

1 Introduction

These are games with ordinal sequences of moves.

TODO: Talk about what they are and why we are interested, being careful to point out that our $\omega + 1$ games correspond to games with winning conditions.

2 Our starting category of games

Before studying games with transfinite sequences of moves, we shall illustrate some of the choices we have made by defining a category of games with finite sequences of moves. We have chosen these definitions because they extend particularly well to the transfinite case.

2.1 Games and strategies

We shall use the notation introduced in [AJ13] to describe games. All our games A will have, at their heart, the following three pieces of information:

- A set M_A of possible moves
- A function $\lambda_A: M_A \rightarrow \{O, P\}$ assigning to each move the player who is allowed to make that move
- A prefix-closed set $P_A \subset M_A^*$ of finite sequences of moves.

We shall normally insist on an *alternating condition* on P_A :

Alternating condition If $a, b \in M_A$ are moves and $s \in M_A^*$ is a sequence of moves such that $sa, sab \in P_A$, then $\lambda_A(a) = \neg\lambda_A(b)$.

As in [AJ13], we identify a *strategy* for a game A with the set of sequences of moves that can occur when player P is playing according to that strategy so that a typical definition of a (partial) strategy might be a set $\sigma \subset P_A$ such that (for all $s \in M_A^*, a, b \in M_A$):

- $\epsilon \in \sigma$ (ensures that σ is non-empty)
- If $sa \in \sigma$, $\lambda_A(a) = P$ and $sab \in P_A$ then $sab \in \sigma$ (σ contains all legal replies by player O)

- If $s, sa, sb \in \sigma$ and $\lambda_A(a) = P$ then $a = b$ (σ contains at most one legal reply by player P)

We can impose additional constraints on σ that will ensure that σ is total, strict, history free and so on. The definition given immediately above is not the only definition of a strategy found in the literature, however. For example, the games described in [AJ13] have the curious property that the set P_A may contain plays that cannot actually occur when A is being played; in particular, all plays must start with a move by player O , but the set P_A may contain positions that start with a P -move. These plays do not affect the strategies for A , but they might come into play if we perform operations on A such as forming the negation $\neg A$ or the implication $A \multimap B$.

This behaviour is made implicit in Abramsky and Jagadeesan's definitions, which do not impose any conditions upon the set P_A beyond the basic alternation condition given above, but which mandate that any play occurring *in a strategy* must begin with an O -move. For the sake of clarity, we adopt a different, but completely equivalent, approach. For a game A , we define a set L_A , regarded as the set of *legal plays* occurring in P_A . In some games models, such as that found in [Bla92], L_A may be defined to be the whole of P_A , while in [AJ13] it is defined to be that subset of P_A consisting of plays that begin with an O -move.

The point of specifying L_A separately is that it allows us to unify the definition of a *strategy*, while making clearer the behaviour observed above, whereby certain plays in P_A may not occur 'in normal play'; this behaviour was previously only implicit in the definition of a strategy. Our unified definition then becomes:

Definition 2.1. If $A = (M_A, \lambda_A, P_A)$ is a game and L_A is its associated set of legal plays (in a particular games model) then a (partial) *strategy* for A is a subset $\sigma \subset L_A$ such that for all $s \in L_A$ and all $a, b \in M_A$:

- $\epsilon \in \sigma$
- If $s \in \sigma$ and a is an O -move, and if $sa \in L_A$, then $sa \in \sigma$
- If $s \in \sigma$ and a, b are P -moves, and if $sa, sb \in \sigma$, then $a = b$

2.2 Positive and negative games, ownership of plays and connectives

Abramsky-Jagadeesan games, as described in [AJ13], may admit both plays that start with a P -move and plays that start with an O -move. Other games models, such as those found in [Bla92] and [Cur92], are more restrictive. The games in [Cur92] only contain plays starting with an O -move. The plays in [Bla92] may start with either a P -move or an O -move, but a play starting with a P -move and a play starting with an O -move may not occur in the same game.

Definition 2.2. We say that a game $A = (M_A, \lambda_A, P_A)$ is *positive* if every play in P_A begins with a P -move. We say that A is *negative* if every play in P_A begins with an O -move.

So the Curien model found in [Cur92] admits only negative games, the Blass model in [Bla92] admits positive and negative games, while the Abramsky-Jagadeesan model found in [AJ13] admits not only positive and negative games, but also games that are neither negative nor positive. We shall now examine the reasons for and drawbacks of each of these choices.

The earliest games model, found in [Con76], did not include a definition of which player is to move at a given position; rather, games are defined recursively as pairs of games $\{L|R\}$, where L represents the positions that the left player may move into, while R represents the positions that the right player may move into. Blass's definition departs completely from this tradition; now, at every position s only one of the two players is allowed to move; extending this logic on to the empty position ϵ , it follows that all games are either positive or negative. This property means that we may freely define $L_A = P_A$, since there is never any question about whose turn it is to play. By contrast, if we were to define $L_A = P_A$ for Abramsky-Jagadeesan games, then a strategy might end up containing two branches, one of plays beginning with an O -move and one of plays beginning with a P -move, which is undesirable. The alternative definition of L_A avoids this problem.

In the case of a Blass game A , we may define a function $\zeta_A: P_A \rightarrow \{O, P\}$ that says which player owns each play; the idea is that if we are in position s , then the next player to move is given by $\neg\zeta_A(s)$; i.e., the opposing player to the player who has just made the move. One might want to define ζ_A by setting $\zeta_A(sa) = \lambda_A(a)$, so that ownership of a play is decided by who

has made the last move in the play, but this definition does not extend in an obvious way to the empty position ϵ (and, as we shall see in the next chapter, it does not extend to plays over limit ordinals). In this case, $\zeta_A(\epsilon)$ is part of the game's data, and it determines whether the game is positive or negative: if $\zeta_A(\epsilon) = P$ then all plays must start with an O -move, and the game is negative – and vice versa.

An important question then arises: how should we extend the function ζ_A to games formed from connectives? The solution adopted by Blass is to use binary conjunctions to deduce the ownership of a play from the ownership of the restrictions of that play to the two component games. In the case of the tensor product $A \otimes B$ of two games A and B , we define $\zeta_{A \otimes B}: P_{A \otimes B} \rightarrow \{O, P\}$ by setting

$$\zeta_{A \otimes B}(s) = (\zeta_A(s|_A) \wedge \zeta_B(s|_B))$$

where $\wedge: \{O, P\} \times \{O, P\} \rightarrow \{O, P\}$ is as in Figure 1.

a	b	$a \wedge b$	a	b	$a \vee b$	a	b	$a \Rightarrow b$
O	O	O	O	O	O	O	O	P
O	P	O	O	P	P	O	P	P
P	O	O	P	O	P	P	O	O
P	P	P	P	P	P	P	P	P

Figure 1: Truth tables for binary conjunctions on $\{O, P\}$

Similarly, we may extend ζ to the implication $A \multimap B$ and the par $A \wp B$ by setting

$$\begin{aligned}\zeta_{A \multimap B}(s) &= (s|_A \Rightarrow s|_B) \\ \zeta_{A \wp B}(s) &= (s|_A \vee s|_B)\end{aligned}$$

Note that if we use these definitions then the owner $\zeta_C(sa)$ of a play sa might not correspond to the player $\lambda_C(a)$ who played the last move a . For example, let A, B be two positive games and form their tensor product $A \otimes B$. Then we have

$$\zeta_{A \otimes B}(\epsilon) = (\zeta_A(\epsilon) \wedge \zeta_B(\epsilon)) = O \wedge O = O$$

and so $A \otimes B$ is a positive game. Player P plays an opening move in one of the two games - let us say she plays the move a in the game A . But then we have

$$\zeta_{A \otimes B}(a) = (\zeta_A(a) \wedge \zeta_B(\epsilon)) = P \wedge O = O$$

In other words, it is still player P 's turn to play! Blass embrace this possibility and allows player P to make these two moves. In his paper, he introduces the notions of *strict* and *relaxed* games, where the strict games are the objects of study but the relaxed games are often used since they allow more manipulations. In this case, the game $A \otimes B$ is defined as a relaxed game that might not satisfy the alternating condition; in the process of converting it into a strict game, these two opening moves by player P are combined into a single move.

This ‘double move’ can only occur at the start of the game, and Blass treats it as a special case in his proofs. Perhaps unsurprisingly, this inconsistency causes major problems if we try to compose strategies. We do not get an associative composition of strategies for $A \multimap B$ with strategies for $B \multimap C$ and so we do not get a categorical semantics. An example of the failure of associativity in Blass’s games model is given towards the end of [AJ13].

By contrast, Abramsky-Jagadeesan games may admit moves by both players at the same position (specifically, at the beginning of the game, before any moves have been played), but this does not cause problems since we insist that our legal plays start with an O -move and be strictly alternating. The authors of [AJ13] note that their model can be considered as an intermediate between Conway’s games, where the position tells you nothing about which player is to move, and Blass games, where the position completely determines which player is to move. In Abramsky-Jagadeesan games, one can deduce which player is to move (by looking at which player made the last move) in every position except the empty starting position.

Curien’s game model ([Cur92]) is similar to Abramsky’s and Jagadeesan’s, but involves only negative games. The only slight problem is that negative Abramsky-Jagadeesan games are not closed under implication: if N, L are negative games then $N \multimap L$ may be neither negative nor positive. We may fix this by modifying the definition of $N \multimap L$ so that we delete from $P_{N \multimap L}$ all plays that start with a P -move - or, equivalently, by requiring that all plays start in L . This is the approach taken in [CLM13], where it fits well with the paper’s treatment of the *sequoid* operator \oslash , which is a version of the tensor product that has been modified so that play is required to start in the left-hand game.

We shall adopt elements of both the Blass and the Abramsky-Jagadeesan games models; specifically, we shall use Blass’s games and Abramsky-Jagadeesan’s strategies. This means that our games model will be more restrictive than either the Blass or the Abramsky-Jagadeesan models, but this lack of flex-

ibility will be just what we need in order to extend these games over the transfinite ordinals. We will later consider ways we can relax our model to recover Abramsky and Jagadeesan's games model.

Definition 2.3. A *game* is a triple $(M_A, \lambda_A, \zeta_A, P_A)$ where

- M_A is a set of moves,
- $\lambda_A: M_A \rightarrow \{O, P\}$ is a function that assigns a player to each move,
- $P_A \subset M_A^*$ is a non-empty prefix-closed set of plays that can occur in the game and
- $\zeta_A: P_A \rightarrow \{O, P\}$ is a function that assigns a player to each position

such that

- If $a \in M_A$ and $sa \in P_A$ then $\zeta_A(sa) = \lambda_A(a)$.
- If $a \in M_A$ and $sa \in P_A$ then $\zeta_A(s) = \neg \zeta_A(sa)$.

Remark 2.4 (Notes on the definition). Given a game $A = (M_A, \lambda_A, \zeta_A, P_A)$, define $b_A = \neg \zeta_A(\epsilon)$. Then every play in P_A must start with a b_A -move, so A is either positive or negative.

Note that ζ_A is now completely specified by λ_A and b_A , so we could have specified our games more efficiently by replacing ζ_A with b_A in our definition, as done in [CLM13]. The slightly more unwieldy ζ_A will be useful when we come to extend our games over the ordinals, though, so we retain it.

If $a \in M_A$ then we may recover $\lambda_A(a)$ from ζ_A so long as a occurs in some play in P_A . Since moves that can never be played do not affect the game at all, we do not really need λ_A in our definition, but we keep it to make the connection to earlier work clearer.

Given the game A , we define L_A to be the set of all plays $s \in P_A$ such that $s_n = O$ for all odd n and $s_n = P$ for all even n - in other words, the set of plays in P_A that start with an O -move.

References

- [AJ13] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *CoRR*, abs/1311.6057, 2013.

- [Bla92] Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56(1–3):183 – 220, 1992.
- [CLM13] Martin Churchill, Jim Laird, and Guy McCusker. Imperative programs as proofs via game semantics. *CoRR*, abs/1307.2004, 2013.
- [Con76] John H. Conway. *On Numbers and Games*, volume 6 of *London Mathematical Society Monographs*. Academic Press, London, 1976.
- [Cur92] Pierre-Louis Curien. Concrete data structures, sequential algorithms and linear logic. Email communication on types mailing list, 1992.