

1 A Fully Abstract Game Semantics for Idealized Algol

To introduce our material, we will go back over some old ground, namely the fully abstract game semantics for Idealized Algol developed by Abramsky and McCusker in [AM96]. In keeping with the spirit of this thesis, we will aim to use category theoretic methods, and so our proofs of soundness and adequacy will depart from those given by Abramsky and McCusker, and will instead involve coalgebraic ideas developed by Laird in [Lai02] and [Lai].

1.1 Idealized Algol

The ground types of Idealized Algol are called **com**, **bool**, **nat** and **Var**. The first three are data types corresponding to the sets $\mathbb{C} = \{a\}$, $\mathbb{B} = \{\text{t}, \text{f}\}$ and $\mathbb{N} = \{0, 1, 2, \dots\}$. **com** takes the role of a command or void type; typically, although the return value of a function $T \rightarrow \text{com}$ will not convey any information, the function will have side effects that *do* make a difference.

The type **Var** is the type of a variable that holds elements of \mathbb{N} . It is best understood as corresponding to the following pseudo-Java ‘interface’.

```
public interface Var
{
  nat read();
  com write(nat value);
}
```

We now present the typing rules for the language. Here, Γ will represent a *context*; i.e., a list $x_1 : T_1, \dots, x_n : T_n$ of variable names together with their types.

First, we have the usual rules for the simply typed lambda calculus.

$$\frac{}{\Gamma, x : T \vdash x : T} \quad \frac{\Gamma \vdash M : S \rightarrow T \quad \Gamma \vdash N : S}{\Gamma \vdash MN : T} \quad \frac{\Gamma, x : S \vdash M : T}{\Gamma \vdash \lambda x^S. M : S \rightarrow T}$$

We then have rules for each of the base types. At type **com** we have:

$$\frac{}{\Gamma \vdash \text{skip} : \text{com}} \quad \frac{\Gamma \vdash M : \text{com} \quad \Gamma \vdash N : T}{\Gamma \vdash M; N : T} \quad T \in \{\text{com}, \text{bool}, \text{nat}\}.$$

Here, **skip** is a generic command with no side-effects that returns the unique element of the singleton set \mathbb{C} . $M; N$ represents the sequential composition of M with N ; i.e., the term that first evaluates M , performing any of its side-effects, and then evaluates N and returns the result.

At type **bool** we have true/false values and conditionals.

$$\frac{}{\Gamma \vdash \mathbf{t} : \mathbf{bool}} \qquad \frac{}{\Gamma \vdash \mathbf{f} : \mathbf{bool}}$$

$$\frac{\Gamma \vdash M : \mathbf{bool} \quad \Gamma \vdash N : T \quad \Gamma \vdash P : T}{\Gamma \vdash \text{If } M \text{ then } N \text{ else } P : T} \quad T \in \{\mathbf{com}, \mathbf{bool}, \mathbf{nat}\}$$

At type **nat** we have numerals, arithmetic operators and a conditional that tests whether a number is equal to 0 or not.

$$\frac{}{\Gamma \vdash n : \mathbf{nat}} \qquad \frac{\Gamma \vdash M : \mathbf{nat}}{\Gamma \vdash \text{succ } M : \mathbf{nat}} \qquad \frac{\Gamma \vdash M : \mathbf{nat}}{\Gamma \vdash \text{pred } M : \mathbf{nat}}$$

$$\frac{\Gamma \vdash M : \mathbf{nat} \quad \Gamma \vdash N : T \quad \Gamma \vdash P : T}{\Gamma \vdash \text{If0 } M \text{ then } N \text{ else } P : T} \quad T \in \{\mathbf{com}, \mathbf{bool}, \mathbf{nat}\}$$

At type **Var**, we have terms that call the read and write ‘methods’ to dereference the variable or to assign a new value to it.

$$\frac{\Gamma \vdash V : \mathbf{Var}}{\Gamma \vdash !V : \mathbf{nat}} \qquad \frac{\Gamma \vdash V : \mathbf{Var} \quad \Gamma \vdash E : \mathbf{nat}}{\Gamma \vdash V \leftarrow E : \mathbf{com}}$$

We also have the ability to create a new variable.

$$\frac{\Gamma, x : \mathbf{Var} \vdash M : T}{\Gamma \vdash \text{new}_T \lambda x. M : T}$$

The idea here is that M is a term that refers to some free variable x of type **Var**; then $\text{new } \lambda x. M$ makes x behave like an actual storage cell (so, for instance, the result of the computation $\text{new}_{\mathbf{nat}} \lambda x. (x \leftarrow 5); !x$ will be 5).

We have another way of creating variables, using the **mkvar** keyword. If we think back to our illustration of the **Var** type as an interface, this becomes clearer. **mkvar** creates a new anonymous instance of the **Var** interface, using the ‘methods’ supplied through its arguments.

$$\frac{\Gamma \vdash M : \mathbf{nat} \quad \Gamma \vdash N : \mathbf{nat} \rightarrow \mathbf{com}}{\Gamma \vdash \text{mkvar } MN : \mathbf{Var}}$$

Lastly, we have fixpoint combinators at all types that we use to implement recursion.

$$\frac{\Gamma \vdash M : T \rightarrow T}{\Gamma \vdash \mathbf{Y}_T M : T}$$

1.2 Games and Strategies

We adopt the game semantics from [AM96]; these are based on the game semantics developed in [HO00], with a modification to make them into a linear category.

Definition 1.1. An *arena* is a tuple $A = (M_A, \lambda_A, \vdash_A)$, where

- M_A is a set of *moves*,
- $\lambda_A: M_A \rightarrow \{O, P\} \times \{Q, A\}$ is a function that identifies each move as either an *O-move* or a *P-move*, and as either a *question* or an *answer*, and
- \vdash_A is a relation between $M_A + \{*\}$ and M_A such that
 - if $* \vdash_A a$, then $\lambda_A(a) = (O, Q)$, and if $b \vdash_A a$ then $b = *$,
 - if $a \vdash_A b$ and a is an answer, then b is a question, and
 - if $a \vdash_A b$ and $a \neq *$, then either a is an *O-move* and b a *P-move*, or the other way round.

If $* \vdash_A a$, then we say that a is an *initial move* in A . If $a \vdash_A b$, then we say that a *enables* b .

As a shorthand, we write $\lambda_A^{OP}: M_A \rightarrow \{O, P\}$ for $\text{pr}_1 \circ \lambda_A$ and $\lambda_A^{QA}: M_A \rightarrow \{Q, A\}$ for $\text{pr}_2 \circ \lambda_A$.

Definition 1.2. A *justified sequence* in an arena A is a finite sequence s of moves together with, for each non-initial move a occurring in s , a pointer back to some move b occurring earlier in s such that $b \vdash_A a$. We say that b *justifies* a or that b is the *justifier* of a .

Given such a justified sequence, we define the *P-view* $\lceil s \rceil$ and *O-view* $\lfloor s \rfloor$ of s inductively as follows.

$$\begin{aligned}
 \lceil \epsilon \rceil &= \epsilon \\
 \lceil sa \rceil &= \lceil s \rceil a && \text{if } a \text{ is a } P\text{-move} \\
 \lceil sa \rceil &= a && \text{if } a \text{ is initial} \\
 \lceil sbta \rceil &= \lceil s \rceil ba && \text{if } a \text{ is an } O\text{-move justified by } b \\
 \\
 \lfloor \epsilon \rfloor &= \epsilon \\
 \lfloor sa \rfloor &= \lfloor s \rfloor a && \text{if } a \text{ is an } O\text{-move} \\
 \lfloor sbta \rfloor &= \lfloor s \rfloor ba && \text{if } a \text{ is a } P\text{-move justified by } b
 \end{aligned}$$

A justified sequence s is *well-bracketed* if whenever a question q justifies some answer a , then any question q' occurring after q and before a must justify some answer a' occurring between q' and a , and moreover a is the only answer justified

by q . We say that a justified sequence s is *alternating* if it alternates between O -moves and P -moves, and that it is *well-formed* if it is both well-bracketed and alternating.

We say that a well-formed justified sequence is *visible* if whenever $ta \sqsubseteq s$, and a is a P -move, then the justifier of a occurs in the P -view of t , and if whenever $tb \sqsubseteq s$, and b is a non-initial O -move, then the justifier of b occurs in the O -view of t .

We say that a justified sequence s is *legal* if it is well-formed and visible, and write L_A for the set of legal sequences occurring in A .

Note that since every non-initial move in a justified sequence s must be justified by some previous move, then the first move in the sequence must be initial and therefore an O -question. If s is alternating, this means that s ends with an O -move if it has odd length and with a P -move if it has even length.

Definition 1.3. Given a legal sequence $s \in L_A$, and a move b in s , we say that a move a in s is *hereditarily justified by b* if there is a chain of justification pointers going back from a to b .

We write $s|_b$ for the subsequence of s given by all moves in s that are hereditarily justified by b . Given a set I of initial moves, we write $s|_I$ for the subsequence of s given by all moves that are hereditarily justified by some $b \in I$.

A *game* is given by a tuple $A = (M_A, \lambda_A, \vdash_A, P_A)$ where $(M_A, \lambda_A, \vdash_A)$ is an arena and P_A is a non-empty prefix-closed subset of L_A such that if $s \in P_A$ and I is a set of initial moves, then $s|_I \in P_A$.

We shall call an odd-length sequence $s \in P_A$ an *O-position* and an even-length sequence a *P-position*.

Example 1.4 (Empty game). The *empty game* I is given by the tuple

$$(\emptyset, \emptyset, \emptyset, \{\epsilon\}),$$

where ϵ is the empty sequence.

Example 1.5 (Data-type games). Let X be some set. Then we have a game, which we shall also call X , given by:

- $M_X = \{q\} + X$,
- $\lambda_X(q) = (O, Q)$ and $\lambda_X(x) = (P, A)$ for all $x \in X$,
- $q \vdash_X x$ for each $x \in X$, and
- $P_X = \{\epsilon, q\} \cup \{qx : x \in X\}$, where the x in qx is justified by q .

In particular, we have games \mathbb{C} , \mathbb{B} and \mathbb{N} , which we shall use to model the datatypes `com`, `bool` and `nat` of Idealized Algol.

Definition 1.6. Let A be a game. Then a *strategy* for A is a non-empty even-prefix-closed set $\sigma \subseteq P_A$ of P -positions in A such that if $sab, sac \in \sigma$ then $b = c$ and the justifier of b is the justifier of c .

Here, we have identified a strategy for a game with the set of P -positions that can occur when player P plays according to that strategy. So the condition we have given is one of *determinism*: in any O -position sa that can occur in the strategy, player P must have at most one reply.

Note that there may be O -positions for which player P has no reply at all; we use these to model non-terminating computations.

We write $\sigma : A$ to denote that σ is a strategy for the game A .

Definition 1.7. A strategy σ for a game A is called *innocent* if player P 's moves only depend on the current P -view; i.e., if whenever $sab \in \sigma$, $t \in \sigma$ and $ta \in P_A$ such that $\ulcorner sa \urcorner = \ulcorner ta \urcorner$, then we have $tab \in \sigma$.

1.3 Connectives on Games

In the *product* $A \times B$ of games A and B , player O chooses either A or B on the first move and subsequent play is that game.

Definition 1.8. Given games A, B , define a game $A \times B$ by

- $M_{A \times B} = M_A + M_B$,
- $\lambda_{A \times B} = [\lambda_A, \lambda_B]$,
- $* \vdash_{A \times B} a$ if and only if $* \vdash_A a$ or $* \vdash_B a$ and $a \vdash_{A \times B} b$ if and only if $a \vdash_A b$ or $a \vdash_B b$, and
- $P_{A \times B} = \{s \in L_{A \times B} : s|_A \in P_A \text{ and } s|_B = \epsilon \text{ or } s|_A = \epsilon \text{ and } s|_B \in P_B\}$.

We extend this to arbitrary products $\prod_i A_i$ in the obvious way. In particular, the product 1 of the empty collection is the same as the empty game I defined in Example 1.4.

Here, we have written $s|_A$ for the subsequence of s consisting of all moves from M_A and $s|_B$ for the subsequence consisting of all moves from M_B .

In the *tensor product* $A \otimes B$ of games A and B , the games A and B are played in parallel, and player O may switch between games when it is his turn.

Definition 1.9. Given games A, B , define a game $A \otimes B$ by

- $M_{A \otimes B} = M_A + M_B$,
- $\lambda_{A \otimes B} = [\lambda_A, \lambda_B]$,
- $* \vdash_{A \otimes B} a$ if and only if $* \vdash_A a$ or $* \vdash_B a$ and $a \vdash_{A \otimes B} b$ if and only if $a \vdash_A b$ or $a \vdash_B b$, and

- $P_{A \otimes B} = \{s \in L_{A \otimes B} : s|_A \in P_A \text{ and } s|_B \in P_B\}$.

In the *linear implication* $A \multimap B$, the game B is played in parallel with a version of A in which the two players' roles have been switched around, and player P may switch between the two games when it is her turn.

Definition 1.10. Given games A, B , define a game $A \multimap B$ by

- $M_{A \multimap B} = M_A + M_B$,
- $\lambda_{A \multimap B} = [\neg \circ \lambda_A, \lambda_B]$,
- $* \vdash_{A \multimap B} a$ if and only if $* \vdash_B a$, and $a \vdash_{A \multimap B} b$ if and only if $a \vdash_A b$ or $a \vdash_B b$, or if a is initial in B and b is initial in A , and
- $P_{A \multimap B} = \{s \in L_{A \multimap B} : s|_A \in P_A \text{ and } s|_B \in P_B\}$.

Here, $\neg : \{O, P\} \times \{Q, A\} \rightarrow \{O, P\} \times \{Q, A\}$ is the function that reverses O and P , while leaving $\{Q, A\}$ unchanged.

In the *exponential* of a game A , infinitely many copies of A are played in parallel, and player O may switch between copies whenever it is his move.

Definition 1.11. Given a game A , define a game $!A$ by

- $M_{!A} = M_A$,
- $\lambda_{!A} = \lambda_A$,
- $\vdash_{!A} = \vdash_A$ and
- $P_{!A} = \{s \in L_{!A} : s|_b \in P_A \text{ for each initial move } b \text{ occurring in } s\}$.

Lastly, the *sequoid* $A \oslash B$ of two games A and B behaves like the tensor product $A \otimes B$, except that the opening move must take place in A .

Definition 1.12. Given games A, B , define a game $A \oslash B$ by

- $M_{A \oslash B} = M_{A \otimes B}$,
- $\lambda_{A \oslash B} = \lambda_{A \otimes B}$,
- $\vdash_{A \oslash B} = \vdash_{A \otimes B}$ and
- $P_{A \oslash B} = \{s \in P_{A \otimes B} : s = \epsilon \text{ or } s \text{ begins with a move from } A\}$.

1.4 Composition of strategies

Definition 1.13. Let A, B, C be arenas. An *interaction sequence* between A, B, C is a justified sequence \mathfrak{s} of moves drawn from M_A, M_B and M_C such that $\mathfrak{s}|_{A,B} \in L_{A \multimap B}$ and $\mathfrak{s}|_{B,C} \in L_{B \multimap C}$. Here, $\mathfrak{s}|_{A,B}$ is the subsequence of \mathfrak{s} consisting of those moves from \mathfrak{s} that occur in A or B , together with all justification pointers between moves in A and B , and $\mathfrak{s}|_{B,C}$ is defined similarly.

We write $\text{int}(A, B, C)$ for the set of all interaction sequences between A, B, C .

Given $\mathfrak{s} \in \text{int}(A, B, C)$, we write $\mathfrak{s}|_{A,C}$ for the subsequence of \mathfrak{s} consisting of those moves from \mathfrak{s} that occur in A or B . A move b in $\mathfrak{s}|_{A,C}$ justifies a move a either if b justifies a in either the A or the C components, or if b justifies in \mathfrak{s} some initial move c in B , which itself justifies a .

Definition 1.14. Let A, B, C be games, let σ be a strategy for $A \multimap B$ and let τ be a strategy for $B \multimap C$. We define $\sigma \parallel \tau$ to be given by the set

$$\{\mathfrak{s} \in \text{int}(A, B, C) : \mathfrak{s}|_{A,B} \in \sigma \text{ and } \mathfrak{s}|_{B,C} \in \tau\}.$$

Then we define the *composition* $\sigma; \tau$ of σ and τ to be given by the set

$$\{\mathfrak{s}|_{A,C} : \mathfrak{s} \in \sigma \parallel \tau\}.$$

We need some small lemmata and definitions to help us show that this is a strategy.

Lemma 1.15. *We extend the function λ_A^{OP} to sequences of moves by*

- $\lambda_A^{OP}(\epsilon) = P$ and
- $\lambda_A^{OP}(sa) = \lambda_A(a)$.

If $s \in P_{A \multimap B}$, then $\lambda_{A \multimap B}^{OP}(s) = (\lambda_A^{OP}(s|_A) \Rightarrow \lambda_B^{OP}(s|_B))$, where \Rightarrow is the binary operation on $\{O, P\}$ defined by

P	Q	$P \Rightarrow Q$
P	P	P
O	P	P
P	O	O
O	O	P

Moreover, if $\lambda_A^{OP}(s|_A) = O$ then $\lambda_A^{OP}(s|_B) = O$.

Proof. Induction on the length of s . If $s = \epsilon$, then $s|_A = s|_B = \epsilon$, and so $(\lambda_A^{OP}(s|_A) \Rightarrow \lambda_B^{OP}(s|_B)) = (P \Rightarrow P) = P = \lambda_{A \multimap B}^{OP}(s)$.

Suppose then that $s = ta$, and that $\lambda_{A \multimap B}^{OP}(t) = O$. This means that $\lambda_A^{OP}(t|_A) = P$ and $\lambda_B^{OP}(t|_B) = O$. Then, whether a is a move in A or a move in B , adding it will flip exactly one of these components – so $\lambda_{A \multimap B}(s|_A) = O$ and $\lambda_{A \multimap B}^{OP}(s|_B) = O$ if a is a move in A and $\lambda_{A \multimap B}(s|_A) = P$ and $\lambda_{A \multimap B}^{OP}(s|_B) = P$ if a is a move in C .

Suppose instead that $\lambda_{A \multimap B}^{OP}(t) = P$. By induction, this means that either $\lambda_A^{OP}(t|_A) = P$ and $\lambda_B^{OP}(t|_B) = P$ or that $\lambda_A^{OP}(t|_A) = O$ and $\lambda_B^{OP}(t|_B) = O$. In the first case, this means that either $t|_A$ is empty or its last move is a P -move in A (and therefore an O -move in $A \multimap B$), and so the move a must take place in C , meaning that $\lambda_A^{OP}(s|_A) = P$ and $\lambda_B^{OP}(s|_B) = O$.

Similarly, in the second case, the last move in $t|_C$ must be an O -move in B (and therefore an O -move in $A \multimap B$, and so the move a must take place in A , meaning that $\lambda_A^{OP}(s|_A) = P$ and $\lambda_B^{OP}(s|_B) = O$. \square

It follows that

Corollary 1.16 (Switching condition). *Only player P may switch between games in $A \multimap B$; i.e., if $tab \in P_{A \multimap B}$, and a occurs in A and b in B , or if a occurs in B and b in A , then b is a P -move.*

Proof. Otherwise, $\lambda_{A \multimap B}(t) = O$, so $\lambda_A(t|_A) = P$ and $\lambda_B(t|_B) = O$. But we must also have $\lambda_{A \multimap B}(tab) = O$, so $\lambda_A(tab|_A) = P$ and $\lambda_B(tab|_B) = O$. But this is a contradiction, since $tab|_A$ and $tab|_B$ are both one move longer than the plays $t|_A$ and $t|_B$. \square

Definition 1.17 ([Har06, §3.1]). Given $\mathfrak{s} \in \text{int}(A, B, C)$, we define the P -view $\ulcorner \mathfrak{s} \urcorner$ of \mathfrak{s} inductively as follows.

$$\begin{aligned} \ulcorner \epsilon \urcorner &= \epsilon \\ \ulcorner \mathfrak{s}a \urcorner &= \ulcorner \mathfrak{s} \urcorner a \quad \text{if } a \text{ is a move in } B, \text{ an } O\text{-move in } A \text{ or a } P\text{-move in } C \\ \ulcorner \mathfrak{s}c \urcorner &= c \quad \text{if } c \text{ is an initial move of } C \\ \ulcorner \mathfrak{s}bta \urcorner &= \ulcorner \mathfrak{s} \urcorner ba \quad \text{if } a \text{ is a } P\text{-move of } A \text{ or an } O\text{-move of } C \text{ and is justified by } b \end{aligned}$$

Lemma 1.18. *If $\mathfrak{s} \in \text{int}(A, B, C)$, then $\ulcorner \mathfrak{s} \urcorner|_{A,C} = \ulcorner \mathfrak{s} \urcorner|_{A,C}$.*

Proof. Induction on the length of \mathfrak{s} . This is clear if $\mathfrak{s} = \epsilon$.

If a is an O -move in A or a P -move in C , then a is a P -move in $A \multimap C$. We have $\ulcorner \mathfrak{s}a \urcorner|_{A,C} = \ulcorner \mathfrak{s} \urcorner|_{A,C}a$, which by the inductive hypothesis is equal to $\ulcorner \mathfrak{s} \urcorner|_{A,C}a$, which is the same as $\ulcorner \mathfrak{s}a \urcorner|_{A,C}$. If b is a move in B , then $\ulcorner \mathfrak{s}b \urcorner|_{A,C} = \ulcorner \mathfrak{s} \urcorner|_{A,C}b = \ulcorner \mathfrak{s} \urcorner|_{A,C}b$, by the inductive hypothesis.

If c is initial in C , then $\ulcorner \mathfrak{s}c \urcorner|_{A,C} = c = \ulcorner \mathfrak{s}c \urcorner|_{A,C}$.

Suppose a is a P -move of A or an O -move of C – so a is an O -move in $A \multimap C$ – and suppose that a is justified by b in the sequence $\mathfrak{s}bta$. Since a cannot be an initial move in A , b must occur in the same game as a , and in particular must not occur in B . Then we have $\ulcorner \mathfrak{s}bta \urcorner|_{A,C} = \ulcorner \mathfrak{s} \urcorner|_{A,C}ba = \ulcorner \mathfrak{s} \urcorner|_{A,C}ba$, which by the inductive hypothesis is equal to $\ulcorner \mathfrak{s} \urcorner|_{A,C}ba = \ulcorner \mathfrak{s}ba \urcorner|_{A,C}$. \square

Lemma 1.19 ([Har06, §3.1]). *Let $\mathfrak{s}a \in \text{int}(A, B, C)$ (so, in particular, $\mathfrak{s}|_{A,B}$ and $\mathfrak{s}|_{B,C}$ satisfy the visibility condition). If a is a move in B , an O -move in A or a P -move in C , then $\ulcorner \mathfrak{s}a \urcorner \in \text{int}(A, B, C)$.*

Proof. Induction on the length of \mathfrak{s} . If $\mathfrak{s} = \epsilon$, then this is clear. Otherwise, suppose that \mathfrak{s} is non-empty.

First, we claim that $\lceil \mathfrak{s} \rceil \in \text{int}(A, B, C)$. If \mathfrak{s} ends with a move in B , an O -move in A or a P -move in C , then this follows immediately from the inductive hypothesis. Otherwise, suppose that \mathfrak{s} ends with a P -move in A or an O -move in C . If this last move is initial, then $\lceil \mathfrak{s} \rceil$ is a single move, so the claim is trivial. Otherwise, write $\mathfrak{s} = \mathfrak{t}pr$, where p justifies r . By the inductive hypothesis, we have $\lceil \mathfrak{t}p \rceil \in \text{int}(A, B, C)$, and then $\lceil \mathfrak{s} \rceil = \lceil \mathfrak{t}pr \rceil = \lceil \mathfrak{t} \rceil pr = \lceil \mathfrak{t}p \rceil r \in \text{int}(A, B, C)$.

Now, since a is a P -move in $A \multimap B$ or in $B \multimap C$, its predecessor b is an O -move and has some justifier c contained in $\lceil \mathfrak{s} \rceil_X$, where $X \in \{A \multimap B, B \multimap C\}$ is that component in which a is a P -move. Then this c is preceded by some other O -move b' , which is necessarily also contained in $\lceil \mathfrak{s} \rceil$, and so has some justifier c' , contained in $\lceil \mathfrak{s} \rceil_X$ by visibility. Continuing in this way until we reach an initial move, we build up the whole of the sequence $\lceil \mathfrak{s} \rceil_X$ as a subsequence of $\lceil \mathfrak{s} \rceil$. Therefore, the justifier of a must be contained in $\lceil \mathfrak{s} \rceil$, and so $\lceil \mathfrak{s}a \rceil = \lceil \mathfrak{s} \rceil a \in \text{int}(A, B, C)$. \square

Lemma 1.20 (*O*-views in the linear implication, [HO00, 4.2,4.3]). *Let A, B be games, and let bs be a non-empty play in $A \multimap B$ beginning with an initial move b in B .*

- i) If bs ends with a P -move in B , then $\downarrow bs \downarrow_{A \multimap B} = \downarrow bs \downarrow_{B \multimap B}$.*
- ii) If bs ends with a P -move in A , then $\downarrow bs \downarrow_{A \multimap B} = b^\lceil s \rceil_A^A$.*

Proof. Induction on the length of s . If $s = \epsilon$, then bs ends with an O -move in B , and we have $\lceil b \rceil_{A \multimap B} = b = \lceil b \rceil_B$.

Otherwise, suppose that bs ends with a P -move c in B . Let d be the justifier of c . Then d must be an O -move in B . Write $bs = tduc$, where t, u are sequences. Then $\downarrow tduc \downarrow_{A \multimap B} = \downarrow t \downarrow_{A \multimap B} dc$ and $\downarrow tduc \downarrow_{B \multimap B} = \downarrow t \downarrow_{B \multimap B} dc$. By Corollary 1.16, t must end with a P -move in B , or be empty, so by the inductive hypothesis we have $\downarrow t \downarrow_{A \multimap B} = \downarrow t \downarrow_{B \multimap B}$. Therefore, $\downarrow bs \downarrow_{A \multimap B} = \downarrow tduc \downarrow_{A \multimap B} = \downarrow t \downarrow_{B \multimap B} dc = \downarrow tduc \downarrow_{B \multimap B} = \downarrow bs \downarrow_{B \multimap B}$.

Next, suppose that bs ends with a P -move a in A . Let c be the justifier of a . Then c must be an O -move in A . Write $s = tcua$, where t, u are sequences. Then $\downarrow btcua \downarrow_{A \multimap B} = \downarrow bt \downarrow_{A \multimap B} ca$ and $\lceil tcua \rceil_A^A = \lceil t \rceil_A^A ca$, since the roles are reversed in A . By Corollary 1.16, t must end in a P -move in A , or be empty, so by the inductive hypothesis we have $\downarrow bt \downarrow_{A \multimap B} = b^\lceil t \rceil_A^A$. Therefore, $\downarrow bs \downarrow_{A \multimap B} = \downarrow btcua \downarrow_{A \multimap B} = \downarrow bt \downarrow_{A \multimap B} ca = b^\lceil t \rceil_A^A ca = b^\lceil tcua \rceil_A^A = b^\lceil s \rceil_A^A$. \square

Proposition 1.21. $\sigma; \tau$ is a strategy for $A \multimap C$.

Proof. First, we claim that $\mathfrak{s}|_{A,C} \in P_{A \multimap B}$ for any $\mathfrak{s} \in \sigma \parallel \tau$. Since we certainly have $\mathfrak{s}|_{A,C}|_A = \mathfrak{s}|_{A,B}|_A \in P_A$ and $\mathfrak{s}|_{A,C}|_C = \mathfrak{s}|_C = \mathfrak{s}|_{B,C}|_C \in P_C$, it suffices to show that $\mathfrak{s}|_{A,C} \in L_{A \multimap C}$.

Suppose that $ta \sqsubseteq \mathfrak{s}|_{A,C}$. We claim that $\lambda_{A \multimap C}(t) = \neg \lambda_{A \multimap C}(a)$. By Lemma 1.15, we are in one of the following configurations.

$\lambda_A^{OP}(t _A)$	$\lambda_B^{OP}(t _B)$	$\lambda_C^{OP}(t _C)$	$\lambda_{A \multimap B}^{OP}(t _{A,B})$	$\lambda_{B \multimap C}^{OP}(t _{B,C})$	$\lambda_{A \multimap C}^{OP}(t _{A,C})$
P	P	P	P	P	P
P	P	O	P	O	O
P	O	O	O	P	O
O	O	O	P	P	P

In the configuration PPP , the move a cannot be a move in A , since that would leave $ta|_{A \multimap B}$ in the configuration OP , which is impossible by Lemma 1.15. Therefore, it must be a move in C , and must therefore be an O -move in C and hence an O -move in $A \multimap C$.

In the configuration PPO , once again the move a cannot take place in A , since this would leave $ta|_{A \multimap B}$ in an illegal configuration. Therefore, it must occur in C , and must be a P -move in C and hence a P -move in $A \multimap C$.

In the configuration POO , the move a cannot take place in C , or it would leave $ta|_{B,C}$ in the illegal configuration OP , so the move a takes place in A . Therefore, it must be an O -move in A and hence a P -move in $A \multimap C$.

Lastly, in the configuration OOO , the move a cannot occur in C , or it would leave $ta|_{B,C}$ in the configuration OP , and so it must take place in A . Therefore, it must be a P -move in A , and hence an O -move in $A \multimap C$.

Having established that $\mathfrak{s}|_{A,C}$ is alternating, we show that it is well-bracketed. Suppose that a question move q in $\mathfrak{s}|_{A,C}$ justifies some answer move a . q and a must occur in the same component, since the only case in which a move from one of A and C can justify a move in the other is when both moves are initial, and hence questions. Suppose first that q and a both occur in the game C . Suppose that some other question move q' occurs between q and a in $\mathfrak{s}|_{A,C}$. If q' occurs in C , then it must be answered by some a' occurring between q' and a , since $\mathfrak{s}|_C$ is a well-bracketed sequence. Otherwise, suppose that q' occurs in A .

By examining the table above, we see that there must be some move in B occurring between q and q' in \mathfrak{s} , since moves in A move between configurations OOO and POO , while moves in C move us between configurations PPP and PPO . Let b be the earliest such move. Then b must be a question; indeed, if it is an answer, then it is non-initial and so can only be justified by questions in B . But such a question must occur earlier in $\mathfrak{s}|_{B,C}$ than q , which would mean that q was an unanswered question when the move b was played, contradicting well-bracketedness of $\mathfrak{s}|_{B,C}$. Since b is a question, it must be answered by some a'' occurring between b and a . Therefore, since $\mathfrak{s}|_{A,B}$ is well-bracketed, the

move q' must be answered by some a' occurring between a' and a'' in $\mathfrak{s}|_{A,B}$, and therefore between a' and a in $\mathfrak{s}|_{A,C}$.

The case when q and a both occur in A is similar.

Lastly, we need to show that $\mathfrak{s}|_{A,C}$ satisfies the visibility condition. Let $ta \sqsubseteq \mathfrak{s}|_{A,C}$. Choose some $\mathfrak{t} \sqsubseteq \mathfrak{s}$ such that $\mathfrak{t}|_{A,C} = t$.

Suppose a is a P -move. Then by Lemma 1.19, $\ulcorner ta \urcorner \in \text{int}(A, B, C)$. By Lemma 1.18, $\ulcorner t \urcorner a = \ulcorner ta \urcorner = \ulcorner ta \urcorner|_{A,C}$, and therefore that the justifier of a must be inside $\ulcorner t \urcorner$.

Secondly, suppose that a is an O -move. If a is an O -move in C , then either it is initial or t ends with some P -move in B , and therefore $\ulcorner t \urcorner|_{A \multimap C} = \ulcorner t \urcorner|_{B \multimap B} = \ulcorner t \urcorner|_{B,C \multimap B}$. Therefore, since $t|_{B,C}$ satisfies visibility, the justifier of a must lie in $\ulcorner t \urcorner|_{A \multimap C}$. If a is an O -move in A , then write $t = cu$ and $\mathfrak{t} = cu$, where c is the starting move in C . We have $\ulcorner cua \urcorner|_{A \multimap C} = c \ulcorner u \urcorner|_{A \multimap A} \ulcorner a \urcorner^A = \ulcorner cu \urcorner|_{A,B \multimap A \multimap B}$. Therefore, the justifier of a must lie in $\ulcorner t \urcorner|_{A \multimap C}$.

Therefore, $\mathfrak{s}|_{A,C} \in L_{A \multimap C}$, so $\mathfrak{s}|_{A,C} \in P_{A \multimap C}$.

It is fairly clear that $\sigma; \tau$ is even-prefix closed, since σ and τ are. Indeed, if $\mathfrak{s}|_{A,C} \in \sigma; \tau$ and $t \sqsubseteq \mathfrak{s}|_{A,C}$, then we may choose some prefix \mathfrak{t} of \mathfrak{s} such that $t = \mathfrak{t}|_{A,C}$. Then $\mathfrak{t}|_{A,B} \sqsubseteq \mathfrak{s}|_{A,B} \in \sigma$ and $\mathfrak{t}|_{B,C} \sqsubseteq \mathfrak{s}|_{B,C} \in \tau$, so $\mathfrak{t} \in \sigma \parallel \tau$.

We claim that every sequence in $\sigma; \tau$ has even length. Indeed, if $\mathfrak{s}|_{A,B} \in \sigma$ and $\mathfrak{s}|_{B,C} \in \tau$, then both $\mathfrak{s}|_{A,B}$ and $\mathfrak{s}|_{B,C}$ must have even length, so must be in configuration OO or PP . This means that \mathfrak{s} as a whole must be in configuration OOO or PPP , and so $\mathfrak{s}|_{A,C}$ must be in configuration OO or PP , so must have even length.

Lastly, we need to show that $\sigma; \tau$ is deterministic. Suppose that $sab, sac \in \sigma; \tau$, and suppose that $b \neq c$. Suppose that $\mathfrak{s}|_{A,C} = sab$ and $\mathfrak{t}|_{A,C} = sac$, for $\mathfrak{s}, \mathfrak{t} \in \sigma \parallel \tau$, and let u be the longest common prefix of $\mathfrak{s}, \mathfrak{t}$. \mathfrak{s} and \mathfrak{t} are certainly incomparable under the prefix ordering, since $\mathfrak{s}|_{A,C}$ and $\mathfrak{t}|_{A,C}$ are, so we have $up \sqsubseteq \mathfrak{s}$ and $uq \sqsubseteq \mathfrak{t}$, where $p \neq q$. Now p and q cannot be O -moves in A , P -moves in C or moves in B , or they would have to be equal by determinism of σ and τ . Therefore, they are P -moves in A or O -moves in C , but this contradicts $\mathfrak{s}|_{A,C} = sab$ and $\mathfrak{t}|_{A,C} = sac$.

Therefore, the composition $\sigma; \tau$ is a strategy. \square

We also want to show that the composition of innocent strategies is innocent. We follow the proof given in [Har06]. First, we use a lemma.

Lemma 1.22 ([Har06, 3.3.3]). *Let $\mathfrak{s}a \in \text{int}(A, B, C)$.*

- i) If a is a P -move of A or an O -move of B , then $\ulcorner \mathfrak{s}a \urcorner|_{A,B} = \ulcorner \ulcorner \mathfrak{s}a \urcorner \urcorner|_{A,B}$.*
- ii) If a is a P -move of B or an O -move of C , then $\ulcorner \mathfrak{s}a \urcorner|_{B,C} = \ulcorner \ulcorner \mathfrak{s}a \urcorner \urcorner|_{B,C}$.*

Proof. Induction on the length of \mathfrak{s} . We prove (i); the proof of (ii) is exactly the same.

If a is a P -move of A or an O -move of B , then it is an O -move of $A \multimap B$. If a is an initial move of $A \multimap B$, then we have $\lceil \mathfrak{s} \rceil_{A,B} a^\neg = a = \lceil a \rceil_{A,B} = \lceil \lceil \mathfrak{s} a \rceil \rceil_{A,B}$. Otherwise, write $\mathfrak{s} = \mathfrak{t}bu$, where b justifies a . Then $\lceil \mathfrak{s} a \rceil_{A,B} = \lceil \mathfrak{t} \rceil_{A,B} bu \lceil a \rceil_{A,B} = \lceil \mathfrak{t} \rceil_{A,B} ba$, which by the inductive hypothesis is equal to $\lceil \lceil \mathfrak{t} \rceil_{A,B} ba \rceil$, which is equal to $\lceil \lceil \mathfrak{t} bu a \rceil_{A,B} \rceil = \lceil \lceil \mathfrak{s} a \rceil_{A,B} \rceil$. \square

Proposition 1.23. *If $\sigma: A \multimap B$ and $\tau: B \multimap C$ are innocent strategies, then $\sigma; \tau: A \multimap C$ is innocent.*

Proof. Suppose there are $sab, t \in \sigma; \tau$ such that $ta \in P_{A \multimap C}$, $\lceil sa \rceil = \lceil ta \rceil$. Let $\mathfrak{s}'b$ be such that $\mathfrak{s}'b \lceil a \rceil_{A,C} = sab$ and choose the minimal prefix $\mathfrak{s} \sqsubseteq \mathfrak{s}'$ such that $\mathfrak{s}a \lceil a \rceil_{A,C} = sa$.

Let ta be such that $ta \lceil a \rceil_{A,C} = ta$. Since $\lceil sa \rceil = \lceil ta \rceil$, we have $\lceil \mathfrak{s} a \rceil_{A,C} = \lceil \mathfrak{s} a \rceil_{A,C} = \lceil sa \rceil = \lceil ta \rceil = \lceil \mathfrak{t} a \rceil_{A,C} = \lceil \mathfrak{t} a \rceil_{A,C}$ by Lemma 1.18. Let \mathfrak{u} be the longest common prefix of $\lceil \mathfrak{s} a \rceil$ and $\lceil \mathfrak{t} a \rceil$. If $\mathfrak{s}a$ and $\mathfrak{t}a$ are not equal, then without loss of generality there is some $\mathfrak{u}p \sqsubseteq \mathfrak{s}$, where $\mathfrak{u}p \not\sqsubseteq \mathfrak{t}$. Then, by determinism of σ and τ , this p cannot be a P -move in either $A \multimap B$ or $B \multimap C$, so it must be a P -move in A or an O -move in C , and is therefore preceded by another move in A or C , which contradicts $\lceil \mathfrak{s} a \rceil_{A,C} = \lceil \mathfrak{t} a \rceil_{A,C}$. Therefore, $\lceil \mathfrak{s} a \rceil = \lceil \mathfrak{t} a \rceil$.

Now write $\mathfrak{s}' = sab_1 \cdots b_n b$, where each b_i is a move in B . We show by induction that $tab_1 \cdots b_j \in \sigma \parallel \tau$. Indeed, if $tab_1 \cdots b_{j-1} \in \sigma \parallel \tau$, then b_j (or b) is a P -move in either $A \multimap B$ or $B \multimap C$, and b_{j-1} is an O -move in that same component. Write X for the component ($A \multimap B$ or $B \multimap C$) in which b_j is a P -move. Repeating the argument above, we see that $\lceil \mathfrak{t} ab_1 \cdots b_{j-1} \rceil = \lceil \mathfrak{s} ab_1 \cdots b_{j-1} \rceil$, and so we have that $\lceil \mathfrak{t} ab_1 \cdots b_{j-1} \rceil_X = \lceil \mathfrak{s} ab_1 \cdots b_{j-1} \rceil_X$ by Lemma 1.22. Therefore, by innocence of σ (if $X = A \multimap B$) or τ (if $X = B \multimap C$), we see that $\mathfrak{t} ab_1 \cdots b_j \in \sigma \parallel \tau$. It follows that $\mathfrak{t} ab_1 \cdots b_n b \in \sigma \parallel \tau$, and therefore that $tab \in \sigma; \tau$. \square

1.5 Associativity of composition

In this section, we will prove that composition is associative; i.e., that if $\sigma: A \multimap B$, $\tau: B \multimap C$ and $v: C \multimap D$ are strategies, then $(\sigma; \tau); v = \sigma; (\tau; v)$. To do this, if A, B, C, D are arenas, we define the set $\text{int}(A, B, C, D)$ to be the set of all sequences \mathfrak{u} of moves such that $\mathfrak{u}|_{A,B} \in L_{A \multimap B}$, $\mathfrak{u}|_{B,C} \in L_{B \multimap C}$ and $\mathfrak{u}|_{C,D} \in L_{C \multimap D}$. Given such a sequence \mathfrak{u} , we define $\mathfrak{u}|_{A,D}$ as before; i.e., we take all moves from \mathfrak{u} occurring in A and D , together with justification pointers within these games, and if an initial move in A is justified by an initial move in B , which is justified by an initial move in C , which is justified by an initial move in D , then we add a justification pointer from that move in A to that move in D .

Given strategies σ, τ, v as above, we define $\sigma \parallel \tau \parallel v$ to be the set of all $\mathfrak{u} \in \text{int}(A, B, C, D)$ such that $\mathfrak{u}|_{A,B} \in \sigma$, $\mathfrak{u}|_{B,C} \in \tau$ and $\mathfrak{u}|_{C,D} \in v$. We then claim

that:

Lemma 1.24.

$$(\sigma; \tau); v = \{u|_{A,C} : u \in \sigma \parallel \tau \parallel v\} = \sigma; (\tau; v).$$

Proof. Firstly, if $u \in \sigma \parallel \tau \parallel v$, then it is clear to see that $u|_{A,B,C} \in \sigma \parallel \tau$ and that $u|_{B,C,D} \in \tau \parallel v$, and therefore that $\{u|_{A,C} : u \in \sigma \parallel \tau \parallel v\} \subseteq (\sigma; \tau); v$ and $\{u|_{A,C} : u \in \sigma \parallel \tau \parallel v\} \subseteq \sigma; (\tau; v)$.

Conversely, suppose that $t \in (\sigma; \tau) \parallel v$, so that $t|_{A,C} \in \sigma; \tau$ and $t|_{C,D} \in v$, and choose some $s \in \sigma \parallel \tau$ such that $s|_{A,C} = t|_{A,C}$. We may write

$$s = c_1 b_1 a_1 \cdots c_n b_n a_n$$

for some (possibly empty) sequences of moves a_i from A , b_i from B and c_i from C . We may then write

$$t = d_1 c_1 a_1 \cdots d_n c_n a_n$$

(for the same a_i, c_i), and we can therefore interleave these sequences into the sequence

$$u = d_1 c_1 b_1 a_1 \cdots d_n c_n b_n a_n,$$

which is in $\sigma \parallel \tau \parallel v$. Then we have $u|_{A,D} = t|_{A,D}$, and it follows that $(\sigma; \tau); v \subseteq \{u|_{A,C} : u \in \sigma \parallel \tau \parallel v\}$, and the case for $\sigma; (\tau; v)$ is identical. \square

1.6 Copycat strategies

Definition 1.25. Let A, B be games. Then a *subset inclusion* of A into B is a partial injection $i: M_A \hookrightarrow M_B$ such that

- if i is defined at a and b then $* \vdash_A a$ if and only if $* \vdash_B i(a)$, and $a \vdash_A b$ if and only if $i(a) \vdash_B i(b)$;
- $i(a)$ is defined for every move a occurring in a play in P_A ; and
- $i_*(s) \in P_B$ for every $s \in P_A$.

Here, $i_*(s)$ means the function i applied pointwise to the elements of the string s .

If i is a subset inclusion of A into B , then we get an innocent strategy $\text{subs}_i: B \multimap A$ defined by

$$\text{subs}_i = \{s \in P_{B \multimap A} : \text{for all even-length } t \sqsubseteq s, t|_B = i_*(t|_A)\}.$$

If $P_B = \{i_*(s) : s \in P_A\}$, then we call it a *structural isomorphism*, and we write cc_i ('copycat') for subs_i .

Proposition 1.26. subs_i is an innocent strategy.

Moreover, if $\sigma: C \multimap B$ is a strategy, then

$$\sigma; \text{subs}_i = \{[\text{id}_{M_C}, i^{-1}]_*(s) : s \in \sigma, s|_B \in i_*(P_A)\},$$

where $i^{-1}: M_B \multimap M_A$ is the canonical partial right-inverse to i , and if $\tau: A \multimap D$ is a strategy, then

$$\text{subs}_i; \tau = \{[i, \text{id}_{M_D}]_*(s) : s \in \tau\}.$$

Proof. subs_i is clearly prefix-closed by definition. Suppose that $sab, sac \in \text{subs}_i$; then $s|_A = i_*(s|_B)$ and $sab|_A = i_*(sab|_B)$. It follows that $ab|_A = i_*(ab|_B)$, so either a is a move in A and $b = i(a)$ or a is a move in B and $a = i(b)$. Since the same applies to c , and since i is injective, we have $b = c$.

This argument also shows that subs_i is *history-free* – i.e., that its reply to an O -position is entirely determined by the last O -move – and therefore it is certainly innocent.

Now let $\sigma: C \multimap B$ be a strategy. Suppose that $\mathfrak{s} \in \sigma \parallel \text{subs}_i$. Then $\mathfrak{s}|_{C,B} \in \sigma$ and $\mathfrak{s}|_B = i_*(\mathfrak{s}|_A)$; i.e., $\mathfrak{s}|_{C,B} = [\text{id}_{M_C}, i]_*(\mathfrak{s}|_{C,A})$, and therefore $\mathfrak{s}|_{C,A} = [\text{id}_{M_C}, i]_*(\mathfrak{s}|_{C,B})$, where $\mathfrak{s}|_B \in i_*(P_A)$.

Conversely, given $s \in \sigma$, where $s|_B \in i_*(P_A)$, for each P -move $b = i(a)$ in s occurring in the component B , insert the move a immediately after it, and for each O -move $b' = i(a')$ in s occurring in the component B , insert the move a' immediately before it. Let these extra moves in B be justified according to the original moves in A , and let all initial moves in B be justified by the initial moves in A that occur immediately before them. Then the resulting sequence \mathfrak{s} is contained in $\sigma \parallel \text{subs}_i$, and $\mathfrak{s}|_{A,C} = [\text{id}_{M_C}, i^{-1}]_*(s)$.

The case for composition in the other direction is similar. \square

An easy corollary of this fact is that composition of copycat strategies respects composition of the underlying subset inclusions.

Corollary 1.27. Let i be a subset inclusion from A to B and let j be a subset inclusion from B to C . Then $j \circ i$ is a subset inclusion from A to C and $\text{cc}_{j \circ i} = \text{subs}_j; \text{subs}_i: C \multimap A$.

It is also easy to see from Proposition 1.26 that the identity function $\text{id}: M_A \rightarrow M_A$ is a structural isomorphism from A to itself, and that the resulting copycat strategy cc_{id} is an identity for composition. Combining this with our result for associativity in the previous section, we get that

Theorem 1.28. *The collection of games forms a category \mathcal{G} , where the morphisms $A \rightarrow B$ are strategies for $A \multimap B$, composition is as above and the identity morphisms are the copycat strategies induced from the identity functions on moves.*

In this setting, Proposition 1.26 tells us that a structural isomorphism gives rise to an isomorphism in \mathcal{G} .

Proposition 1.29. *Let f be a structural isomorphism from a game A to a game B . Then cc_f is an isomorphism in \mathcal{G} from A to B .*

Proof. The underlying partial injection $f: M_A \hookrightarrow M_B$ has an inverse partial injection $f^{-1}: M_B \rightarrow M_A$, inducing a structural isomorphism from B to A . Then Proposition 1.26 tells us that cc_f and $\text{cc}_{f^{-1}}$ are inverses in \mathcal{G} . \square

General subset inclusions are not, of course, isomorphisms, but we can still say something category-theoretic about them.

Proposition 1.30. *Let i be a subset inclusion from a game A to a game B . Then the strategy subs_i is an epimorphism from B to A .*

Proof. In fact, it is a split epimorphism: we can define a retract

$$\text{ret}_i = \{s \in P_{A \multimap B} : \text{for all even-length } t \sqsubseteq s, t|_A = i_*(t|_B)\}.$$

The same argument as in Proposition 1.26 tells us that this is indeed a strategy for $A \multimap B$. Note that although subs_i is always a total strategy (i.e., if $s \in \text{subs}_i$ and $sa \in P_{B \multimap A}$, then there is always $sab \in \text{subs}_i$ for some b), the same is not in general true about ret_i .

In any case, if $\mathfrak{s} \in \text{ret}_i \parallel \text{subs}_i$, then $\mathfrak{s}|_{A^L} = i_*(\mathfrak{s}|_B) = \mathfrak{s}|_{A^R}$, and the same is true of any even-length substring of \mathfrak{s} , and so $\mathfrak{s}|_{A,A} \in \text{id}_A$. Conversely, given any $s \in \text{id}_A$, we can form some $\mathfrak{s} \in \text{ret}_i \parallel \text{subs}_i$ such that $\mathfrak{s}|_{A,A} = s$ as in Proposition 1.26.

We can also prove that subs_i is an epimorphism directly, which might be useful, for example, in a setting in which non-total strategies such as ret_i are disallowed. In this setting, let $\sigma, \tau: A \multimap C$ be strategies such that $\text{subs}_i; \sigma = \text{subs}_i; \tau$. Then, by Proposition 1.26, we know that

$$\{[i, \text{id}_{M_C}]_*(s) : s \in \sigma\} = \{[i, \text{id}_{M_C}]_*(s) : s \in \tau\}.$$

Then, since the function $[i, \text{id}_{M_D}]_*: P_{A \multimap C} \rightarrow P_{B \multimap C}$ is an injection, we deduce that $\sigma = \tau$. \square

1.7 Tree Embeddings

We can generalize the subset inclusions of the previous section to *tree embeddings*. Tree embeddings are similar to subset inclusions, but generated by a function between plays, rather than between moves. A consequence of this is that while tree embeddings do give rise to strategies, these strategies are not in general innocent.

Definition 1.31. Let A, B be games. A *tree embedding* from A to B is a length-preserving function $\phi: P_A \hookrightarrow P_B$ such that for all sequences $s, t \in P_A$, if $t \sqsubseteq s$ then $\phi(t) \sqsubseteq \phi(s)$ and such that if $\phi(sb) = \phi(sc)$, where b, c are P -moves in A , then $b = c$.

Given a tree embedding ϕ from A to B , we define a strategy $\text{tree}_\phi: B \multimap A$ by

$$\text{tree}_\phi = \{s \in P_{B \multimap A} : \text{for all even-length } t \sqsubseteq s, t|_B = \phi(t|_A)\}.$$

Example 1.32. If i is a subset inclusion from A to B , then i_* is a tree embedding from A to B .

Proposition 1.33. tree_ϕ is a strategy.

If ϕ is a tree embedding from A to B and ψ is a tree embedding from B to C , then $\psi \circ \phi$ is a tree embedding from A to C and $\text{tree}_{\psi \circ \phi} = \text{tree}_\psi; \text{tree}_\phi$.

Proof. tree_ϕ is a prefix-closed subset of $P_{B \multimap A}$ by definition. If $sab, sac \in \text{tree}_\phi$, then we have $s|_B = \phi(s|_A)$ and $sab|_B = \phi(sab|_A)$. Since ϕ is length-preserving, $s|_A$ and $s|_B$ must have the same length, and the same is true of $sab|_A$ and $sab|_B$. Therefore, either a is a move in A and $s|_B b = \phi(s|_A a)$ or a is a move in B and $s|_B a = \phi(s|_A b)$. The same applies to c : so either $s|_B b = \phi(s|_A a) = s|_B c$ or $\phi(s|_A b) = s|_B a = \phi(s|_A c)$. In either case, we have $b = c$.

Now suppose that ϕ is a tree embedding from A to B and that ψ is a tree embedding from B to C . Then $\psi \circ \phi$ certainly preserves length and the prefix relation. Let $\mathfrak{s} \in \text{tree}_\psi \parallel \text{tree}_\phi$, and let $t \sqsubseteq \mathfrak{s}|_{C,A}$ have even length. Let \mathfrak{t} be a prefix of \mathfrak{s} such that $\mathfrak{t}|_{C,A} = t$. Then, by the argument above, $\mathfrak{t}|_{C,B}$ and $\mathfrak{t}|_{B,A}$ have even length, and so $\mathfrak{t}|_C = \psi(\mathfrak{t}|_B)$ and $\mathfrak{t}|_B = \phi(\mathfrak{t}|_A)$. Therefore, $\mathfrak{t}|_C = (\psi \circ \phi)(\mathfrak{t}|_A)$. Since t was arbitrary, this means that $\mathfrak{s}|_{A,C} \in \text{tree}_{\psi \circ \phi}$.

Conversely, suppose that $s \in \text{tree}_{\psi \circ \phi}$: so $t|_C = \psi(\phi(t|_A))$ for all even-length $t \sqsubseteq s$. By induction on the length of s , we build up a sequence $\mathfrak{s}(s) \in \text{tree}_\psi \parallel \text{tree}_\phi$ such that $\mathfrak{s}(s)|_{C,A} = s$.

- $\mathfrak{s}(\epsilon) = \epsilon$.
- Given a sequence $sab \in \text{tree}_{\psi \circ \phi}$, we have $s|_C = \psi(\phi(s|_A))$. By our discussion above, we know that $sab|_A$ is one move longer than $s|_A$, so $\phi(sab|_A) = \phi(s|_A)c$ for some move c in B , justified by some move in B . Then we set $\mathfrak{s}(sab) = \mathfrak{s}(s)acb$, where the justifier of c is as above.

It is clearly true that $\mathfrak{s}(s)|_{C,A} = s$ for all sequences $s \in \text{tree}_{\psi \circ \phi}$.

We claim that $\mathfrak{s}(s) \in \text{tree}_\psi \parallel \text{tree}_\phi$ for all s . This is clearly true for the empty sequence; for sab , we have $\mathfrak{s}(sab)|_{B,A} = \mathfrak{s}(s)acb|_{B,A}$, where $\phi(sab|_A) = \phi(s|_A)c$.

By induction, $\mathfrak{s}(s) \in \psi(\phi(s|_A))$, and so for all even-length $t \sqsubseteq \mathfrak{s}(s)|_{B,A}$ we have $t|_B = \phi(t|_A)$, and for all even-length $u \sqsubseteq \mathfrak{s}(s)|_{C,B}$ we have $u|_C = \psi(u|_B)$. All we need to show is that $\mathfrak{s}(s)acb|_B = \phi(\mathfrak{s}(s)acb|_A)$ and that $\mathfrak{s}(s)acb|_C = \psi(\mathfrak{s}(s)acb|_B)$. Indeed, $\phi(\mathfrak{s}(s)acb|_A) = \phi(sab|_A) = \phi(s|_A)c = \phi(\mathfrak{s}(s)|_A)c = \mathfrak{s}(s)|_{Bc} = \mathfrak{s}(s)acb|_V$, and, in addition, we have that $\psi(\mathfrak{s}(s)acb|_B) = \psi(\mathfrak{s}(s)|_{Bc}) = \psi(\phi(\mathfrak{s}(s)|_A)c) = \psi(\phi(sab|_A)) = sab|_C = \mathfrak{s}(s)acb|_C$. \square

Definition 1.34. We say that a tree embedding ϕ is a *tree isomorphism* if it is a bijection.

Proposition 1.35. *If ϕ is a tree isomorphism from a game A to a game B , then tree_ϕ is an isomorphism in \mathcal{G} .*

Proof. If ϕ is a tree isomorphism, then its inverse ϕ^{-1} is also a tree isomorphism, and Proposition 1.33 tells us that tree_ϕ and $\text{tree}_{\phi^{-1}}$ are inverses in \mathcal{G} . \square

1.8 \mathcal{G} as a Symmetric Monoidal Category

We now claim that the tensor product connective \otimes makes \mathcal{G} into a symmetric monoidal closed category, with internal hom given by \multimap .

Definition 1.36. Let $\sigma: A \multimap B$ and $\tau: C \multimap D$ be strategies. We define a strategy $\sigma \otimes \tau: (A \otimes C) \multimap (B \otimes D)$ by

$$\sigma \otimes \tau = \{s \in P_{(A \otimes C) \multimap (B \otimes D)} : s|_{A,B} \in \sigma \text{ and } s|_{C,D} \in \tau\}.$$

To prove that this is a strategy, we prove a lemma analogous to our Lemma 1.15.

Lemma 1.37. *Let $s \in P_{A \otimes B}$. Then $\lambda_{A \otimes B}^{OP}(s) = \lambda_A^{OP}(s|_A) \wedge \lambda_B^{OP}(s|_B)$, where \wedge is the binary operator on $\{O, P\}$ given by*

p	q	$p \wedge q$
P	P	P
O	P	O
P	O	O
O	O	O

Moreover, either $\lambda_A^{OP}(s|_A) = P$ or $\lambda_B^{OP}(s|_B) = P$.

Proof. Mutual induction on the length of s . This is obvious if s is empty. Suppose that $sa \in P_{A \otimes B}$, where a is an O -move. By induction, since $\lambda_{A \otimes B}(s) = P$, we must have $\lambda_{A \otimes B}(s|_A) = P$ and $\lambda_{A \otimes B}(s|_B) = P$. Therefore, depending on which game a is played in, either $\lambda_A(sa|_A) = O$ and $\lambda_B(sa|_B) = P$ or $\lambda_A(sa|_A) = P$ and $\lambda_B(sa|_B) = O$.

If $sb \in P_{A \otimes B}$, where b is a P -move, then by induction either $\lambda_A(s|_A) = O$ and $\lambda_B(s|_B) = P$ or $\lambda_A(s|_A) = P$ and $\lambda_B(s|_B) = O$. In either case, player P must play in whichever game is currently in an O -position, returning us to configuration PP . \square

The above proof gives us the following result analogous to Corollary 1.16.

Corollary 1.38 (Switching condition for \otimes). *Player O switches games in $A \otimes B$; i.e., if $sab \in P_{A \otimes B}$, where a and b take place in different games (i.e., a in A and b in B or a in B and b in A), then b is an O -move.*

Proposition 1.39. $\sigma \otimes \tau$ is a strategy for $(A \otimes C) \multimap (B \otimes D)$.

Proof. $\sigma \otimes \tau$ is certainly an even-prefix-closed subset of $P_{(A \otimes C) \multimap (B \otimes D)}^{\text{even}}$.

Let s be a play of $P_{(A \otimes B) \multimap (C \otimes D)}$. We consider the possible configurations of s ; i.e., the tuples $(\lambda_A(s|_A), \lambda_B(s|_B), \lambda_C(s|_C), \lambda_D(s|_D))$.

By Lemma 1.15 we must avoid the overall configuration OP for the linear implication, and by Lemma 1.37 we must avoid the configuration OO inside either tensor product, so we end up with the following possibilities.

$\lambda_A(s _A)$	$\lambda_C(s _C)$	$\lambda_B(s _B)$	$\lambda_D(s _D)$	$\lambda_{A \otimes C}(s _{A,C})$	$\lambda_{B \otimes D}(s _{B,D})$	$\lambda_{(A \otimes C) \multimap (B \otimes D)}(s)$
P	P	P	P	P	P	P
P	P	P	O	P	O	O
P	P	O	P	P	O	O
P	O	P	O	O	O	P
O	P	O	P	O	O	P
P	O	O	P	O	O	P
O	P	P	O	O	O	P

Now, if $s \in \sigma \otimes \tau$, or an odd-length sequence formed by adding an O -move to the end of a sequence in $\sigma \otimes \tau$, then we also know that $s|_{A,B} \in \sigma \subseteq P_{A \multimap B}$ and that $s|_{C,D} \in \tau \subseteq P_{C \multimap D}$. This means that we can discount the last two configurations in the table above, since one contains the illegal configuration OP in $C \multimap D$ and the other contains the illegal configuration OP in $A \multimap B$.

Now suppose that $sab, sac \in \sigma \otimes \tau$. Then sa is an O -position in $P_{(A \otimes C) \multimap (B \otimes D)}$, and is therefore in configuration $PPPO$ or $PPOP$. By inspecting the table above, we see that if sa is in configuration $PPPO$, then b and c must both occur either in C or in D , and that if sa is in configuration $PPOP$, then b and c must both occur either in A or in B . In either case, we must have $b = c$, by determinism of τ (in the first case) or of σ (in the second case). \square

We need a lemma to prove that the tensor product of two innocent strategies is innocent.

Lemma 1.40. *Let $s \in \sigma \otimes \tau$.*

i) If s ends with a move in A or B , then $\ulcorner s \urcorner^{(A \otimes C) \multimap (B \otimes D)} = \ulcorner s|_{A,B} \urcorner^{A \multimap B}$.

ii) If s ends with a move in C or D , then $\ulcorner s \urcorner^{(A \otimes C) \multimap (B \otimes D)} = \ulcorner s|_{C,D} \urcorner^{C \multimap D}$.

Proof. Induction on the length of s . We prove (i); (ii) is exactly the same.

If a is a P -move, then we have $\ulcorner sa \urcorner = \ulcorner s \urcorner a$. By our analysis in the proof of Proposition 1.39, player P only switches moves between A and B , and between C and D , so s must end with a move from A or B . Therefore, by the inductive hypothesis, $\ulcorner s \urcorner = \ulcorner s|_{A,B} \urcorner$. Then $\ulcorner sa \urcorner = \ulcorner s \urcorner a = \ulcorner s|_{A,B} \urcorner a = \ulcorner sa|_{A,B} \urcorner$.

If a is an initial move, then $\ulcorner sa \urcorner = a = \ulcorner sa|_{A,B} \urcorner$.

If a is an O -move justified by b in $sbta$, then $\ulcorner sbta \urcorner = \ulcorner s \urcorner ba$. Then b is a P -move, so s must end with a move in A or B , as before. Therefore, by the inductive hypothesis, $\ulcorner s \urcorner = \ulcorner s|_{A,B} \urcorner$. Then $\ulcorner sbta \urcorner = \ulcorner s \urcorner ba = \ulcorner s|_{A,B} \urcorner ba = \ulcorner sbta|_{A,B} \urcorner$. \square

Proposition 1.41. *Let $\sigma: A \rightarrow B$, $\tau: C \rightarrow D$ be innocent strategies. Then $\sigma \otimes \tau$ is innocent.*

Proof. Suppose $sab, t \in \sigma \otimes \tau$ such that $ta \in P_{(A \otimes C) \multimap (B \otimes D)}$ and $\ulcorner sa \urcorner = \ulcorner ta \urcorner$. Suppose without loss of generality that a is a move in A or B . Then $\ulcorner sa \urcorner = \ulcorner sa|_{A,B} \urcorner$ and $\ulcorner ta \urcorner = \ulcorner ta|_{A,B} \urcorner$ by Lemma 1.40, and therefore $tab|_{A,B} \in \sigma$ by innocence of σ , and so $tab \in \sigma \otimes \tau$. \square

The most important thing we need to prove is that \otimes is a functor.

Proposition 1.42. *Let $\sigma': A'' \multimap A'$, $\sigma: A' \multimap A$, $\tau': B'' \multimap B'$ and $\tau: B' \multimap B$ be strategies. Then $(\sigma' \otimes \tau'); (\sigma \otimes \tau) = (\sigma'; \sigma) \otimes (\tau'; \tau)$.*

Moreover, if A', A, B', B are games, i is a subset inclusion from A to A' and j is a structural isomorphism from B to B' , then $\text{subs}_i \otimes \text{subs}_j = \text{subs}_{[i,j]}$. In particular, if A and B are games, then $\text{id}_A \otimes \text{id}_B = \text{id}_{A \otimes B}$.

Proof. First suppose that $s \in (\sigma' \otimes \tau'); (\sigma \otimes \tau)$; so $s = \mathfrak{s}|_{A'', B'', A, B}$, where $\mathfrak{s} \in (\sigma' \otimes \tau') \| (\sigma \otimes \tau)$. Then $\mathfrak{s}|_{A'', A'} \in \sigma'$ and $\mathfrak{s}|_{A', A} \in \sigma$, so $\mathfrak{s}|_{A'', A', A} \in \sigma' \| \sigma$ and therefore $\mathfrak{s}|_{A'', A} = \mathfrak{s}|_{A'', A} \in \sigma'; \sigma$. Similarly, $\mathfrak{s}|_{B'', B} \in \tau'; \tau$, and therefore $s \in (\sigma'; \sigma) \otimes (\tau'; \tau)$.

Conversely, suppose that $s \in (\sigma'; \sigma) \otimes (\tau'; \tau)$. Choose some $\mathfrak{s} \in \sigma' \| \sigma$, $\mathfrak{t} \in \tau' \| \tau$ such that $\mathfrak{s}|_{A'', A} = \mathfrak{s}|_{A'', A}$ and $\mathfrak{s}|_{B'', B} = \mathfrak{s}|_{B'', B}$. By our analysis, the only time we switch from the A'' , A -component to the B'' , B component in s , or *vice versa*, is when player O switches between the games A and B . Thus, we may divide s up into blocks, each starting and ending with a move in the outer component

$A \otimes B$. This then gives us a way to divide up \mathfrak{s} and \mathfrak{t} into blocks, such that each block of \mathfrak{s} or \mathfrak{t} projects on to a block of s . Lastly, we can string these blocks together to give us some $u \in (\sigma' \otimes \tau') \parallel (\sigma \otimes \tau)$ such that $u|_{A'', B'', A, B} = s$.

For the second part, let A', A, B', B be games, let i be a structural isomorphism from A to A' and let j be a structural isomorphism from B to B' . Suppose that $s \in \text{subs}_i \otimes \text{subs}_j$. Then $s|_{A', A} \in \text{subs}_i$ and $s|_{B', B} \in \text{subs}_j$ – so if $u \sqsubseteq s|_{A, A}$ has even length, then $u|_{A'} = i_*(u|_A)$, and if $v \sqsubseteq s|_{B, B}$ has even length, then $v|_{B'} = j_*(v|_B)$. Suppose that $t \sqsubseteq s$ is of even length. Then, since only player O switches between the A', A -component and the B', B -component, both $t|_{A', A}$ and $t|_{B', B}$ are of even length, it follows that $t|_{A', B'} = [i, j]_*(t|_{A, B})$. Since t was arbitrary, this means that $s \in \text{subs}_{[i, j]}$.

Conversely, suppose that $s \in \text{subs}_{[i, j]}$. Then for all even-length $t \sqsubseteq s$, $t|_{A'} = i_*(t|_A)$ and $t|_{B'} = j_*(t|_B)$. Since any play in σ or in τ is itself a play of $\sigma \otimes \tau$, then if $u \sqsubseteq s|_{A', A}$ has even length, then $u|_{A'} = i_*(u|_A)$, and if $v \sqsubseteq s|_{B', B}$, then $v|_{B'} = j_*(v|_B)$. It follows that $s|_{A', A} \in \text{subs}_i$ and $s|_{B', B} \in \text{subs}_j$, and therefore that $s \in \text{subs}_i \otimes \text{subs}_j$. \square

Now it is fairly clear that if A, B, C are games, then we have structural isomorphisms

$$\begin{aligned} (A \otimes B) \otimes C &\cong A \otimes (B \otimes C) \\ A &\cong A \otimes I & A &\cong I \otimes A \\ A \otimes B &\cong B \otimes A, \end{aligned}$$

induced by the associators, unitors and symmetry of the category of sets with coproduct. We claim that these are natural transformations.

Proposition 1.43. *The families of morphisms*

$$\begin{aligned} \text{cc}_{\text{assoc}}_{M_A, M_B, M_C} &: (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C) \\ \text{cc}_{\text{lunit}}_{M_A} &: A \rightarrow I \otimes A & \text{cc}_{\text{runit}}_{M_A} &: A \rightarrow A \otimes I \\ \text{cc}_{\text{sym}}_{M_A, M_B} &: A \otimes B \rightarrow B \otimes A \end{aligned}$$

are natural transformations in \mathcal{G} .

Proof. We prove this for the associator; the other cases are similar.

Let $\sigma: A' \multimap A$, $\tau: B' \multimap B$, $v: C' \multimap C$ be strategies. By Proposition 1.26, we

have

$$\begin{aligned}
& ((\sigma \otimes \tau) \otimes v); \text{cc}_{\text{assoc}_{M_A, M_B, M_C}} \\
&= \{ [\text{id}_{M_{(A' \otimes B') \otimes C'}}, \text{assoc}_{M_A, M_B, M_C}]_*(s) : s \in (\sigma \otimes \tau) \otimes v \} \\
&= \left\{ [\text{id}_{M_{(A' \otimes B') \otimes C'}}, \text{assoc}_{M_A, M_B, M_C}]_*(s) \left| \begin{array}{l} s \in P_{((A' \otimes B') \otimes C') \multimap ((A \otimes B) \otimes C)} \\ s|_{A', A} \in \sigma, s|_{B', B} \in \tau, s|_{C', C} \in v \end{array} \right. \right\} \\
&= \{ s \in P_{((A' \otimes B') \otimes C') \multimap (A \otimes (B \otimes C))} : s|_{A', A} \in \sigma, s|_{B', B} \in \tau, s|_{C', C} \in v \} \\
&= \left\{ [\text{assoc}_{M_{A'}, M_{B'}, M_{C'}}, \text{id}_{M_{A \otimes (B \otimes C)}}]_*(s) \left| \begin{array}{l} s \in P_{(A' \otimes (B' \otimes C')) \multimap (A \otimes (B \otimes C))} \\ s|_{A', A} \in \sigma, s|_{B', B} \in \tau, s|_{C', C} \in v \end{array} \right. \right\} \\
&= \{ [\text{assoc}_{M_{A'}, M_{B'}, M_{C'}}, \text{id}_{M_{A \otimes (B \otimes C)}}]_*(s) : s \in \sigma \otimes (\tau \otimes v) \} \\
&= \text{cc}_{\text{assoc}_{M_{A'}, M_{B'}, M_{C'}}} ; (\sigma \otimes (\tau \otimes v)). \quad \square
\end{aligned}$$

Then, by Proposition 1.26 again, these natural transformations satisfy the same coherence diagrams (pentagon, triangles, hexagon etc.) satisfied by the original associators, unitors and symmetry in $(\mathbf{Set}, +)$.

It follows that \otimes makes \mathcal{G} into a symmetric monoidal category.

1.9 \mathcal{G} as a Symmetric Monoidal Closed Category

Definition 1.44. Let A, B, C, D be games, let σ be a strategy for $A \multimap B$ and let τ be a strategy for $C \multimap D$. Then we define a strategy $\sigma \multimap \tau : (B \multimap C) \multimap (A \multimap D)$ by

$$\sigma \multimap \tau = \{ s \in P_{(B \multimap C) \multimap (A \multimap D)} : s|_{A, B} \in \sigma, s|_{C, D} \in \tau \}.$$

Proposition 1.45. $\sigma \multimap \tau$ is a strategy for $(B \multimap C) \multimap (A \multimap D)$.

Proof. $\sigma \multimap \tau$ is certainly a prefix-closed subset of $P_{(B \multimap C) \multimap (A \multimap D)}^{\text{even}}$.

We examine the sign configuration of a play in $(B \multimap C) \multimap (A \multimap D)$, using Lemma 1.15. Since we must avoid the configuration OP in either $B \multimap C$, $A \multimap D$ or in $(B \multimap C) \multimap (A \multimap D)$, we arrive at the following list of possibilities.

$\lambda_B^{OP}(s _B)$	$\lambda_C^{OP}(s _C)$	$\lambda_A^{OP}(s _A)$	$\lambda_D^{OP}(s _D)$	$\lambda_{B \multimap C}^{OP}(s _{B, C})$	$\lambda_{A \multimap D}^{OP}(s _{A, D})$	$\lambda_{(B \multimap C) \multimap (A \multimap D)}^{OP}(s)$
P	O	P	O	O	O	P
P	P	P	O	P	O	O
O	O	P	O	P	O	O
P	P	P	P	P	P	P
O	O	O	O	P	P	P
P	P	O	O	P	P	P
O	O	P	P	P	P	P

If $s \in \sigma \multimap \tau$, then we can immediately discount the last two of these possibilities, since one includes the illegal configuration OP in $A \multimap B$, and the other includes the illegal configuration OP in $B \multimap D$.

By examining the remaining possibilities, we arrive at the conclusion that any O -position in configuration $PPPO$ constrains player P to play in C (to reach configuration $POPO$) or to play in D (to reach configuration $PPPP$), and that any O -position in configuration $OOPO$ constrains player P to play in A (to reach configuration $OOOO$) or to play in C (to reach configuration $POPO$).

Now suppose that $sab, sac \in \sigma \multimap \tau$. Then, by our above analysis, b and c must either both take place in the B, A -component, in which case $b = c$ by determinism of σ , or both in the C, D -component, in which case $b = c$ by determinism of τ . \square

To prove that $\sigma \multimap \tau$ is innocent if σ and τ are, we need a lemma analogous to Lemma 1.40.

Lemma 1.46. *Let $s \in \sigma \multimap \tau$.*

- i) *If s ends with a move in A or B , then $\ulcorner s \urcorner^{(B \multimap C) \multimap (A \multimap D)} = \ulcorner s|_{A,B} \urcorner^{A \multimap B}$.*
- ii) *If s ends with a move in C or D , then $\ulcorner s \urcorner^{(B \multimap C) \multimap (A \multimap D)} = \ulcorner s|_{C,D} \urcorner^{C \multimap D}$.*

Proof. Exactly the same as in Lemma 1.40, using the analysis from the proof of Proposition 1.45 to show that player P only switches moves between A and B , and between C and D , in $\sigma \multimap \tau$. \square

Proposition 1.47. *Let $\sigma: A \multimap B$, $\tau: C \multimap D$ be innocent strategies. Then $\sigma \multimap \tau$ is innocent.*

Proof. Suppose $sab, t \in \sigma \multimap \tau$ such that $ta \in P_{(B \multimap C) \multimap (A \multimap D)}$ and $\ulcorner sa \urcorner = \ulcorner ta \urcorner$. Suppose without loss of generality that a is a move in A or B . Then $\ulcorner sa \urcorner = \ulcorner sa|_{A,B} \urcorner$ and $\ulcorner ta \urcorner = \ulcorner ta|_{A,B} \urcorner$ by Lemma 1.46, and therefore $tab|_{A,B} \in \sigma$ by innocence of σ , and so $tab \in \sigma \multimap \tau$. \square

We now need to prove that \multimap is a functor $\mathcal{G}^{\text{op}} \times \mathcal{G} \rightarrow \mathcal{G}$.

Proposition 1.48. *Let $\sigma': A'' \multimap A'$, $\sigma: A' \multimap A$, $\tau': B'' \multimap B'$ and $\tau: B' \multimap B$ be strategies. Then $(\sigma \multimap \tau'); (\sigma' \multimap \tau) = (\sigma'; \sigma) \multimap (\tau'; \tau)$.*

Moreover, if A', A, B', B are games, f is a structural isomorphism from A' to A and g is a structural isomorphism from B' to B , then $\text{cc}_f \multimap \text{cc}_g = \text{cc}_{[f^{-1}, g]}$. In particular, if A, B are games then $\text{id}_A \multimap \text{id}_B = \text{id}_{A \multimap B}$.

Proof. As in Proposition 1.42. \square

Now it is easy to see that the associator $\text{assoc}_{M_A, M_B, M_C}$ is a structural isomorphism from $(A \otimes B) \multimap C$ to $A \multimap (B \multimap C)$, so it induces a copycat isomorphism $\Lambda_{A,B,C} = \text{cc}_{\text{assoc}_{M_A, M_B, M_C}}: (A \otimes B) \multimap C \rightarrow A \multimap (B \multimap C)$.

Proposition 1.49. $\Lambda_{A,B,C}$ is natural in A, B, C .

Proof. The same argument as in Proposition 1.43. \square

We have proved the following.

Theorem 1.50. \mathcal{G} is a symmetric monoidal closed category, with tensor product given by \otimes and internal hom given by \multimap .

1.10 Products in \mathcal{G}

Proposition 1.51. Given some family A_i of games, the game $\prod_i A_i$, as defined in Definition 1.8, is the category-theoretic product of the A_i .

Proof. We have natural injections $\text{in}_j : M_{A_j} \hookrightarrow M_{\prod_i A_i}$ giving rise to subset inclusions. Then our projections are given by the morphisms

$$\text{pr}_j := \text{subs}_{\text{in}_j} : \prod_i A_i \rightarrow A_j.$$

Now suppose we have some game B , and strategies $\sigma_i : B \multimap A_i$ for each i . Define a strategy

$$\langle \sigma_i \rangle = \bigcup_i [\text{id}_{M_B}, \text{in}_i]_*(\sigma_i).$$

We claim that this is indeed a strategy for $B \multimap \prod_i A_i$. Indeed, it is certainly a prefix-closed subset of $P_{B \multimap \prod_i A_i}$.

Moreover, if $sab, sac \in \langle \sigma_i \rangle$, then there is some unique j such that a comes from a move in A_j , and therefore sab, sac are both plays in σ_j , so $b = c$.

Next, we claim that $\langle \sigma_i \rangle; \text{pr}_j = \sigma_j$. Indeed, we have

$$\begin{aligned} \langle \sigma_i \rangle; \text{pr}_j &= \langle \sigma_i \rangle; \text{subs}_{\text{in}_j} \\ &= \{[\text{id}_{M_B}, \text{in}_j^{-1}]_*(s) : s \in \langle \sigma_i \rangle, s|_{\prod_i A_i} \in (\text{in}_j)_*(P_{A_j})\} \quad \text{Prop. 1.26} \\ &= \sigma_j. \end{aligned}$$

Lastly, suppose $\tau : B \multimap \prod_i A_i$ is a strategy such that $\tau; \text{pr}_j = \sigma_j$ for each j . We claim that $\tau = \langle \sigma_i \rangle$. Indeed, by the argument above, we must have

$$\{[\text{id}_{M_B}, \text{in}_j^{-1}]_*(s) : s \in \tau, s|_{\prod_i A_i} \in (\text{in}_j)_*(P_{A_j})\} = \sigma_j$$

for each j . Suppose that $s \in \tau$. Then $s|_{\prod_i A_i} \in (\text{in}_j)_*(P_{A_j})$ for some j , by the definition of $\prod_i A_i$. Therefore, $s \in [\text{id}_{M_B}, \text{in}_j]_*(\sigma_j)$.

Conversely, let $t \in \sigma_j$. By the above equation, we know that there is some $s \in \tau$ such that $s|_{\prod_i A_i} \in (\text{in}_j)_*(P_{A_j})$ and $[\text{id}_{M_B}, \text{in}_j^{-1}]_*(s) = t$. It follows that $[\text{id}_{M_B}, \text{in}_j]_*(t) = s \in \tau$. \square

An examination of the definitions tells us that

Proposition 1.52. *Let A_i, B be games and let ϕ_i be tree embeddings from A_i to B . Then $\langle \text{tree}_{\phi_i} \rangle = \text{tree}_\phi$, where ϕ is the tree embeddings from $\prod_i A_i$ to B given by*

$$\phi(s) = \begin{cases} \epsilon & \text{if } s = \epsilon \\ \phi_i(s|_{A_i}) & \text{if } s \text{ starts with a move from } A_i \end{cases}$$

Proof. The only thing we really need to check is that this is indeed a tree embedding. Let sb, sc be positions in $\prod_i A_i$, where b, c are P -moves. Then sb, sc must start with the same move, so if $\phi(sb) = \phi(sc)$ then we have $\phi_i(sb) = \phi_i(sc)$ for some i and therefore $b = c$. \square

Note that $\langle \sigma_i \rangle$ is not in general innocent, even if all the σ_i are, and there is no version of Proposition 1.52 that works for subset inclusion strategies. Of course, since a subset inclusion is a special case of a tree embedding, then $\langle \text{subs}_i \rangle$ is always a tree embedding strategy.

1.11 Sequoidal categories

We have now given the category-theoretic properties of all the connectives from Section 1.3, with the exception of the sequoid \otimes and the exponential $!$.

We would like to say that $_ \otimes _$ is a functor from $\mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$, as is the case with the tensor product $_ \otimes _$. However, this does not quite work: given strategies $\sigma: A \multimap B$ and $\tau: C \multimap D$, we may not get a well-formed strategy for $(A \otimes C) \multimap (B \otimes D)$ by ‘playing according to σ in A and B and according to τ in C and D ’. The reason is that the constraint that player O plays in B before D is not strong enough to force player P to play in A before C ; indeed, suppose that σ tells player P to respond to an initial move in B with another move in B . Suppose that player O then decides to make a move in D . If τ tells player P to respond to this move in D with a move in C , then she will be stuck, unable to play this move because no move has yet been played in A .

We can fix this problem by imposing some constraints on the strategies σ and τ . The problem occurs when player O ’s initial move in B is not reflected by an initial move by player P in A ; therefore, if σ is such that player P always responds to the initial move in B with a move in A , then we can form a strategy $\sigma \otimes \tau$ for $(A \otimes C) \multimap (B \otimes D)$. Moreover, this strategy $\sigma \otimes \tau$ will inherit this property that the first move on the right is always replied to by a move on the left.

Definition 1.53. Let A, B be games. A *strict morphism* from A to B is a strategy σ for $A \multimap B$ such that any player P response to an opening move in B is a move in A ; i.e., such that if b is an initial O -move in B and $ba \in \sigma$, then a is a move in A .

We will call such a σ a *strict strategy* for $A \multimap B$, although this is a slight abuse of language, since the definition depends on the constituent games A and B , which may not be recoverable from $A \multimap B$.

It is clear that the composition of strict morphisms is again a strict morphism, as is any morphism of the form subs_i , and so we get a wide subcategory \mathcal{G}_s of \mathcal{G} whose objects are games and where the morphisms are the strict strategies. We then have a natural inclusion functor $J: \mathcal{G}_s \rightarrow \mathcal{G}$.

Definition 1.54. Given games A, B, C, D , a strict morphism $\sigma: A \multimap B$ and a strategy $\tau: C \multimap D$, we define a strict morphism $\sigma \otimes \tau: (A \otimes C) \multimap (B \otimes D)$ by

$$\sigma \otimes \tau = \{s \in P_{(A \otimes C) \multimap (B \otimes D)} : s|_{A,B} \in \sigma, s|_{C,D} \in \tau\}.$$

Proposition 1.55. $\sigma \otimes \tau$ is a strategy.

Proof. $\sigma \otimes \tau$ is certainly a prefix-closed subset of $P_{(A \otimes C) \multimap (B \otimes D)}$. Moreover, if $sab, sac \in \sigma \otimes \tau$, then $sab, sac \in \sigma \otimes \tau$, so $b = c$. \square

Of course, $P_{A \otimes B}$ is a subset of $P_{A \otimes B}$, which means that the identity function $M_A + M_B \rightarrow M_A + M_B$ gives us a subset inclusion from $A \otimes B$ to $A \otimes B$, and hence a strategy $\text{subs}_{\text{id}_{M_A + M_B}}$ for $A \otimes B \multimap A \otimes B$, which we shall refer to as $\text{wk}_{A,B}$.

Proposition 1.56. Let A, B, C, D be games, let $\sigma: A \multimap B$ be a strict strategy and let $\tau: C \multimap D$ be a strategy. Then the following diagram commutes.

$$\begin{array}{ccc} A \otimes C & \xrightarrow{\sigma \otimes \tau} & B \otimes D \\ \text{wk}_{A,C} \downarrow & & \downarrow \text{wk}_{B,D} \\ A \otimes C & \xrightarrow{\sigma \otimes \tau} & B \otimes D \end{array}$$

Proof. By Proposition 1.26 and the definition of wk , we know that

$$\begin{aligned} \sigma \otimes \tau; \text{wk}_{B,D} &= \{s \in \sigma \otimes \tau, s|_{B,D} \in P_{B \otimes D}\} \\ \text{wk}_{A,C}; \sigma \otimes \tau &= \sigma \otimes \tau, \end{aligned}$$

as sets of plays.

Now we know that $\sigma \otimes \tau = \{s \in \sigma \otimes \tau : s|_{A \otimes C} \in P_{(A \otimes C) \multimap (B \otimes D)}\}$, so it suffices to show that if $s \in \sigma \otimes \tau$ is such that $s|_{B,D} \in P_{B \otimes D}$ then $s \in P_{(A \otimes C) \multimap (B \otimes D)}$.

Indeed, if $s|_{B,D} \in P_{B \otimes D}$ then s begins with an initial O -move in B . Then, since σ is strict, the next move in s must be a move in A , and therefore $s|_{A,C}$ begins with a move in A . Since we also have $s|_{A,C} \in P_{A \otimes C}$, we must have that $s|_{A,C} \in P_{A \otimes C}$. \square

Remark 1.57. This is the main place where we have used the assumption that σ is a strict strategy: if we drop the strictness requirement from Definition 1.54, then we get a valid (if nonsensical) strategy that has a partiality (‘gives up’) if playing according to σ and τ would lead to it creating an invalid play. But such a strategy would not satisfy the conclusion of Proposition 1.56, since $\sigma \otimes \tau; \text{wk}_{B,D}$ would contain these extra plays where $\text{wk}_{A,C}; \sigma \otimes \tau$ had ‘given up’.

Remark 1.58. Of course, we would *like* to restate Proposition 1.56 by saying that wk is some sort of natural transformation, but that doesn’t make sense until we’ve shown that $_ \otimes _$ is a functor.

Proposition 1.59. *If we have strict strategies $\sigma': A'' \multimap A'$ and $\sigma: A' \multimap A$, and strategies $\tau': B'' \multimap B'$ and $\tau: B' \multimap B$, then we have*

$$(\sigma' \otimes \tau'); (\sigma \otimes \tau) = (\sigma'; \sigma) \otimes (\tau'; \tau).$$

If A', A, B', B are games, i is a subset inclusion from A into A' and j is a subset inclusion from B into B' , then

$$\text{subs}_i \otimes \text{subs}_j = \text{subs}_{[i,j]}: A' \otimes B' \multimap A \otimes B.$$

In particular, if A, B are games, then $\text{id}_A \otimes \text{id}_B = \text{id}_{A \otimes B}$.

Proof. Let A'', A', A, B'', B', B and $\sigma', \sigma, \tau', \tau$ be as above.

We have

$$\begin{aligned} \text{wk}_{A'',B''}; (\sigma' \otimes \tau'); (\sigma \otimes \tau) &= (\sigma' \otimes \tau'); \text{wk}_{A',B'}; (\sigma \otimes \tau) && \text{Prop. 1.56} \\ &= (\sigma' \otimes \tau'); (\sigma \otimes \tau); \text{wk}_{A,B} && \text{Prop. 1.56} \\ &= ((\sigma'; \sigma) \otimes (\tau'; \tau)); \text{wk}_{A,B} && \text{Prop. 1.42} \\ &= \text{wk}_{A'',B''}; ((\sigma'; \sigma) \otimes (\tau'; \tau)). && \text{Prop. 1.56} \end{aligned}$$

By Proposition 1.30, $\text{wk}_{A'',B''}$ is an epimorphism, and therefore we have that

$$(\sigma' \otimes \tau'); (\sigma \otimes \tau) = (\sigma'; \sigma) \otimes (\tau'; \tau).$$

Now let A', A, B', B be games, let i be a subset inclusion from A into A' and let j be a subset inclusion from B into B' . Then, since subset inclusion strategies are automatically strict, we have

$$\begin{aligned} \text{wk}_{A',B'}; (\text{subs}_i \otimes \text{subs}_j) &= (\text{subs}_i \otimes \text{subs}_j); \text{wk}_{A,B} && \text{Prop. 1.56} \\ &= \text{subs}_{[i,j]}; \text{wk}_{A,B} && \text{Prop. 1.42} \\ &= \text{subs}_{[i,j]} && \text{Prop. 1.26} \\ &= \text{wk}_{A',B'}; \text{subs}_{[i,j]}. && \text{Prop. 1.26} \end{aligned}$$

As before, we know from Proposition 1.30 that $\text{wk}_{A',B'}$ is an epimorphism, and so

$$\text{subs}_i \otimes \text{subs}_j = \text{subs}_{[i,j]} . \quad \square$$

Proposition 1.59 tells us that $_ \otimes _$ is a functor $\mathcal{G}_s \times \mathcal{G} \rightarrow \mathcal{G}$. As before, write J for the inclusion functor $\mathcal{G}_s \hookrightarrow \mathcal{G}$. Then we can restate Proposition 1.56 in a more *natural* way.

Proposition 1.60. $\text{wk}_{A,B}$ is a natural transformation $JA \otimes B \rightarrow J(A \otimes B)$.

We have some additional structure on the \otimes and \otimes operators. By inspecting the definitions that if A, X, Y are games then the associator $\text{assoc}_{M_A, M_X, M_Y}$ and unitor runit_{M_A} in $(\mathbf{Set}, +)$ give rise to structural isomorphisms

$$(A \otimes X) \otimes Y \cong A \otimes (X \otimes Y) \quad A \cong A \otimes I .$$

Indeed, in the first case, both games are the game in which A, X and Y are played in parallel, but where the first move must take place in A . In the second case, we have $A \otimes I = A \otimes I$, because there are no moves in I anyway, and the copycat morphism induced from the right unitor in $(\mathbf{Set}, +)$ is the same strategy as the right unitor $A \xrightarrow{\cong} A \otimes I$.

We formalize the structure we have uncovered so far in the concept of a *sequoidal category*.

Definition 1.61. A *sequoidal category* \mathcal{C} is given by

- a monoidal category $(\mathcal{C}, \otimes, I)$ (with coherences $\text{assoc}, \text{lunit}, \text{runit}$);
- a (strong) right action of \mathcal{C} on a category \mathcal{C}_s ; i.e., a functor $_ \otimes _ : \mathcal{C}_s \times \mathcal{C} \rightarrow \mathcal{C}_s$ together with natural isomorphisms

$$\text{passoc}_{a,x,y} : (a \otimes x) \otimes y \xrightarrow{\cong} a \otimes (x \otimes y) \quad r_a : a \xrightarrow{\cong} a \otimes I$$

that make the diagrams

$$\begin{array}{ccc} ((a \otimes x) \otimes y) \otimes z & \xrightarrow{\text{passoc}_{a,x,y} \otimes z} & (a \otimes (x \otimes y)) \otimes z \xrightarrow{\text{passoc}_{a,x \otimes y,z}} a \otimes ((x \otimes y) \otimes z) \\ \downarrow \text{passoc}_{a \otimes x,y,z} & & \swarrow a \otimes \text{assoc}_{x,y,z} \\ (a \otimes x) \otimes (y \otimes z) & \xrightarrow{\text{passoc}_{a,x,(y \otimes z)}} & a \otimes (x \otimes (y \otimes z)) \end{array}$$

$$\begin{array}{ccc} a \otimes x & \xrightarrow{a \otimes \text{lunit}_x} & a \otimes (I \otimes x) \\ r_a \otimes x \downarrow & \nearrow \text{passoc}_{a,I,x} & \\ (a \otimes I) \otimes x & & \end{array} \quad \begin{array}{ccc} a \otimes x & \xrightarrow{a \otimes \text{runit}_x} & a \otimes (x \otimes I) \\ r_a \otimes x \downarrow & \nearrow \text{passoc}_{a,x,I} & \\ (a \otimes x) \otimes I & & \end{array}$$

commute; and

- a lax morphism of actions from $_ \otimes _$ to the right tensor multiplication action $_ \otimes _$ of \mathcal{C} on itself; i.e., a functor $J: \mathcal{C}_s \rightarrow \mathcal{C}$ and a natural transformation $\text{wk}_{a,x}: Ja \otimes x \rightarrow J(a \otimes x)$ that makes the following diagrams commute.

$$\begin{array}{ccc}
(Ja \otimes x) \otimes y & \xrightarrow{\text{wk}_{a,x} \otimes y} & J(a \otimes x) \otimes y \xrightarrow{\text{wk}_{a \otimes x, y}} J((a \otimes x) \otimes y) \\
\text{assoc}_{Ja, x, y} \downarrow & & \swarrow J \text{ passoc}_{a, x, y} \\
Ja \otimes (x \otimes y) & \xrightarrow{\text{wk}_{a, x \otimes y}} & J(a \otimes (x \otimes y))
\end{array}$$

$$\begin{array}{ccc}
Ja & \xrightarrow{J r_a} & J(a \otimes I) \\
\text{runit}_{Ja} \downarrow & & \swarrow \text{wk}_{a, I} \\
Ja \otimes I & &
\end{array}$$

Remark 1.62. The definitions of *lax action* can be found at the start of Section ??, while that of an *oplax morphism of actions* is found at Definition ?. The definitions we have used are similar: a *strong action* is a lax action in which the coherences (called m and e in Section ?? and **passoc** and **r** here) are isomorphisms. A *lax morphism of actions* is defined in the same way as an oplax morphism, except that the coherence (called μ in Definition ?? and **wk** here) goes in the opposite direction.

Proposition 1.63. *The monoidal category \mathcal{G} , together with the category \mathcal{G}_s , the natural transformations*

$$\text{passoc}_{A, X, Y} = \text{cc}_{\text{assoc}_{M_A, M_X, M_Y}} : (A \otimes X) \otimes Y \xrightarrow{\cong} A \otimes (X \otimes Y)$$

$$r_A = \text{cc}_{\text{runit}_{M_A}} : A \xrightarrow{\cong} A \otimes I,$$

the inclusion functor $J: \mathcal{G}_s \rightarrow \mathcal{G}$ and the natural transformation

$$\text{wk}_{A, X} = \text{subs}_j : JA \otimes X = A \otimes X \rightarrow A \otimes X = J(A \otimes X)$$

form a sequoidal category.

Proof. We have shown most of this already; all that remains is to show that **passoc** and **r** are natural transformations and that the five diagrams in Definition 1.61 commute.

Let us start with the diagrams. By Proposition 1.26, commutativity of these diagrams follows from commutativity of the diagrams formed from the corresponding subset inclusion functions in **Set**. For example, to show that the first diagram commutes in \mathcal{G} , we must show that the following diagram commutes in

Set.

$$\begin{array}{ccccc}
((M_A + M_X) + M_Y) + M_Z & \xrightarrow{[\text{assoc}_{M_A, M_X, M_Y}, \text{id}_{M_Z}]} & (M_A + (M_X + M_Y)) + M_Z & \xrightarrow{\text{assoc}_{M_A, M_X + M_Y, M_Z}} & M_A + ((M_X + M_Y) + M_Z) \\
\downarrow \text{assoc}_{M_A + M_X, M_Y, Z} & & & & \swarrow [\text{id}_{M_A}, \text{assoc}_{M_X, M_Y, M_Z}] \\
(M_A + M_X) + (M_Y + M_Z) & \xrightarrow{\text{assoc}_{M_A, M_X, M_Y + M_Z}} & M_A + (M_X + (M_Y + M_Z)) & &
\end{array}$$

This diagram is, of course, none other than the pentagram diagram for the coproduct $+$ in **Set**. Similarly, the second and third diagrams in Definition 1.61 reduce in this case to the triangle diagrams for the coproduct $+$ in **Set**.

For the fourth diagram in Definition 1.61, since wk is a subset inclusion strategy induced from an identity map, Proposition 1.26 tells us that both arms of the diagram are the strategy induced by the subset inclusion $\text{assoc}_{M_A, M_X, M_Y} : (M_A + M_X) + M_Y \rightarrow M_A + (M_X + M_Y)$ from $(A \otimes X) \otimes Y$ to $A \otimes (X \otimes Y)$. Similarly, both arms of the last diagram in Definition 1.61 are the strategies induced by the subset inclusion $\text{runit}_{M_A} : M_A \rightarrow M_A + \emptyset$ from A to $A \otimes I$.

It now remains only to show that passoc and r are natural transformations. For passoc , suppose that A', X', Y', A, X, Y are games, that $\sigma : A' \multimap A$ is a strict strategy and that $\tau : B' \multimap B, v : C' \multimap C$ are strategies. Then we need to show that the following diagram commutes.

$$\begin{array}{ccc}
(A' \otimes X') \otimes Y' & \xrightarrow{\text{passoc}_{A', X', Y'}} & A' \otimes (X' \otimes Y') \\
(\sigma \otimes \tau) \otimes v \downarrow & & \downarrow \sigma \otimes (\tau \otimes v) \\
(A \otimes X) \otimes Y & \xrightarrow{\text{passoc}_{A, X, Y}} & A \otimes (X \otimes Y)
\end{array}$$

Indeed, we have

$$\begin{aligned}
& (\text{wk}_{A', X'} \otimes Y'); \text{wk}_{A' \otimes X', Y'}; \text{passoc}_{A', X', Y'}; (\sigma \otimes (\tau \otimes v)) \\
&= \text{assoc}_{A', X', Y'}; \text{wk}_{A', X' \otimes Y'}; (\sigma \otimes (\tau \otimes v)) && \text{(see above)} \\
&= \text{assoc}_{A', X', Y'}; (\sigma \otimes (\tau \otimes v)); \text{wk}_{A, X \otimes Y} && \text{Prop. 1.60} \\
&= ((\sigma \otimes \tau) \otimes v); \text{assoc}_{A, X, Y}; \text{wk}_{A, X \otimes Y} && \text{Prop. 1.43} \\
&= ((\sigma \otimes \tau) \otimes v); (\text{wk}_{A, X} \otimes Y); \text{wk}_{A \otimes X, Y}; \text{passoc}_{A, X, Y} && \text{(see above)} \\
&= (\text{wk}_{A', X'} \otimes Y'); ((\sigma \otimes \tau) \otimes v); \text{wk}_{A \otimes X, Y}; \text{passoc}_{A, X, Y} && \text{Prop. 1.60} \\
&= (\text{wk}_{A', X'} \otimes Y'); \text{wk}_{A' \otimes X', Y'}; ((\sigma \otimes \tau) \otimes v); \text{passoc}_{A, X, Y} . && \text{Prop. 1.60}
\end{aligned}$$

Now observe that $\text{wk}_{A', X'} \otimes Y = \text{subs}_{[\text{id}_{M_{A'} + M_{X'}}, \text{id}_{M_{Y'}}]}$ by Proposition 1.42, so it is an epimorphism by Proposition 1.30. Proposition 1.30 also tells us that $\text{wk}_{A' \otimes X', Y'}$ is an epimorphism. Therefore, we have

$$\text{passoc}_{A', X', Y'}; (\sigma \otimes (\tau \otimes v)) = ((\sigma \otimes \tau) \otimes v); \text{passoc}_{A, X, Y}$$

for any $A', X', Y', A, X, Y, \sigma, \tau, v$ as above. It follows that passoc is a natural transformation.

The proof that r is a natural transformation is similar. Let A', A be games and let $\sigma: A' \multimap A$ be a strict strategy. We need to show that the following diagram commutes.

$$\begin{array}{ccc} A' & \xrightarrow{r_{A'}} & A' \otimes I \\ \sigma \downarrow & & \downarrow \sigma \otimes I \\ A & \xrightarrow{r_A} & A \otimes I \end{array}$$

Indeed, we have

$$\begin{aligned} r_{A'}; (\sigma \otimes I) &= \text{runit}_{A'}; \text{wk}_{A', I}; (\sigma \otimes I) && \text{(see above)} \\ &= \text{runit}_{A'}; (\sigma \otimes I); \text{wk}_{A, I} && \text{Prop. 1.60} \\ &= \sigma; \text{runit}_A; \text{wk}_{A, I} && \text{Prop. 1.43} \\ &= \sigma; r_A && \text{(see above)} \end{aligned}$$

Therefore, r is a natural transformation, which completes our check of the criteria required by Definition 1.61. \square

1.12 Sequoidally decomposable categories

We will now consider some important additional category-theoretic properties of the sequoid operator on games that do not follow from the fact that \mathcal{G} is a sequoidal category.

Definition 1.64 ([CLM13]). Let \mathcal{C} be a sequoidal category such that C_s has arbitrary products (including a terminal object 1). We say that \mathcal{C} is *distributive* if whenever a_i is a collection of objects of \mathcal{C}' and x is an object of \mathcal{C} , the morphism

$$\text{dec}_{(a_i), x} = \langle \text{pr}_i \otimes x \rangle: \prod_i a_i \otimes x \rightarrow \prod_i (a_i \otimes x)$$

is an isomorphism.

Remark 1.65. In particular, taking (a_i) to be the empty collection, the morphism $\text{l}_x = () : 1 \otimes x \rightarrow 1$ is an isomorphism.

Proposition 1.66. \mathcal{G} is a distributive sequoidal category.

Proof. Let $(A_i), X$ be games. By Proposition 1.52, the morphism $\langle \text{pr}_i \otimes X \rangle$ is given by the tree embedding $\phi: P_{\prod_i A_i \otimes X} \rightarrow P_{\prod_i (A_i \otimes X)}$ defined as follows.

$$\phi(s) = \begin{cases} \epsilon & \text{if } s = \epsilon \\ [\text{in}_{A_j}, \text{in}_{X^j}]_*(s) & \text{if } s \text{ begins with a move in } A_j \end{cases}$$

When we say $[\text{in}_{A_j}, \text{in}_{X^j}]_*(s)$, we have considered s as a sequence in $(M_{A_j} + X)^*$.

We claim that ϕ is a bijection. Indeed, it is certainly injective. Then, if we have a non-empty sequence $s \in P_{\prod_i (A_i \otimes X)}$, then s must start with a move in

some A_j , and must thereafter take place in the games A_j and X^j . Then $s = \phi([\text{in}_{A_j}, \text{in}_X]_*(s))$, where we have considered s as a sequence in $(M_{A_j} + M_{X^j})^*$.

Therefore, ϕ is a tree isomorphism, so $\text{dec}_{(A_i), X} = \text{tree}_\phi$ is an isomorphism by Proposition 1.33. \square

We can get a distributivity result in the other direction, but this one is not as strong, since the morphism we get is only a monomorphism, not an isomorphism.

Definition 1.67. Let \mathcal{C} be a distributive sequoidal category. We say that \mathcal{C} is *strongly distributive* if whenever A, B, C are objects of \mathcal{C}_s , the morphism

$$\langle A \otimes J(\text{pr}_1), A \otimes J(\text{pr}_2) \rangle : A \otimes J(B \times C) \rightarrow (A \otimes JB) \times (A \otimes JC)$$

is a monomorphism.

Proposition 1.68. \mathcal{G} is a strongly distributive sequoidal category.

Proof. Let $\sigma, \tau : D \multimap (A \otimes (B \times C))$ be strategies and suppose that

$$\sigma; \langle A \otimes \text{pr}_1, A \otimes \text{pr}_2 \rangle = \tau; \langle A \otimes \text{pr}_1, A \otimes \text{pr}_2 \rangle.$$

Let $s \in \sigma$. Without loss of generality, suppose that $s|_{B \times C} \in P_B$. Then we have

$$[\text{id}_{M_D}, \text{in}_{M_{A^L}}, \text{in}_{M_B}, \perp]_*(s) \in \sigma; \langle A \otimes \text{pr}_1, A \otimes \text{pr}_2 \rangle = \tau; \langle A \otimes \text{pr}_1, A \otimes \text{pr}_2 \rangle.$$

It follows that $s \in \tau$. Since s was arbitrary, we get $\sigma \subseteq \tau$. By an identical argument, we get that $\tau \subseteq \sigma$, and therefore that $\sigma = \tau$. \square

Definition 1.69. A sequoidal category \mathcal{C} is *inclusive* if \mathcal{C}_s is a full-on-objects subcategory of \mathcal{C} containing wk and all isomorphisms of \mathcal{G} , and the functor J is the inclusion functor.

In such a situation, we will sometimes drop the mention of the functor J .

Proposition 1.70. \mathcal{G} is an inclusive sequoidal category.

Proof. The only thing we really need to check is that isomorphisms in \mathcal{G} are always strict strategies. Indeed, if σ is a strategy for $A \multimap B$ and τ a strategy for $B \multimap A$ such that $\sigma; \tau = \text{id}_A$, then for any opening move a in A on the right of τ there is some $\mathfrak{s} \in \int(A, B, A)$ such that $\mathfrak{s}|_{A, A} = aa$, and therefore the reply to a in τ must take place in B . \square

An important fact about the sequoid operator for games is that it gives us a way to decompose the tensor product as

$$A \otimes B \cong (A \otimes B) \times (B \otimes A).$$

Informally, this is because both sides allow player O to start either in A or in B , and thereafter to continue in that game.

Definition 1.71. Let \mathcal{C} be a distributive inclusive sequoidal category, where \mathcal{C} is a symmetric monoidal category. We say that \mathcal{C} is *decomposable* if the morphisms

$$\text{dec}_{a,b} = \langle \text{wk}_{a,Jb}, \text{sym}_{Ja,Jb}; \text{wk}_{b,Ja} \rangle: Ja \otimes Jb \rightarrow (a \otimes Jb) \times (b \otimes Ja) \quad (): I \rightarrow 1$$

are isomorphisms in \mathcal{C}_s .

Proposition 1.72. Let \mathcal{C} be a decomposable sequoidal category and suppose that a_1, \dots, a_n is a list of objects of \mathcal{C}_s . Then we have an isomorphism

$$a_1 \otimes \dots \otimes a_n \cong \prod_{i=1}^n (a_i \otimes (a_1 \otimes \dots \otimes a_{i-1} \otimes a_{i+1} \otimes \dots \otimes a_n)).$$

Proof. Induction on n . If $n = 0$, then we have the isomorphism $() : I \rightarrow 1$. More generally, we have

$$\begin{aligned} & a_1 \otimes \dots \otimes a_{n+1} \\ \cong & (a_1 \otimes \dots \otimes a_n) \otimes a_{n+1} \\ \xrightarrow{\text{dec}} & (a_1 \otimes \dots \otimes a_n) \otimes a_{n+1} \times a_{n+1} \otimes (a_1 \otimes \dots \otimes a_n) \\ \cong & \left(\prod_{i=1}^n \left(a_i \otimes \bigotimes_{\substack{j \leq n \\ j \neq i}} a_j \right) \right) \otimes a_{n+1} \times a_{n+1} \otimes (a_1 \otimes \dots \otimes a_n) \\ \xrightarrow{\text{dist} \times \text{id}} & \prod_{i=1}^n \left(\left(a_i \otimes \bigotimes_{\substack{j \leq n \\ j \neq i}} a_j \right) \otimes a_{n+1} \right) \times a_{n+1} \otimes (a_1 \otimes \dots \otimes a_n) \\ \xrightarrow{\langle \text{passoc} \rangle \times \text{id}} & \prod_{i=1}^n \left(a_i \otimes \left(\bigotimes_{\substack{j \leq n \\ j \neq i}} a_j \otimes a_{n+1} \right) \right) \times a_{n+1} \otimes (a_1 \otimes \dots \otimes a_n) \\ \cong & \prod_{i=1}^{n+1} \left(a_i \times \bigotimes_{\substack{j \leq n+1 \\ j \neq i}} a_j \right) \times a_{n+1} \otimes (a_1 \otimes \dots \otimes a_n) \\ \cong & \prod_{i=1}^{n+1} (a_i \otimes (a_1 \otimes \dots \otimes a_{i-1} \otimes a_{i+1} \otimes \dots \otimes a_{n+1})), \end{aligned}$$

where each of the arrows is an isomorphism. \square

The specific isomorphism in Proposition 1.72 is rather complicated at the moment, but we can simplify it.

Definition 1.73. Given objects a_1, \dots, a_n of a monoidal category, we write sym_i^n for the unique symmetric coherence isomorphism

$$a_1 \otimes \dots \otimes a_n \cong a_i \otimes a_1 \otimes \dots \otimes a_{i-1} \otimes a_{i+1} \otimes \dots \otimes a_n.$$

Proposition 1.74. *The isomorphism in the proof of Proposition 1.72 is given by*

$$\text{dec}_{(a_i)}^n = \langle \text{sym}_i^n; \text{wk}_{a_i, a_1 \otimes \cdots \otimes a_{i-1} \otimes a_{i+1} \otimes \cdots \otimes a_n} \rangle.$$

Proof. Induction on n . We will make use of the coherence theorem for symmetric monoidal categories [Mac71, §11] to allow us to elide associators. The base case is obviously true, because $() : I \rightarrow 1$ is the unique morphism between these objects. Otherwise, we observe that the morphism into $\prod_{i=1}^{n+1} \left(a_i \otimes \bigotimes_{j \neq i}^{j \leq n+1} a_j \right)$ is given component-wise by morphisms $a_1 \otimes \cdots \otimes a_{n+1} \rightarrow a_i \otimes \bigotimes_{j \neq i}^{j \leq n+1} a_j$ for each $i = 1, \dots, n+1$; we need to check that each of these components is equal to $\text{sym}_i^{n+1}; \text{wk}_{a_i, \bigotimes_{j \neq i} a_j}$.

If $i \leq n$, then the i -th component of the morphism in the proof of Proposition 1.72 is given by the composite thick dashed arrows in Figure 1, and is therefore equal to the composite of the solid arrows, which is $\text{sym}_i^{n+1}; \text{wk}_{a_i, \bigotimes_{j \neq i} a_j}$ as desired. The $n+1$ -th component of the morphism in the proof of Proposition 1.72 is given by the composite

$$\bigotimes_{j=1}^{n+1} a_j \rightarrow \bigotimes_{j=1}^n a_j \otimes a_{n+1} \xrightarrow{\text{sym}_{\bigotimes_{j=1}^n a_j, a_{n+1}}} a_{n+1} \otimes \bigotimes_{j=1}^n a_j \xrightarrow{\text{wk}_{a_{n+1}, \bigotimes_{j=1}^n a_j}} a_{n+1} \otimes \bigotimes_{j=1}^n a_j,$$

and then we use the fact that the leftmost two morphisms in this composite compose to give us sym_{n+1}^{n+1} . \square

Proposition 1.75. *\mathcal{G} is a decomposable sequoidal category.*

Proof. Let A, B be games. By Proposition 1.52, the strategy

$$\langle \text{wk}_{A,B}, \text{sym}_{A,B}; \text{wk}_{A,B} \rangle$$

is given by the tree embedding ϕ from $(A \otimes B) \times (B \otimes A)$ to $A \otimes B$ given by

$$\phi(s) = \begin{cases} \epsilon & \text{if } s = \epsilon \\ s|_{A \otimes B} & \text{if } s \text{ takes place entirely within } A \otimes B \\ s|_{B \otimes A} & \text{if } s \text{ takes place entirely within } B \otimes A \end{cases}.$$

We claim that this tree embedding is a bijection. Indeed, it is certainly injective. Now let $s \in P_{A \otimes B}$ be a non-empty play. Then, if s begins with a move in A , we have $s = \phi((\text{in}_{A \otimes B})_*(s))$, and if s begins with a move in B , we have $s = \phi((\text{in}_{B \otimes A})_*(s))$. Therefore, ϕ is a tree isomorphism, so $\text{dec}_{A,B} = \text{tree}_\phi$ is an isomorphism in \mathcal{G} .

Lastly, we have $I = 1$ in \mathcal{G} , and the unique morphism $I \rightarrow 1$ is the identity. \square

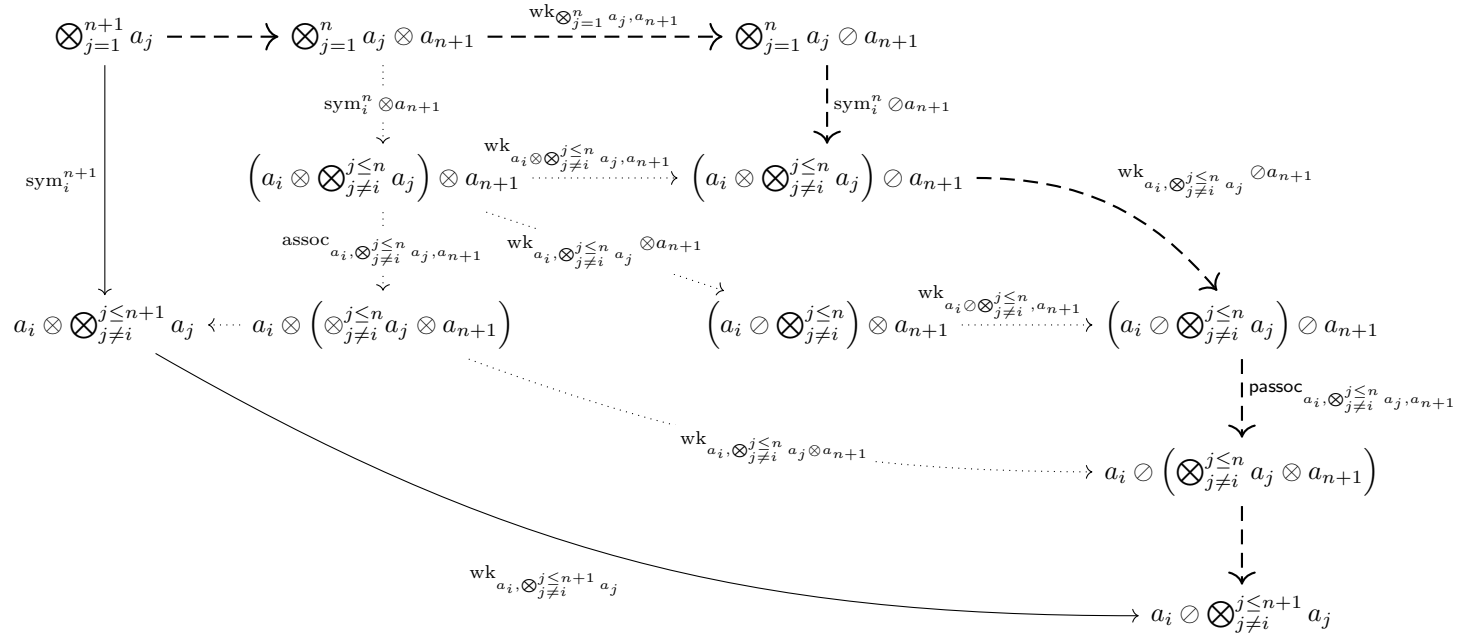


Figure 1: Diagram used in the proof of Proposition 1.74. The pentagon at the heart of the diagram is the coherence diagram for passoc and wk from Definition 1.61.

1.13 A Formula for the Exponential

Definition 1.76. Let \mathcal{C} be a symmetric monoidal category. Given objects A_1, \dots, A_n of \mathcal{C} and a permutation $\pi \in \mathfrak{S}_n$, there is a unique canonical symmetry isomorphism

$$\text{sym}^\pi: A_1 \otimes \dots \otimes A_n \xrightarrow{\cong} A_{\pi(1)} \otimes \dots \otimes A_{\pi(n)}.$$

Given an object A of \mathcal{C} , an n -th *symmetrized tensor power* of A is an equalizer (A^n, eq^n) for the diagram given by all morphisms of the form

$$\text{sym}_\pi: A^{\otimes n} \rightarrow A^{\otimes n}.$$

We say that the symmetrized tensor power A^n *commutes with the tensor product* if $(B \otimes A^n, B \otimes \text{eq}_n)$ is an equalizer for the diagram given by morphisms of the form

$$B \otimes \text{sym}_\pi: B \otimes A^{\otimes n} \rightarrow B \otimes A^{\otimes n}.$$

Proposition 1.77 ([GL17]). *Let \mathcal{C} be an inclusive, strongly distributive, decomposable sequoidal category. Then \mathcal{C} has all symmetrized tensor powers.*

Proof. Let A be an object of \mathcal{C} (equivalently, an object of \mathcal{C}_s). We inductively define objects $A^{\otimes n}$ by

- $A^{\otimes 0} = I$; and
- $A^{\otimes(n+1)} = J(A \otimes A^{\otimes n})$.

We claim that $A^{\otimes n}$ is a symmetrized tensor power of A .

Given n , we inductively define a morphism $\text{wk}^n: A^{\otimes n} \rightarrow A^{\otimes n}$, where $\text{wk}^0 = \text{id}_I$, and wk^{n+1} is given by the composite

$$A^{\otimes(n+1)} \rightarrow A \otimes A^{\otimes n} \xrightarrow{A \otimes \text{wk}^n} A \otimes A^{\otimes n} \xrightarrow{\text{wk}_{A, A^{\otimes n}}} A \otimes A^{\otimes n}.$$

We show by induction on n that if B is an object of \mathcal{C} and $k \geq 0$ then the composite

$$B \otimes (A \otimes _)^k A^{\otimes n} \xrightarrow{\langle B \otimes (A \otimes _)^k, \text{sym}^\pi \rangle} (B \otimes (A \otimes _)^k A^{\otimes n})^{n!} \xrightarrow{(B \otimes (A \otimes _)^k, \text{wk}^n)^{n!}} (B \otimes A^{\otimes(k+n)})^{n!}$$

(i.e., the morphism $\langle B \otimes (A \otimes _)^k, \text{sym}^\pi; \text{wk}^n \rangle$) is a monomorphism. In particular, taking $k = 0$, we will have shown that $\langle B \otimes \text{sym}^\pi; \text{wk}^n \rangle$ is a monomorphism.

The hypothesis is clearly true for $n = 0$; otherwise, we have a composite

$$\begin{aligned} B \otimes (A \otimes _)^k A^{\otimes(n+1)} &\xrightarrow{B \otimes (A \otimes _)^k \langle \text{sym}_i^{n+1}; \text{wk}_{A, A^{\otimes n}} \rangle} B \otimes (A \otimes _)^k (A \otimes \\ A^{\otimes n})^{n+1} &\xrightarrow{\langle B \otimes (A \otimes _)^k; \text{pr}_i \rangle} (B \otimes (A \otimes _)^{k+1} A^{\otimes n})^{n+1} \rightarrow (B \otimes A^{\otimes(k+n+1)})^{(n+1)!}, \end{aligned}$$

where the last arrow is the tensor product of B with the $(n+1)$ -th power of the composite given by

$$(A \otimes _)^{k+1} A^{\otimes n} \xrightarrow{\langle (A \otimes _)^{k+1} \text{sym}^\sigma \rangle} ((A \otimes _)^{k+1} A^{\otimes n})^{n!} \xrightarrow{\langle (A \otimes _)^{k+1} \text{wk}^n \rangle^{n!}} (A^{\otimes (k+n+1)})^{n!},$$

which is a monomorphism by the induction hypothesis. Then the previous composite is the composite of monomorphisms (by our assumptions on \mathcal{C}), and is therefore itself a monomorphism. Now this composite may be written as

$$\langle B \otimes (A \otimes _)^k (\text{sym}_i^{n+1}; \text{wk}_{A, A^{\otimes n}}; (A \otimes \text{sym}^\sigma); (A \otimes \text{wk}^n)) \rangle,$$

which, since wk is a natural transformation, is equal to

$$\langle B \otimes (A \otimes _)^k (\text{sym}_i^{n+1}; (A \otimes \text{sym}^\sigma); (A \otimes \text{wk}^n); \text{wk}_{A, A^{\otimes n}}) \rangle,$$

where σ ranges over the permutations in \mathfrak{S}_n . Moreover, by the definition of wk^n , this is equal to

$$\langle B \otimes (A \otimes _)^k (\text{sym}_i^{n+1}; (A \otimes \text{sym}^\sigma); \text{wk}^{n+1}) \rangle.$$

Now, given $i \in \{1, \dots, n+1\}$ and $\sigma \in \mathfrak{S}_n$, there is a unique permutation $\pi \in \mathfrak{S}_{n+1}$ such that $\text{sym}_i^{n+1}; (A \otimes \text{sym}^\sigma = \text{sym}^\pi)$; moreover, this defines a bijection from $\{1, \dots, n+1\} \times \mathfrak{S}_n \rightarrow \mathfrak{S}_{n+1}$. Therefore (after choosing an appropriate enumeration of our permutations), we see that this composite is in fact equal to

$$\langle B \otimes (A \otimes _)^k (\text{sym}^\pi; \text{wk}^{n+1}) \rangle.$$

Therefore, $\langle B \otimes (A \otimes _)^k (\text{sym}^\pi; \text{wk}^{n+1}) \rangle$ is a monomorphism as desired, completing the induction.

Next, we define morphisms $\text{eq}_n: A^{\otimes n} \rightarrow A^{\otimes n}$ inductively, where $\text{eq}_0 = \text{id}$ and eq_{n+1} is defined by the following composite

$$A^{\otimes (n+1)} = A \otimes A^{\otimes n} \xrightarrow{\langle (A \otimes \text{eq}_n)_1^n \rangle} (A \otimes A^{\otimes n})^n \cong A^{\otimes (n+1)},$$

where the final isomorphism is as in Propositions 1.72 and 1.74.

First, we show inductively that $\text{eq}_n; \text{sym}^\pi; \text{wk}^n = \text{id}_{A^{\otimes n}}$ for all permutations π of S_n . This is certainly true for $n=0$; otherwise, let $\pi \in \mathfrak{S}_{n+1}$ be a permutation. Let $j = \pi^{-1}(1)$ be the element sent to 1 by π and let σ be the permutation of $1, \dots, n$ such that applying σ to the elements $2, \dots, n+1$ and composing with π gives us the j -cycle $(1 \dots j)$. Then we have

$$\text{sym}_j^{n+1}; (A \otimes \text{sym}^\sigma) = \text{sym}^\pi.$$

Now we get

$$\begin{aligned} & \text{eq}_{n+1}; \text{sym}^\pi; \text{wk}^{n+1} \\ &= \langle (A \otimes \text{eq}_n)_1^n; (\text{dec}_A^{n+1})^{-1}; \text{sym}^\pi; (A \otimes \text{wk}^n); \text{wk}_{A, A^{\otimes n}} \rangle \\ &= \langle (A \otimes \text{eq}_n)_1^n; (\text{dec}_A^{n+1})^{-1}; \text{sym}_j^{n+1}; (A \otimes \text{sym}^\sigma); (A \otimes \text{wk}^n); \text{wk}_{A, A^{\otimes n}} \rangle \\ &= \langle (A \otimes \text{eq}_n)_1^n; (\text{dec}_A^{n+1})^{-1}; \text{sym}_j^{n+1}; \text{wk}_{A, A^{\otimes n}}; (A \otimes \text{sym}^\sigma); (A \otimes \text{wk}^n) \rangle \\ &= \langle (A \otimes \text{eq}_n)_1^n; (\text{dec}_A^{n+1})^{-1}; \langle \text{sym}_i^{n+1}; \text{wk}_{A, A^{\otimes n}} \rangle; \text{pr}_j; (A \otimes \text{sym}^\sigma); (A \otimes \text{wk}^n) \rangle \\ &= A \otimes (\text{eq}_n; \text{sym}^\sigma; \text{wk}^n), \end{aligned}$$

which is equal to the identity on $A^{\otimes(n+1)}$ by the induction hypothesis.

Now let ρ be a permutation in \mathfrak{S}_n . We claim that $\mathbf{eq}_n = \mathbf{eq}_n; \text{sym}^\rho$.

Indeed, we have

$$\begin{aligned} \mathbf{eq}_n; \langle \text{sym}^\pi; \text{wk}^n \rangle &= \langle \mathbf{eq}_n; \text{sym}^\pi; \text{wk}^n \rangle \\ &= \langle \text{id} \rangle \\ &= \langle \mathbf{eq}_n; \text{sym}^{\rho\pi}; \text{wk}^n \rangle \\ &= \mathbf{eq}_n; \text{sym}^\rho; \langle \text{sym}^\pi; \text{wk}^n \rangle. \end{aligned}$$

Since $\langle \text{sym}^\pi; \text{wk}^n \rangle$ is a monomorphism, this means that $\mathbf{eq}_n = \mathbf{eq}_n; \text{sym}^\rho$, as desired. Therefore, \mathbf{eq}_n equalizes the morphisms \mathbf{eq}_n . We claim that it is an equalizer, and that this equalizer is preserved by the tensor product.

Indeed, let B, C be objects of \mathcal{C} , and let $f: C \rightarrow B \otimes A^{\otimes n}$ be a morphism such that $f = f; (B \otimes \text{sym}^\pi)$ for all $\pi \in \mathfrak{S}_n$.

Let $\tilde{f} = f; (B \otimes \text{wk}^n): C \rightarrow B \otimes A^{\otimes n}$. We claim that $\tilde{f}; (B \otimes \mathbf{eq}_n) = f$; indeed, we have

$$\begin{aligned} \tilde{f}; (B \otimes \mathbf{eq}_n); \langle B \otimes (\text{sym}^\pi; \text{wk}^n) \rangle &= \langle f; (B \otimes (\text{wk}^n; \mathbf{eq}_n; \text{sym}^\pi; \text{wk}^n)) \rangle \\ &= \langle f; (B \otimes \text{wk}^n) \rangle \\ &= f; \langle B \otimes (\text{sym}^\pi; \text{wk}^n) \rangle. \end{aligned}$$

Therefore, since $\langle B \otimes (\text{sym}^\pi; \text{wk}^n) \rangle$ is a monomorphism, we know that $\tilde{f}; (B \otimes \mathbf{eq}_n) = f$.

Now suppose that $h: C \rightarrow B \otimes A^{\otimes n}$ is such that $h; (B \otimes \mathbf{eq}_n) = f$. We claim that $h = \tilde{f}$. Indeed, we have

$$\tilde{f} = f; (B \otimes \text{wk}^n) = h; (B \otimes \mathbf{eq}_n); (B \otimes \text{wk}^n) = h.$$

Therefore, $(B \otimes A^{\otimes n}, B \otimes \mathbf{eq}_n)$ is an equalizer of the arrows $B \otimes \text{sym}^\pi: B \otimes A^{\otimes n} \rightarrow B \otimes A^{\otimes n}$, as desired. \square

We are interested in symmetrized tensor powers because of an important result of Melliès, Tabareau and Tasson. Suppose \mathcal{C} is a monoidal category, and that \mathcal{C} has symmetrized tensor powers that commute with the tensor product. Given n , we have a morphism

$$() \otimes A^{\otimes n}: A^{\otimes(n+1)} \rightarrow A^{\otimes n};$$

then, if A^n and A^{n+1} are the n -th and $n+1$ -th symmetrized tensor powers of A , and $\mathbf{eq}_{n+1}, \mathbf{eq}$ the corresponding equalization, for any $\pi \in \mathfrak{S}_n$ we have a commutative diagram

$$\begin{array}{ccccc} B \otimes A^{n+1} & \xrightarrow{B \otimes \mathbf{eq}_{n+1}} & B \otimes A^{\otimes(n+1)} & \xrightarrow{B \otimes () \otimes A^{\otimes n}} & B \otimes A^{\otimes n} \\ & & \text{sym}^{\pi'} \downarrow & & \downarrow \text{sym}^\pi \\ & & B \otimes A^{\otimes(n+1)} & \xrightarrow{B \otimes () \otimes A^{\otimes n}} & B \otimes A^{\otimes n} \end{array},$$

where π' is the permutation of $1, \dots, n+1$ that fixes 1 and applies π to the remaining elements $2, \dots, n+1$.

This means that for each $\pi \in \mathfrak{S}_n$ we have

$$\begin{aligned} (B \otimes \mathbf{eq}_{n+1}); (B \otimes () \otimes A^{\otimes n}) &= (B \otimes \mathbf{eq}_{n+1}); (B \otimes \text{sym}^{\pi'}); (B \otimes () \otimes A^{\otimes n}) \\ &= (B \otimes \mathbf{eq}_{n+1}); (B \otimes () \otimes A^{\otimes n}); (B \otimes \text{sym}^{\pi}), \end{aligned}$$

and that there is therefore an induced morphism

$$B \otimes A^{n+1} \rightarrow B \otimes A^n,$$

by the universal property of the equalizer.

Theorem 1.78 ([MTT09]). *Let \mathcal{C} be a monoidal category in which the monoidal unit is a terminal object. Suppose that \mathcal{C} has symmetrized tensor powers that commute with the tensor product.*

Then, for any objects A, B of \mathcal{C} , there is a natural sequence

$$B \leftarrow B \otimes A \leftarrow B \otimes A^2 \leftarrow B \otimes A^3 \leftarrow \dots.$$

In particular, there is a sequence

$$I \leftarrow A \leftarrow A^2 \leftarrow A^3 \leftarrow \dots.$$

Suppose that this sequence has a limit $!A$, and that $B \otimes !A$ is the limit of the first sequence for all B . Then $!A$ has the structure of the cofree commutative comonoid on A .

References

- [AM96] Samson Abramsky and Guy McCusker. Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions: Extended abstract. *Electronic Notes in Theoretical Computer Science*, 3:2 – 14, 1996. Linear Logic 96 Tokyo Meeting.
- [CLM13] Martin Churchill, Jim Laird, and Guy McCusker. Imperative programs as proofs via game semantics. *CoRR*, abs/1307.2004, 2013.
- [GL17] William John Gowers and James Laird. Sequoidal categories and transfinite games: A coalgebraic approach to stateful objects in game semantics. In *CALCO*, 2017.
- [Har06] Russell Harmer. Innocent game semantics. 2006.
- [HO00] J.M.E. Hyland and C.-H.L. Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285 – 408, 2000.
- [Lai] James Laird. Functional programs as coroutines: A semantic analysis. *Logical methods in computer science*. To appear.

- [Lai02] J. Laird. A categorical semantics of higher-order store. In *Proceedings of CTCS '02*, number 69 in ENTCS. Elsevier, 2002.
- [Mac71] Saunders MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.
- [MTT09] Paul-André Melliès, Nicolas Tabareau, and Christine Tasson. *An Explicit Formula for the Free Exponential Modality of Linear Logic*, pages 247–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.