EPiC Series in Computing

Volume 100, 2024, Pages 341–359

Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning

A Simple Token Game and its Logic

Chris Fermüller[1], Robert Freiman[1], and Timo Lang[2]

[1] TU Wien, Wien, Austria
{chrisf,robert}@logic.at
[2] University College London, London, U.K.
timo.lang@ucl.ac.uk

Abstract

We introduce a simple game of resource-conscious reasoning. In this two-player game, players **P** and **O** place tokens of positive and negative polarity onto a game board according to certain rules. **P** wins if she manages to match every negative token with a corresponding positive token.

We study this token game using methods from computational complexity and proof theory. Specifically, we show complexity results for various fragments of the game, including **PSPACE**-completeness of a finite restriction and undecidability of the full game featuring non-terminating plays. Moreover, we show that the finitary version of the game is axiomatisable and can be embedded into exponential-free linear logic. The full game is shown to satisfy the exponential rules of linear logic, but is not fully captured by it. Finally, we show determinacy of the game, that is the existence of a winning strategy for one of the players.

1 Introduction

Games and logic have a long-standing and fruitful relationship. We mention just a few products of this happy union and refer to [HV19] for a more detailed account. Ehrenfeucht-Fraïssé games or model comparison games, which characterize elementary equivalence, go back to [Ehr61]. In the 1960s and 70s Lorenzen and Lorenz [Lor60, LL78, Lor01] used a dialogue game to provide pragmatic foundations for constructive thinking. A little later, Hintikka [Hin74] characterized truth in a model by a game for classical first-order logic. Also in the 1970s, Giles [Gil74, Gil77] characterized infinite-valued Łukasiewicz logics by a game involving bets on the outcomes of dispersive elementary experiments. More recently, and closer to our current concern, formal games have been used to model resource-conscious reasoning, leading to the rich field of game semantics for substructural logics, in particular linear logic, see e.g. [Bla92, AJ94]. We also draw inspiration from Japaridze's research program on 'computability logic' [Jap03], which explores a game model for many different kinds of interactions between machine and environment.

Like Japaridze, we follow a 'games first' approach here: Instead of coming up with a logic that exhibits aspects of resource-conscious reasoning, we devise a game that clearly has these features, and then study its logic afterwards. Our guiding principle is simplicity: What is a minimalistic model of interaction that enforces resource-conscious reasoning and supports

the basic moves of *choice* and *repetition*? Although simple, the *token game* we present here gives rise to an intricate interaction. This is corroborated through its high computational complexity: The repetition-free fragment of the game is **PSPACE**-complete (Theorem 27), while the full game is already undecidable (Theorem 31). We establish a relationship between the game and logic by showing that the rules of linear logic with mix are sound for the token game (Theorem 35) and complete for its repetition-free fragment (Prop. 16). Finally, we derive determinacy of the game (Theorem 37) from the Borel determinacy theorem.

2 The token game

In this section we introduce the *token game*. We start from an intuitive description before moving on to the formal study.

2.1 An informal description

Two players, **P** and **O**, place polarized *tokens* $a^+, a^-, b^+, b^-, \ldots$ and *tasks* (defined below) on a board. At any point **P** can take a matching pair a^+, a^- and remove it from the board. She wins by ensuring that no negative token remains unmatched indefinitely.[1] As an example, **P** will win if the board is $\{a^+, a^+, a^-, b^+\}$ as she can match a^- with one of the a^+'s, but she loses in $\{a^+, b^+, a^-, b^-, b^-\}$ since one of the b^-'s cannot be matched. Apart from polarized tokens, the game board may contain *choice tasks* of the form

$$\{s_1, \ldots, s_n\} \oplus_{\mathbf{P}} \{t_1, \ldots, t_m\} \qquad \text{and} \qquad \{s_1, \ldots, s_n\} \oplus_{\mathbf{O}} \{t_1, \ldots, t_m\}$$

that **P** (respectively **O**) can replace by either all of the s_i's or all of the t_j's, each s_i and t_j again being a task or a polarized token. Moreover, we have *repetition tasks*

$$[\![s_1, \ldots, s_n]\!]_{\mathbf{P}} \qquad \text{and} \qquad [\![s_1, \ldots, s_n]\!]_{\mathbf{O}}$$

that allow **P** (respectively **O**) to add all the s_i's simultaneously to the game board. The repetition task itself remains on the board, and so adding the s_i's can be repeated.

As the players' actions—matching, choice and repeat—do not interfere with each other, the order in which they move can essentially be left unspecified; we just require that every player gets to move eventually. This condition trivially holds in a turn-based game model, but it can also be realized in an asynchronous game satisfying a fairness condition on plays.

More importantly however, we stipulate the following principle that rules out certain deadlocks: **P** *does not lose as long as there is an* $\oplus_{\mathbf{O}}$*-task on the board.* Effectively, this allows **P** to wait for **O** to move in $\oplus_{\mathbf{O}}$.[2]

This concludes the informal description of the game. The reader familiar with linear logic will of course see the connection from choice and repetition tasks to the additive and exponential connectives of linear logic. Our chief interest will be whether **P** has a winning strategy in a given game. Below are some examples of games and their analysis.[3]

- $a^- \oplus_{\mathbf{O}} b^-$, $a^+ \oplus_{\mathbf{P}} b^+$
  P wins by waiting for **O**'s choice in his task, and then answering appropriately using her own choice task. If **O** never makes a choice, **P** wins by definition.

[1] In finite plays this is equivalent to saying that the board is eventually free of negative tokens; in infinite plays the board may never be free of negative tokens, but **P** must at least make sure that every single negative token eventually disappears.

[2] At the end of Section 2.2 we give an intuitive motivation for this particular choice.

[3] We often identify singleton (multi)sets with their elements, thus writing $a^+ \oplus_{\mathbf{O}} b^-$ for $\{a^+\} \oplus_{\mathbf{O}} \{b^-\}$.

- d^- , $\{(a^- \oplus_{\mathbf{O}} c^-), d^+\} \oplus_{\mathbf{P}} \{(b^- \oplus_{\mathbf{O}} c^-), d^+\}$, $(a^+ \oplus_{\mathbf{O}} b^+) \oplus_{\mathbf{P}} c^+$
  P must move as otherwise d^- remains unmatched. If for example she chooses $(a^- \oplus_{\mathbf{O}} c^-), d^+$ in the first choices task, **O** can make a further choice to place a^- on the board. To match this a^- **P** will then choose $a^+ \oplus_{\mathbf{O}} b^+$ in the third choice task, but her attempt is thwarted by **O** moving into b^+. The other cases are similar, and we conclude that **P** has no winning strategy in this game.

- $[\![a^-, a^-]\!]_{\mathbf{O}}$, $[\![a^+]\!]_{\mathbf{P}}$
  P wins this game: For every number of a^-'s that **O** puts on the board, she adds the same number of a^+'s and does the matching. Even if **O** never stops, every occurrence of a^- on the game board will eventually be matched.

- $[\![a^-, b^+]\!]_{\mathbf{P}}$, $[\![a^+, b^-, b^-]\!]_{\mathbf{O}}$
  Here **O** wins by unpacking the repeat exactly once, thus placing a^+, b^-, b^- on the game board. Now **P** needs to unpack her repeat task twice to match the two b^-'s. This results in two a^-'s on the board, one of which will remain unmatched.

- $a^- \oplus_{\mathbf{P}} a^-$
  P wins this game by simply not moving.

2.2 A formal account

In this section, we formalize the game introduced in Section 2.1. The purpose of this is to eliminate any potential misunderstandings about the informal presentation and to provide a solid basis for formal reasoning about the game. However, the reader may also skip this and the subsequent Section 2.3 on first reading, taking only note of Prop. 10 and 11, and then refer back as needed.

We first formalize a *liberal* version, where players move independently (asynchronously). We will later see that, if one cares only about the existence of winning strategies for **P**, the game is equivalent to a *strict* version, with **P** as the scheduler of moves. The strict game is easily analyzed using a calculus, and we often work with it in subsequent sections. The reason that we nevertheless start with the liberal version is that, in our opinion, it is more intuitive as a game and closer to applications. Moreover, by showing the equivalence of the liberal and strict semantics the scope of all subsequent results is extended.

We assume an countably infinite set $a, b, c, \ldots$ of *tokens*. *Polarized tokens* are of the form a^+ (positive) or a^- (negative). Tasks and Multitasks are defined via mutual recursion:

Definition 1 (tasks and multitasks)**.** Multitasks *(denoted by uppercase letters $S, T, \ldots$) are finite families of tasks.* Tasks *(denoted by lowercase letters $s, t, \ldots$) are of the following form:*

$$s \quad ::= \quad a^+ \mid a^- \mid S \oplus_{\mathbf{P}} T \mid S \oplus_{\mathbf{O}} T \mid [\![S]\!]_{\mathbf{P}} \mid [\![S]\!]_{\mathbf{O}}$$

For multitasks we use multiset notation, e.g. $S = \{a^+, a^+, (\{b^-, c^+\} \oplus_{\mathbf{P}} \emptyset), [\![b^+, a^-]\!]_{\mathbf{O}}\}$. Multitasks are allowed to be empty. Since multitasks are families rather than multisets, we retain the possibility of distinguishing between different occurrences of the same task within a multitask.[4]

We write S, T for the disjoint union of multitasks S and T, and let $S, s := S, \{s\}$. Consequently, the line between tasks and multitask is somewhat blurred in the notation. Keeping in

[4]Families are formalised as functions $X : I \to \mathrm{dom}(X)$, where $|\{i \in I \mid f(i) = a\}|$ denotes the multiplicity of a in X. Different occurrences of a can be distinguished, as they are associated to different indices i.

mind the following should prevent any confusion: The arguments of $\oplus_{\mathbf{O}}$, $\oplus_{\mathbf{P}}$, $[\![]\!]_{\mathbf{P}}$ and $[\![]\!]_{\mathbf{O}}$ are *always* multitasks, and the elements of multitasks are *always* tasks.

Definition 2 (moves)**.** *The relations $\to_{\mathbf{P}}, \to_{\mathbf{O}}$ denote single moves by players* **P** *and* **O** *and are defined through the following clauses ($i = 1, 2$):*

$$S, a^-, a^+ \to_{\mathbf{P}} S \quad \textit{(match)}$$

$$S, T_1 \oplus_{\mathbf{P}} T_2 \to_{\mathbf{P}} S, T_i \quad (\mathbf{P}\textit{-choice}) \qquad S, T_1 \oplus_{\mathbf{O}} T_2 \to_{\mathbf{O}} S, T_i \quad (\mathbf{O}\textit{-choice})$$

$$S, [\![T]\!]_{\mathbf{P}} \to_{\mathbf{P}} S, [\![T]\!]_{\mathbf{P}}, T \quad (\mathbf{P}\textit{-repeat}) \qquad S, [\![T]\!]_{\mathbf{O}} \to_{\mathbf{O}} S, [\![T]\!]_{\mathbf{O}}, T \quad (\mathbf{O}\textit{-repeat})$$

Their reflexive transitive closures are denoted $\to^*_{\mathbf{P}}$ and $\to^*_{\mathbf{O}}$.

Here are some examples of moves:

$$\{s, t, a^-, a^-, a^+\} \to_{\mathbf{P}} \{s, t, a^-\} \qquad \{s, t, \{s_1, s_2\} \oplus_{\mathbf{O}} T\} \to_{\mathbf{O}} \{s, t, s_1, s_2\}$$
$$\{s, t, [\![s_1, s_2]\!]_{\mathbf{P}}\} \to_{\mathbf{P}} \{s, t, [\![s_1, s_2]\!]_{\mathbf{P}}, s_1, s_2\}$$

Definition 3. *$S^{\mathbf{P}}$ denotes the submultitask of S where* **P** *can move: Tokens and tasks with principal connective $\oplus_{\mathbf{P}}$ or $[\![]\!]_{\mathbf{P}}$. $S^{\mathbf{O}}$, its complement, contains tasks with principal connective $\oplus_{\mathbf{O}}$ or $[\![]\!]_{\mathbf{O}}$.*

Every **P**-move in S is essentially a move in $S^{\mathbf{P}}$, and likewise for **O**. Formally: Whenever $S \to^*_{\mathbf{P}} T$, there exists U such that $S^{\mathbf{P}} \to^*_{\mathbf{P}} U$ and $T = U, S^{\mathbf{O}}$ (and similar for **O**).

We consider a game where **P** and **O** may move simultaneously. A round in the game consists of both players looking at the history of the play so far and deciding on a (possibly empty) finite sequence of moves they will carry out independently of the other player's concurrent moves. Once both players have moved, they inspect the game state again (now seeing the effect of the other player's moves) and decide on the next move, and so on. This notion of asynchronous play is relatively liberal, and in particular it allows players simply to wait for the other to move.

For purely technical reasons, in Definition 4 below the players will take turns with **O** taking the lead, instead of playing concurrently as described above. When we define strategies we will however enforce that **P** cannot take **O**'s preceding move into account. This has the desired effect that we can think of a pair of subsequent moves by **O** and **P** as taking place simultaneously.

Definition 4 (play)**.** *A* play *on S is an infinite sequence $(S_1, S_2, S_3, \ldots)$ such that*

$$S = S_1 \to^*_{\mathbf{O}} S_2 \to^*_{\mathbf{P}} S_3 \to^*_{\mathbf{O}} \cdots$$

An initial segment $(S_1, \ldots, S_k)$ of a play is called a position*, and we say that it* ends in S_k*. A play is said to* terminate in T *if for some $k > 0$, $S_i = T$ for all $i \geq k$. The set of all plays on S is denoted $\mathcal{G}^\infty(S)$. A position is* realized *in a set of plays if it appears as an initial segment of some play.*

Note that if neither **P** nor **O** has a move in S_k then the only play realizing the position $(S_1, \ldots, S_k)$ is $(S_1, \ldots, S_k, S_k, S_k, \ldots)$, which according to Def. 4 terminates in S_k. In this way the definition also captures finite games.

Definition 5 (strategy)**.** *An* **O***-strategy in S is a nonempty subset $\mathcal{H} \subseteq \mathcal{G}^\infty(S)$ with the following property: If $(S_1, \ldots, S_k)$ is realized in $\mathcal{H}$ and k is even, then for all $S_k \to^*_{\mathbf{P}} T$ there exists a unique $T \to^*_{\mathbf{O}} U$ such that $(S_1, \ldots, S_k, T, U)$ is realized in $\mathcal{H}$.*

A **P***-strategy in S is a nonempty subset $\mathcal{H} \subseteq \mathcal{G}^\infty(S)$ with the following property: If $(S_1, \ldots, S_k)$ is realized in $\mathcal{H}$ and k is odd, then there exists a unique $S^{\mathbf{P}}_k \to_{\mathbf{P}} U$ such that for all $S^{\mathbf{O}}_k \to^*_{\mathbf{O}} T$, $(S_1, \ldots, S^{\mathbf{P}}_k \cup S^{\mathbf{O}}_k, S^{\mathbf{P}}_k \cup T, U \cup T)$ is realized in $\mathcal{H}$.*

As mentioned earlier, the somewhat complicated definition of a **P**-strategy guarantees that her moves do not depend on the immediately preceding move of **O**.

Lemma 6. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be* **P***- and* **O***-strategies in $\mathcal{G}^\infty(S)$ respectively. Then $\mathcal{H}_1 \cap \mathcal{H}_2$ contains a single play.*

Proof. Straightforward. □

Definition 7 (guarantees and winning conditions)**.** *Let $h \in \mathcal{G}^\infty(S)$.*

1. *h is* **O**-responsive *if for all $\sigma \subseteq h$ containing an occurrence of $S \oplus_{\mathbf{O}} T$, there exists a $\sigma \subseteq \sigma' \subseteq h$ where* **O** *has moved in that occurrence.*
2. *h is* **P**-responsive *if for every $\sigma \subseteq h$ containing an occurrence of a^-, there exist an $\sigma \subseteq \sigma' \subseteq h$ where that occurrence has been matched.*
3. *h is* **O**-winning *if it is* **O***-responsive but not* **P***-responsive; it is* **P***-winning otherwise.*

Put differently, being **P**-winning means satisfying the implication 'if **O**-responsive, then **P**-responsive'.

Definition 8 (winning strategies)**.** *A strategy $\mathcal{H}$ for* **P** (**O**) *is winning if it contains only* **P***-winning* (**O***-winning*) *plays. We write $\models_{\mathbf{P}} S$ ($\models_{\mathbf{O}} S$) if* **P** (**O**) *has a winning strategy in S.*

The asymmetric nature of the token game, as apparent in Definition 7, warrants further explanation. We envision a scenario where **P** is a system providing a resource-sensitive service, as exemplified by condition **P**-responsiveness. **O** is the user of that system. A task $S \oplus_{\mathbf{O}} T$ can be understood as a *prompt* requiring input from **O**. In practice, there might be various reasons why such input does not arrive (e.g., the user drops out of the connection), and in that case, it is unfair to require **P** to deliver its service. Formally, we therefore declare plays failing **O**-responsiveness to be won by **P**.

There is no similar requirement for the task $S \oplus_{\mathbf{P}} T$, which in our scenario corresponds to a **P**-computation waiting to be continued. This is because we consider **P**'s basic function to guarantee **P**-responsiveness; if this can be achieved without all computations terminating, then this is considered acceptable. Note also that since we are working with multitasks, an unanswered $\oplus_{\mathbf{P}}$ corresponds to only one computation of the system being halted, and not the whole system coming to a halt.

Of course, different scenarios might call for different connectives and guarantees. We believe that the combination of operators and guarantees as given here is interesting in its own right, but also note that some alternative connectives are definable in the present setup:

defined connective	interpretation	definition
$S \circledast_{\mathbf{P}} T$	forced **P**-choice (Def. 23)	$a^-, \{S, a^+\} \oplus_{\mathbf{P}} \{T, a^+\}$ (a fresh)
$[\![S]\!]^*_{\mathbf{P}}$	forced **P**-repetition	$[\![a^-]\!]_{\mathbf{O}}, [\![S, a^+]\!]_{\mathbf{P}}$ (a fresh)
$S \oplus^l_{\mathbf{O}} T$	liberal **O**-choice	$(S \oplus_{\mathbf{O}} T) \oplus_{\mathbf{O}} \{\}$

2.3 Some basic properties of the token game

We begin with two simple observations, whose proof can be found in the appendix. First, **P** never loses by letting **O** move before her. Second, if **P** has winning strategies in two games then she can play them in parallel.

Lemma 9. *If $\models_{\mathbf{P}} S$ and $S \to^*_{\mathbf{O}} T$, then $\models_{\mathbf{P}} T$.*

Proposition 10. *If* $\models_{\mathbf{P}} S$ *and* $\models_{\mathbf{P}} T$*, then* $\models_{\mathbf{P}} S, T$*.*

Proposition 11.

1. *If* $\models_{\mathbf{P}} S, T_1$ *or* $\models_{\mathbf{P}} S, T_2$*, then* $\models_{\mathbf{P}} S, (T_1 \oplus_{\mathbf{P}} T_2)$*.*
2. $\models_{\mathbf{P}} S, (T_1 \oplus_{\mathbf{O}} T_2)$ *iff* $(\models_{\mathbf{P}} S, T_1$ *and* $\models_{\mathbf{P}} S, T_2)$

Proof. For (1), let **P** have a winning strategy in, say, S, T_1. Then she can win in $S, T_1 \oplus_{\mathbf{P}} T_2$ as follows. First, she makes the move $S^{\mathbf{P}}, T_1 \oplus_{\mathbf{P}} T_2 \to_{\mathbf{P}} S, T_1$, while **O** moves $S^{\mathbf{O}} \to^*_{\mathbf{O}} T$. The resulting multitask is $S^{\mathbf{P}} T_1 \cup T$. This multitask is obtainable from S, T_1 by **O**-moves. Since **P** has a winning strategy for the latter multitask, she can win in the former by Lem. 9.

The left-to-right direction of (2) is a direct application of Lemma 9. For the other direction, note that **P** can simply wait until **O** makes his move in $T_1 \oplus_{\mathbf{O}} T_2$ (if **O** never makes a move in this task, he loses). Then at some point, **P** finds a multitask T where $S, T_i \to^*_{\mathbf{O}} T$ for some $i = 1, 2$. By the assumption and Lem. 9, **P** has a winning strategy for this multitask. □

3 The elementary token game

We begin our in-depth examination of the token game by focussing on its elementary fragment.

Definition 12. *A multitask is* elementary *if it contains no repeat tasks.*

If S is elementary, then every play in $\mathcal{G}^\infty(S)$ terminates in some multitask T.

3.1 Axiomatisability

With the groundwork laid in Section 2.3, it is now straightforward to provide a calculus **WS** that derives exactly those elementary multitasks where **P** has a winning strategy. **WS** has three rules that correspond to moves in the game:

$$(match)\ \frac{S}{S, a^-, a^+} \qquad (\oplus_{\mathbf{P}})_{i=1,2}\ \frac{S, T_i}{S, T_1 \oplus_{\mathbf{P}} T_2} \qquad (\oplus_{\mathbf{O}})\ \frac{S, T_1 \qquad S, T_2}{S, T_1 \oplus_{\mathbf{O}} T_2}$$

Initial multitasks are elementary multitasks without $\oplus_{\mathbf{O}}$'s and without negative tokens. Intuitively, initial multitasks are multitasks where **O** cannot move and **P** wins without continuing the game. A *proof* in **WS** of a multitask S is a tree rooted in S, leaves are initial multitasks, and each node is obtained from its parent by applying the above rules bottom-up.

Theorem 13. *Let* S *be elementary. Then* $\models_{\mathbf{P}} S$ *iff there is a proof of* S *in* **WS***.*

Proof. $\Leftarrow$: It suffices to show that the rules and axioms of **WS** are sound with respect to $\models_{\mathbf{P}}$, i.e. they preserve **P**-winnability. Initial multitasks are sound because **P** wins by doing nothing. The soundness of $(\oplus_{\mathbf{P}})$ and $(\oplus_{\mathbf{O}})$ has been established in Prop. 11.1 and 11.2, respectively. Finally, soundness of (*match*) is obvious.

$\Rightarrow$: By induction on the total number of connectives and tokens occurring (nested) in S. Suppose $\models_{\mathbf{P}} S$. If S contains $\oplus_{\mathbf{O}}$s, i.e., $S = S', T_1 \oplus_{\mathbf{O}} T_2$ then **P** has winning strategies for S', T_1 and S', T_2, by Prop. 11.2. The inductive hypothesis gives us **WS**-proofs for S', T_1 and S', T_2, which can be combined into a proof of S by applying $(\oplus_{\mathbf{O}})$.

We can therefore assume that **O** cannot move in S, i.e. $S = S^{\mathbf{P}}$. If S wins by not moving, then S is initial multitask and therefore has a trivial proof in **WS**. Otherwise, according to her

MALL $+ (mix)$:

$$\frac{}{\vdash a, a^{\perp}} \qquad \frac{}{\vdash} \qquad \frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}\ (mix) \qquad \frac{\vdash \Gamma}{\vdash \Gamma, 1}\ (1)$$

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ (\otimes) \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ (\parr) \qquad \frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus A_2}\ (\oplus) \qquad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B}\ (\&)$$

LL $+ (mix)$: All rules above plus

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ (w?) \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ (c?) \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ (pr!) \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ (dr?)$$

Figure 1: **LL** $+ (mix)$ and **MALL** $+ (mix)$.

winning strategy, **P** makes a nontrivial sequence of moves $S \to_{\mathbf{P}} S' \to_{\mathbf{P}} \cdots$. Obviously, **P** has a winning strategy in S', and so by induction hypothesis S' has a **WS**-proof. We can extend this to a **WS**-proof of S by adding an inference $(match)$ or $(\oplus_{\mathbf{P}})$ below that mirrors the move $S \to_{\mathbf{P}} S'$. □

From the completeness of **WS** it follows immediately that the following strict version of the game is equivalent to the liberal one: Instead of moving asynchronously, **P** now acts as a scheduler, that is, in every round she decides in which tokens to match or which $\oplus_{\mathbf{P}}$ to resolve. Additionally, she may point at a $\oplus_{\mathbf{O}}$ that **O** then has to resolve. **P** wins if she manages to arrive at an initial multitask this way (note that this implies that all $\oplus_{\mathbf{O}}$'s have been resolved).

Proposition 14. *The following rules are admissible in* **WS***:*

$$\frac{S}{S, a^{+}}\ (weak_{+}) \qquad \frac{S}{S, T_1 \oplus_{\mathbf{P}} T_2}\ (weak_{\oplus_{\mathbf{P}}}) \qquad \frac{S \quad T}{S, T}\ (mix)$$

Proof. Obviously, the rules $(weak_{+})$ and $(weak_{\oplus_{\mathbf{P}}})$ preserve **P**-winnability, as **P** can simply ignore the additional positive token or **P**-choice. Admissibility of (mix) is Prop. 10. □

3.2 A connection to linear logic

Having linked the elementary game to a relatively conventional calculus, we can also draw a comparison to linear logic. This will be achieved by translating elementary multitasks to formulas of linear logic, and then showing that **P**-winnability of a multitask is equivalent to provability of its translation in linear logic. The most natural translation goes into the variant of linear logic called **MALL** $+ (mix)$, that is, exponential-free linear logic with the *mix rule*[5], as presented in Figure 1. Translations in other variants of linear logic are also possible.

Definition 15 (from elementary multitasks to formulas of linear logic)**.**

$$\tau(\{s_1, \ldots, s_n\}) := \tau(s_1) \parr \ldots \parr \tau(s_n) \qquad \tau(a^{-}) := a^{\perp}$$
$$\tau(\emptyset) := 1 \qquad \tau(S \oplus_{\mathbf{P}} T) := (\tau(S) \oplus \tau(T)) \oplus 1$$
$$\tau(a^{+}) := a \oplus 1 \qquad \tau(S \oplus_{\mathbf{O}} T) := \tau(S) \& \tau(T)$$

[5]Following [FR94], we include $\vdash$ (the empty sequent) in **MALL** $+ (mix)$, which can be seen as a 0-ary version of (mix). We also omit the logical constant $\top$.

The $\oplus 1$ in $\tau(a^+)$ and $\tau(S \oplus_{\mathbf{P}} T)$ indicates that **P** can choose to not match a positive token, or to not make a **P**-choice. Note that this translation does not use the tensor $\otimes$ of linear logic.

Proposition 16. *For elementary* $S = \{s_1, \ldots, s_n\}$, $\models_{\mathbf{P}} S$ *iff* **MALL** $+ (mix)$ *proves* $\vdash \tau(s_1), \ldots, \tau(s_n)$.

Proof. By Thm. 13, it suffices to show that S is provable in **WS** iff $\tau(s_1), \ldots, \tau(s_n)$ is provable in **MALL** $+ (mix)$. This is established in a straightforward manner by showing how the rules and axioms of both calculi simulate each other.

From **WS** *to* **MALL** $+ (mix)$. If S is an initial multitask, then each $\tau(s_i)$ is of the form $* \oplus 1$, and therefore the whole sequent is derivable from $\vdash$ using the rules (1) and $(\oplus)$. The $(match)$-rule of **WS**, below left, is simulated by the **MALL** $+ (mix)$-derivation below right:

$$\frac{S}{S, a^-, a^+}\ (match) \qquad \frac{\vdash \tau(S) \quad \dfrac{\overline{\vdash a^\perp, a}}{\vdash a^\perp, a \oplus 1}\ (\oplus)}{\vdash \tau(S), a^\perp, a \oplus 1}\ (mix)$$

The simulation of $(\oplus_{\mathbf{P}})$ and $(\oplus_{\mathbf{O}})$ in **MALL** $+ (mix)$ is straightforward.

From **MALL** $+ (mix)$ *to* **WS**. Straightforward, using Prop. 14. □

This relation between the token game and linear logic should not be overemphasized, as the image of the τ-translation is a rather odd fragment of linear logic. It is also worth pointing out here that the game is not closed under substitution (tokens are not *variables*), and thus fails to be a logic according to at least some definitions.

Nevertheless, some simple conclusions can be drawn from the embedding, and we discuss one that relates to cut-admissibility in **MALL** $+ (mix)$. For this, call a pair s, t of elementary tasks *pseudo-dual* if **MALL** $+ (mix)$ proves $\vdash \tau(s)^\perp, \tau(t)^\perp$. This holds in particular if $\tau(t) = \tau(s)^\perp$.

Corollary 17. *If* s, t *are pseudo-dual, then the following rule is admissible in* **WS***:*

$$\frac{S, s \quad T, t}{S, T}\ (cut)$$

Proof. Assume $\models_{\mathbf{P}} S, s$ and $\models_{\mathbf{P}} T, t$ where $S = \{s_1, \ldots, s_n\}$ and $T = \{t_1, \ldots, t_m\}$. By Prop. 16 the sequents $\vdash \tau(s_1), \ldots, \tau(s_n), \tau(s)$ and $\vdash \tau(t_1), \ldots, \tau(t_m), \tau(t)$ are provable in **MALL**$+(mix)$. By provability of $\vdash \tau(s)^\perp, \tau(t)^\perp$ and cut-admissibility in **MALL** $+ (mix)$ we conclude that also $\vdash \tau(t_1), \ldots, \tau(t_m), \tau(s)^\perp$ is provable in **MALL** $+ (mix)$, and by yet another cut it follows that the same holds for $\vdash \tau(s_1), \ldots, \tau(s_n), \tau(t_1), \ldots, \tau(t_m)$. Thus $\models_{\mathbf{P}} S, T$ by Prop. 16. □

As an example, we can show that $\models_{\mathbf{P}} S, a^+$ and $\models_{\mathbf{P}} T, a^-$ implies $\models_{\mathbf{P}} S, T$. Indeed, $\models_{\mathbf{P}} T, a^-$ implies $\models_{\mathbf{P}} T, a^- \oplus_{\mathbf{O}} \emptyset$. But[6] $\tau(a^- \oplus_{\mathbf{O}} \emptyset) = a^\perp \& 1 = (a \oplus 1)^\perp = \tau(a^+)^\perp$, which implies that a^+ and $a^- \oplus_{\mathbf{O}} \emptyset$ are pseudo-dual, and hence by Cor. 17 we get $\models_{\mathbf{P}} S, T$.

3.3 Computational complexity

Definition 18 (token count)**.** *The* a-count *of an elementary (multi)task is defined as follows:*

$$a{:}S = \sum_{s \in S} a{:}s \qquad a{:}(S \oplus_{\mathbf{O}} T) = \min\{a{:}S, a{:}T\} \qquad a{:}(S \oplus_{\mathbf{P}} T) = \max\{0, a{:}S, a{:}T\}$$

$$a{:}a^+ = 1 \qquad a{:}a^- = -1 \qquad a{:}b = 0 \quad (b \notin \{a^+, a^-\})$$

[6] Note here that $1^\perp = 1$ in the presence of (mix).

If S does no contain choice connectives, then $a:S$ simply counts the number of a^+'s in S minus the number of a^-'s. More generally, we have the following:

Lemma 19. *Let S be elementary. Then $a:S$ is the minimal integer k such that* **O** *can enforce $a:T \leq k$ in every play terminating in T.*

Proof. Straightforward; the 0 appears in $\max\{0, a:S, a:T\}$ because **P** can choose to never move in $S \oplus_{\mathbf{P}} T$. □

We first observe that **O**-winnability in the elementary token game without **P**-choices is a simple optimisation task that is independent across subgames, and thus efficiently decidable.

Proposition 20. **O**-*winnability in the $\oplus_{\mathbf{O}}$-fragment is* **PTIME**.

Proof. For any elementary multitask S, $\models_{\mathbf{O}} S$ iff **O** has a strategy σ such that in every play in σ terminating in T we have $a:T < 0$ for some token a. If S is in the $\oplus_{\mathbf{O}}$-fragment, then there is only one play once **O**'s strategy σ is fixed, and so a is determined by σ. Together with Lem. 19, it follows that $\models_{\mathbf{O}} S$ iff for some token a occurring in S, $a:S < 0$.[7] A **PTIME**-algorithm for determining whether $\models_{\mathbf{O}} S$ now proceeds as follows: For every token a appearing in S, calculate $a:S$. If one of the $a:S$'s is negative, accept; Otherwise reject. □

As games in the $\oplus_{\mathbf{O}}$-fragment are finite and thus determined, we obtain immediately:

Corollary 21. **P**-*winnability in the $\oplus_{\mathbf{O}}$-fragment is* **PTIME**.

For the next proposition, we need two additional definitions.

Definition 22. *For a natural number k and a task s, $k \cdot s = \{s, \ldots, s\}$ (k times).*

Definition 23 (forced **P**-choice)**.** $S \mathbin{\textcircled{\bullet}_{\mathbf{P}}} T := \{t^-, (t^+, S) \oplus_{\mathbf{P}} (t^+, T)\}$, *where t is a fresh token.*

The multitask $S \mathbin{\textcircled{\bullet}_{\mathbf{P}}} P$ is like $S \oplus_{\mathbf{P}} T$, only that **P** loses if she does not make a choice.

Proposition 24. **P**-*winnability in the $\oplus_{\mathbf{P}}$-fragment is* **NP**-*complete.*

Proof. A strategy in the $\oplus_{\mathbf{P}}$-fragment of the token game is a sequence of consecutive choices by **P**, bounded in length by the number of $\oplus_{\mathbf{P}}$'s in the multitask. Therefore we can guess in polynomial time a strategy for **P**, and then check (also in polynomial time) whether the resulting multiset of tokens has a matching. Hence winnability in the $\oplus_{\mathbf{P}}$-fragment is in **NP**.

We show **NP**-hardness by reducing 3SAT to winnability for the $\oplus_{\mathbf{P}}$-fragment. For this, let φ be a boolean formula of the form

$$\bigwedge_{i=1}^{n} (l_i \vee m_i \vee r_i)$$

where each l_i, m_i, r_i is a literal. φ is satisfiable if and only if one can choose one literal in each clause of the conjunction (intuitively: making it true) in a consistent way, meaning that no literal and its converse are chosen.

It is this property that we now model as a multitask. For this, we identify $x \in vars(\varphi)$ with a token; we also introduce a token $\bar{x}$ that stands for $\neg x$. In particular, all literals in φ are identified with a token. Using one additional token t, define:

$$C_x := (n \cdot x^+) \mathbin{\textcircled{\bullet}_{\mathbf{P}}} (n \cdot \bar{x}^+) \qquad S_i := l_i^- \mathbin{\textcircled{\bullet}_{\mathbf{P}}} m_i^- \mathbin{\textcircled{\bullet}_{\mathbf{P}}} r_i^-$$
$$\varphi^\dagger := \{S_1, \ldots, S_n\} \cup \{C_x \mid x \in vars(\varphi)\}$$

[7] This equivalence would not hold if **P**-choices were involved: For example, **O** could win by enforcing that in every play either the a-count or the b-count for some tokens a, b is negative, but which one it is might depend on choices by **P**.

Clearly, the map $\varphi \mapsto \varphi^\dagger$ is computable in polynomial time, and $\varphi^\dagger$ is in the $\oplus_{\mathbf{P}}$-fragment. To win $\varphi^\dagger$, $\mathbf{P}$ has to make choices in all of the C_x's, and we interpret these choices as picking a variable assigment: $\mathbf{P}$'s choice of $n \cdot x^+$ makes x true, and $\mathbf{P}$'s choice of $n \cdot \bar{x}^+$ makes x false. She also has to make choices in all of the S_i's, thereby she effectively chooses a literal from each clause (l_i, m_i or r_i).

The consistency of $\mathbf{P}$'s choices is enforced as follows. If $\mathbf{P}$ chooses both a literal x and its converse $\bar{x}$, then the resulting multitask contains both tokens x^- and $\bar{x}^-$. They cannot both be matched using the positive tokens in C_x, and so $\mathbf{P}$ will inevitably lose. On the other hand, if this is not the case—say only $\bar{x}$ occurs in the multitask—then there are at most n such $\bar{x}$'s in total (one from each S_i) and so C_x contains enough positive tokens x^+ to obtain a matching. □

Sequential choices. Consider a sequence $(S_1 \oplus_{\mathbf{X}_1} T_1), (S_2 \oplus_{\mathbf{X}_2} T_2), \ldots, (S_n \oplus_{\mathbf{X}_n} T_n)$ of choice tasks by either player. We want to come up with a game that $\mathbf{P}$ can only win if the choices are made in the indicated order, that is, first $\mathbf{X}_1$ chooses between S_1 and T_1, then $\mathbf{X}_2$ chooses between S_2 and T_2 and so forth. (We do not impose anything about the order of the subsequent moves in the multitasks S_i, T_i.)

The obvious solution to this is to use nested choices. For example, the multitask $(S_1, (S_2 \oplus_{\mathbf{X}_2} T_2)) \oplus_{\mathbf{X}_1} (T_1, (S_2 \oplus_{\mathbf{X}_2} T_2))$ enforces that $\mathbf{X}_1$ chooses between S_1 and T_1 *before* $\mathbf{X}_2$ chooses between S_2 and T_2. It is however easy to see that this approach leads to a multitask whose size is exponential in the number n of consecutive choices. For $n = 3$ we already have:

$$(S_1,(S_2,(S_3 \oplus_{\mathbf{X}_3} T_3)) \oplus_{\mathbf{X}_2} (T_2,(S_3 \oplus_{\mathbf{X}_3} T_3))) \oplus_{\mathbf{X}_1} (T_1,(S_2,(S_3 \oplus_{\mathbf{X}_3} T_3)) \oplus_{\mathbf{X}_2} (T_2,(S_3 \oplus_{\mathbf{X}_3} T_3)))$$

We now discuss an encoding that avoids the exponential blowup. This will be key to showing **PSPACE**-hardness of the elementary game.

Definition 25. *$(S_i \oplus_{\mathbf{X}_i} T_i)_{1 \le i \le n}$ be a sequence of choice tasks. The modified task $\langle S_i \oplus_{\mathbf{X}_i} T_i \rangle$ is defined as follows, where $a_1, b_1, \ldots, a_n, b_n$ is a sequence of 'fresh' tokens:*

$$\begin{aligned}
\text{for } 1 < i < n\text{:}\quad \langle S_i \oplus_{\mathbf{X}_i} T_i \rangle := {} & ((S_i, (a_i^- \oplus_{\mathbf{O}} b_i^-), a_{i-1}^+) \oplus_{\mathbf{X}_i} (T_i, (a_i^- \oplus_{\mathbf{O}} b_i^-), a_{i-1}^+)) \\
& \oplus_{\mathbf{P}} ((S_i, (a_i^- \oplus_{\mathbf{O}} b_i^-), b_{i-1}^+) \oplus_{\mathbf{X}_i} (T_i, (a_i^- \oplus_{\mathbf{O}} b_i^-), b_{i-1}^+)) \\
\langle S_n \oplus_{\mathbf{X}_n} T_n \rangle := {} & ((S_n, a_{n-1}^+) \oplus_{\mathbf{X}_n} (T_n, a_{n-1}^+)) \\
& \oplus_{\mathbf{P}} ((S_n, b_{n-1}^+) \oplus_{\mathbf{X}_n} (T_n, b_{n-1}^+)) \\
\langle S_1 \oplus_{\mathbf{X}_1} T_1 \rangle := {} & (S_1, (a_1^- \oplus_{\mathbf{O}} b_1^-)) \oplus_{\mathbf{X}_1} (T_1, (a_1^- \oplus_{\mathbf{O}} b_1^-))
\end{aligned}$$

We define $\langle S_i \oplus_{\mathbf{X}_i} T_i \rangle_{1 \le 1 \le n}$ as the multitask $\{\langle S_1 \oplus_{\mathbf{X}_1} T_1 \rangle, \langle S_2 \oplus_{\mathbf{X}_2} T_2 \rangle, \ldots, \langle S_n \oplus_{\mathbf{X}_n} T_n \rangle\}$.

Unlike the naive encoding, $\langle S_i \oplus_{\mathbf{X}_i} T_i \rangle_{1 \le 1 \le n}$ grows only linearly in the size of n. We now claim that it enforces that all choices are made in the right order.

The idea for the $\langle\rangle$-encoding is the following. For $i > 0$, $\langle S_i \oplus_{\mathbf{X}_i} T_i \rangle$ is a $\mathbf{P}$-choice between two modified copies of the task $S_i \oplus_{\mathbf{X}_i} T_i$, one of which has an a_{i-1}^+ appended to it while the other one has b_{i-1}^+. However $\mathbf{P}$ only knows which one of these positive tasks she needs to win the game once $\mathbf{O}$ has made his choice in the task $a_{i-1}^- \oplus_{\mathbf{O}} b_{i-1}^-$, and this $\mathbf{O}$-choice can only happen after $\mathbf{P}$ has moved in $\langle S_{i-1} \oplus_{\mathbf{X}_{i-1}} T_{i-1} \rangle$.

Lemma 26. *Let $(S_i \oplus_{\mathbf{X}_i} T_i)_{1 \le i \le n}$ be a sequence of choice tasks. Assume that at some point during a play on $S, \langle S_i \oplus_{\mathbf{X}_i} T_i \rangle_{1 \le 1 \le n}$, the choice between S_k and T_k is being made, where the choice between S_{k+1} and T_{k+1} has already taken place. Then $\mathbf{O}$ has a winning strategy for the remaining game.*

Proof. The choice between S_k and T_k adds $a_k^- \oplus_{\mathbf{O}} b_k^-$ to the multitask; and as the choice between S_{k+1} and T_{k+1} has been made, there is exactly one of a_k^+ and b_k^+ in the multitask. Now **O** wins by choosing the token in $a_k^- \oplus_{\mathbf{O}} b_k^-$ that does not match. □

Theorem 27. **P**-*winnability in the elementary game is* **PSPACE**-*complete.*

Proof. Containment in **PSPACE** follows, for example, from Prop. 16 and the fact that **MALL**+ (*mix*) is in **PSPACE** [Hor15].

The hardness proof is an extension of the **NP**-hardness proof in Prop. 24. We translate a quantified boolean formula φ of the form

$$Q_1 x_1 \dots Q_m x_m \bigwedge_{i=1}^{n} (l_i \vee m_i \vee r_i)$$

where $Q_i \in \{\forall, \exists\}$ to **P** to an elementary game $\varphi^\dagger$ such that validity of φ is equivalent to **P**-winnability of $\varphi^\dagger$.

Without loss of generality, we assume that all variables in φ are bound by a unique quantifier. We will use again the multitask S_i defined as in the proof of Prop. 24. Moreover, we define

$$D_{x_i} := (n \cdot x_i^+) \oplus_{\mathbf{X}_i} (n \cdot \bar{x}_i^+) \qquad \text{and} \qquad \varphi^\dagger := \langle D_{x_i} \rangle_{1 \le i \le n}, S_1, \dots, S_n.$$

Just like in Prop. 24, we interpret a choice in D_{x_i} as fixing the value of x_i to be either true (left disjunct) or false (right disjunct). This time the choice is by **P** if x_i is existentially quantified and by **O** if x_i is universally quantified. To make sure that both players choices are made in an order respecting the quantifier string $Q_1 x_1 \dots Q_m x_m$ we now use the $\langle\rangle$-encoding. The map $\phi \mapsto \phi^\dagger$ is computable in polynomial time. Using the same argument as in the proof of Prop. 24 and Lem. 26, it is easy to see that $\models_P \varphi^\dagger$ iff φ is valid. □

A similar but different 'trick' for encoding the quantifier order appears in the **PSPACE**-hardness proof for **MALL** in [LMSS92]. However the encoding in [LMSS92] uses the tensor $\otimes$ which does not have a direct correspondent in the game, see Section 3.2.

4 The full token game

4.1 Undecidability of the full game

In this section, we show the undecidability of the full game by encoding the halting problem of a Turing-complete machine model, namely and-branching two counter machines. The same model is used in the undecidability proof [LMSS92] of linear logic, and we loosely follow the outline given there.

An *and-branching two counter machine* (ACM) $X = (Q, T, q_I, q_E)$ consists of a finite set Q of states with distinguished initial state $q_I \in Q$ and final state $q_E \in Q$ and a finite set T of transitions (to be defined below). ACMs are models of parallel computation, and therefore one is interested in multisets of states. An *instantaneous description* (ID) of an ACM is a finite multiset of *machine states* (q, a, b) where $q \in Q$ is the current state and $a, b \in \mathbb{N}_0$ are the values of the two counters. An ID is accepting if all of its elements are $(q_E, 0, 0)$. Each transition replaces one state within an ID by one or two states. The possible forms of transitions and their effect on an ID are described in the first two columns of Figure 2. Note in particular that a *decrease* instruction can only be carried out if the respective counter is positive.

Figure 2: ACM instructions and their encoding

$t \in T$	effect of instruction	encoding $\hat{t}$
$(p, inc(A), q)$	$(p, a, b) \to_t (q, a+1, b)$	$\{p^-, a_0^+, a_1^-, q^+\}$
$(p, dec(A), q)$	$(p, a+1, b) \to_t (q, a, b)$	$\{p^-, a_1^+, a_0^-, q^+\}$
$(p, inc(B), q)$	$(p, a, b) \to_t (q, a, b+1)$	$\{p^-, b_0^+, b_1^-, q^+\}$
$(p, dec(B), q)$	$(p, a, b+1) \to_t (q, a, b)$	$\{p^-, b_1^+, b_0^-, q^+\}$
$(p, fork, q_1, q_2)$	$(p, a, b) \to_t (q_1, a, b), (q_2, a, b)$	$\{p^-, q_1^+ \oplus_{\mathbf{O}} q_2^+\}$

Definition 28. *An ACM X is* accepting *if there is a computation sequence from $\{(q_I, 0, 0)\}$ to an accepting ID.*

Note that there are two types of counter tests in an ACM: First, whenever we apply $dec(A)$ or $dec(B)$, the counter A or B is implicitly tested for positivity. Second, there is a zero test for both counters at the end of the computation.

For technical reasons, in Figure 2 we assume $p \neq q_E$ (no transition applies to the final state) and $p \notin \{q, q_1, q_2\}$ (no simple loops). It is easy to see that both requirements are not essential.[8]

We now move towards the encoding of ACMs into multitasks. We first discuss the encoding of counter values, as this is the tricky bit. An idea is to assign to a counter value A a token a, and model the value $A = n$ by a multitask $k \cdot a^+, l \cdot a^-$ where $k - l = n$. This idea does not work as we must be able to implement a zero test in final states, i.e. a test for $k = l$. But increasing k always preserves **P**-winnability and increasing l preserves **O**-winnability, while violating the equation $k = l$. Thus, with this encoding, it seems impossible to relate a successful zero test with winnability by either player.

Instead, we will use two tokens a_0, a_1 and encode $A = n$ by $k \cdot \{a_0^+, a_1^-\}, l \cdot \{a_1^+, a_0^-\}$ where $n = k - l$. It is easy to see that this multitask is **P**-winnable iff $k = l$, i.e. $A = 0$. This is progress, but we also need to test for $A \geq 0$, i.e $k \geq l$ for the $dec(A)$ instruction. We do so by allowing **O** to abort the game at any point where he suspects a previous $dec(A)$ was illegal, and therefore $A < 0$. However, when **O** does so he must grant **P** a $[\![a_1^+]\!]_{\mathbf{P}}$. Thus **P** has to win $k \cdot \{a_0^+, a_1^-\}, l \cdot \{a_1^+, a_0^-\}, [\![a_1^+]\!]_{\mathbf{P}}$ to prove **O** wrong, and indeed this game is **P**-winnable iff $k \geq l$.

Consider now an ACM $X = (Q, T, q_I, q_E)$ where $T = \{t_1, \ldots, t_k\}$. We identify each state $q \in Q$ with a *state token*, and add four *counter tokens* a_0, a_1, b_0, b_1 for encoding the two counters A and B.

Definition 29. *The multitask encoding of an ACM X is*

$$\hat{X} := \{q_I^+, q_E^-, [\![C \oplus_{\mathbf{O}} (\hat{t_1} \oplus_{\mathbf{P}} \ldots \oplus_{\mathbf{P}} \hat{t_k})]\!]_{\mathbf{P}}\}$$

where $C := \{q_E^+, [\![a_1^+]\!]_{\mathbf{P}}, [\![b_1^+]\!]_{\mathbf{P}}\}$ is the control multitask *and $\hat{t_i}$ is defined as in Figure 2.*

Proposition 30. *X is accepting iff $\models_{\mathbf{P}} \hat{X}$.*

Theorem 31. *It is undecidable whether* **P** *has a winning strategy in a given multitask S.*

Proof. Follows from Prop. 30 and the undecidability of the halting problem for ACMs [LMSS92]. □

We will need some preparations before we can prove Prop. 30.

[8] Both properties are used in the proof of Lem. 33, see appendix.

Definition 32. *A* quasi-simulation state *is any multitask of the form*

$$S = R, U, q_E^-, [\![C \oplus_{\mathbf{O}} (\hat{t_1} \oplus_{\mathbf{P}} \ldots \oplus_{\mathbf{P}} \hat{t_k})]\!]_{\mathbf{P}}$$

where R contains polarized counter tokens such that $a_0 : R = -a_1 : R$ and $b_0 : R = -b_1 : R$, and U contains polarized state tokens such that $\sum_{p \in Q} p : U = 1$ and $q_E : U \geq 0$.

A quasi-simulation state S is p-proper, *if $a_0 : R \geq 0$, $b_0 : R \geq 0$, $p : U = 1$, and $q : U = 0$ for all $q \in Q \setminus \{p\}$. To such a proper simulation state we associate the machine state $\rho(S) = (p, a_0 : R, b_0 : R)$. A p_E-proper simulation state is called* final.

$\hat{X}$ is a simulation state with $R = \emptyset$, $U = \{q_I^+\}$ and $\rho(\hat{X}) = (q_I, 0, 0)$. Note also that a proper simulation state S is final iff $q_E : S = 0$, as $q_E : S = q_E : U - 1$.

The quasi-simulation states serve as an over-approximation to proper states encoding a machine state. In our encoding we cannot prevent **P** from choosing an instruction that decreases a register that is already zero, or one that does not match the current state. However, such 'illegal' moves lead from proper states to improper ones (Lem. 33) and we will make sure that **O** wins all improper states (Lem. 34.1).

Lemma 33. *Let S be a proper simulation state and $t \in T$.*

- *If t is non-forking, then $S, \hat{t}$ is a quasi–simulation state. $S, \hat{t}$ is proper iff $\rho(S) \to_t \rho(S, \hat{t})$.*
- *If $t = (p, fork, q_1, q_2)$, then $S, \hat{t}_1$ and $S, \hat{t}_2$ are quasi-simulation states, where $\hat{t}_i = \{p^-, q_i^+\}$. They are both proper iff $\rho(S) \to_t \rho(S, \hat{t}_1), \rho(S, \hat{t}_2)$.*

Proof. See appendix. □

Lemma 34 (properties of simulation states)**.**

1. *If S is an improper simulation state, then $\models_{\mathbf{O}} S$.*
2. *If S is a proper simulation state, then $\models_{\mathbf{P}} S, C$.*
3. *If S is a final simulation state S, then $\models_{\mathbf{P}} S \iff \rho(S) = (q_E, 0, 0)$.*

Proof. (1) There are three possible sources of improperness in a quasi-simulation state: Either $a_0 : S < 0$, $b_0 : S < 0$, or $q : U < 0$ for some $q \in Q$ (where by the properties of a quasi-simulation state, $q \neq q_E$). In any case, **O**'s strategy is to choose the control task C whenever prompted. This means that all three aforementioned counts remain unchanged in the remainder of the game. But then **P** loses, as one of the counts is negative.

(2) By assumption, the only tokens that can have negative counts in S are q_E (being at least -1) and a_1, b_1. However, these can all be matched using positive tokens in C.

(3) As we assumed that no transitions start in q_E, Lem. 33 implies that $S, \hat{t}$ is improper for all $t \in T$, and therefore $\models_{\mathbf{O}} S, \hat{t}$ by (1). Therefore **P** has no incentive to unpack her repeat task. It follows that **P** needs to win in the remaining multitask, i.e. in R, U, q_E^-. As S is final, there is a matching for the state tokens in U, q_E^-; and the counter tokens a_0, a_1, b_0, b_1 in R can be matched if and only if their counts are all non-negative. But as $a_0 : R = -a_1 : R$ and $b_0 : R = -b_1 : R$ this can only be the case if all token counts are 0, which amounts to $\rho(S) = (q_e, 0, 0)$. □

Proof of Prop. 30. It is straightforward to encode an accepting run of X as a **P**-winning strategy in $\hat{X}$. We therefore focus on the direction from $\models_{\mathbf{P}} \hat{X}$ to X being accepting.

Consider the game on any proper and non-final simulation state S. Assuming that both players move in the best of their interest, the game proceeds as follows: **P** is the only player who

can move; and she must move, as by assumption $q_E : S = -1$. She will then unpack one task from the repeat and waits for **O**'s choice (it is easy to see that is not beneficial for **P** to unpack multiple times before getting **O**'s answer). **O** now chooses the right disjunct $\hat{t_1} \oplus_{\mathbf{P}} \ldots \oplus_{\mathbf{P}} \hat{t_k}$, as otherwise the game becomes **P**-winnable (Lem. 34.2). Then **P** chooses a transition in such a way, if possible, that the resulting quasi-simulation state (or both states, in the case of a fork instruction) is again proper, as otherwise **O** would have a winning strategy (Lem. 34.1). Together with Lem. 33 this shows that if both players play in the best of their interests, the arising sequence $S_1, S_2, \ldots$ of simulation states is such that $\rho(S_1) \to_{t_1} \rho(S_2) \to_{t_2} \ldots$ for suitable transitions $t_i \in T$. Moreover **P** can only win if the sequence arrives at a final simulation state S_n with $\rho(S) = (q_E, 0, 0)$ (Lem. 34.3). □

4.2 Some rules for the full game

Extending the τ-translation of Definition 15 to repeat tasks via $\tau([\![S]\!]_{\mathbf{P}}) = ?\tau(S)$ and $\tau([\![S]\!]_{\mathbf{O}}) = !\tau(S)$ we have the following partial extension of Prop. 16.

Theorem 35. *Let $S = \{s_1, \ldots, s_n\}$. If $\mathbf{LL} + (mix)$ proves $\vdash \tau(s_1), \ldots, \tau(s_n)$, then $\models_{\mathbf{P}} S$.*

Proof sketch. It is not difficult to show that all rules below preserve **P**-winnability. Modulo τ, they correspond to the exponential rules of linear logic $(pr!)$, $(c?)$, $(dr?)$ and $(w?)$ (see Figure 1).

$$\frac{[\![S_1]\!]_{\mathbf{P}}, \ldots, [\![S_n]\!]_{\mathbf{P}}, T}{[\![S_1]\!]_{\mathbf{P}}, \ldots, [\![S_n]\!]_{\mathbf{P}}, [\![T]\!]_{\mathbf{O}}} \quad \frac{[\![S]\!]_{\mathbf{P}}, [\![S]\!]_{\mathbf{P}}, T}{[\![S]\!]_{\mathbf{P}}, T} \quad \frac{S, T}{[\![S]\!]_{\mathbf{P}}, T} \quad \frac{T}{[\![S]\!]_{\mathbf{P}}, T}$$ □

On the other hand, these rules do not fully characterize the infinite game: The multitask $[\![a^-, a^+]\!]_{\mathbf{P}}, a^-$ (corresponding to $?(a^\perp \parr (a \wedge 1)), a^\perp$ in linear logic) is **P**-winnable, but not provable using the rules given so far. **P**'s strategy consists in unpacking the repeat once, matching one a^+ with the preexisting a^-, and then repeat the whole procedure to match the newly introduced a^-; the resulting non-terminating play is **P**-winning. $[\![a^-, a^+]\!]_{\mathbf{P}}, a^-$ becomes provable if we allow the following rule, which can easily seen to be sound in the token game:

$$\frac{S, T}{[\![S, T]\!]_{\mathbf{P}}, T} \ (interleave)$$

The rule (*interleave*) implements a simple kind of cyclic reasoning, and it might indeed be the case that the full power of cyclic proof systems (see e.g. [BS11]) is needed to fully capture the token game.

4.3 Determinacy

In this section, we use the Borel Determinacy Theorem (BDT) [Kec12, Chapter 20] to prove the determinacy of the token game: For every multitask S, one of the players has a winning strategy. Strictly speaking the BDT applies only to sequential games, and so we show in the appendix (Prop. 40) that the token game can be reformulated in a sequential format.

Let us for now assume that the BDT applies to the token game as-is. By the BDT, determinacy follows if we can show that the set of winning plays (Definition 7) forms a Borel set in the product topology on $\mathcal{G}^\infty(S)$. Recall that the basic open sets of the product topology on $\mathcal{G}^\infty(S)$ are sets consisting of all plays that realize a given position.

Lemma 36. *The set of winning plays for **P** in $\mathcal{G}^\infty(S)$ is a Borel set.*

Proof. Let $\mathbf{O}_{\oplus}(S)$ and $T^{-}(S)$ be the set of all occurrences of $\mathbf{O}$-choices and negative tokens that can appear in a token game starting on S; by a suitable encoding, both sets are countable.[9] So if $t \in \mathbf{O}_{\oplus}(S)$ and $(S_1, S_2, \ldots) \in \mathcal{G}^{\infty}(S)$ is a play with $t \in S_k$ and $t \in S_l$ for $l > k$, this indicates that the $\mathbf{O}$-choice in t has not been made between S_k and S_l. Similarly, if $a^{-} \in T^{-}(S)$ and $(S_1, S_2, \ldots)$ is a play with $a^{-} \in S_k$ and $a^{-} \in S_l$ for $l > k$, this indicates that the occurrence of a^{-} has not been matched between S_k and S_l. For any $t \in \mathbf{O}_{\oplus}(S) \cup T^{-}(S)$, we let

$$[\![t \in \rho_i]\!] = \{(S_1, S_2, \ldots) \in \mathcal{G}^{\infty}(S) : t \in S_i\}$$

be the set of all plays that contain t at the i-th position. It is not difficult to see that $[\![t \in \rho_i]\!]$ can be written as a union of basic open sets and is thus itself open. Its complement $[\![t \notin \rho_i]\!] := \mathcal{G}^{\infty}(S) \setminus [\![t \in \rho_i]\!]$ is therefore closed. The set of $\mathbf{O}$-responsive and $\mathbf{P}$-responsive (Definition 7) plays in $\mathcal{G}^{\infty}(S)$ can now be written as

$$\mathrm{ORes} = \bigcap_{i \geq 1} \bigcap_{t \in \mathbf{O}_{\oplus}(S)} \bigcup_{j \geq i} [\![t \notin \rho_j]\!] \qquad \text{and} \qquad \mathrm{PRes} = \bigcap_{i \geq 1} \bigcap_{a^{-} \in T^{-}(S)} \bigcup_{j \geq i} [\![a^{-} \notin \rho_j]\!].$$

Both ORes and PRes are obtained from closed sets by the operations of countable union and countable intersection, and are therefore Borel sets. Hence $(\mathcal{G}^{\infty}(S) \setminus \mathrm{ORes}) \cup \mathrm{PRes}$, the set of winning plays for $\mathbf{P}$ in $\mathcal{G}^{\infty}(S)$, is Borel as well (in fact, it is a $\mathbf{\Delta}^0_4$ set in the Borel hierarchy). □

Theorem 37. *For every multitask S, the token game starting at S is determined.*

Proof. The Borel Determinacy Theorem deals with sequential games only, but by Prop. 40, we can investigate an equivalent sequential game instead. As demonstrated in Lem. 36, the set of winning plays for $\mathbf{P}$ is a Borel set. Hence, the game is determined. □

Corollary 38. *It is undecidable whether $\mathbf{O}$ has a winning strategy in a given multitask S.*

Proof. By Thm. 37, $\models_{\mathbf{O}} S$ iff *not* $\models_{\mathbf{P}} S$, and the latter is undecidable by Thm. 31. □

5 Conclusion

Token games and the logic of tasks. The token game presented here shares some ideas with Japaridze's *logic of tasks* [Jap02, Jap06], especially in its presentation as a two-player 'coin game' given in [MV08]. In both games a player has to obtain matchings between positive and negative occurrences of atoms, and both players make certain choice moves. A crucial difference is that the coin game features an additional level of arbitrariness: Once all choice moves are made, truth values are assigned to all atom occurrences, and then these post-assigned truth values are used to determine which player has won. The token game is simpler than the coin game, but the latter is more 'logical' (for example it is closed under substitution). Ultimately, the logic of the coin game unravels in quite a different way (it turns out to be a model of affine logic) and it is not clear to us if there is a correspondence between both games that goes beyond the basic conceptual ideas.

[9] For example, we could formalise a task occurrence t as function that takes a multitask T as an argument and returns the index of t in T if $t \in T$, and a default value if $t \notin T$.

Further work. In Section 4.2 we discussed a sound calculus for the full game. The most obvious open question is that of the axiomatisability of the full game; in fact we do not even know if the set of **P**-winnable multitasks is computably enumerable. It could also be enlightening to describe the subclass of games for which reasoning in linear logic is not only sound (cf. Thm. 35), but complete.

In light of the undecidability of the full token game, it could be instructive to investigate the decidability for some of its fragments, e.g. those using only tasks of a specific shape or only a fixed number of tokens. For example, it should not be too difficult to show that the fragment based on a single token is decidable.

Acknowledgements. We thank the anonymous reviewers for their careful reading and helpful suggestions.

References

[AJ94] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2):543–574, 1994.

[Bla92] Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56(1–3):183–220, 1992.

[BS11] James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.

[Ehr61] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicæ*, 49(129-141):13, 1961.

[FR94] Arnaud Fleury and Christian Retoré. The mix rule. *Mathematical Structures in Computer Science*, 4(2):273–285, 1994.

[Gil74] Robin Giles. A non-classical logic for physics. *Studia Logica*, 4(33):399–417, 1974.

[Gil77] Robin Giles. A non-classical logic for physics. In R. Wojcicki and G. Malinkowski, editors, *Selected Papers on Łukasiewicz Sentential Calculi*, pages 13–51. Polish Academy of Sciences, 1977.

[Hin74] Jaakko Hintikka. Quantifiers vs. quantification theory. *Linguistic inquiry*, 5(2):153–177, 1974.

[Hor15] Ross James Horne. The consistency and complexity of multiplicative additive system virtual. *Scientific Annals of Computer Science*, 25(2):245–316, 2015.

[HV19] Wilfrid Hodges and Jouko Väänänen. Logic and Games. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2019 edition, 2019.

[Jap02] Giorgi Japaridze. The logic of tasks. *Annals of Pure and Applied Logic*, 117(1-3):261–293, 2002.

[Jap03] Giorgi Japaridze. Introduction to computability logic. *Annals of Pure and Applied Logic*, 123(1-3):1–99, 2003.

[Jap06] Giorgi Japaridze. Introduction to cirquent calculus and abstract resource semantics. *Journal of Logic and Computation*, 16(4):489–532, 2006.

[Kec12] Alexander Kechris. *Classical descriptive set theory*, volume 156. Springer Science & Business Media, 2012.

[LL78] Paul Lorenzen and Kuno Lorenz. *Dialogische Logik*. Wissenschaftliche Buchgesellschaft, 1978.

[LMSS92] Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of pure and applied logic*, 56(1-3):239–311, 1992.

[Lor60] Paul Lorenzen. Logik und Agon. In *Atti del XII Congresso Internazionale di Filosofia*, volume 4, pages 187–194. Sansoni, 1960.

[Lor01] Kuno Lorenz. Basic objectives of dialogue logic in historical perspective. *Synthese*, 127(1-2):255–263, 2001.

[MV08] Ilya Mezhirov and Nikolay Vereshchagin. On game semantics of the affine and intuitionistic logics. In *International Workshop on Logic, Language, Information, and Computation*, pages 28–42. Springer, 2008.

Appendix

Proof of Lemma 9

Lemma. *If* $\models_{\mathbf{P}} S$ *and* $S \to^*_{\mathbf{O}} T$*, then* $\models_{\mathbf{P}} T$*.*

Proof. Let $S \to^*_{\mathbf{O}} T = S^{\mathbf{P}} \cup U'$ and let $\mathcal{H}$ be **P**'s winning strategy in S. According to $\mathcal{H}$, **P** makes the move $S^{\mathbf{P}} \to^*_{\mathbf{P}} U$ in S. By the definition of a strategy, the position $(S, S^{\mathbf{P}} \cup U', U \cup U')$ is realized in $\mathcal{H}$. Let $V = U \cup U'$. Following $\mathcal{H}$, **P** then chooses some $V^{\mathbf{P}} \to^*_{\mathbf{P}} W$.

P's strategy in T is as follows. She starts by moving directly to $T^{\mathbf{P}} \to^*_{\mathbf{P}} W \cup U^{\mathbf{O}}$. This is possible since

$$T^{\mathbf{P}} = (S^{\mathbf{P}} \cup U')^{\mathbf{P}} = S^{\mathbf{P}} \cup U'^{\mathbf{P}} \to^*_{\mathbf{P}} U \cup U'^{\mathbf{P}} = U^{\mathbf{P}} \cup U^{\mathbf{O}} \cup U'^{\mathbf{P}} = V^{\mathbf{P}} \cup U^{\mathbf{O}} \to^*_{\mathbf{P}} W \cup U^{\mathbf{O}}.$$

After an arbitrary **O**-move $T^{\mathbf{O}} =\to^*_{\mathbf{O}} W'$, the new position in the T-game is $(T, T^{\mathbf{P}} \cup W', W \cup U^{\mathbf{O}} \cup W')$. Back to the S-game, we note that also $V^{\mathbf{O}} = U^{\mathbf{O}} \cup (U')^{\mathbf{O}} = U^{\mathbf{O}} \cup T^{\mathbf{O}} \to^*_{\mathbf{O}} U^{\mathbf{O}} \cup W'$ is a legal **O**-move. Hence, the position $(S, S^{\mathbf{P}} \cup U', V, V^{\mathbf{P}} \cup (U^{\mathbf{O}} \cup W'), W \cup (U^{\mathbf{O}} \cup W'))$ is realized in $\mathcal{H}$. Now the two games are synchronized and **P** can use $\mathcal{H}$ to continue playing in the T-game. Clearly, this strategy is winning, since none of the winning conditions depend on finite initial segments. □

Proof of Proposition 10

Proposition. *If* $\models_{\mathbf{P}} S$ *and* $\models_{\mathbf{P}} T$*, then* $\models_{\mathbf{P}} S, T$*.*

Proof. Let $\mathcal{H}_S$ and $\mathcal{H}_T$ be winning strategies for **P** in S and T, respectively. Intuitively, **P**'s strategy in S, T is to play $\mathcal{H}_S$ on S and $\mathcal{H}_T$ on T. We define this strategy $\mathcal{H}$ to consist of plays

$$(S_1 \cup T_1, S_2 \cup T_2, \ldots)$$

such that $(S_1, S_2, \ldots) \in \mathcal{H}_S$ and $(T_1, T_2, \ldots) \in \mathcal{H}_T$. To show that $\mathcal{H}$ is indeed a well-defined strategy for **P** according to Definition 5, assume that $(S_1 \cup T_1, \ldots, S_k \cup T_k)$ is realized in $\mathcal{H}$ and k is odd. Since $\mathcal{H}_S$ and $\mathcal{H}_T$ are **P**-strategies, there are unique moves $S_k^{\mathbf{P}} \to^*_{\mathbf{P}} U$ and $T_k^{\mathbf{P}} \to_{\mathbf{P}} V$ such that for all $S_k^{\mathbf{O}} \to_{\mathbf{O}} U'$, $T_k^{\mathbf{O}} \to_{\mathbf{O}} V'$, the position $(S_1, \ldots, S_k, S_k^{\mathbf{P}} \cup U', U \cup U')$ is realized in $\mathcal{H}_S$ and $(T_1, \ldots, T_k, T_k^{\mathbf{P}} \cup V', V \cup V')$ is realized in $\mathcal{H}_T$. Let $W = U \cup V$. We note that $(S_k^{\cup} T_k)^{\mathbf{P}} = S_k^{\mathbf{P}} \cup T_k^{\mathbf{P}} \to_{\mathbf{P}} U' \cup T' = W$ is the unique **P**-move dictated by $\mathcal{H}$ in the current position. To see that it satisfies the required independence of **O**'s preceding move, let $(S_k \cup T_k)^{\mathbf{O}} \to_{\mathbf{O}} W'$ be an arbitrary **O**-move. We easily see that this move decomposes into $S_k \to_{\mathbf{O}} U'$ and $T_k^{\mathbf{O}} \to_{\mathbf{O}} V'$ with $W' = U' \cup T'$. By the above,

$$(S_1 \cup T_1, \ldots, S_k \cup T_k, (S_k \cup T_k)^{\mathbf{P}} \cup W', W \cup W')$$

is realized in $\mathcal{H}$. The fact that $\mathcal{H}$ is winning follows directly from the the corresponding facts for $\mathcal{H}_S$ and $\mathcal{H}_T$. □

Proof of Lemma 33

Lemma. *Let S be a proper simulation state and $t \in T$.*

- *If t is non-forking, then $S, \hat{t}$ is a quasi–simulation state. $S, \hat{t}$ is proper iff $\rho(S) \to_t \rho(S, \hat{t})$.*
- *If $t = (p, fork, q_1, q_2)$, then $S, \hat{t}_1$ and $S, \hat{t}_2$ are quasi-simulation states, where $\hat{t}_i = \{p^-, q_i^+\}$. They are both proper iff $\rho(S) \to_t \rho(S, \hat{t}_1), \rho(S, \hat{t}_2)$.*

Proof. We consider only the case where $t = (p, dec(A), q)$, and consequently $S, \hat{t} = S, p^-, a_1^+, a_0^-, q^+$. The other cases are similar.

The equations $a_0 : R = -a_1 : R$ and $\sum_{p \in Q} p : U = 1$ are both preserved by adding the four polarized tokens in $\hat{t}$. Moreover $q_E : (S, \hat{t}) = q_E : S + 1$ if $q = q_E$ and $q_E : (S, \hat{t}) = q_E : S$ otherwise (recall that always $p \neq q_E$). Thus $S, \hat{t}$ is a quasi-simulation state.

We observe that $a_0 : (S, \hat{t}) = a_0 : S - 1$, $b_0 : (S, \hat{t}) = b_0 : S$, $q' : (S, \hat{t}) = q' : S$ for $q' \in Q \setminus \{p, q\}$, $p : (S, \hat{t}) = p : S - 1$ and $q : (S, \hat{t}) = p : S + 1$ (recall that always $p \neq q$ by assumption).

Therefore $S, \hat{t}$ is proper iff the following are satisfied: $a_0 : (S, \hat{t}) \geq 0$, and thus $a_0 : S \geq 1$; $p : (S, \hat{t}) = 0$ and thus $p : S = 1$; and $q : (S, \hat{t}) = 1$. But this means that $\rho(S) = (p, a_0 : (S, \hat{t}) - 1, b_0 : S)$ and $\rho(S, \hat{t}) = (q, a_0 : (S, \hat{t}), b_0 : S)$, which is equivalent to $\rho(S) \to_t \rho(S, \hat{t})$. □

Sequentialising the token game

We show how to reformulate the token game as a sequential game, as the BDT only holds for such games. In this game, **O** starts by making finitely many moves on the multitask S. Unlike in the original game, **P** is now informed about the new multitask and makes finitely many moves herself. The resulting sequence of multitasks is a play, in the sense of Definition 4. However, the notion of strategy must be adapted to capture the sequentiality of moves:

Definition 39 (sequential strategy)**.** *A* sequential **O**-*strategy (***P**-*strategy) in S is a nonempty subset $\mathcal{H} \subseteq \mathcal{G}^\infty(S)$ with the following property: If $(S_1, \ldots, S_k)$ is realized in $\mathcal{H}$ and k is even (odd), then there exists a unique $S_k \to_{\mathbf{O}} S_{k+1}$ ($S_k \to_{\mathbf{P}} S_{k+1}$) such that $(S_1, \ldots, S_k, S_{k+1})$ is realized in $\mathcal{H}$.*

The definition of guarantees and winning strategies is like in Definition 7. For technical reasons, we introduce a second sequential game. This game is exactly as before, only that now **P** starts. The definition of play and strategies are adapted accordingly, while the definition of guarantees and winning strategies remains the same. For reference, we call the game where **O** starts the **OP**-*sequential game* (or just **OP**-*game*), the game where **P** starts the **PO**-*sequential game*, and our original game the asynchronous game. The order of turns for all three games is depicted schematically in Figure 3.

Strategically speaking, the games get easier for **P** from left to right in Figure 3. Indeed, in the **PO**-game, every move of **P** is reported to **O** before he makes his move. Hence, if **P** has a winning strategy in this game, certainly she has a winning strategy in the asynchronous game, where **O** does not have this informational advantage[10].

Similarly, the **OP**-game is easier for **P** than the asynchronous game. If she can win the asynchronous game, she can simply ignore the additional information she receives in the **OP**-game about **O**'s preceding move.

Completing the circle, **P** can also use a winning strategy for the **OP**-game in the **PO**-game by skipping the first round in the latter. We thus have:

[10] This is not always true in game theory: more information does not always give the player an advantage.

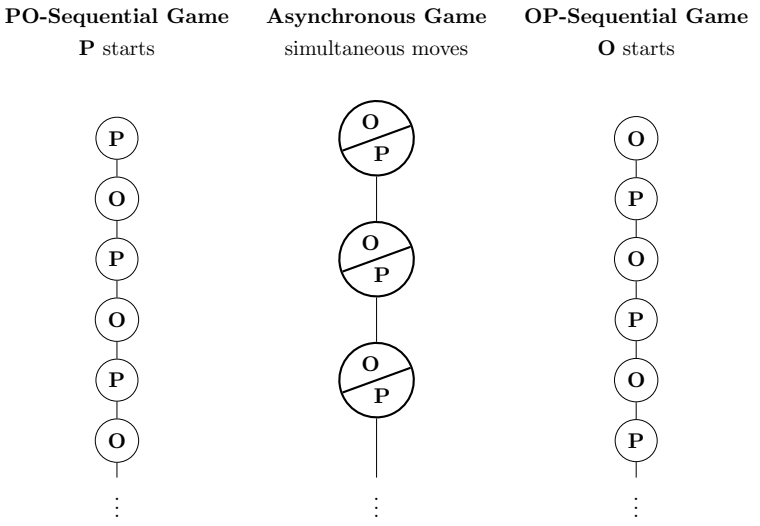

Figure 3: Three variants of the game.

Proposition 40. *Let T be a multitask. The following are equivalent:*

1. **P** *has a sequential winning strategy in the* **PO***-game starting in T.*
2. **P** *has a winning strategy the asynchronous game starting in T.*
3. **P** *has a sequential winning strategy in the* **OP***-game starting in T.*

The same holds for **O***'s winning strategies.*

Proof. $1 \to 2$: Let $\mathcal{H}$ be a winning strategy for **P** in the **PO**-game. Let us call a play $(S_1, S_2, \ldots)$ of that game *essentially asynchronous* if **O** plays without taking **P**'s latest move into consideration, i.e., for k odd we have $S_{k+1} = U \cup U'$, with $S^{\mathbf{O}}_{k-1} \to^*_{\mathbf{O}} U'$, whenever $S^{\mathbf{P}}_{k-1} \to^*_{\mathbf{P}} U$. We construct the strategy $\mathcal{H}'$ in the asynchronous game by adding for every such play a sequence $(S_1, T_2, S_3, T_4, \ldots)$ to $\mathcal{H}'$ as follows. Remember that in the asynchronous game, **O** formally moves at odd positions, while **P** at even positions but without taking **O**'s previous move into account. Since in $(S_1, S_2, \ldots)$, **O** plays essentially asynchronous, we can change the order of moves. We set for k even, $T_k = S^{\mathbf{P}}_{k-1} \cup U'$ with $S^{\mathbf{O}}_k \to^*_{\mathbf{O}} U'$ as above. **P** then moves to $S_{k+1} = U \cup U'$ with $S^{\mathbf{P}}_{k-1} \to^*_{\mathbf{P}} U$. It is easily seen that this is a well-defined winning strategy in the asynchronous game.

$2 \to 3$: Every sequential winning strategy for **P** in the asynchronous game is also a winning strategy in the **PO**-game.

$3 \to 1$: If $\mathcal{H}$ is a winning strategy for **P** in the **OP**-game, then $\mathcal{H}'$ consisting of all plays $(S_1, S_1, S_2, S_3, \ldots)$ where $(S_1, S_2, S_3, \ldots)$ is a play in $\mathcal{H}$ is a winning strategy for **P** in the **PO**-game. □