(1010) = 11, u_3(1010) = u_4(1010) = 10$$

Let us consider profiles starting and ending as $a = (1010, b_1, \dots, b_4, 100000)$. Consider for instance $a = (1010, 10, 11, 00, 10, 100000)$. As $b_1 = 10 \neq u_1(1010)$, player 1 and players in block B_1 gets utility 0. As $b_2 = 11 = u_2(1010)$, player 2 and players in block B_2 get 1 utility. Following this argument

$$u(a) = \underbrace{0101}_{B_0} \underbrace{00}_{B_1} \underbrace{11}_{B_2} \underbrace{00}_{B_3} \underbrace{11}_{B_4} \underbrace{000000}_{B_5}$$

- In all the remaining cases, all the players get utility 1.

Lemma 2. Let Γ_1, Γ_2 be two binary actions games given in general form, set $t = \max\{t_1, t_2, 3\}$, where t_i is the time allowed to the utility TM of game Γ_i , and $m = \max\{t, n_1, n_2\}$, where n_i is the number of players in game Γ_i . There is a strong isomorphism between Γ_1 and Γ_2 iff there is a strong isomorphism between $\text{BINARY}(\Gamma_1, t, m)$ and $\text{BINARY}(\Gamma_2, t, m)$.

Proof. Given a mapping $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ of Γ_1 into Γ_2 , consider the mapping $\psi' = (p, f_1, \dots, f_k)$ of $\text{BINARY}(\Gamma_1, t, m)$ into $\text{BINARY}(\Gamma_2, t, m)$ in which, for any $1 \leq i \leq n$, $f_i = \varphi_i$, and, for any $i > n$, f_i is the identity. The permutation p on $B_{1,0}$ is exactly π . For any $1 \leq i \leq n$, block $B_{1,i}$ is mapped to block $B_{2,\pi(i)}$ and block $B_{1,n+1}$ is mapped to block $B_{2,n+1}$. Inside each block the players are assigned preserving the relative order of positions in the block. It is straightforward to show that if ψ is an isomorphism then ψ' is also an isomorphism.

For the reverse implication, assume that $\psi' = (p, f_1, \dots, f_k)$ is a strong isomorphism between the games $\text{BINARY}(\Gamma_1, t, m)$ and $\text{BINARY}(\Gamma_2, t, m)$. Observe that in such a case Γ_1 and Γ_2 have the same number n of players. In this case we can show that permutation p preserves blocks, and relative positions inside interior blocks, therefore we can define a mapping $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ in which π is the restriction of p to block $B_{1,0}$ and, for any $1 \leq i \leq n$, $\varphi_i = f_i$. In Claim 4 of Appendix B we prove that ψ is an isomorphism. \square

Given a binary game $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ with n players, such that for any $1 \leq i \leq n$, utility u_i has range $\{0, 1\}$ and $A_i = \{0, 1\}$. We construct a circuit C_Γ on $4n + 2$ variables. Recall that, when $u_i(x)$ is computed by a Turing machine in polynomial time, Ladner’s construction [23] gives us a polynomial size circuit computing the same function.

Circuit C_Γ . The variables in C_Γ are grouped in four blocks, the X -block contains the first n -variables, the Y -block is formed by the variables in positions $n + 1$ to $2n$, the C -block contain the following $n + 2$ variables, and the D -block the remaining variables. For sake of readability we split the set of variables into four parts $a = (x, y, c, d)$ where $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n)$, $c = (c_1, \dots, c_{n+2})$, and $d = (d_1, \dots, d_n)$.

We define C_Γ with the help of $n + 2$ following circuits.

$$\begin{aligned} C_1(x, y, d) &= [(x_1 = \overline{d_1}) \wedge \dots \wedge (x_n = \overline{d_n}) \wedge (u_1(x) = y_1) \wedge \dots \wedge (u_n(x) = y_n)] \\ C_2(y) &= [y_1 \vee \dots \vee y_n] \\ C_{i+2}(x_i, y_i, d_i) &= [\overline{y_i} \wedge (x_i = \overline{d_i})] \quad \text{for } 1 \leq i \leq n. \end{aligned}$$

Finally

$$C_\Gamma(x, y, c, d) = \begin{cases} 0 & \text{if } \sum_{1 \leq i \leq n+2} c_i = 0 \text{ or } \sum_{1 \leq i \leq n+2} c_i > 1 \\ C_j & \text{if } \sum_{1 \leq i \leq n+2} c_i = 1 \text{ and } c_j = 1 \end{cases}$$

The previous construction is used to reduce the ISO problem to the CIRCUITCONG problem.

Lemma 3. *Let Γ and Γ' be two binary games in general form with at least two players each. There is a congruence isomorphism between C_Γ and $C_{\Gamma'}$ iff there is a strong isomorphism between Γ and Γ' .*

Proof. Assume that $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ is a strong isomorphism from Γ to Γ' and consider the following variable transformation. The transformation preserves blocks. Variable x_i is mapped to variable $x'_{\pi(i)}$ with permutation φ_i , the same happens with block D . Variables y_i is mapped to variable $y'_{\pi(i)}$ with permutation the identity function, variable c_1 is mapped to variable c'_1 , c_2 to c'_2 , and c_{2+i} to $c'_{2+\pi(i)}$ all the block c with permutation the identity function.

For the reverse implication, let $\psi' = (p, f_1, \dots, f_{4n+2})$ be a congruence morphism between C_Γ and $C_{\Gamma'}$. Given $a = (a_1, \dots, a_{4n+2})$, for $1 \leq i \leq 4n+2$ the value $p(i)$ points to the image of a_i . When $\psi(a) = (a'_1, \dots, a'_{4n+2})$ it holds $a'_{p(i)} = f(a_i)$.

When $a = (x, y, c, d)$, to avoid confusions we note $p(x_i)$ the position of the image of x_i and we take similar conventions for $p(y_i)$, $p(c_i)$ and $p(d_i)$. Similarly the value of the image of x_i will be $f(x_i)$. The congruence verifies that for any truth assignment a to the variables of C_Γ , we have that $C_\Gamma(a) = C_{\Gamma'}(\psi'(a))$. Congruence Ψ' allows us (as proved in Claim 5 of Appendix B) to prove that ψ preserves the structure of the C and Y blocks. Furthermore the functions f_i , for i in block C or Y , is the identity. This allows us to consider the permutation π on $\{1, \dots, n\}$ such that $p(c_{i+2}) = c'_{\pi(i)+2}$. Moreover this permutation verifies (proof in Claim 6 of Appendix B) $p(x_i) = x'_{\pi(i)}$ iff $p(d_i) = d'_{\pi(i)}$ and $p(x_i) = d'_{\pi(i)}$ iff $p(d_i) = x'_{\pi(i)}$.

Consider the mapping $\psi'' = (p', f'_1, \dots, f'_{4n+2})$ such that the behavior of the permutation and bijections coincides with ψ' in blocks Y and C . In blocks X and D is given by

$$p'(x_i) = \begin{cases} x'_{\pi(i)} & \text{if } p(x_i) = x'_{\pi(i)} \\ x'_{\pi(i)} & \text{if } p(x_i) = d'_{\pi(i)} \end{cases} \quad \text{and} \quad p'(d_i) = \begin{cases} d'_{\pi(i)} & \text{if } p(d_i) = d'_{\pi(i)} \\ d'_{\pi(i)} & \text{if } p(d_i) = x'_{\pi(i)} \end{cases}$$

then $p' : X \rightarrow X'$ and $p' : D \rightarrow D'$ and the behavior of p' is the same in both blocks, that is $p'(x_i) = x'_{\pi(i)}$ iff $p'(d_i) = d'_{\pi(i)}$. The bijections are defined as

$$f'(x_i) = \begin{cases} f(x_i) & \text{if } p(x_i) = x'_{\pi(i)} \\ \neg f(x_i) & \text{if } p(x_i) = d'_{\pi(i)} \end{cases} \quad \text{and} \quad f'(d_i) = \begin{cases} f(d_i) & \text{if } p(d_i) = d'_{\pi(i)} \\ \neg f(d_i) & \text{if } p(d_i) = x'_{\pi(i)} \end{cases}$$

The morphism ψ'' is a congruence. This trivially happens because for any strategy profile a it holds $\psi'(a) = \psi''(a)$.

Finally, it is easy to prove that the morphism ψ given by $\psi = (\pi, f'_1, \dots, f'_n)$ is an isomorphism between Γ and Γ' . \square

It is easy to show that CIRCUITCONG is reducible to ISO, just consider a game with as many players as variables in which the utilities for all the players are identical and coincide with the evaluation of the circuit. Taking into account that CIRCUITCONG is equivalent to CIRCUITISO putting all together we have:

Theorem 3. *The strong isomorphism problem for strategic games in general form is polynomially equivalent to the circuit isomorphism problem.*

Observe that, for games in general form, the ISO problem for strong isomorphism remains equivalent to the ISO problem for strong isomorphism when the games are restricted to be binary actions or binary games, as the identity trivially reduces the latest problem to the former one.

We consider now formula games in general form, the results also apply to WCFG games [14] where actions are given in explicit way. The proof follows the same steps as for the previous case. Now we have to show that a description of the games provided in the reduction as formula games can be computed in polynomial time.

Theorem 4. *The strong isomorphism problem for formula games and WCFG in general form are equivalent to the boolean formula isomorphism problem.*

Proof. The game $\text{BINARYACT}(\Gamma, \mu)$ when Γ is a formula game in general form is a formula game. The game $\text{BINARY}(\Gamma, t, m)$ when Γ is a binary actions formula game in general form is a formula game. A description in general form of the games $\text{BINARYACT}(\Gamma, \mu)$ and $\text{BINARY}(\Gamma, t, m)$ can be computed in polynomial time. Furthermore, a description of the circuit C_Γ , for a binary formula game Γ , can be obtained in polynomial time. A detailed proof all of this is given in Claim 7 of Appendix B. \square

Proving NP-completeness in the case of explicit form appears to be a difficult task. Observe, that a game in explicit form can be seen as a graph with edge labels and weights. As the total number of different weights appearing in both games is polynomial the problem can be reduced to the **Graph isomorphism (GI)** problem [30]. Therefore the NP-hardness of ISO will imply the NP-hardness of GI. We have to prove the opposite direction. We start by constructing a game from a graph.

Given an undirected graph G , let us define a strategic game $\Gamma(G)$ associated to this graph.

Game $\Gamma(G)$. Assume that $G = (V, E)$ is a nondirected graph with $V = \{1, \dots, n\}$ and m edges. Given $e \in E$ we write $e = \{i, j\}$ to denote an edge connecting i and j . The game has 4 players with $A_1 = A_2 = \{0, 1, \dots, n\}$, $A_3 = \{0, 1\}$, $A_4 = E \cup \{0\}$. Let $A = A_1 \times A_2 \times A_3 \times A_4$ and $(i, j, k, l) \in A$, the utilities are:

$$u_1(i, j, k, l) = u_2(i, j, k, l) = \begin{cases} 1 & \text{if } l = \{i, j\} \text{ and } k = 0, \\ 0 & \text{otherwise} \end{cases}$$

$$u_3(i, j, k, l) = \begin{cases} 1 & \text{if } i = 0, j \neq 0, k = 1 \text{ and } l \neq 0, \\ 0 & \text{otherwise} \end{cases}$$

$$u_4(i, j, k, l) = \begin{cases} 1 & \text{if } i = j = k = 0, \\ 0 & \text{otherwise} \end{cases}$$

Observe that the graph isomorphism problem is equivalent to the problem restricted to connected graphs G and G' that have $n > 2$ vertices. Otherwise, we add one after another new vertices connected to all the other vertices in the graph (in both graphs) until the condition is fulfilled. This type of vertex addition preserves isomorphism. With this construction we can also assume that, any vertex has at least one outgoing edge.

Lemma 4. *Let G, G' two connected undirected graphs, with at least two vertices. The games $\Gamma(G)$ and $\Gamma(G')$ are strongly isomorphic iff G and G' are isomorphic.*

Proof. Assume that p is an isomorphism between graphs G and G' . Let $\psi = (\pi, \varphi_1, \dots, \varphi_4)$ be an game mapping defined as follows π and φ_3 are the identity, $\varphi_1 = \varphi_2 = p$, $\varphi_4(0) = 0$ and, for any edge $\{u, v\} \in E(G)$, $\varphi_4(\{u, v\}) = \{p(u), p(v)\}$. It is straightforward to show that ψ is a strong isomorphism between $\Gamma(G)$ and $\Gamma(G')$.

Let $\psi = (\pi, \varphi_1, \dots, \varphi_4)$ be a strong isomorphism between $\Gamma(G)$ and $\Gamma(G')$, this verifies the following.

The player's permutation verifies $\pi : \{1, 2\} \rightarrow \{1, 2\}$, $\pi : \{3\} \rightarrow \{3\}$ and finally $\pi : \{4\} \rightarrow \{4\}$. Denoting by $\#$ the cardinality of a set, we have $\#\{a | u_1(a) = 1\} = \#\{a | u_2(a) = 1\} = 2m$, $\#\{a | u_3(a) = 1\} = mn$ and $\#\{a | u_4(a) = 1\} = m + 1$. When $n > 2$ and $m > 2$, these sets have different cardinality. As $\#\{a | u_i(a) = 1\} = \#\{\psi(a) | u'_{\pi(i)}(\psi(a)) = 1\}$, we get the result.

As players 1 and 2 have the same behavior we can assume $\pi(1) = 1$ and $\pi(2) = 2$, therefore π is the identity.

The action's bijection φ_3 is the identity. As π is the identity, it holds the following bijection $\psi(A_1 \times A_2 \times \{0\} \times A_4) = A'_1 \times A'_2 \times \{1\} \times A'_4$. Therefore, for any i', j', l' holds $u'_3(i', j', 1, l') = 0$ because $u_3(\varphi_1^{-1}(i'), \varphi_2^{-1}(j'), 0, \varphi_4^{-1}(l')) = 0$. This is a contradiction because $u'_3(i', 0, 1, e') = 1$ when $1 \leq i \leq n$ and $e' \in E'$.

The action's bijections for players 1 and 2 verify $\varphi_1(0) = 0$ and $\varphi_2(0) = 0$. This is forced by the rigid the structure of u_4 . As $u_4(0, 0, 0, l) = 1$ we should have $u'_4(\varphi_1(0), \varphi_2(0), 0, \varphi_4(l)) = 1$ and this force $\varphi_1(0) = \varphi_2(0) = 0$. Therefore φ_1 and φ_2 are permutations on vertices $\{1, \dots, n\}$.

The action's bijection φ_4 verifies $\varphi_4(0) = 0$. When this does not hold, as φ_4 is a bijection, there exists e such that $\varphi_4(e) = 0$. For $j \neq 0$ it holds that $u_3(0, j, 1, e) = 1$ and, as ψ is a morphism, $u'_3(0, \varphi_2(j), 1, 0) = 1$, but this is a contradiction. Therefore φ_4 is a permutation on the m edges.

When $\varphi_4(e) = e'$ and $e = \{i, j\}$ it holds that $e' = \{\varphi_1(i), \varphi_2(j)\}$. As $u_1(i, j, 0, \{i, j\}) = 1$ and ψ is a morphism, $u'_3(\varphi_1(i), \varphi_2(j), 0, \varphi_4(\{i, j\})) = 1$, and $\varphi_4(\{i, j\}) = \{\varphi_1(i), \varphi_2(j)\}$.

φ_1 and φ_2 are the same permutation on $\{1, \dots, n\}$. We have to prove that, for all $i \in \{1, \dots, n\}$ it hold $\varphi_1(i) = \varphi_2(i)$. Let i be a vertex, as every node has positive degree, exists j such that $e = \{i, j\}$ is an edge in G . As $u_1(i, j, 0, e) = u_1(j, i, 0, e) = 1$ it holds $\varphi_4(e) = \{\varphi_1(i), \varphi_2(j)\} = \{\varphi_1(j), \varphi_2(i)\}$. There are two possibilities, $\varphi_1(i) = \varphi_2(i)$ and $\varphi_1(j) = \varphi_2(j)$ or $\varphi_1(i) = \varphi_1(j)$ and $\varphi_2(j) = \varphi_2(i)$. But $\varphi_1(i) = \varphi_1(j)$ is impossible because φ_1 is a permutation.

When ψ is an isomorphism, the mapping $\varphi_1 : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ induces a graph isomorphism. Consider and edge $e = \{i, j\}$ in G as ψ is a game morphism $u_1(i, j, 0, e) = u'_1(\varphi_1(i), \varphi_1(j), 0, \varphi_4(e))$ and this forces $\varphi_4(e) = \{\varphi_1(i), \varphi_1(j)\}$. \square

As a consequence of the previous results we get the following.

Theorem 5. *The strong isomorphism problem for games given in explicit form is equivalent to the graph isomorphism problem.*

4. Weak isomorphisms

Replacing strong by weak isomorphisms does not modify complexity bounds. In this section we show that, for the case of weak isomorphism, the ISISO problem is coNP-complete and that the ISO problem is equivalent to the ISO problem for strong isomorphisms. The last equivalence will hold for any of the considered representations of the games.

Theorem 6. *The ISISO problem for weak isomorphism is coNP-complete, for games given in general form (strategic, formula and WCFG), and it belongs to NC when the games are given in explicit form. The ISO problem belongs to Σ_2^P , when the games are given in general form (strategic, formula and WCFG) and it belongs to NP when the games are given in explicit form.*

Proof. We adapt the proofs given in Theorems 1 and 2. Membership in coNP of the ISISO problem for weak isomorphism and for games given in explicit or general form follows from the definitions.

When the games and the morphism are given in explicit form, a direct adaptation of the proof given in Theorem 1 give us that ISISO belongs to NC for weak isomorphism.

To prove hardness, given a boolean formula F with n variables, we define a variation of the game WINWHENTRUE(F), WINWHENTRUEW(F) in which we redefine utilities as follows:

$$u_i(a_1, \dots, a_n) = \begin{cases} 2^n & \text{if } F(a_1, \dots, a_n) \text{ is true,} \\ \sum_{i=1}^n a_i 2^{n-i} & \text{if } F(a_1, \dots, a_n) \text{ is false.} \end{cases}$$

Observe that for any pair of strategy profiles $a \neq a'$, $a \sim_i a'$ holds when both $F(a) = F(a') = 1$. When $F(a) = F(a') = 0$, $a \prec_i a'$ if and only if $a < a'$ in lexicographic order. When $F(a) \neq F(a')$ player i prefers the satisfying assignment. On the other hand we consider the ALWAYSWIN game in which $a \sim_i a'$ always holds.

Observe that the IDENTITY morphism is a weak isomorphism between the games ALWAYSWIN and WINWHENTRUEW(F) iff F is valid. Thus, the ISISO problem for weak isomorphism and games in general form is coNP-complete.

Finally observe that a description in general form of the WINWHENTRUEW(F) and the ALWAYSWIN games can be computed in polynomial time, both when the utility functions are described by Turing machines or by formulas. \square

When considering the weak isomorphism we will show that the ISO problem, for strategic games in general form, is equivalent to the boolean circuit isomorphisms, for formula games in general form to the boolean formula isomorphism, and for strategic games in explicit form to the graph isomorphism problem. Before proving those results we provide a series of game transformations that preserve weak isomorphism establishing equivalence with the strong isomorphism. Later on we will show that those transformations are indeed polynomial time reduction for the considered representations.

Assume that $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ is a binary game where $N = \{1, \dots, n\}$. We consider the following game.

$\text{CHECKW}(\Gamma) = (N', (A'_i)_{i \in N'}, (u'_i)_{i \in N'})$

where $N' = \{1, \dots, n, n+1\}$ and, for any $1 \leq i \leq n$, $A'_i = \{0, 1\}$ and $A'_{n+1} = \{0, 1, 2, 3\}$. The utilities are defined as follows, for a player i , $1 \leq i \leq n$,

$$u'_i(a') = \begin{cases} 1 & \text{if } u_i(a'_1, \dots, a'_n) = (a'_{n+1} \bmod 2), \\ 0 & \text{otherwise,} \end{cases}$$

For the last player,

$$u'_{n+1}(a') = a'_{n+1}.$$

We can look at the equality $(u_i(a'_1, \dots, a'_n) = (a'_{n+1} \bmod 2))$ as a boolean expression taking values $\{0, 1\}$, under this point of view we write shortly $u'_i(a') = (u_i(a'_1, \dots, a'_n) = (a'_{n+1} \bmod 2))$.

Also note that Γ is a binary game (both, actions and utilities are binary), $\text{CHECKW}(\Gamma)$ is not a binary game either a binary action game due to the last player. Player $n+1$ in $\text{CHECKW}(\Gamma)$ has four actions and u_{n+1} takes four values.

Lemma 5. *Let Γ_1 and Γ_2 be two binary games. Γ_1 and Γ_2 are strongly isomorphic iff the games $\text{CHECKW}(\Gamma_1)$ and $\text{CHECKW}(\Gamma_2)$ are weakly isomorphic.*

Proof. Let $\Gamma'_1 = \text{CHECKW}(\Gamma_1)$ and $\Gamma'_2 = \text{CHECKW}(\Gamma_2)$. Assume that $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ is a strong isomorphism between Γ_1 and Γ_2 . Define the mapping $\psi' = (p, f_1, \dots, f_{n+1})$ where, for $1 \leq i \leq n$, $p(i) = \pi(i)$ and $f_i = \varphi_i$, $p(n+1) = n+1$ and f_{n+1} is the identity function. As we see in Claim 8 ψ' is a strong (therefore also a weak) isomorphism between Γ'_1 and Γ'_2 .

Assume now that $\psi' = (p, f_1, \dots, f_{n+1})$ is a weak isomorphism between Γ'_1 and Γ'_2 . As p maps between them players having the same number of actions, and the only player with 4 actions is the last one we are forced to have $p(n+1) = n+1$. Let ψ be ψ' restricted to players $1, \dots, n$, that is $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ such that for $1 \leq i \leq n$, $\pi(i) = p(i)$ and $\varphi_i = f_i$. For any $a' = (a, a'_{n+1})$ with $a = (a_1, \dots, a_n)$ we have that $\psi'(a') = (\psi(a), f_{n+1}(a'_{n+1}))$. The definition of the preference relation of player $n+1$ forces $f_{n+1} = \text{Id}$ (see Claim 9). After that we have the factorization $\psi'(a') = (\psi(a), a'_{n+1})$. Note that for any $a' = (a, a_{n+1})$ it holds

$$u'_{n+1}(\psi'(a')) = u'_{n+1}(\psi(a), a_{n+1}) = a_{n+1} = u'_{n+1}(a')$$

Given $a = (a'_1, \dots, a'_n)$, as Γ_1 is a binary game it holds $u_i(a) \in \{0, 1\}$. We define $\bar{u}_i(a) = 1 - u_i(a)$ and in this case $u_i(a) = (u_i(a) \bmod 2)$ and $\bar{u}_i(a) = (\bar{u}_i(a) \bmod 2)$. In $\text{CHECKW}(\Gamma_1)$, given $a = (a'_1, \dots, a'_n)$ for any player $1 \leq i \leq n$ it holds,

$$\begin{aligned} u'_i(a, \bar{u}_i(a)) &= (u_i(a) = (\bar{u}_i(a) \bmod 2)) = (u_i(a) = \bar{u}_i(a)) = 0 \\ u'_i(a, u_i(a)) &= (u_i(a) = (u_i(a) \bmod 2)) = (u_i(a) = u_i(a)) = 1 \end{aligned}$$

Therefore $(a, \bar{u}_i(a)) \prec_i (a, u_i(a))$. As ψ' is a weak isomorphism $\psi'(a, \bar{u}_i(a)) \prec_{\pi(i)} \psi'(a, u_i(a))$ as f_{n+1} is the identity $(\psi(a), \bar{u}_i(a)) \prec_{\pi(i)} (\psi(a), u_i(a))$. This forces $u'_{\pi(i)}(\psi(a), \bar{u}_i(a)) < u'_{\pi(i)}(\psi(a), u_i(a))$ and consequently $u'_{\pi(i)}(\psi(a), \bar{u}_i(a)) = 0$ and $u'_{\pi(i)}(\psi(a), u_i(a)) = 1$. According to the definition of $u'_{\pi(i)}$ it holds

$$u'_{\pi(i)}(\psi(a), u_i(a)) = (u_{\pi(i)}(\psi(a)) = (u_i(a) \bmod 2)) = (u_{\pi(i)}(\psi(a)) = u_i(a)) = 1$$

therefore $u_{\pi(i)}(\psi(a)) = u_i(a)$ for any $1 \leq i \leq n$ and ψ is a strong isomorphism. \square

As we have done in the previous section we start by defining a transformation to binary actions game. The construction of the game follows the same lines as in the $\text{BINARYACT}(\Gamma)$ (see Page 9), but now we have to guarantee an adequate preference relation for each player. Assume that $\Gamma = (N, A_1, \dots, A_n, (u_i)_{1 \leq i \leq n})$. We can assume, without loss of generality, that utilities are non negative. If this happens it is enough to add a “big positive number”. When utilities computed by a TM with time t we can add 2^t .

$\text{BINARYACTW}(\Gamma) = (N', (A'_i)_{i \in N'}, (u'_i)_{i \in N'})$ where $N' = \{1, \dots, k\}$ and, for any $i \in N'$, $A'_i = \{0, 1\}$ and thus the set of action profiles is $A' = \{0, 1\}^k$. We associate to A_i a block B_i of $k_i = |A_i|$ players each one taking care of one bit. Thus, $k = k_1 + \dots + k_n$. We split a strategy profile a' into n blocks, thus $a' = (b_1, \dots, b_n)$ where $b_i \in \{0, 1\}^{k_i}$. We keep a'_j to refer to the strategy of player j . Recall that, if $A^i = \{0, 1\}^{k_i}$, $\text{good}(A^i) = \{\text{binify}(a) \mid a \in A_i\}$ where $\text{binify}(j) = 0^{j-1}10^{k_i-j}$, $\text{good}(A') = \{\text{binify}(a_1) \cdots \text{binify}(a_n) \mid a_1 \in A_1, \dots, a_n \in A_n\}$, and that, for $a' \in \text{good}(A')$, $\text{binify}^{-1}(a') = (\text{binify}^{-1}(b_1), \dots, \text{binify}^{-1}(b_n))$. For a player α that occupies position j in block B_i , the player partitions A' in the following sets:

$$\begin{aligned} X_0(\alpha) &= \{a' \mid b_i \notin \text{good}(A^i)\} \\ X_1(\alpha) &= \{a' \mid b_{i_j} = 0 \text{ and } b_i \in \text{good}(A^i) \text{ and } a' \in \text{bad}(A')\} \\ X_2(\alpha) &= \{a' \mid b_{i_j} = 1 \text{ and } b_i \in \text{good}(A^i) \text{ and } a' \in \text{bad}(A')\} \\ X_3(\alpha) &= \text{good}(A') \end{aligned}$$

and the utility function is defined as

$$u'_\alpha(a') = \begin{cases} 0 & \text{if } a' \in X_0(\alpha), \\ 1 & \text{if } a' \in X_1(\alpha), \\ 2 & \text{if } a' \in X_2(\alpha), \\ 3 + u_i(\text{binify}^{-1}(a')) & \text{if } a' \in X_3(\alpha), \end{cases}$$

When $|A_i| = k_i = 1$, we have $A^i = \{0, 1\}$, $\text{binify}(1) = 1$ and B_i has just one player. Let α be such a player, in this case $X_1(\alpha) = \emptyset$. When $k_i > 1$ all the sets $X_0(\alpha), \dots, X_3(\alpha)$ are non empty.

Observe that player α prefers profiles in $X_3(\alpha)$ to profiles in $X_2(\alpha)$, profiles in $X_2(\alpha)$ to profiles in $X_1(\alpha)$, and profiles in $X_1(\alpha)$ to profiles in $X_0(\alpha)$. Moreover, player α is indifferent among two profiles belonging to the same set $X_0(\alpha)$, $X_1(\alpha)$, or $X_2(\alpha)$. For profiles a'_1 and a'_2 both in $X_3(\alpha)$, player α keeps the preferences of player i in Γ among the profiles $\text{binify}^{-1}(a'_1)$ and $\text{binify}^{-1}(a'_2)$.

Lemma 6. *Let Γ_1 and Γ_2 be two strategic games. Γ_1 and Γ_2 are weakly isomorphic iff $\text{BINARYACTW}(\Gamma_1)$ and $\text{BINARYACTW}(\Gamma_2)$ are weakly isomorphic.*

Proof. Let $\Gamma'_1 = \text{BINARYACTW}(\Gamma_1)$ and $\Gamma'_2 = \text{BINARYACTW}(\Gamma_2)$.

Assume that $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ is a weak isomorphism between Γ_1 and Γ_2 . Consider the mapping $\psi' = (p, f_1, \dots, f_k)$ where, for $1 \leq i \leq n$, p maps the bits in block i of Γ'_1 to the bits in block $\pi(i)$ of Γ'_2 so that the j -th bit of B_i goes to bit $\varphi_i(j)$ of $B_{p(i)}$, and, for $1 \leq j \leq k$, f_j is the identity function. It is straightforward to show that ψ' is a weak isomorphism between Γ'_1 and Γ'_2 .

Let us consider the reverse part. Assume that $\psi' = (p, f_1, \dots, f_k)$ is a weak isomorphism between Γ'_1 and Γ'_2 . As we prove in the Claim 10 of Appendix C, all the f_α , $1 \leq \alpha \leq k$, are identities and p induces a permutation into the blocks, therefore we consider a player permutation π on $\{1, \dots, n\}$. For a player α in position j inside block B_i , let $\varphi(j)$ be the position of player $p(\beta)$ in block $\pi(i)$. It holds that $(\pi, \varphi_1, \dots, \varphi_n)$ is a weak isomorphism between Γ_1 and Γ_2 . \square

Next is to transform weakly isomorphic games into strongly isomorphic games. The transformation consists on coding precedence relations into utilities. Given a binary actions game $\Gamma = (N, (A_i)_{i \in N}, (u_i)_{i \in N})$ where $N = \{1, \dots, n\}$ and $A_i = \{0, 1\}$, consider the following game.

$\text{FLIPW}(\Gamma) = (N, (A'_i)_{i \in N}, (u'_i)_{i \in N})$ where $A'_i = \{0, 1\}^2$. Let $a' = (a_1 b_1, \dots, a_n b_n)$ be a strategy profile in game $\text{FLIPW}(\Gamma)$, define $\text{driver}(a') = (a_1, \dots, a_n) = a$ and $\text{flipper}(a') = (b_1, \dots, b_n) = b$. We note shortly $a' = a \uparrow b$. For $xy \in \{0, 1\}^2$ define

$$\text{flip}(xy) = \begin{cases} x & \text{if } y = 0, \\ \bar{x} & \text{if } y = 1. \end{cases}$$

Let $a' = (a_1 b_1, \dots, a_n b_n)$ be a strategy profile in game $\text{FLIPW}(\Gamma)$, define

$$\text{flip}(a') = (\text{flip}(a_1 b_1), \dots, \text{flip}(a_n b_n)).$$

Observe that $\text{flip}(a')$ is a strategy profile in game Γ . Given a strategy profile $a' = (a_1b_1, \dots, a_nb_n)$, for any player i , $1 \leq i \leq n$, we define:

$$u'_i(a') = \begin{cases} 5 & \text{if } u_i(\text{flip}(a')) < u_i(\text{driver}(a')) \text{ and } b_i = 1 \\ 4 & \text{if } u_i(\text{flip}(a')) = u_i(\text{driver}(a')) \text{ and } b_i = 1 \\ 3 & \text{if } u_i(\text{flip}(a')) > u_i(\text{driver}(a')) \text{ and } b_i = 1 \\ 2 & \text{if } u_i(\text{flip}(a')) < u_i(\text{driver}(a')) \text{ and } b_i = 0 \\ 1 & \text{if } u_i(\text{flip}(a')) = u_i(\text{driver}(a')) \text{ and } b_i = 0 \\ 0 & \text{if } u_i(\text{flip}(a')) > u_i(\text{driver}(a')) \text{ and } b_i = 0 \end{cases}$$

Example 6. Consider the game Γ

		Player 2	
		0	1
Player 1	0	0, 0	1, 0
	1	0, 1	0, 0
Γ			

It holds $(1, 0) \prec_1 (0, 1)$. Let us see how this preference is coded as an utility in $\text{FLIPW}(\Gamma)$. To transform $a = (1, 0)$ into $(0, 1)$, both bits in $(1, 0)$ have to be flipped, therefore the flipper is $b = (1, 1)$ and we code the transformation in $\text{FLIPW}(\Gamma)$ with the strategy profile $a' = (11, 01) = (1, 0) \uparrow (1, 1) = a \uparrow b$. We have $\text{driver}(a') = (1, 0)$, $\text{flipper}(a') = (1, 1)$ and $\text{flip}(a') = (0, 1)$. To compute $u'_1(a')$, we look at the flipper of first player, as $b_1 = 1$ and $u_1(1, 0) < u_1(0, 1)$ we get $u'_1(a') = 3$. Consider a case of indifference, for instance $(0, 0) \sim_2 (0, 1)$. The driver is $a = (0, 0)$, the flipper is $b = (0, 1)$ and $a' = (00, 01) = a \uparrow b$. As $b_2 = 1$ we get $u'_2(a') = 4$.

Lemma 7. Let Γ_1 and Γ_2 be two binary actions games. Γ_1 and Γ_2 are weakly isomorphic iff the games $\text{FLIPW}(\Gamma_1)$ and $\text{FLIPW}(\Gamma_2)$ are strongly isomorphic.

Proof. Let $\Gamma'_1 = \text{FLIPW}(\Gamma_1)$ and $\Gamma'_2 = \text{FLIPW}(\Gamma_2)$.

Let $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ be a mapping between two binary action games Γ_1 and Γ_2 . Let $\psi' = (\pi, f_1, \dots, f_n)$ be a mapping between Γ'_1 and Γ'_2 verifying $f_i(a_ib_i) = \varphi_i(a_i)b_i$ for $1 \leq i \leq n$. Taking $\mu = (\pi, id_1, \dots, id_n)$, for any $a' = a \uparrow b$ it holds $\psi'(a') = \psi(a) \uparrow \mu(b)$. Moreover $\text{flip}(\psi'(a')) = \psi(\text{flip}(a'))$, look at Claim 11 of Appendix C.

Assume that $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ is a weak isomorphism between games Γ_1 and Γ_2 . Consider the games Γ'_1 and Γ'_2 , and the morphism $\psi' = (\pi, f_1, \dots, f_n)$ where, for $1 \leq i \leq n$, $f_i(a_ib_i) = \varphi_i(a_i)b_i$. It holds that ψ' is a strong isomorphism between Γ'_1 and Γ'_2 , look at the Claim 12 of Appendix C.

If $\psi' = (\pi, f_1, \dots, f_n)$ is a strong isomorphism between Γ'_1 and Γ'_2 the definition of the utility functions of player i forces that for any action a_ib_i , $f_i(a_ib_i) = \varphi_i(a_i)b_i$ for some permutation φ_i on $\{0, 1\}$. Consider the morphism $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ and a profile $a' = (a_1b_1, \dots, a_nb_n)$, observe that $\text{flip}(\psi'(a')) = \psi(\text{flip}(a'))$, look at Claim 11 of Appendix C. Taking into account this fact and that ψ' preserves utilities, we can show that ψ is a weak isomorphism between Γ_1 and Γ_2 , look at the Claim 13 of the Appendix C. \square

Taking into account all the previous results together, it remains to show that the previous transformation can be performed in polynomial time when the input and output game representation is fixed to be one of the considered in this paper, as stated in the following complexity equivalence.

Theorem 7. For strategic games given in general form, the ISO problem for weak isomorphism is equivalent to the circuit isomorphism problem. For formula games given in general form, the ISO problem for weak isomorphism is equivalent to the boolean formula isomorphism problem. For strategic games given in explicit form, the ISO problem for weak isomorphism is equivalent to the graph isomorphism problem.

Proof. It is straightforward to show that for a strategic game in general form a description in general form the games constructed in this section can be computed in polynomial time. The same happens when the original and the target representation is a formula game in general form or a game in explicit form.

In consequence all the game constructions this section show polynomial time reductions between different isomorphism problems.

Lemma 5 reduces strong isomorphism for boolean games to weak isomorphism. Lemma 6 reduces weak isomorphism to weak isomorphism for binary actions games. Lemma 7 reduces weak isomorphism to strong isomorphism. Finally, Lemmas 1 and 2 establish the reduction from strong isomorphism to strong isomorphism for binary games. Therefore, all the problems, for the same game representation, are polynomially equivalent.

According to the complexity equivalences stated in in Section 3, the claim follows. \square

5. Further results and open problems

We are working towards extending the definitions of game isomorphism. There are still some other ways to relax the notion of isomorphism while maintaining some structure of the Nash equilibria, besides the strong and weak isomorphisms considered in this paper. In particular we are interested in isomorphism with minimum requirement on maintaining partially the structure of Nash equilibria. On another line it is of interest to extend the notion of game isomorphisms to other game families, in particular for strategic games given in implicit form. The main difficulty here is to select a suitable succinct representation of permutations on the set $\{0, 1\}^k$ to being able to represent a morphism. We expect the ISO problem for games in implicit form (with utilities given by TM, circuits or formulas) to be computationally harder than for the case of games given in general form. Observe, that for strategic games in implicit form the reductions in this paper will not longer be polynomial time computable as the number of strategy profile will be exponential in the size of the representation. Another family of interest is that of extensive games. We would like to study the isomorphism problem for such games avoiding the use of strategic forms. An interesting open question is finding suitable definition of game isomorphisms for games without perfect information.

A second line of research is to obtain more information about the classification of strategic games with the same number of players according to the structure of classes induced by the type of isomorphism. the pure Nash equilibria. A first naive approach is to consider games as equivalent if they have the same number of Nash equilibria. The counting of pure Nash equilibria has been undertaken via probabilistic analysis by I. Y. Powers [22]. She studied the limit distributions of the number of pure strategies Nash equilibria for n players strategic games. Further results in [15]. Just counting pure Nash equilibria is different from strong and weak isomorphism as the notions provide a finer classification.

There are several fields in computer science developping games, strategic or extensive, that can be used to attain different goals in the Semantic Web. One clear example of this direction is the *games with a purpose* approach [3, 27, 28]. Those games are used, for instance, to label a image, thus facilitating the acquisition of terms for the semantic web. Games, strategic or extensive, are used in this approach to learn from the strategic behaviour of the players. The games are defined in such a way that term agreement provides higher utility. Observe that in this setting games designed by different research teams might lead to different definitions on the game corresponding to the labeling of the same image. To asses the validity of the final results we should check the equivalence among the games. This might lead to different notions of equivalence from the ones presented in this paper. We believe that the results on this paper will provide the basis for the analysis of the complexity of equivalence of such games and other web games.

Acknowledgement

The authors want to thank an anonymous referee for the careful reading that help us to improve the readability of the paper and correct several inaccuracies.

References

- [1] M. Agrawal and T. Thierauf. The formula isomorphism problem. *SIAM Journal on Computing*, 30(3):990–1009, 2000.

- [2] C. Álvarez, J. Gabarro, and M. Serna. Pure Nash equilibrium in strategic games with a large number of actions. In *Mathematical Foundations of Computer Science, MFCS 2005*, volume 3618 of *LNCS*, pages 95–106. Springer Verlag, 2005.
- [3] L. von Ahn. Games with a Purpose, *Computer*, Vol 39, Num 6, 92-94, 2006
- [4] A. Beimel and E. Weinreb. Monotone circuits for monotone weighted threshold functions. *Inf. Process. Lett.*, 97(1):12–18, 2006.
- [5] E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini. Boolean games revisited. In *ECAI 2006, 17th European Conference on Artificial Intelligence*, pages 265–269, 2006.
- [6] B. Borchet, D. Ranjan, and F. Stephan. On the computational complexity of some classical equivalence relations on boolean functions. *Theory Comput. Systems*, 31:679–693, 1998.
- [7] B. de Bruin. Game transformations and game equivalence. Technical Report X-1999-01, IIL Technical Note, 1999.
- [8] A. Casajus. Weak isomorphism of extensive games. *Mathematical Social Sciences*, Elsevier, 46(3), 267-290, 2003.
- [9] Ch. Chang and S. Tijs. A note on isomorphism and strategic equivalence of cooperative games. *TOP*, 14(2): 333-342, 2006.
- [10] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H.Freeman and Co, 1979.
- [11] J. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games*. MIT Press, 1988.
- [12] J. McKinsey. Isomorphism of games and strategic equivalence. In *Contributions to the theory of games*, Vol 1 (Annals of Mathematics Studies, 24). Edited by H. W. Kuhn, A. W. Tucker. Princeton University Press, pag 117-130, 1950.
- [13] J. Kobler, U. Schoning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhauser, 1993.
- [14] M. Mavronicolas, B. Monien, and K. W. Wagner. Weighted boolean formula games. In *Internet and Network Economics, Third International Workshop, WINE 2007*, volume 4858 of *LNCS*, pages 469–481, 2007.
- [15] A. McLennan and J. Berg. Asymptotic expected number of Nash equilibria of two-player normal form games. *Games and Economic Behavior*, 51:264–295, 2005.
- [16] J. Nash. Non-cooperative games. In *Classics in Game Theory*, pages 14–26, 1997.
- [17] N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani. *Algorithmic Game Theory*, Cambridge University Press, 2007.
- [18] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [19] Ch. Papadimitriou. Algorithms, games, and the internet. STOC, 749-753, 2001,
- [20] B. Peleg, J. Rosenmüller, and P. Sudhölter. The canonical extensive form of a game form: Part I symmetries. In *Current Trends in Economics, Advancement of Studies in Economics*, pages 367–387, 1999.
- [21] N. Pippenger. The complexity of computations by networks. *IBM Journal of Research and Development*, 31(2):235–243, 1987.

- [22] I. Powers. Limiting distribution of the number of pure strategy Nash equilibria in n -person games. *International Journal of Game Theory*, 19(3):277–286, 1990.
- [23] J. J. R. Greenlaw and W. Ruzzo. *Limits to Parallel Computation*. Oxford University Press, 1995.
- [24] J. H. Reif and S. R. Tate. On threshold circuits and polynomial computation. *SIAM J. Comput.*, 21(5):896–908, 1992.
- [25] G. Schoenebeck and S. P. Vadhan. The computational complexity of Nash equilibria in concisely represented games. In *ACM Conference on Electronic Commerce*, pages 270–279, 2006.
- [26] Y. Shoham. Computer science and game theory. *CACM* 51(8), 74-79, 2008.
- [27] K. Siorpaes and M. Hepp. Games with a purpose for the Semantic Web. *IEEE Intelligent Systems*:23(3), 50-60, 2008.
- [28] K. Siorpaes and M. Hepp. OntoGame: Weaving the Semantic Web by online games. In *The Semantic Web: Research and Applications*, LNCS 5021, 50-60, 2008.
- [29] P. Sudhölter, J. Rosenmüller, and B. Peleg. The canonical extensive form of a game form. Part II. Representation. *Journal of Mathematical Economics*, 33(3):299–338, 2000.
- [30] J. Torán. Personal Communication.

Appendix

A. Technical results for Section 2

Claim 1. *Given a WCFG $\Gamma = \langle 1^n, 1^r, 1^\ell, (\mathfrak{F}_i)_{1 \leq i \leq n} \rangle$ we can build in polynomial time in the size of Γ a Boolean Formula Game $\Gamma' = \langle 1^n, 1^\ell, (\varphi_{i,j})_{1 \leq i \leq n, 0 \leq j < \ell} \rangle$ with the same utilities and reciprocally.*

Proof. We transform a WCFG to a formula game through a sequence of steps. Let us start to consider a restricted form of utility.

Given $A = \{0, 1\}^r$, $a_i \in \{0, 1\}$ and $a = (a_1, \dots, a_r) \in A$ and given $v(a) = \sum_{1 \leq i \leq r} w_i a_i$, where each w_i has ℓ bits, we can compute in time $O(\ell r)$ formulas φ_j , $0 \leq j < \ell + \log r$, such that $v(a) = \sum_{0 \leq j < \ell + \log r} \varphi_j(a) 2^j$.

To prove the preceding fact, we use the ideas given in [4]. We define $x_{i,j}$ to take care in the future, of the bit j of the word w_i , eventually $x_{i,j} = w_{i,j} a_i$. Note that $y_i = (x_{i,\ell-1}, \dots, x_{i,0})$ is like a number of ℓ bits. Using a result given in [21] or [24], we can easily build in polynomial time a nonuniform TC^0 circuit $\text{IteratedSum}(y_1, \dots, y_r)$ giving the sum of the r numbers each one of ℓ bits. Let us compute the number of outputs of such a circuit.

With ℓ bits the biggest number written with ℓ bits has value $2^\ell - 1$. Therefore the sum of r numbers each one of ℓ bits is at most $k(2^\ell - 1)$ and this sum can be written with $\ell + \log k$ bits. Therefore $\text{IteratedSum}(y_1, \dots, y_r)$ outputs $\ell + \log k$ bits. From this circuit can be easily obtained a TC^0 circuit giving the bit j of this iterated sum. As $\text{TC}^0 \subseteq \text{NC}^1$ and circuits in NC^1 have logarithmic depth and polynomial size, in polynomial time we can find a formula $\phi_j(y_1, \dots, y_r)$ giving the bit j of $\text{IteratedSum}(y_1, \dots, y_r)$. To get the final result we have to substitute $x_{i,j}$ for $w_{i,j} a_i$ and we get

$$\varphi_j(a_1, \dots, a_r) = \phi_j((w_{1,\ell-1} a_1, \dots, w_{r,\ell-1} a_r), \dots, (w_{1,0} a_1, \dots, w_{r,0} a_r))$$

Let us consider another fact.

Given $\mathfrak{F} = \{(f_1, w_1), \dots, (f_r, w_r)\}$ such that, each w_i has ℓ bits and $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$, we can compute in polynomial time in the size of \mathfrak{F} formulas φ_j such that $u(a) = \sum_{1 \leq i \leq r} w_i f_i(a) = \sum_{0 \leq j < \ell + \log r} \varphi_j(a) 2^j$.

Let us prove it. Given $b \in \{0, 1\}^r$ consider the utility $v(b) = \sum_{1 \leq i \leq r} w_i b_i$ and using the preceding fact, find ϕ_j such that $v(a') = \sum_{1 \leq j \leq \ell + \log r} \phi_j(b) 2^j$. Now we identify $b_i = f_i(a)$ and $\varphi_j(a) = \phi_j(f_1(a), \dots, f_r(a))$. The transformation can be done in polynomial time.

Finally, to transform WCFG into boolean formula games we apply the last fact to each \mathfrak{F}_i .

Transform a boolean formula game into a WCFG is easy. We define $w_{i,j} = 2^j$ and $f_{i,j} = \varphi_{i,j}$. \square

B. Technical results of Section 3

Recall that, a strong isomorphism $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ of Γ into Γ' , allow us to define a mapping $\psi' = (p, f_1, \dots, f_k)$ from game $\text{BINARYACT}(\Gamma, \mu)$ to game $\text{BINARYACT}(\Gamma', \mu)$. If in ψ we have $\pi(i) = j$, as $\varphi_i : A_i \rightarrow A'_j$ is a bijection, we ask p to be a bijection $p : B_i \rightarrow B'_j$. Writing $A_i = A'_j = \{1, \dots, \ell\}$ and $B_i = \{i_1, \dots, i_\ell\}$ and $B'_j = \{j_1, \dots, j_\ell\}$, the action bijection $\varphi_i(p) = q$, $1 \leq p \leq \ell$, induces the bijection $p(i_p) = j_q$ between both blocks. This concludes the definition of the bijection between the players. All the f_i for $1 \leq i \leq k$ are identities.

Claim 2. *Given a strong isomorphism $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ from Γ to Γ' , the mapping $\psi' = (p, f_1, \dots, f_k)$ from $\text{BINARYACT}(\Gamma, \mu)$ to $\text{BINARYACT}(\Gamma', \mu)$ is a strong isomorphism.*

Proof. We state the proof as a sequence of claims.

ψ' maps any strategy profile b_i for players in B_i into a strategy profile $b'_{\pi(i)}$ for players in $B'_{\pi(i)}$, we write $\psi' : B_i \rightarrow B'_{\pi(i)}$ and $\psi'(b_i) = b'_{\pi(i)}$. This is clear because p gives a bijection between B_i and $B'_{\pi(i)}$.

If $|B_i| = \ell$, profile $b_i = 0^{p-1} 10^{\ell-p} = \text{binify}(p)$ maps into $\psi'(b_i) = 0^{q-1} 10^{\ell-q} = \text{binify}(q)$ iff $\varphi_i(p) = q$. As all the f_i 's are identities and p is a permutation of $\{1, \dots, \ell\}$, a binified action is mapped into a binified

action. Moreover as $p(i_p) = j_q$ the 1 in position p is mapped into the 1 in position q . As ψ' maps strategy profiles between blocks B_i and $B_{\pi(i)}$, we write $\psi'(\text{binify}(p)) = \text{binify}(\varphi_i(p))$.

Given $a = (a_1, \dots, a_n) \in A$ in Γ and $\text{binify}(a) = (\text{binify}(a_1), \dots, \text{binify}(a_n))$, it holds for binary action games $\psi'(\text{binify}(a)) = (\text{binify}(a'_1), \dots, \text{binify}(a'_n))$ such that $\text{binify}(a'_{\pi(i)}) = \text{binify}(\varphi_i(a_i))$.

For any $a \in A$ in Γ , it holds $\psi'(\text{binify}(a)) = \text{binify}(\psi(a))$. For $a = (a_1, \dots, a_n) \in A$ in Γ , it holds $\psi(a) = (a'_1, \dots, a'_n)$ with $a'_{\pi(i)} = \varphi_i(a_i)$. As $\text{binify}(\psi(a)) = (\text{binify}(a'_1), \dots, \text{binify}(a'_n))$ we get $\text{binify}(\psi(a)) = \psi'(\text{binify}(a))$.

For $p(i_p) = j_q$ it holds $u'_{j_q}(\psi'(\text{binify}(a))) = u'_{i_p}(\text{binify}(a))$. Observe that, as we have that $u'_{j_q}(\psi'(\text{binify}(a))) = u'_{j_q}(\text{binify}(\psi(a))) = u_j(\psi(a))$ and $u'_{i_p}(\text{binify}(a)) = u_i(a)$ and ψ is a morphism the identity holds.

As ψ' maps bijectively bad strategy profiles, in this case utilities are the penalty payoff μ and ψ' is a morphism. \square

Assume that $\psi' = (p, f_1, \dots, f_k)$ is an strong isomorphism between the games $\text{BINARYACT}(\Gamma_1, \mu)$ and $\text{BINARYACT}(\Gamma_2, \mu)$. As we will prove in the following lemma, p maps bijectively blocks of players and we can assume that all the f_i , $1 \leq i \leq k$ are identities. Therefore we define a mapping $(\pi, \varphi_1, \dots, \varphi_n)$ of Γ_1 to Γ_2 as follows. If B_i is mapped to $B'_{p(i)}$ we set $\pi(i) = p(i)$. If $\text{binify}(j)$ in block B_i is mapped into $\text{binify}(j')$ in $B'_{p(i)}$, we define $\varphi_i(j) = j'$.

Claim 3. *Given a strong isomorphism $\psi' = (p, f_1, \dots, f_k)$ between the games $\text{BINARYACT}(\Gamma_1, \mu)$ and $\text{BINARYACT}(\Gamma_2, \mu)$, the mapping $(\pi, \varphi_1, \dots, \varphi_n)$ is an strong isomorphism from Γ_1 to Γ_2 .*

Proof. Assume that $\psi' = (p, f_1, \dots, f_k)$ is an strong isomorphism between the games $\text{BINARYACT}(\Gamma_1, \mu)$ and $\text{BINARYACT}(\Gamma_2, \mu)$ having players N'_1 and N'_2 with $N'_1 = N'_2$. The strategy profiles in both binary action games are A'_1 and A'_2 . We state the proof as a sequence of claims.

$\psi' : A'_1 \rightarrow A'_2$ induces a bijection between $\psi' : \text{bad}(A'_1) \rightarrow \text{bad}(A'_2)$. Let $a' \in \text{bad}(A'_1)$ then $u'_i(a') = \mu$, as μ is a penalty payoff and ψ' is a morphism, $u'_{p(i)}(\psi(a')) = \mu$ but this forces $\psi(a') \in \text{bad}(A'_2)$ and $\psi'(\text{bad}(A'_1)) \subseteq \text{bad}(A'_2)$. Given $a' \in \text{bad}(A'_2)$, any player gets μ and then $\psi^{-1}(a) \in \text{bad}(A'_1)$. As $\text{good}(A'_1) = A'_1 \setminus \text{bad}(A'_1)$ there is also a bijection $\psi' : \text{good}(A'_1) \rightarrow \text{good}(A'_2)$.

Note that N'_1 can be partitioned into the different blocks of players as $N'_1 = B_1 \cup \dots \cup B_n$ and $N_2 = B'_1 \cup \dots \cup B'_n$ being n the number of players in Γ and Γ' .

Given a block B_k in N'_1 and $i, j \in B_k$ with $i \neq j$, it is impossible that $p(i)$ and $p(j)$ belongs to different blocks of N'_2 . Suppose that B_k has ℓ players and $1 \leq i < j \leq \ell$. Consider the strategy profile $b_k = \text{binify}(i) = 0^{i-1}10^{j-i-1}00^{\ell-j}$ for block B_k . All other blocs take the corresponding $\text{binify}(1)$. Then $a' = (b_1, \dots, b_n) \in \text{good}(A'_1)$ and $\psi'(a') \in \text{good}(A'_2)$. Therefore we have a factorization $\psi'(a) = (b'_1, \dots, b'_n)$ corresponding each b'_i , $1 \leq i \leq n$, to a binify process. For block B_k define the profile $c_k = \text{binify}(j) = 0^{i-1}00^{j-i-1}10^{\ell-j}$ and all other blocks keeps as before $\text{binify}(1)$ then $c = (c_1, \dots, c_n)$ is good, $\psi(c)$ is also good and factorizes as $\psi(c) = (c'_1, \dots, c'_k)$. Let us compare $\psi'(a) = (b'_1, \dots, b'_n)$ with $\psi'(c) = (c'_1, \dots, c'_k)$. Let $p(i) = i' \in B'_{k_1}$ and $p(j) = j' \in B'_{k_2}$ with $k_1 \neq k_2$. In fact the bits in $\psi'(a)$ and $\psi'(c)$ coincides anywhere except in positions corresponding to the players $i' \in B'_{k_1}$ and $j' \in B'_{k_2}$. Suppose $|B'_{k_1}| = \ell_1$. Consider the bijection f associated to the position i in B_k . This bijection can be an identity or a negation. When f is the identity, the 1 appearing in position i of block b_k is mapped into the 1 in position i' of b'_{k_1} , as this profile is binified , we have $b'_{k_1} = 0^{i'-1}10^{\ell_1-i'}$. Unfortunately the 0 appearing in position i of c_k will give $c'_{k_1} = 0^{\ell_1}$ turning a valid profile into an invalid profile. When f is a negation b'_{k_1} has a 0 in position i' and c'_{k_1} will have two 1's, giving a contradiction.

Permutation p maps bijectively each block B_i into another B'_j . Let $k \in B_i$ and $p(k) \in B'_j$ then $p(B_i) \subseteq B'_j$. Suppose that exists $l \in B'_j \setminus p(B_i)$, then $i' = \text{block}(p^{-1}(l))$ verifies $B_i \cap B_{i'} = \emptyset$. Let f and f' the bijections associated to the positions k in B_i and $p^{-1}(l)$ in $B_{i'}$. Let us consider two cases depending on the size of $B_{i'}$.

- Case $|B_{i'}| = 1$. When f' is the identity, defining $B'_j = \text{binify}(p(k))$ we force $B_{i'} = 0$. Fulfilling all the other blocks in $\text{BINARYACT}(\Gamma_2, \mu)$ with $\text{binify}(1)$ we get that ψ' maps a bad profile into a good one, but this is a contradiction. Consider the case, f' is a negation. In this case take $B'_j = \text{binify}(l)$ and we get a similar contradiction. The same argument allow us to assume that $|B_i| > 1$.

- Case $|B_{i'}| > 1$ and $|B_i| > 1$. When f' is the identity, any bijection associated to a position m in B_i is a negation. Take $B_{i'} = \text{binify}(p^{-1}(l))$ and $B'_j = \text{binify}(l)$ and $B_i = \text{binify}(m)$ as good profiles map into good profiles, the 1 in position m in B_i is transformed into a 0 in B'_j . Therefore $B_i = \text{binify}(1) = 10^{|B_i|-1}$ will give $|B_i| - 1 > 0$ 1's in B'_j and the number of 1's in such a block will be at least 2. Consider the case, f' is a negation. As B'_i has at least two positions, take a position m in such a block such that $m \neq p^{-1}(l)$ and fix $B_i = \text{binify}(m)$. This profile fix a 0 in position $p^{-1}(l)$ of $B_{i'}$ and a 1 in position l of B_j and we apply the preceding argument.

We can assume that all the bijections f are identities. Suppose that $p(B_i) = B'_j$. We consider three cases depending on the size of B_i .

- Case $|B_i| = 1$. In this $B_{j'}$ has also 1 element. As good profiles map into good profiles, the bijection associated to this element has to be the identity.
- Case $|B_i| = 2$. See in detail the different possibilities. Call $B_i = \{1, 2\}$ and $B'_j = \{1', 2'\}$ and call the corresponding bijections f_1 and f_2 . There are two possibilities for p . Consider the case $p(1) = 1'$ and $p(2) = 2'$ and look at the different possibilities for f_i .
 - When $f_1 = f_2$ are identities the property holds.
 - When f_1 is the identity and f_2 is a negation we get a contradiction because $B_i = \text{binify}(2)$ is mapped into $B'_j = 00$. When f_1 is a negation and f_2 is the identity, the same argument applies with $B_i = \text{binify}(1)$.
 - When f_1 and f_2 are negations, we have to deal with care. Remark that bad profiles maps into bad profiles because 00 maps to 11 and 11 maps to 00. Also good profiles maps into good profiles because 10 maps to 01 and 01 maps into 10. Nothing bad happens in this case. To get identities we define another morphism ψ'' such that $p(1) = 2'$, $p(2) = 1'$ and $f_1 = f_2$ identities. Note that, under ψ'' , profiles bad maps into bad profiles because 00 maps to 00 and 11 maps into 11. Much better good profiles map in ψ' and in ψ'' in the same because 10 maps to 01 and 01 maps to 10. Note that ψ'' and ψ' are isomorphic, therefore we can take ψ'' where f_1 and f_2 are identities.

When $p(1) = 2'$ and $p(2) = 1'$ the proof is similar to the preceding case.

- Case $|B_i| > 2$. If all the f 's associated to B_i are identities the property holds, otherwise there is a position l such that f_l is a negation. As $B_i = \text{binify}(l)$ maps to a good profile, exists l' such that $B'_j = \text{binify}(l')$, moreover as f_l is a negation $l' \neq p(l)$ and $f_{p^{-1}(l')}$ is also a negation. As $|B_i| > 2$, there is a position $k \notin \{l, p^{-1}(l')\}$ in B_i and $B_i = \text{binify}(k)$ give at least two 1's corresponding to positions l and $p^{-1}(l')$, therefore we get a contradiction.

To summarize, given a strong isomorphism $\psi' = (p, f_1, \dots, f_k)$ between the games $\text{BINARYACT}(\Gamma_1, \mu)$ and $\text{BINARYACT}(\Gamma_2, \mu)$ we have that p maps bijectively blocks of players and that we can assume that all the f_i , $1 \leq i \leq k$ are identities.

let us consider the mapping $(\pi, \varphi_1, \dots, \varphi_n)$ of Γ_1 to Γ_2 . The permutation of players π mimics the block permutation induced by p , thus if B_i is mapped to $B'_{p(i)}$ we set $\pi(i) = p(i)$. The i action bijection is defined as follows. The action j in A_i corresponds in $\text{BINARYACT}(\Gamma_1, \mu)$ to the profile $\text{binify}(j)$ in block B_i . As this block is mapped into $B'_{p(i)}$, the profile is mapped into another good profile $\text{binify}(j')$ and we define $\varphi_i(j) = j'$. It is straightforward to show that the mapping $(\pi, \varphi_1, \dots, \varphi_n)$ is an strong isomorphism from Γ_1 to Γ_2 . \square

Let $\psi' = (p, f_1, \dots, f_k)$ be an isomorphism between $\text{BINARY}(\Gamma_1, t, m)$ and $\text{BINARY}(\Gamma_2, t, m)$. Observe that in such a case Γ_1 and Γ_2 have the same number n of players. Based on ψ' we can define an isomorphism π from Γ to Γ' as we see in the following claim.

Claim 4. *Let $\psi' = (p, f_1, \dots, f_k)$ be a strong isomorphism between the two games $\text{BINARY}(\Gamma_1, t, m)$ and $\text{BINARY}(\Gamma_2, t, m)$. The mapping $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ in which π is the restriction of p to block $B_{1,0}$ and, for any $1 \leq i \leq n$, $\varphi_i = f_i$ is a strong isomorphism from Γ to Γ' .*

Proof. Let $a = x b_1 \dots b_n z$ be a strategy profile for $\text{BINARY}(\Gamma_1, t, m)$, where $x = x_1 \dots x_n$, $b_i = b_{i_1} \dots b_{i_t}$ and $z = z_0 \dots z_{m+1}$ are binary words.

We represent the utilities of a as a binary string $u(a) = u_1, \dots, u_{n+tn+m+2}$, when we speak in general about a property of the construction we will not use subindices, however we will use $u_1(a)$ and $u_2(\psi(a))$ to denote vector utilities for the first or second game. As usual, for a binary string w we use $|w|_1$ to denote the number of 1's present in w . Observe that, for a strategy profile a , $|u_1(a)|_1 = |u_2(\psi(a))|_1$. According with the definition of utilities for $\text{BINARY}(\Gamma, t, m)$ we have that, for any profile a ,

1. if $z = 0^{m+2-\ell} 1^\ell$, then $|u(a)|_1 = \ell$, and at least one player in the block B_{n+1} gets utility 1.
2. if $z = 00^{j-1} 10^{m+1-j}$, for some $1 \leq j < t$, then $n + m + 2 \leq |u(a)|_1 \leq n + t + m + 2$ and all the players in the block B_{n+1} get utility 1.
3. if $z = 10^{m+1}$, then $|u(a)|_1$ is a multiple of $t + 1$ and all the players in the block B_{n+1} get utility 0.
4. In the remaining cases, $|u(a)|_1 = n + tn + m + 2$.

The permutation p maps the block $n + 1$ of Γ_1 to the block $n + 1$ of Γ_2 , furthermore the restriction of p to $B_{1,n+1}$ is the identity and, for any $j \in B_{1,n+1}$, f_j is the identity. The claim follows from condition 1, as this is needed to guarantee that, when $z = 0^{m+2-\ell} 1^\ell$, $|u_1(a)|_1 = |u_2(\psi(a))|_1$.

Let $\text{BIT}(j)$ be the set of players that appear at the j -th position in some block B_1, \dots, B_n .

For any $1 \leq j \leq t$, the permutation p maps $\text{BIT}_1(j)$ to $\text{BIT}_2(j)$. Furthermore for any $i \in \text{BIT}_1(j)$, f_i is the identity. The rigidity of ψ on block $B_{1,n+1}$ forces that the profile a in which all the player $i \in \text{BIT}_1(j)$ select action 1 and $z = 00^{j-1} 10^{m+1-j}$, creates an utility string with exactly $2n + m$ ones, therefore the unique possibility for ψ to remain as an isomorphism is the one expressed in the claim.

As a consequence of the previous claims we have that the permutation p maps the players in block $B_{1,0}$ to block $B_{2,0}$.

For any $1 \leq i \leq n$, the permutation p maps the block $B_{1,i}$ to the block $B_{2,p(i)}$. Furthermore, for any $1 \leq j \leq n$, the player in the j -th position of $B_{1,i}$ is mapped by ψ to the j -th position of $B_{2,p(i)}$. Consider the profile a in which $z = 10^{m+1}$, and x verifies that, $b_i = u_i(x)$ and, for any $\ell \neq i$, $b_\ell \neq u_\ell(x)$. The rigidity of ψ on block $B_{1,n+1}$ forces that in $|u_1(a)|_1 = t + 1$. In $u_2(\psi(a))$ we know that the utility for player $p(i)$ has to be one and therefore all the utilities of all the players in $B_{2,p(i)}$ must be one. Again the unique possibility is the one expressed in the first part of claim. The second part follows as a consequence of the first part and the previous claim.

Putting all together, we can define a morphism $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ in which π is the restriction of p to block $B_{1,0}$ and, for any $1 \leq i \leq n$, $\varphi_i = f_i$. Consider the profile a in which $z = 10^{m+1}$, and x verifies that, for any $1 \leq i \leq n$, verifies $b_{1,i} = u_i(x)$, then $u_1(a)$ has a 1 in all positions except the last $m + 2$ that hold a 0. Furthermore, $\psi(a) = \pi(x) b_{2,1} \dots b_{2,n} z$ and, for any $1 \leq i \leq n$, if $b_{2,\pi(i)} = b_{1,i}$. Therefore, we have that, for any $1 \leq i \leq n$, $u_1(x) = u_2(\psi(x))$, therefore ψ is an isomorphism. \square

Let us consider some technical detail for the reverse implication of Lemma 3. Let $\psi' = (p, f_1, \dots, f_{4n+2})$ be a congruence morphism between C_Γ and $C_{\Gamma'}$. When $a = (x, y, c, b)$, we note $p(x_i)$ the position of the image of x_i and we take similar conventions for $p(y_i)$, $p(c_i)$ and $p(d_i)$. Similarly the value of the image of x_i will be $f(x_i)$. For simplicity we use $C(x, y, c, d)$ to denote a boolean circuit or the output of a boolean circuit when the input is set to a particular value. In general we use a minterm notation, for input assignments instance. For example $c = 010^n$ is the input assignment in which only the second variable of block C is activated.

Claim 5. Let $\psi' = (p, f_1, \dots, f_{4n+2})$ be a congruence morphism between C_Γ and $C_{\Gamma'}$. Then ψ preserves the structure of the C and Y blocks. Furthermore the functions f_i , for i in block C or Y , is the identity.

Proof. The values of the variables c_1, \dots, c_{n+2} are used to activate the different circuits C_1, \dots, C_{n+2} that form C_Γ . Each of those circuits has different properties. As before we state the proof as a series of claims.

Permutation p maps the variables in the C -block of C_Γ to the C -block of $C_{\Gamma'}$. Assume for contradiction that there are $k \geq 1$ variables mapped from outside the C -block of C_Γ to the C -block of $C_{\Gamma'}$. Let us see that we can force an assignment a in which there is only one 1 in position 2 of the C -block for which $C_\Gamma(a)$ is true, while in $\psi'(a)$ there are at least two 1's, and that is impossible. Consider in detail the case $k = 1$.

As in $a = (a_1, \dots, a_{4n+2})$ one position leaves the block C , there is a_i in blocks X, Y or D entering the block C' in $\psi'(a)$ and $c'_{p(i)} = f(a_i)$. We have to consider two cases based on $p(c_2)$.

- Case $p(c_2)$ is a position in C' . In this case $c'_{p(c_2)} = f(c_2)$. When the bijection f is the identity, $c'_{p(c_2)} = c_2$, we have to consider the different possible origins of a_i .
 - When a_i is located in X we have $a_i = x_i$. Fix $x_i \in \{0, 1\}$ to the value such that $f(x_i) = 1$. Consider $a = (x, y, c, d)$ such that $x = 0^{i-1}x_i0^{n-i}$, $y = 10^n$, $c = 010^n$ and $d = 0^n$. It holds $C_\Gamma(a) = F_2(a) = 1$ but $C_{\Gamma'}(\psi(a)) = 0$ because C' contains at least two ones in $c'_{p(i)}$ and $c'_{p(c_2)}$. When a_i is located in D we have $a_i = d_{i-3n+2}$, the analysis is similar.
 - When a_i is located in Y it holds $y_{i-n} = a_i$. Fix the value of y_{i-n} such that $f(y_{i-n}) = 1$. As the Y block has at least two positions, there is j such that $j \neq i-n$ and we can fix $y_j = 1$. Then $a = (x, y, c, d)$ with $x = 0^n$, $y = 0 \dots 0y_{i-n}0 \dots 0y_j0 \dots 0$ (case $i-n \geq j$, other cases are similar) $c = 010^n$ and $d = 0^n$ verifies $C_\Gamma(a) \neq C_{\Gamma'}(\psi(a))$.

Now we have to consider what happens when f is a negation, $c'_{p(c_2)} = -c_2$. As just one position in C is mapped out C' , exists j such that $p(c_j)$ is not a position in C' , therefore $c_j \neq c_2$. Taking C also as a set, we have that for any $c_k \in C \setminus \{c_2, c_j\}$ it holds that $p(c_k)$ is located in C' . We consider two cases

- For all $c_k \in C \setminus \{c_2, c_j\}$ it holds that $f(c_k)$ is the identity, $c'_{p(c_k)} = c_k$. We can force the value $c'_{p(i)}$ to be 0, therefore for $C = 010^n$ we get $C' = 0^{n+2}$. As we can choose always choose the block Y having at least one 1 we can easily build a profile a such that $C_\Gamma(a) \neq C_{\Gamma'}(\psi(a))$.
- Exists $c_k \in C \setminus \{c_2, c_j\}$ such that $f(c_k)$ is a negation, $c'_{p(c_k)} = -c_k$. Fixing $C = 010^n$ we get $c_k = 1$. Forcing $c'_{p(i)}$ to be 1, the block C' will have at least two 1. As Y has at least two positions, we can easily build a not fulfilling the congruence.
- Case $p(c_2)$ is a position in X', Y' or D' . Intuitively c_2 leaves the C block and we have to look at the elements $c_k \in C \setminus \{c_2\}$. We consider two cases.
 - For any $c_k \in C \setminus \{c_2\}$, the bijection $f(c_k)$ is the identity. Fixing $c'_{p(i)}$ to be 0 and $C = 010^n$ we have $C' = 0^{n+2}$ and we can build a not fulfilling the congruence.
 - Exists $c_k \in C \setminus \{c_2\}$ such that $f(c_k)$ is a negation. Fixing $c'_{p(i)}$ to be 1 and $C = 010^n$ we have that C' has at least two 1 and we can build a not fulfilling the congruence.

This concludes analysis of the impossibility when $k = 1$. When $k > 1$ the analysis follows the same ideas.

All the functions associated to variables in the C -block are the identity. If there are more than two negations, ψ' transform an input with exactly one 1 in block C to a situation with two 1's in block C . If there is one negation, the situation in which all the bits in C are set to 0, is transformed into another one in which there is only one 1.

We have $p(c_1) = c'_1$ and $p(c_2) = c'_2$. Let us consider the possible misplacements for $p(c_1)$. We have two cases

- The index $p(c_1) = c_i$ is located in one of the last n positions of C' . Then $C = 10^{n+1}$ is mapped into $C' = 0^{i-1}10^{n+2-i}$ and the activated circuits are $C_\Gamma = C_1$ and $C_{\Gamma'} = C_i$. Fix $x'_{i-2} = y'_{i-2} = 0$ and $d'_{i-2} = 1$ and $C_{\Gamma'} = 1$. As p maps C into C' it also maps $X \cup Y \cup D$ bijectively into $X' \cup Y' \cup D'$. As $|X| + |D| \geq 4$ and $x'_{i-2} y'_{i-2} d'_{i-2}$ are fixed, only three antiimages have been fixed and there is at least a “free” position in the $X \cup D$ blocks. Suppose that x_j is the free position (the case d_j is similar) and look at the corresponding d_j . If d_j is fixed take $x_j = d_j$, if d_j is free define $d_j = x_j = 1$. In both cases $C_1 = 0$.

- The index $p(c_1)$ points to c'_2 , that is $c_2 = c_1$. The block $C = 10^{n+1}$ is mapped to 010^n and under this situation $C_\Gamma = C_1$ and $C_{\Gamma'} = C_2$. Fix an arbitrary bit y'_i in Y' and look at the possible antiimage of y'_i . In the bijection $p : X \cup Y \cup D \rightarrow X' \cup Y' \cup D'$ only the antiimage of y'_i fixed. Suppose that the antiimage is one is an x_j , then fix $d_j = x_j$. When the antiimage belongs to Y , choose one x_k and fix it and the corresponding d_k to 1. When the antiimage is d_j fix x_j to the same value. In all the cases $C_1 = 0$.

We have proved $p(c_1) = c'_1$. To prove $p(c_2) = c'_2$ follow the same ideas.

Permutation p maps the variables in the Y -block of C_Γ to the Y -block of $C_{\Gamma'}$. Furthermore, the functions associated to the variables in the Y -block are the identity. This is a consequence of the rigidity of ψ' on c_2 and the definition of the formula C_2 . \square

Claim 6. Let $\psi' = (p, f_1, \dots, f_{4n+2})$ be a congruence morphism between C_Γ and $C_{\Gamma'}$. Let π to be the permutation on $\{1, \dots, n\}$ such that $p(c_{i+2}) = c'_{\pi(i)+2}$, then $p(x_i) = x'_{\pi(i)}$ iff $p(d_i) = d'_{\pi(i)}$ and $p(x_i) = d'_{\pi(i)}$ iff $p(d_i) = x'_{\pi(i)}$.

Proof. Consider π to be the permutation on $\{1, \dots, n\}$ such that $p(c_{i+2}) = c'_{\pi(i)+2}$.

For any $1 \leq i \leq n$, positions i in blocks X , Y and D of C_Γ are mapped to positions $\pi(i)$ of blocks X , Y and D of $C_{\Gamma'}$. Furthermore, $p(y_i) = y'_{\pi(i)}$. This result is insured by the definition of the n formulas C_{i+2} , as each of them forces to combine the input bits x_i and y_i with d_i . The last part follows taking into account that the Y -block of C_Γ is mapped to the Y -block of $C_{\Gamma'}$.

The above result implies that, for any $1 \leq i \leq n$, either $p(x_i) = x_{\pi(i)}$ or $p(x_i) = d_{\pi(i)}$. Furthermore, the permutation associated to x_i and d_i must be the same,

$$p(x_i) = x'_{\pi(i)} \text{ iff } p(d_i) = d'_{\pi(i)} \text{ and } p(x_i) = d'_{\pi(i)} \text{ iff } p(d_i) = x'_{\pi(i)},$$

otherwise, we can find an input for which $C_{i+2}(x_i, y_i, d_i) = 1$ while we also have $C_{\pi(i)+2}(\psi'(x_i, y_i, d_i)) = 0$. \square

Claim 7. The game $\text{BINARYACT}(\Gamma, \mu)$ when Γ is a formula game in general form is a formula game. The game $\text{BINARY}(\Gamma, t, m)$ when Γ is a binary actions formula game in general form is a formula game. A description in general form of the games $\text{BINARYACT}(\Gamma, \mu)$ and $\text{BINARY}(\Gamma, t, m)$ can be computed in polynomial time. Furthermore, a description of the circuit C_Γ , for a binary formula game Γ , can be obtained in polynomial time.

Proof. To prove this theorem we show first that the game $\text{BINARYACT}(\Gamma, \mu)$, for a given formula game in general form $\Gamma = \langle 1^n, A_1, \dots, A_n, 1^\ell, (\varphi_{i,j})_{1 \leq i \leq n, 0 \leq j < \ell} \rangle$, as defined in Page 9, is a formula game whose description can be computed in polynomial time.

Recall that the utility functions of $\text{BINARYACT}(\Gamma, \mu)$ are defined as follows:

$$u'_i(a') = \begin{cases} u_{\text{block}(i)}(\text{binify}^{-1}(a')) & \text{if } a' \in \text{good}(A'), \\ \mu & \text{if } a' \in \text{bad}(A'). \end{cases}$$

Where $\text{good}(A') = \{\text{binify}(a) \mid a \in A\}$ and $\text{binify}(j) = 0^{j-1}10^{k_i-j}$. Observe that to compute the utilities we will need to show that the function binify^{-1} can be represented by boolean formula as well as the property $a' \in \text{good}(A')$. For doing so, we show that they can be computed in NC^1 , and use the same argument used in the proof of Lemma 1 to construct the formulas.

For $a' \in \text{good}(A')$ it must happen that the sum of all its bits is one. And of course this can be computed in NC^1 . To compute $\text{binify}^{-1}(a')$ for some $a' \in \text{good}(A')$, let's assume that $a' = 0^{j-1}10^{k_i-j}$. We compute the suffix sum of the bits of a' , thus getting $b = 1^j0^{k_i-j}$. Then j is the sum of the bits of b .

Finally, using the formula for $a' \in \text{good}(A')$ and the ones that compute the bits of $\text{binify}^{-1}(a')$, combined with the fact that μ is a constant, and the formulas describing the utilities of the player's in Γ , we can construct the set of formulas that describe the utilities for $\text{BINARYACT}(\Gamma, \mu)$ in polynomial time. Therefore,

we have that the ISO problem for formula games in general form is equivalent to the the ISO problem for formula games in general form with binary actions, according to Lemma 1.

Now we show that given $\Gamma = \langle 1^n, A_1, \dots, A_n, 1^\ell, (\varphi_{i,j})_{1 \leq i \leq n, 0 \leq j < \ell} \rangle$, a formula game in general form with binary actions, the game $\text{BINARY}(\Gamma, t, m)$, as defined in Page 10, is a formula game whose description can be computed in polynomial time.

Recall that $\text{BINARY}(\Gamma, t, m)$ is the game $(N', (A'_i)_{i \in N'}, (u'_i)_{i \in N'})$ where $N' = \{1, \dots, k\}$ and, for any $i \in N'$, $A'_i = \{0, 1\}$ where $k = n + tn + m + 2$. The set N' is partitioned into $n + 2$ consecutive intervals B_0, \dots, B_n, B_{n+1} so that the interval B_0 has exactly n players, for $1 \leq i \leq n$, the block B_i has t players, finally block B_{n+1} has $m + 2$ players. As before a strategy profile a is usually factorized as $a = x b_1 \dots b_n z$ where now $x = x_1 \dots x_n$, $b_i = b_{i\ell-1} \dots b_{i0}$ and $z = z_0 \dots z_{m+1}$. Observe that if the formula game uses ℓ formulas per player then $t = \ell + 1$.

To express the utilities by a boolean formula, we consider the following auxiliary formulas, for $z = z_0, \dots, z_{m+1}$:

$$\begin{aligned} \text{FROM}_i(z) &= \left(\bigwedge_{j=0}^{i-1} \neg z_j \right) \wedge \left(\bigwedge_{j=i}^{m+1} z_j \right) \quad \text{for } 0 \leq i \leq m+1 \\ \text{ONLY}_i(z) &= \left(\bigwedge_{j=0}^{i-1} \neg z_j \right) \wedge z_i \wedge \left(\bigwedge_{j=i+1}^{m+1} \neg z_j \right) \quad \text{for } 0 \leq i \leq m+1 \end{aligned}$$

The previous formulas allow us to express the different conditions considered in the definition of the game $\text{BINARY}(\Gamma, t, m)$.

$$\begin{aligned} \text{ONE}(z) &= \bigvee_{i=0}^{m+1} \text{FROM}_i(z) \\ \text{TWO}(z) &= \bigvee_{i=1}^t \text{ONLY}_i(z) \\ \text{THREE}(z) &= \text{ONLY}_0(z) \\ \text{FOUR}(z) &= \neg(\text{ONE}(z) \vee \text{TWO}(z) \vee \text{THREE}(z)) \end{aligned}$$

Note that, predicates ONE, TWO, TWO, FOUR give us a partition of the strategy profiles. Recall that the utility of player i in Γ is given by the equation $u_i(a_1, \dots, a_n) = \sum_{0 \leq j < \ell} \varphi_{i,j}(a_1, \dots, a_n) 2^j$. We also consider the formula

$$\text{EQU}_i(x, b_i) = \bigwedge_{j=0}^{\ell} (\varphi_{ij}(x) \wedge b_{ij}) \vee (\neg \varphi_{ij}(x) \wedge \neg b_{ij}),$$

which express the fact that b_i is the utility of player i in game Γ .

Now we provide a formula for each ‘‘type of player’’ that allows to compute their utility in game $\text{BINARY}(\Gamma, t, m)$.

- Utility for player α in position β of block j ($1 \leq j \leq n$). The formula is expressed as disjunction of the four cases.
 - When $\text{ONE}(z)$ holds, the utility is 0. This give us a term $\text{ONE}(z) \wedge 0$ equivalent to 0.
 - When $\text{TWO}(z)$ holds there are two cases. When $\text{ONLY}_\beta(z)$ holds, the position of the player α inside the block j coincides with the position of the 1 in z and then the utility is $b_{j\beta}$. When $\text{ONLY}_\beta(z)$ is false the utility is 0. Therefore this part contributes with a term $\text{TWO}(z) \wedge \text{ONLY}_\beta(z) \wedge b_{j\beta}$.
 - When $\text{THREE}(z)$ holds, all the players is block j have the same boolean utility defined as the value of the expression $(u_j(x) = b_j)$. This part is encoded as $\text{THREE}(z) \wedge \text{EQU}_j(x, b_j)$.
 - When $\text{FOUR}(z)$ holds, the value of the utility is 1, therefore we have a term $\text{FOUR}(z) \wedge 1$.

Using basic properties of boolean functions we obtain

$$\Psi_\alpha(a) = (\text{TWO}(z) \wedge \text{ONLY}_\beta(z) \wedge b_{j\beta}) \vee (\text{THREE}(z) \wedge \text{EQU}_j(x, b_j)) \vee \text{FOUR}(z)$$

- Utility for player α in position β of block 0

$$\Psi_\alpha(a) = \text{TWO}(z) \vee (\text{THREE}(z) \wedge \text{EQU}_\beta(x, b_\beta)) \vee \text{FOUR}(z)$$

- Utility for player α in position β of block $n + 1$

$$\Psi_\alpha(a) = (\text{ONE}(z) \wedge \text{FROM}_\beta(z)) \vee \text{TWO}(z) \vee \text{FOUR}(z)$$

It is straightforward to show that the previous formulas can be written in polynomial time. Thus using Lemma 2, we have that the ISO problem for formula games in general form is equivalent to the the ISO problem for binary formula games.

The last step is to show that given a binary formula game Γ the boolean circuit C_Γ as defined in Page 11 can be described by a formula. From the definition of C_Γ it follows trivially that C_k ($1 \leq k \leq n + 2$) can be described by formulas as the utility for the player are given by a formula. Consider the formulas:

$$\begin{aligned} \text{ONLY}_i(c_1, \dots, c_{n+2}) &= \neg c_1 \wedge \dots \wedge \neg c_{i-1} \wedge c_i \wedge \neg c_{i+1} \wedge \dots \wedge \neg c_{n+2}, \quad 1 \leq i \leq n + 2 \\ \text{EXONE}(c_1, \dots, c_{n+2}) &= c_1 \vee \dots \vee c_{n+2} \\ \text{MOREONE}(c_1, \dots, c_{n+2}) &= \bigvee_{1 \leq i < j \leq n+2} (c_i \wedge c_j) \end{aligned}$$

Then C_Γ can be expressed as a disjunctions of the three cases. When $\neg \text{EXONE}(c)$ or $\text{MOREONE}(c)$ holds the result is 0, otherwise the value is computed by a disjunction of terms $\text{ONLY}_j(c) \wedge C_j(a)$. Therefore C_Γ is expressed as

$$\bigvee_{j=1}^{n+2} \text{ONLY}_j(c) \wedge C_j(a).$$

It is straightforward to show that a description of the previous circuit can be computed in polynomial time. Thus using Lemma 3, we have that the ISO problem for formula games in general form is equivalent to the FORMULAISO problem. \square

C. Technical results of Section 4

Claim 8. Let $\psi : \Gamma_1 \rightarrow \Gamma_2$ be a strong isomorphism between binay games such that $\psi = (\pi, \varphi_1, \dots, \varphi_n)$. The mapping $\psi' : \text{CHECKW}(\Gamma_1) \rightarrow \text{CHECKW}(\Gamma_2)$ defined as $\psi' = (p, f_1, \dots, f_{n+1})$ where, for $1 \leq i \leq n$, $p(i) = \pi(i)$, $f_i = \varphi_i$ and for $n + 1$, $p(n + 1) = n + 1$, f_{n+1} is a strong isomorphism between $\text{CHECKW}(\Gamma_1)$ and $\text{CHECKW}(\Gamma_2)$.

Proof. Let $a' = (a_1, \dots, a_n, a_{n+1})$ be a strategy profile in $\text{CHECKW}(\Gamma_1)$ we write $a' = (a, a_{n+1})$ with $a = (a_1, \dots, a_n)$. By the definition of ψ' it hods $\psi'(a') = (\psi(a), a_{n+1})$ because $f_{n+1}(a_{n+1}) = a_{n+1}$. Let us prove that ψ' is strong isomorphism. Note that for $1 \leq i \leq n$, as ψ is a strong isomorphism, $u_{\pi(i)}(\psi(a)) = u_i(a)$ and therefore

$$u'_{p(i)}(\psi'(a')) = (u_{\pi(i)}(\psi(a)) = (a'_{n+1} \bmod 2)) = (u_i(a) = (a'_{n+1} \bmod 2)) = u'_i(a')$$

It remains the case $n + 1$. As $p(n + 1) = n + 1$ and f_{n+1} is the identity $u'_{n+1}(\psi'(a)) = a'_{n+1} = u_{n+1}(a)$. \square

Claim 9. Let $\psi' = (p, f_1, \dots, f_{n+1})$ be a weak isomorphism $\psi' : \text{CHECKW}(\Gamma_1) \rightarrow \text{CHECKW}(\Gamma_2)$, it holds $f_{n+1} = \text{Id}$.

Proof. As in the proof of Lemma 5 let ψ be ψ' restricted to players $1, \dots, n$, that is $\psi = (\pi, \varphi_1, \dots, \varphi_n)$, then $\pi(i) = p(i)$ and $\varphi_i = f_i$ for $1 \leq i \leq n$ and $\psi'(a') = (\psi(a), f_{n+1}(a'_{n+1}))$. In $\text{CHECKW}(\Gamma_1)$ player $n + 1$ has the following chain of strict preferences

$$(a, 0) \prec_{n+1} (a, 1) \prec_{n+1} (a, 2) \prec_{n+1} (a, 3)$$

As ψ' is a weak morphism, preferences of player $n + 1$ in $\text{CHECKW}(\Gamma_2)$ verify

$$(\psi(a), f_{n+1}(0)) \prec_{n+1} (\psi(a), f_{n+1}(1)) \prec_{n+1} (\psi(a), f_{n+1}(2)) \prec_{n+1} (\psi(a), f_{n+1}(3))$$

This forces $u'_{n+1}(f_{n+1}(0)) < u'_{n+1}(f_{n+1}(1)) < u'_{n+1}(f_{n+1}(2)) < u'_{n+1}(f_{n+1}(3))$ and therefore

$$u'_{n+1}(f_{n+1}(0)) = 0, u'_{n+1}(f_{n+1}(1)) = 1, u'_{n+1}(f_{n+1}(2)) = 2, u'_{n+1}(f_{n+1}(3)) = 3$$

The only possibility to fulfill the preceding equalities is to take $f_{n+1}(a'_{n+1}) = a'_{n+1}$ for $a'_{n+1} \in \{0, 1, 2, 3\}$. \square

Assume that $\psi' = (p, f_1, \dots, f_k)$ is a weak isomorphism between the two games $\text{BINARYACTW}(\Gamma_1)$ and $\text{BINARYACTW}(\Gamma_2)$. As we prove in the following lemma all the f_α are identities and p induces a permutation into the blocks, therefore ψ' induces a permutation π on $\{1, \dots, n\}$. For a player α in position j inside block B_i , let $\varphi(j)$ be the position of player $p(\alpha)$ in block $\pi(i)$.

Claim 10. *Given a weak isomorphism $\psi' = (p, f_1, \dots, f_k)$ between $\text{BINARYACTW}(\Gamma_1)$ and $\text{BINARYACTW}(\Gamma_2)$, the mapping $(\pi, \varphi_1, \dots, \varphi_n)$ is a weak isomorphism between Γ_1 and Γ_2 .*

Proof. If $\psi' = (p, f_1, \dots, f_k)$ is a weak isomorphism between the games $\Gamma'_1 = \text{BINARYACTW}(\Gamma_1)$ and $\Gamma'_2 = \text{BINARYACTW}(\Gamma_2)$. We state the proof as a series of claims.

Given players $\alpha \in B_i$ and $\alpha' \in B_j$ with $i \neq j$ it holds $X_0(\alpha) \neq X_0(\alpha')$. When $k_i \neq k_j$ the proof is direct because $X_0(\alpha)$ has a cardinality $2^{k-k_i}(2^{k_i} - k_i)$ which is different from the cardinality of $X_0(\alpha')$. Consider the case $k_i = k_j$. Assume that block i precedes block j and consider a profile schematized as follows

$$a = (b_1, \dots, b_{i-1}, \text{bad}_i, b_{i+1}, \dots, b_{j-1}, \text{good}_j, b_{j+1}, \dots, b_n)$$

where bad_i is a bad profile in A^i and good_j is a good profile in A^j . It holds $a \in X_0(\alpha)$ but $a \notin X_0(\alpha')$.

Given the permutation p map and a player α , sets $X_0(\alpha)$ and $X_0(p(\alpha))$ are both non empty. Player α occupies forcibly a position into a block; suppose that B_i is such a block. As $\text{good}(A^i)$ has $k_i \geq 1$ elements the set $A^i \setminus \text{good}(A^i)$ contains $2^{k_i} - k_i > 0$ elements. By the same reason $X_0(p(\alpha))$ is not empty.

It holds $\psi'(X_0(\alpha)) = X_0(p(\alpha))$ for any player α . First note that $\psi'(X_0(\alpha)) \subseteq X_0(p(\alpha))$. Otherwise there is $a' \in \psi'(X_0(\alpha)) \setminus X_0(p(\alpha))$ and $a'' \in X_0(p(\alpha))$ (because $X_0(p(\alpha))$ is not empty) such that $a'' \prec_{p(\alpha)} a'$. Then $\psi'^{-1}(a'') \prec_\alpha \psi'^{-1}(a')$ but this is impossible because $\psi'^{-1}(a'') \in X_0(\alpha)$ and therefore, $\psi'^{-1}(a')$ is a less preferred element. Suppose that $\psi'(X_0(\alpha)) \neq X_0(p(\alpha))$. Let $a' \in X_0(p(\alpha)) \setminus \psi'(X_0(\alpha))$ and consider $\psi'^{-1}(a')$. If $\psi'^{-1}(a')$ belongs to $X_0(\alpha)$ we get a contradiction. If we assume $\psi'^{-1}(a') \notin X_0(\alpha)$, exists $b \in X_0(\alpha)$ such that $b \prec_\alpha \psi'^{-1}(a')$. Then $\psi(b) \prec_{p(\alpha)} a'$, but this is impossible because a' is a less preferred element.

It holds $p(B_{\text{block}(\alpha)}) = B_{\text{block}(p(\alpha))}$ for all α . Let α and α' be players in block B_i , that is $\text{block}(\alpha) = \text{block}(\alpha') = i$. As $X_0(\alpha) = X_0(\alpha')$ it holds $\psi(X_0(\alpha)) = X_0(p(\alpha)) = X_0(p(\alpha'))$. Both $X_0(p(\alpha)) = X_0(p(\alpha'))$ iff $\text{block}(p(\alpha)) = \text{block}(p(\alpha'))$.

Thus, ψ' induces a permutation π on $\{1, \dots, n\}$ such that $\pi(B_i) = B_{\pi(i)}$ moreover $\pi(\text{block}(\alpha)) = \text{block}(p(\alpha))$. For a player α in position j inside block B_i , let $\varphi(j)$ be the position of player $p(\alpha)$ in block $\pi(i)$. Therefore we define a mapping $\psi = (\pi, \varphi_1, \dots, \varphi_n)$.

It holds $\psi'(X_1(\alpha)) = X_1(p(\alpha))$ for any player α . There are two cases depending on the values of the k_i corresponding to the block containing α . Consider first the case $k_i = 1$, in this case $B_i = \{0, 1\}$ and $X_i(\alpha) = \emptyset$. We have $p(B_i) = B_{\pi(i)} = \{0, 1\}$ and $X_1(p(\alpha)) = \emptyset$. Consider the case $k_i > 1$. As ψ' is a bijection between strategy profiles and there is a bijection between $X_0(\alpha)$ and $X_0(p(\alpha))$ we have $\psi(X_1(\alpha)) \subseteq X_1(p(\alpha)) \cup X_2(p(\alpha)) \cup X_3(p(\alpha))$. If exists $a' \in X_1(\alpha)$ such that $\psi'(a') \in X_2(p(\alpha)) \cup X_3(p(\alpha))$, exists $b \in X_1(p(\alpha))$ such that $b \prec_{p(\alpha)} \psi'(a')$. Therefore $\psi'^{-1}(b) \prec_\alpha a'$, but this is impossible because $\psi'^{-1}(b)$ cannot be an element of $X_0(\alpha)$. Therefore $\psi(X_1(\alpha)) \subseteq X_1(p(\alpha))$. As $\psi(X_1(\alpha))$ and $X_1(p(\alpha))$ have the same number of elements we conclude $\psi(X_1(\alpha)) = X_1(p(\alpha))$.

It holds $\psi'(X_2(\alpha)) = X_2(p(\alpha))$ for any player α . We have $\psi'(X_2(\alpha)) \subseteq X_2(p(\alpha)) \cup X_3(p(\alpha))$ and by similar arguments we conclude the equality.

It holds $\psi'(X_3(\alpha)) = X_3(p(\alpha))$ for any player α . As ψ' is a bijection and $\psi'(X_2(\alpha)) \subseteq X_2(p(\alpha))$, this forces equality.

We have that f_α is the identity, for any player α . Note that α belongs to a block of players B_i , $i = \text{block}(\alpha)$ having A^i as the corresponding alphabet. We consider two cases depending on the size of A^i . First, we consider the case such that A^i has just one element. In this case $\text{good}(A^i) = \{1\}$. As $\psi(\text{good}(A^i)) =$

$\text{good}(A^{\pi(i)}) = \{1\}$ this forces to f_α to be the identity. Consider the case where A^i contains most than one element. Suppose that α occupies the position j in B_i and consider the profile $a' = (b_{-i}, 0^{j-1}10^{k_i-j})$ belonging to $X_2(\alpha)$ as $\psi(a')$ belongs to $X_2(p(\alpha))$ we need a factorization $\psi(a') = (\psi(b_{-i}), 0^{\varphi_i(j)-1}10^{k_i-\varphi_i(j)})$ and this forces to f_α to be the identity.

Given a strategy profile a in Γ_1 it holds $\psi'(\text{binify}(a)) = \text{binify}(\psi(a))$. Note that $A_i = \{1, \dots, k_i\}$ and for $j \in A_i$ we have $\text{binify}(j) = 0^{j-1}10^{k_i-j} \in A^i$. As $p(B_i) = B_{\pi(i)}$, we have $\psi(\text{binify}(j)) = 0^{\varphi_i(j)-1}10^{k_i-\varphi_i(j)} \in A^{\pi(i)}$ and we conclude the result.

Given two profiles a, a' and a player i in Γ_1 and a player α in Γ'_1 such that $\text{block}(\alpha) = i$ it holds that $a \prec_i a'$ iff $\text{binary}(a) \prec_\alpha \text{binary}(a')$. This happens because we have equalities like $u_\alpha(\text{binary}(a)) = u_i(a) + 3$. The same property holds for Γ_2 and Γ'_2 .

The mapping $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ is a weak morphism between Γ_1 and Γ_2 . Suppose that in Γ_1 we have $a \prec_i a'$, let in Γ'_1 a player α such that $\text{block}(\alpha) = i$, then it holds $\text{binify}(a) \prec_\alpha \text{binify}(a')$. As ψ' is a weak morphism $\psi'(\text{binify}(a)) \prec_{p(\alpha)} \psi'(\text{binify}(a'))$ and therefore, changing ψ' into ψ , $\text{binify}(\psi(a)) \prec_{p(\alpha)} \text{binify}(\psi(a'))$. Then it holds $\psi(a) \prec_{\text{block}(\pi(\alpha))} \psi(a')$, as $\text{block}(\pi(\alpha)) = \pi(\text{block}(\alpha)) = \pi(i)$ we finally obtain $\psi(a) \prec_{\pi(i)} \psi(a')$. \square

Claim 11. Let $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ be a mapping between binary action games Γ_1 and Γ_2 . Let $\psi' = (\pi, f_1, \dots, f_n)$ be a mapping between $\text{FLIPW}(\Gamma_1)$ and $\text{FLIPW}(\Gamma_2)$ such that, for $1 \leq i \leq n$, $f_i(a_i b_i) = \varphi_i(a_i) b_i$. Taking the mapping $\mu = (\pi, id_1, \dots, id_n)$, for any $a' = a \uparrow b$ it holds $\psi'(a') = \psi(a) \uparrow \mu(b)$. Moreover $\text{flip}(\psi'(a')) = \psi(\text{flip}(a'))$.

Proof. Given $a' = (a_1 b_1, \dots, a_n b_n) = a \uparrow b$ with $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$. In all the mappings ψ, μ and ψ' the bijection function π maps player i into player $\pi(i)$. We have $\psi(a) = (\hat{a}_1, \dots, \hat{a}_n)$ with $\hat{a}_{\pi(i)} = \varphi(a_i)$ for all i . We have $\mu(b) = (\hat{b}_1, \dots, \hat{b}_n)$ with $\hat{b}_{\pi(i)} = b_i$ for all i . Moreover $\psi'(a') = (\hat{a}_1 \hat{b}_1, \dots, \hat{a}_n \hat{b}_n)$ with $\hat{a}_i \hat{b}_i = \varphi_i(a_i) b_i$ for all i . Therefore $\psi'(a') = (\hat{a}_1, \dots, \hat{b}_n) \uparrow (\hat{b}_1, \dots, \hat{b}_n) = \psi(a) \uparrow \mu(b)$.

Let us consider the behaviour of the flips. Given $a' = a \uparrow b$, note that $\psi(\text{flip}(a')) = \text{flip}(\psi'(a'))$ is equivalent to $\psi(\text{flip}(a \uparrow b)) = \text{flip}(\psi(a) \uparrow \mu(b))$. Component wise is equatily corresponds to $\varphi_i(\text{flip}(a_i b_i)) = \text{flip}(\varphi_i(a_i) b_i)$. As flip, as a boolean function is $\text{flip}(xy) = x\bar{y} + \bar{x}y$, the preceding equality is $\varphi_i(a_i \bar{b}_i + \bar{a}_i b_i) = \varphi_i(a_i) \bar{b}_i + \varphi_i(\bar{a}_i) b_i$. As φ_i is a permutation on $\{0, 1\}$, the only possibilities are $\varphi(a_i) = a_i$ or $\varphi(a_i) = \bar{a}_i$. The equation trivially holds for identity. When φ_i is a negation it is enough to check that $\neg(a_i \bar{b}_i + \bar{a}_i b_i) = \bar{a}_i \bar{b}_i + a_i b_i$. \square

Claim 12. Let $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ be a weak isomorphism between two binary action games Γ_1 and Γ_2 . The mapping ψ' between $\Gamma'_1 = \text{FLIPW}(\Gamma_1)$ and $\Gamma'_2 = \text{FLIPW}(\Gamma_2)$ defined by $\psi' = (\pi, f_1, \dots, f_n)$ where, for $1 \leq i \leq n$, $f_i(a_i b_i) = \varphi_i(a_i) b_i$ is a strong isomorphism.

Proof. We have to prove $u'_i(a') = u'_{\pi(i)}(\psi'(a'))$. The utility can have 6 different values. Assume that exists a' such that $u_i(a') = 6$. Factorizing $a' = a \uparrow b$, this is equivalent to $u_i(\text{flip}(a')) < u_i(a)$. It holds $u_{\pi(i)}(\psi(\text{flip}(a'))) < u_{\pi(i)}(\psi(a))$ because ψ is a weak isomorphism. From Claim 11, $\psi(\text{flip}(a')) = \text{flip}(\psi'(a'))$ and $\psi'(a') = \psi(a) \uparrow \mu(b)$. Therefore, as $u_{\pi(i)}(\text{flip}(\psi'(a'))) < u_{\pi(i)}(\psi(a))$, we conclude $u'_{\pi(i)}(\psi'(a')) = 6$. Other utility values are similar. \square

Claim 13. Let $\psi' = (\pi, f_1, \dots, f_n)$ be a strong isomorphism between $\text{FLIPW}(\Gamma_1)$ and $\text{FLIPW}(\Gamma_2)$. The player's bijections verify $f_i(a_i b_i) = \varphi_i(a_i) b_i$ for some permutation φ_i on $\{0, 1\}$. Moreover $\psi = (\pi, \varphi_1, \dots, \varphi_n)$ is a weak isomorphism between Γ_1 and Γ_2 .

Proof. Consider the sets $\text{Zero}(i) = \{a' | a' = (a'_{-i}, a_i 0)\}$ and $\text{One}(i) = \{a' | a' = (a'_{-i}, a_i 1)\}$. Note that $a' \in \text{Zero}(i)$ iff $u'_i(a') \in \{0, 1, 2\}$ and $a' \in \text{One}(i)$ iff $u'_i(a') \in \{3, 4, 5\}$, moreover the following holds

- For $n > 0$, every set $\text{Zero}(i)$ and $\text{One}(i)$ contain $4 \cdot 2^{2n-1} > 0$ elements each one.
- Are disjoint, $\text{Zero}(i) \cap \text{One}(i) = \emptyset$ and $\text{Zero}(i) \cup \text{One}(i) = A'_1$
- It holds $\psi'(\text{Zero}(i)) = \text{Zero}(\pi(i))$. If this is false, $\psi'(\text{Zero}(i)) \cap \text{One}(\pi(i)) \neq \emptyset$ and therefore exists $a' \in \text{Zero}(i)$ such that $u'_i(a') \in \{0, 1, 2\}$ but utilities $u'_{\pi(i)}(\psi'(a')) \in \{3, 4, 5\}$. This cannot happen because ψ' is a strong isomorphism.

- Similarly $\psi'(\text{One}(i)) = \text{One}(\pi(i))$.
- The player's bijections verify $f_i(a_i b_i) = \varphi_i(a_i) b_i$ for some permutation φ_i on $\{0, 1\}$. This is just another way to write $\psi'(\text{Zero}(i)) = \text{Zero}(\pi(i))$ and $\psi'(\text{One}(i)) = \text{One}(\pi(i))$.

Let us prove that ψ is a weak isomorphism. Assume that in Γ_1 exists a and \hat{a} such that $u_i(a) < u_i(\hat{a})$, we have to prove $u_{\pi(i)}(\psi(a)) < u_{\pi(i)}(\psi(\hat{a}))$. Given the inequality $u_i(a) < u_i(\hat{a})$, let b be the flipper such that $\text{flip}(a \uparrow b) = \hat{a}$. Assume $b_i = 0$ (case $b_i = 1$ is similar) then in $\text{FLIPW}(\Gamma_1)$ it holds $u'_i(a \uparrow b) = 0$. As ψ' is a strong isomorphism, it holds $u'_i(a \uparrow b) = u'_{\pi(i)}(\psi'(a \uparrow b)) = 0$, therefore $u_{\pi(i)}(\text{flip}(\psi'(a \uparrow b))) > u_{\pi(i)}(\text{driver}(\psi'(a \uparrow b)))$.

As $f_i(a_i b_i) = \varphi_i(a_i) b_i$ it holds (by Claim 11) $\text{flip}(\psi'(a \uparrow b)) = \psi(\text{flip}(a \uparrow b)) = \psi(\hat{a})$. Moreover as $\psi'(a \uparrow b) = \psi(a) \uparrow \mu(b)$ it holds $\text{driver}(\psi'(a \uparrow b)) = \psi(a)$. Finally we get $u_{\pi(i)}(\psi(\hat{a})) > u_{\pi(i)}(\psi(a))$. Other cases are similar. \square