Mathematical Structures in Computer Science (2020), **30**, pp. 892–951
doi:10.1017/S0960129520000250

CAMBRIDGE
UNIVERSITY PRESS

PAPER

Dynamic game semantics

Norihiro Yamada[1,*] and Samson Abramsky[2]

[1]The University of Minnesota, Minneapolis, USA and [2]The University of Oxford, Oxford, UK
*Corresponding author. Email: yamad041@umn.edu

(Received 25 October 2018; revised 21 September 2020; accepted 23 September 2020; first published online 18 December 2020)

Abstract
The present work achieves a mathematical, in particular *syntax-independent*, formulation of *dynamics* and *intensionality* of computation in terms of *games* and *strategies*. Specifically, we give *game semantics* of a higher-order programming language that distinguishes programmes with the same value yet different algorithms (or intensionality) and the *hiding operation* on strategies that precisely corresponds to the (small-step) operational semantics (or dynamics) of the language. Categorically, our games and strategies give rise to a *cartesian closed bicategory*, and our game semantics forms an instance of a bicategorical generalisation of the standard interpretation of functional programming languages in cartesian closed categories. This work is intended to be a step towards a mathematical foundation of intensional and dynamic aspects of logic and computation; it should be applicable to a wide range of logics and computations.

Keywords: Game semantics; categorical logic; intensionality of computation

1. Introduction

Girard et al. (1989) mention the dichotomy between the *static* and the *dynamic* viewpoints in logic and computation; the former identifies terms (i.e., proofs or programmes) with their *denotations* (i.e., results of their computations in an ideal sense), while the latter focuses on their *senses* (i.e., algorithms or intensionality) and dynamics (i.e., proof-normalisation or reduction). This distinction has been reflected in the two mutually complementary semantics of programming languages: *denotational* and *operational* ones (Amadio and Curien, 1998; Gunter, 1992; Winskel, 1993).

Then, Girard et al. (1989) point out that a *mathematical* formulation of the former has been well developed by Scott's beautiful *domain theory* (Abramsky and Jung, 1994; Gierz et al., 2003; Scott, 1976), but it is not the case for the latter; the treatment of senses has been based on *ad hoc syntactic manipulations*. They thus emphasise the importance of *mathematics of senses*:

> The establishment of a truly operational semantics of algorithms is perhaps the most important problem in computer science (Girard et al., 1989, p. 14).

The present work addresses this fundamental problem. Specifically, it gives an interpretation $[\![\_]\!]_{\mathscr{D}}$ of a programming language $\mathscr{L}$ with a small-step operational semantics $\to$ and a syntax-independent operation $\mathscr{H}$ that satisfies the following *dynamic correspondence property (DCP)*:

$$(\mathsf{M}_1 \to \mathsf{M}_2) \Rightarrow ([\![\mathsf{M}_1]\!]_{\mathscr{D}} \neq [\![\mathsf{M}_2]\!]_{\mathscr{D}} \wedge \mathscr{H}([\![\mathsf{M}_1]\!]_{\mathscr{D}}) = [\![\mathsf{M}_2]\!]_{\mathscr{D}})$$

for all programmes M_1 and M_2 in $\mathscr{L}$. Note that this 'only if' direction of the DCP corresponds to certain *soundness* of the interpretation $\mathscr{H}$ of $\to$. (N.b., the opposite or *completeness* does *not* hold: $(\lambda x.\,\mathsf{ff})((\lambda y.\,y)\mathsf{tt}) \to (\lambda x.\,\mathsf{ff})\mathsf{tt}$ and $(\lambda x.\,\mathsf{ff})((\lambda y.\,y)\mathsf{tt}) \not\to (\lambda x.\,\mathsf{ff})\mathsf{ff}$, but

$\mathscr{H}([\![(\lambda x.\, \mathsf{ff})((\lambda y.\, y)\mathsf{tt})]\!]_{\mathscr{D}}) = [\![(\lambda x.\, \mathsf{ff})\mathsf{tt}]\!]_{\mathscr{D}} = [\![(\lambda x.\, \mathsf{ff})\mathsf{ff}\,]\!]_{\mathscr{D}}$, where tt and ff are programmes of the evident truth values. The lack of completeness, however, is not necessarily negative because it implies that the interpretation ignores some superficial syntactic differences as in the example.) Note also that the interpretation $[\![\_]\!]_{\mathscr{D}}$ is *finer* than the usual denotational semantics since $\mathsf{M}_1 \to \mathsf{M}_2$ implies $[\![\mathsf{M}_1]\!]_{\mathscr{D}} \neq [\![\mathsf{M}_2]\!]_{\mathscr{D}}$. Therefore, the interpretation $[\![\_]\!]_{\mathscr{D}}$ and the operation $\mathscr{H}$ capture *intensionality* and *dynamics* of computation, respectively.

Although our framework is intended to be a *general* approach, being applicable to a wide range of logics and computations, as the first step, we focus on a finite fragment of the programming language *PCF* (Plotkin, 1977; Scott, 1993) customised for our aim.

1.1 Game semantics

Our approach is based on *game semantics* (Abramsky et al., 1997; Abramsky and McCusker, 1999; Hyland, 1997), a particular kind of denotational semantics of logic and computation, in which formulas (or types) and proofs (or programmes) are interpreted as *games* and *strategies*, respectively.

We employ game semantics primarily for its conceptual naturality, which has so far led to a deeper understanding of logic and computation, and mathematical precision, which has been demonstrated by various *full completeness/abstraction* results (Curien, 2007) in the literature. Also, game semantics is very flexible: It has modelled a wide range of formal systems and programming languages by simply varying constraints on strategies (Abramsky and McCusker, 1999), which enables us to compare and relate various concepts in logic and computation *syntax-independently*. We utilise game semantics for the present work with the hope that these advantages of game semantics are also true for intensionality and dynamics of logic and computation.

A *game*, roughly, is a certain kind of a directed rooted forest whose branches represent possible 'developments' or *(valid) positions* of a 'game in the usual sense' (such as chess, poker, etc.). *Moves* of a game are nodes of the game, where some moves are distinguished and called *initial*; only initial moves can be the first element or *occurrence* of a position of the game. *Plays* of a game are (finitely or infinitely) increasing sequences $(\boldsymbol{\epsilon}, m_1, m_1 m_2, \ldots)$ of positions of the game, where $\boldsymbol{\epsilon}$ is the *empty sequence*. For our purpose, it suffices to focus on rather standard *sequential* (as opposed to *concurrent* (Abramsky and Melliès, 1999)), *negative* (as opposed to *positive* (Laurent, 2004)) games played by two participants, *Player*, representing a 'computational agent,' and *Opponent*, representing an 'environment,' in each of which Opponent always starts a play (i.e., negative), and then they alternately and separately perform moves (i.e., sequential) allowed by the rules of the game. Strictly speaking, a position of each game is not just a finite sequence of moves: Each occurrence m of Opponent's or O- (resp. Player's or P-) non-initial move in a position is assigned or *points to* a previous occurrence m' of P- (resp. O-) move in the position, meaning that m is performed specifically as a response to m'. The pointers are necessary to distinguish similar yet distinct computations (Abramsky and McCusker, 1999; Hyland and Ong, 2000).

A *strategy* on a game, on the other hand, is what tells Player which move (together with a pointer) she should make at each of her turns in the game. Hence, game semantics $[\![\_]\!]_{\mathscr{G}}$ of a programming language $\mathscr{L}$ interprets a type A in $\mathscr{L}$ as a game $[\![\mathsf{A}]\!]_{\mathscr{G}}$ that specifies possible plays between Player and Opponent, and a term $\mathsf{M} : \mathsf{A}$[1] in $\mathscr{L}$ as a strategy $[\![\mathsf{M}]\!]_{\mathscr{G}}$ on the game $[\![\mathsf{A}]\!]_{\mathscr{G}}$ that describes for Player how to play on $[\![\mathsf{A}]\!]_{\mathscr{G}}$. An execution of the term M is then modelled as a play in the game $[\![\mathsf{A}]\!]_{\mathscr{G}}$ in which Player follows $[\![\mathsf{M}]\!]_{\mathscr{G}}$.

Let us consider simple examples. The simplest game is the *terminal game* T, which has no moves, and thus it has only the trivial position $\boldsymbol{\epsilon}$ and the trivial strategy $\top := \{\boldsymbol{\epsilon}\}$.

As another example, consider the *natural number game* N, which is the following rooted tree (infinite in width):

$$\begin{tikzcd} & & q \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \cdots \\ 0 & 1 & 2 & 3 & \cdots \end{tikzcd}$$

in which a play starts with Opponent's question q ('What is your number?') and ends with Player's answer $n \in \mathbb{N}$ ('My number is n!'), where $\mathbb{N}$ is the set of all natural numbers, and n points to q (though this pointer is omitted in the above diagram). Henceforth, we usually skip drawing arrows that represent edges of a game. A strategy $\underline{10}$ on N, for instance, that corresponds to the natural number $10 \in \mathbb{N}$ can be represented by the map $q \mapsto 10$ equipped with the pointer from 10 to q (though it is the only choice for the pointer). In the following, pointers of most strategies are obvious, and therefore we often omit them.

As yet another example, consider the game $N \multimap N$ of *linear functions* (Girard, 1987), written informally $N_{[0]} \multimap N_{[1]}$ too, on natural numbers, whose typical maximal position is $q_{[1]}q_{[0]}n_{[0]}m_{[1]}$, where $n, m \in \mathbb{N}$, and the subscripts $(\_)_{[i]}$ for $i = 0, 1$ are unspecified 'tags' to distinguish the two copies of N (in the rest of the paper, we employ a similar notation for three or more copies of a game in the obvious manner too), or diagrammatically:[2]

$$\begin{tikzcd} N_{[0]} & \multimap & N_{[1]} \\ & & q_{[1]} \\ q_{[0]} \arrow[urr] & & \\ n_{[0]} \arrow[u, bend left] & & \\ & & m_{[1]} \arrow[uuu, bend right] \end{tikzcd}$$

which can be read as follows:

(1) Opponent's question $q_{[1]}$ for an output ('What is your output?');
(2) Player's question $q_{[0]}$ for an input ('Wait, what is your input?');
(3) Opponent's answer, say, $n_{[0]}$ to $q_{[0]}$ ('OK, here is an input n.');
(4) Player's answer, say, $m_{[1]}$ to $q_{[1]}$ ('Alright, the output is then m.').

A strategy *succ* on this game that corresponds to the (linear) successor function is represented by the map $q_{[1]} \mapsto q_{[0]}, q_{[1]}q_{[0]}n_{[0]} \mapsto n+1_{[1]}$, where n ranges over $\mathbb{N}$, or diagrammatically:

$$\begin{tikzcd} N_{[0]} & \overset{succ}{\multimap} & N_{[1]} \\ & & q_{[1]} \\ q_{[0]} \arrow[urr] & & \\ n_{[0]} \arrow[u, bend left] & & \\ & & n+1_{[1]} \arrow[uuu, bend right] \end{tikzcd}$$

1.2 Static game semantics

Game semantics is often said to be *intensional* and *dynamic* because a category of games and strategies is usually not well pointed, and plays in a game can be regarded as 'intensional, dynamic interactions' between the participants of the game. However, it has been employed as denotational semantics, and hence it is in particular *sound*: If two programmes evaluate to the same value, then their denotations in conventional game semantics are identical. Consequently, the conventional game semantics $[\![\_]\!]_{\mathcal{G}}$ is actually *extensional* and *static* in the sense that if there is a reduction $\mathsf{M}_1 \to \mathsf{M}_2$ in syntax, then the equation $[\![\mathsf{M}_1]\!]_{\mathcal{G}} = [\![\mathsf{M}_2]\!]_{\mathcal{G}}$ holds in the semantics (i.e., it does not capture the dynamics $\mathsf{M}_1 \to \mathsf{M}_2$ or the intensional difference between M_1 and M_2). In other words, the conventional game semantics is *not* intensional or dynamic in the sense that it does not satisfy the DCPs.

Therefore, to establish mathematics of senses (Girard et al., 1989), we need a more dynamic, intensional refinement of game semantics, so that it satisfies the DCPs. To get insights to develop such refined game semantics, let us see how the conventional game semantics fails to be dynamic or intensional. The point in a word is that 'internal communication' between strategies for their composition is *a priori* 'hidden,' and thus the resulting strategy is always in 'normal form.' For instance, the composition $succ; double : N \multimap N$ of strategies $succ : N \multimap N$ and $double : N \multimap N$, implementing the successor and the doubling (linear) functions, respectively,

$$\begin{tikzcd}
N_{[0]} & \overset{succ}{\multimap} & N_{[1]} \\
 & & q_{[1]} \\
q_{[0]} \arrow[urr] & & \\
m_{[0]} \arrow[u, bend left] & & \\
 & & m+1_{[1]} \arrow[uuu, bend right]
\end{tikzcd}
\qquad
\begin{tikzcd}
N_{[2]} & \overset{double}{\multimap} & N_{[3]} \\
 & & q_{[3]} \\
q_{[2]} \arrow[urr] & & \\
n_{[2]} \arrow[u, bend left] & & \\
 & & 2n_{[3]} \arrow[uuu, bend right]
\end{tikzcd}$$

is formed as follows. First, by 'internal communication,' we mean that Player plays the role of Opponent as well in the intermediate component games $N_{[1]}$ and $N_{[2]}$ just by 'copy-catting' her last moves, resulting in the play

$$\begin{tikzcd}
N_{[0]} & \overset{succ}{\multimap} & N_{[1]} & N_{[2]} & \overset{double}{\multimap} & N_{[3]} \\
 & & & & & q_{[3]} \\
 & & & \boxed{q_{[2]}} \arrow[urr] & & \\
 & & \boxed{q_{[1]}} \arrow[ur] & & & \\
q_{[0]} \arrow[urr] & & & & & \\
n_{[0]} \arrow[u, bend left] & & & & & \\
 & & \boxed{n+1_{[1]}} \arrow[uuu, bend left] & & & \\
 & & & \boxed{n+1_{[2]}} \arrow[uuuuu, bend left] & & \\
 & & & & & 2(n+1)_{[3]} \arrow[uuuuuuu, bend right]
\end{tikzcd}$$

where each move for 'internal communication' is marked by a square box just for clarity, and the pointer from $q_{[1]}$ to $q_{[2]}$ is added because the move $q_{[1]}$ is no longer initial. Importantly, it is assumed that Opponent plays on the game $N_{[0]} \multimap N_{[3]}$, 'seeing' only moves of $N_{[0]}$ or $N_{[3]}$. That is, the resulting play is to be read as follows:

(1) Opponent's question $q_{[3]}$ for an output in $N_{[0]} \multimap N_{[3]}$ ('What is your output?');

(2) Player's question $\boxed{q_{[2]}}$ by *double* for an input in $N_{[2]} \multimap N_{[3]}$ ('What is your input?');

(3) $\boxed{q_{[2]}}$ triggers the question $\boxed{q_{[1]}}$ for an output in $N_{[0]} \multimap N_{[1]}$ ('What is your output?');

(4) Player's question $q_{[0]}$ by *succ* for an input in $N_{[0]} \multimap N_{[1]}$ ('What is your input?');

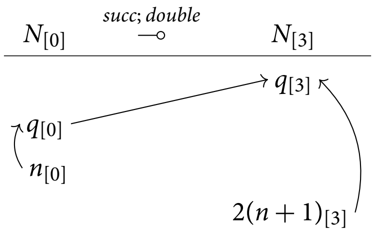
(5) Opponent's answer, say, $n_{[0]}$ to $q_{[0]}$ in $N_{[0]} \multimap N_{[3]}$ ('Here is an input *n*.');

(6) Player's answer $\boxed{n+1_{[1]}}$ to $\boxed{q_{[1]}}$ by *succ* in $N_{[0]} \multimap N_{[1]}$ ('The output is then $n+1$.');

(7) $\boxed{n+1_{[1]}}$ triggers the answer $\boxed{n+1_{[2]}}$ to $\boxed{q_{[2]}}$ in $N_{[2]} \multimap N_{[3]}$ ('Here is the input $n+1$.');

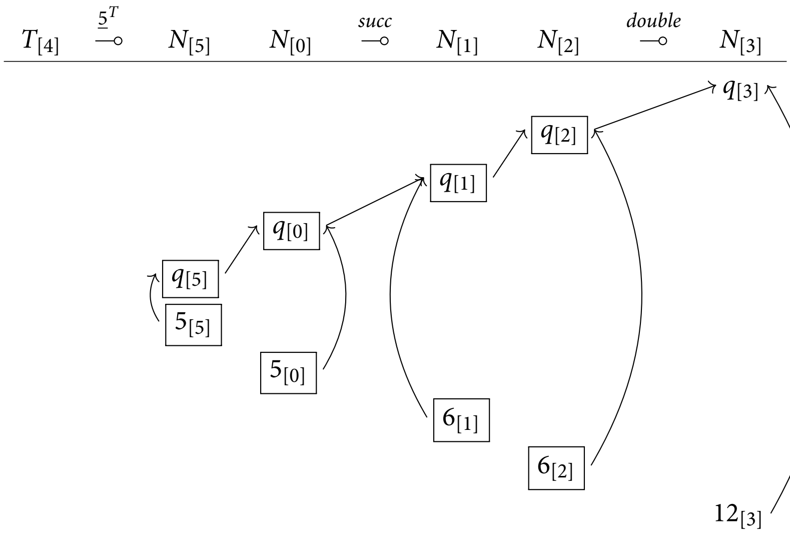
(8) Player's answer $2(n+1)_{[3]}$ to $q_{[3]}$ by *double* in $N_{[2]} \multimap N_{[3]}$ ('The output is then $2(n+1)$!').

Next, 'hiding' means to hide or delete every move with the square box from the play, resulting in the strategy for the (linear) function $n \mapsto 2(n+1)$ as expected:

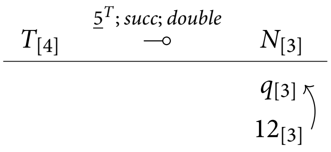


Note that it is 'hiding' that makes the resulting play a valid one in the game $N_{[0]} \multimap N_{[3]}$.

Now, let us plug in the strategy $\underline{5}^T : q_{[5]} \mapsto 5_{[5]}$ on the game $T_{[4]} \multimap N_{[5]}$, which coincides with N up to 'tags.' The composition $\underline{5}^T; succ; double : T \multimap N^3$ is computed again by 'internal communication'



plus 'hiding'



In syntax, on the other hand, assuming that there are a (ground) type ι of natural numbers, the numeral $\underline{n}$ of type ι for each $n \in \mathbb{N}$, and the constructors succ and double on type ι for the successor and the doubling functions, respectively, equipped with the operational semantics $\mathsf{succ}\,\underline{n} \to \underline{n+1}$ and $\mathsf{double}\,\underline{n} \to \underline{2n}$ for all $n \in \mathbb{N}$ in an arbitrary functional programming language, the programme $\mathsf{p}_1 \overset{\mathrm{df.}}{\equiv} \lambda \mathsf{x}.\, (\lambda \mathsf{y}.\, \mathsf{double}\, \mathsf{y})((\lambda \mathsf{z}.\, \mathsf{succ}\, \mathsf{z})\mathsf{x})$ represents the syntactic composition $\mathsf{succ}; \mathsf{double}$ of succ and double. When it is applied to the numeral $\underline{5}$, we have the chain of reductions

$$\begin{aligned} \mathsf{p}_1\underline{5} &\to^* (\lambda \mathsf{x}.\, \mathsf{double}\, (\mathsf{succ}\, \mathsf{x}))\underline{5} \\ &\to^* \mathsf{double}\, (\mathsf{succ}\, \underline{5}) \\ &\to^* \mathsf{double}\, \underline{6} \\ &\to^* \underline{12} \end{aligned}$$

Therefore, it seems that reduction in syntax corresponds in game semantics to 'hiding internal communication.' As seen in the above example, however, this game-semantic reduction is *a priori* executed and thus invisible in the conventional game semantics $[\![\_]\!]_{\mathscr{G}}$. Consequently, the two programmes $\mathsf{p}_1\underline{5}$ and $\underline{12}$ are interpreted by $[\![\_]\!]_{\mathscr{G}}$ as the same strategy. Furthermore, observe that moves with the square box describe *intensionality* or *step-by-step processes* to compute an output from an input, but they are invisible after 'hiding.' Thus, e.g., a programme $\mathsf{p}_2 \overset{\mathrm{df.}}{\equiv} \lambda \mathsf{x}.\, (\lambda \mathsf{y}.\, \mathsf{succ}\, \mathsf{y})(\lambda \mathsf{v}.\, (\lambda \mathsf{z}.\, \mathsf{succ}\, \mathsf{z})((\lambda \mathsf{w}.\, \mathsf{double}\, \mathsf{w})\mathsf{v})\mathsf{x})$, representing the same function as p_1 yet a different algorithm $\mathsf{double}; \mathsf{succ}; \mathsf{succ}$, is modelled as

$$[\![\mathsf{p}_2]\!]_{\mathscr{G}} = [\![\mathsf{double}; \mathsf{succ}; \mathsf{succ}]\!]_{\mathscr{G}} = [\![\mathsf{succ}; \mathsf{double}]\!]_{\mathscr{G}} = [\![\mathsf{p}_1]\!]_{\mathscr{G}}.$$

To sum up, we have observed the following three points:

(1) (Reduction as hiding). Reduction in syntax corresponds in game semantics to 'hiding intermediate moves (i.e., moves with the square box)';
(2) (A priori normalisation). However, the 'hiding' process is *a priori* executed in the conventional game semantics, and hence strategies are always in 'normal form';
(3) (Intermediate moves as intensionality). Finally, 'intermediate moves' constitute *intensionality* of computation; however, they are not captured in the conventional game semantics again due to the *a priori* execution of the 'hiding' operation.

1.3 Dynamic games and strategies

From these observations, we have obtained a promising solution: to define a variant of games and strategies, in which 'intermediate moves' are not a priori 'hidden,' representing intensionality of logic and computation, and the *hiding operations* $\mathscr{H}$ on the games and the strategies that 'hide intermediate moves' in a step-by-step fashion, interpreting dynamics of logic and computation. Let us call such a variant of games (resp. strategies) *dynamic games* (resp. *dynamic strategies*).

In doing so, we shall develop new mathematical structures that are conceptually natural and mathematically elegant. This effort is to inherit the natural, intuitive nature of the conventional game semantics, so that the resulting interpretation would be insightful, convincing and useful. Also, mathematics often leads to a 'correct' formulation: If a definition gives rise to neat mathematical structures, then it is likely to succeed in capturing the essence of concepts and phenomena of concern and subsume various instances (n.b., recall that our aim is to establish *mathematics* of senses). In fact, dynamic games and strategies are a natural generalisation of the conventional games and strategies, and they satisfy some beautiful algebraic laws. As a result, they form a *cartesian closed bicategory (CCB)* in the sense of Ouaknine (1997)[4] $\mathbb{LDG}$ (Definition 4.1), in which 0- (resp. 1-) cells are certain dynamic games (resp. dynamic strategies), and 2-cells are the *extensional equivalence* between 1-cells. The countably-infinite iteration of the hiding operations $\mathscr{H}$ on dynamic games and strategies induces the 2-functor $\mathscr{H}^\omega : \mathbb{LDG} \to \mathbb{LMG}$, where the CCC $\mathbb{LMG}$ of conventional games and strategies can be seen as the 'extensionally collapsed' $\mathbb{LDG}$.

1.4 Dynamic game semantics

We then give, as the main result of the present work, game semantics $[\![\_]\!]_{\mathscr{DG}}$ of *finitary PCF* (i.e., the simply-typed λ-calculus equipped with the boolean type) in $\mathbb{LDG}$ that together with the hiding operation $\mathscr{H}$ satisfies the DCP (Corollary 4.6). We call the pair $([\![\_]\!]_{\mathscr{DG}}, \mathscr{H})$ *dynamic game* *semantics* because it captures dynamics and intensionality of computation better than the conventional game semantics. We select finitary PCF as our target language since a simple language would be appropriate for the first work on dynamic game semantics.

At this point, note that it does not make much sense to ask whether full abstraction holds for dynamic game semantics because its aim is to capture intensionality of computation.

Also, note that our dynamic game semantics is not faithful: Equality (on the nose) between dynamic strategies is of course *finer than β-equivalence* but also *coarser than α-equivalence*, e.g., non-α-equivalent terms $(\lambda x.\, \mathsf{ff})\mathsf{tt}$ and $(\lambda x.\, \mathsf{ff})\mathsf{ff}$ are interpreted to be the same dynamic strategy in the dynamic game semantics. It is because an equation between dynamic strategies captures *algorithmic difference* of the interpreted terms, while α-equivalence distinguishes *how the terms are constructed* even if their algorithms coincide. This point justifies our (syntax-independent) mathematics of senses without the completeness property explained before.

On the other hand, it makes sense to ask if full completeness holds for dynamic game semantics. In fact, we shall establish our *dynamic full completeness* result (Corollary 4.7).

1.5 Our contribution and related work

To the best of our knowledge, the present work is the first syntax-independent characterisation of dynamics and intensionality of computation in the sense of the DCPs.

The work closest in spirit is Girard's *geometry of interaction (GoI)* (Girard, 1989, 1990, 1995, 2003, 2011, 2013). However, GoI appears mathematically *ad hoc* because it does not conform to the standard categorical semantics of type theories (Jacobs, 1999; Lambek and Scott, 1988; Pitts, 2001). Also, it does not capture the *step-by-step* process of reduction in the sense of the DCPs. In contrast, dynamic game semantics refines the standard semantics and does satisfy the DCP.

Next, the idea of exhibiting 'intermediate moves' in the composition of strategies is *nothing new*; there have been game-semantic approaches (Dimovski et al., 2005; Greenland, 2005; Ong, 2006) that give such moves an 'official status.' However, because their aims are rather to develop a *tool* for programme analysis and verification, they do not study in depth mathematical structures thereof, give an intensional game semantics that refines the standard categorical semantics or formulate the *step-by-step* 'hiding' process. Therefore, our contribution for this point is in studying the algebraic structures of games and strategies when we do not a priori 'hide intermediate moves' and refining the standard categorical semantics in such a way that it satisfies the DCP.

Also, there are several approaches to model dynamics of computation by *2-categories* (Hilken, 1996; Mellies, 2005; Seely, 1987). In these papers, however, the horizontal composition of 1-cells is the *normalising* one, which is why the structures are 2-categories rather than bicategories.[5] In addition, the 2-cells of their 2-categories are rewriting, while the 2-cells of our bicategory are the *external equivalence* between 1-cells. Note that 2-cells in a bicategory cannot interpret rewriting unless the horizontal composition is normalising since the associativity of non-normalising composition on such 2-cells does not hold.[6] Thus, although their motivations are similar to ours, our *bicategorical* approach seems novel, interpreting an application of a term by the non-normalising composition, the extensional equivalence of terms by 2-cells, and rewriting by the hiding operation $\mathcal{H}$. Moreover, their frameworks are categorical, while we instantiate our categorical model by game semantics. Furthermore, neither of the previous work satisfies the DCP.

Finally, the present work has some implications from theoretical as well as practical viewpoints. From the theoretical perspective, it enables us to study dynamics and intensionality of computation as purely *mathematical* (or *semantic*) concepts, just like any concepts in pure mathematics such as differentiation and integration in calculus, homotopy in topology, etc. Thus, we would be able to rigorously analyse the essence of these concepts, ignoring superfluous syntactic details. From the practical point, on the other hand, it might become a useful tool for language analysis and design, e.g., our variant of finitary PCF would not exist without the present work.

1.6 Structure of the paper

The rest of the present paper proceeds as follows. First, this introduction ends with fixing some notations. Then, Section 2 defines our target programming language, viz., the finitary PCF, and its bicategorical semantics that satisfies the DCP, so that it remains to establish its game-semantic instance. Next, Section 3 introduces dynamic games and strategies, and studies their basic algebraic structures. Then, Section 4 gives dynamic with game semantics of the programming language. Finally, Section 5 draws a conclusion and proposes some future work.

Notation 1.1. *We use the following notations throughout the present article:*

- *We write $\mathbb{N}$ for the set of all natural numbers and define $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$;*
- *We use bold letters $\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$, etc. for sequences, in particular $\boldsymbol{\varepsilon}$ for the empty sequence, and letters $a, b, c, d, m, n, x, y, z$, etc. for elements of sequences;*
- *Given a natural number $k \in \mathbb{N}$, we write $\overline{k}$ for the finite set $\{1, 2, \dots, k\} \subseteq \mathbb{N}$ (n.b., $\overline{0} = \emptyset$);*
- *We often abbreviate a finite sequence $\boldsymbol{s} = (x_1, x_2, \dots, x_{|\boldsymbol{s}|})$ as $x_1x_2 \dots x_{|\boldsymbol{s}|}$, where $|\boldsymbol{s}|$ denotes the length (i.e., the number of elements) of $\boldsymbol{s}$, and write $\boldsymbol{s}(i)$ as another notation for x_i ($i \in \overline{|\boldsymbol{s}|}$);*
- *A concatenation of sequences is represented by the juxtaposition of them, but we often write $a\boldsymbol{s}$, $\boldsymbol{t}b$, $\boldsymbol{u}c\boldsymbol{v}$ for $(a)\boldsymbol{s}$, $\boldsymbol{t}(b)$, $\boldsymbol{u}(c)\boldsymbol{v}$, etc., and also write $\boldsymbol{s}.\boldsymbol{t}$ for $\boldsymbol{st}$ if it increases readability;*
- *We define $\boldsymbol{s}^n := \underbrace{\boldsymbol{ss}\cdots\boldsymbol{s}}_{n}$ for a sequence $\boldsymbol{s}$ and a natural number $n \in \mathbb{N}$;*
- *We write $\mathsf{Even}(\boldsymbol{s})$ (resp. $\mathsf{Odd}(\boldsymbol{s})$) if a sequence $\boldsymbol{s}$ is of even length (resp. odd length);*
- *We define $S^{\mathsf{P}} := \{\, \boldsymbol{s} \in S \mid \mathsf{P}(\boldsymbol{s}) \,\}$ for a set S of sequences and a predicate $\mathsf{P} \in \{\mathsf{Even}, \mathsf{Odd}\}$;*
- *$\boldsymbol{s} \preceq \boldsymbol{t}$ means $\boldsymbol{s}$ is a prefix of $\boldsymbol{t}$, i.e., $\boldsymbol{t} = \boldsymbol{s}.\boldsymbol{u}$ for some sequence $\boldsymbol{u}$, and given a set S of sequences, we define $\mathsf{Pref}(S) := \{\, \boldsymbol{s} \mid \exists \boldsymbol{t} \in S \,.\, \boldsymbol{s} \preceq \boldsymbol{t} \,\}$;*
- *For a poset $P = (P, \leqslant)$ and a subset $S \subseteq P$, we write $\mathsf{Sup}_P(S)$ (usually abbreviated as $\mathsf{Sup}(S)$) for the supremum of S with respect to P;*
- *Given a set X, we define $X^* := \{\, x_1x_2 \dots x_n \mid n \in \mathbb{N}, \forall i \in \overline{n} \,.\, x_i \in X \,\}$;*
- *Given a function $f : A \to B$ and a subset $S \subseteq A$, we define $f \upharpoonright S : S \to B$ to be the restriction of f to S, and $f^* : A^* \to B^*$ by $f^*(a_1a_2 \dots a_n) := f(a_1)f(a_2) \dots f(a_n) \in B^*$ ($a_1a_2 \dots a_n \in A^*$);*
- *Given sets $X_1, X_2, \dots, X_n$, and $i \in \overline{n}$, we write π_i (or $\pi_i^{(n)}$) for the i^{th}-projection function $X_1 \times X_2 \times \dots \times X_n \to X_i$ that maps $(x_1, x_2, \dots, x_n) \mapsto x_i$;*
- *Given a function $f : X \to Y$ and a subset $Z \subseteq X$, we write $f \downharpoonright Z : X \setminus Z \to Y$ for the restriction of f to the set difference $X \setminus Z \subseteq X$.*

2. Dynamic Bicategorical Semantics

Let us first present in this section a *categorical* description of how dynamic games and strategies model dynamics and intensionality of our target language and show that it *refines* the standard categorical semantics of type theories (Jacobs, 1999; Lambek and Scott, 1988; Pitts, 2001).

2.1 Beta-categories of computation

The categorical structure for our interpretation of logic and computation is *β-categories of computation (BoCs)*, a certain kind of bicategories whose 2-cells are equivalence relations between 1-cells, equipped with an *evaluation* (function) on 1-cells such that the equivalence relations are the equality between 1-cells modulo the evaluation. Note that we do not take the quotient of 1-cells modulo the equivalence relation (so that the BoC becomes a category) since otherwise we would identify 1-cells with the same value even if they have different senses (or algorithms), unable to establish mathematics of senses.

Let us first introduce an auxiliary notion of *β-categories*, which are more general than BoCs. Roughly, β-categories are categories *up to an equivalence relation on morphisms*:

Definition 2.1 (β-categories). *A β-category is a pair $\mathscr{C} = (\mathscr{C}, \simeq)$ that consists of*

- *A class* ob$(\mathscr{C})$ *of* objects, *where we usually write $A \in \mathscr{C}$ for $A \in$* ob$(\mathscr{C})$*;*
- *A class $\mathscr{C}(A, B)$ of β-*morphisms *from A to B for each pair $A, B \in \mathscr{C}$, where we often write $f : A \to B$ for $f \in \mathscr{C}(A, B)$ if the underlying β-category $\mathscr{C}$ is obvious from the context;*
- *A (class) function $\mathscr{C}(A, B) \times \mathscr{C}(B, C) \overset{;_{A,B,C}}{\to} \mathscr{C}(A, C)$, called the β-*composition *on β-morphisms from A to B and those from B to C, for each triple $A, B, C \in \mathscr{C}$;*
- *A β-morphism $id_A \in \mathscr{C}(A, A)$, called the β-*identity *on A, for each object $A \in \mathscr{C}$;*
- *An equivalence (class) relation $\simeq_{A,B}$ on $\mathscr{C}(A, B)$, called the* equivalence *on β-morphisms from A to B, for each pair $A, B \in \mathscr{C}$,*

where we also write $\mathscr{C}(B, C) \times \mathscr{C}(A, B) \overset{\circ_{A,B,C}}{\to} \mathscr{C}(A, C)$ for the β-composition $;_{A,B,C}$ and often omit the subscripts $(\_)_{A,B,C}$ on ; $(\circ)$ and $\simeq$, such that it satisfies the four equations

$$(f; g); h \simeq f; (g; h) \qquad f; id_B \simeq f \qquad id_A; f \simeq f \qquad f \simeq f' \wedge g \simeq g' \Rightarrow f; g \simeq f'; g'$$

for any $A, B, C, D \in \mathscr{C}$, $f, f' : A \to B$, $g, g' : B \to C$ and $h : C \to D$.

Moreover, the β-category $\mathscr{C}$ is cartesian closed *if*

- *There is an object $T \in \mathscr{C}$, called a β-*terminal object, *equipped with a β-morphism $!_A : A \to T$, called the* canonical β-morphism *on A, for each object $A \in \mathscr{C}$, that satisfies $!_A \simeq t$ for any β-morphism $t : A \to T$;*
- *There is an object $A \times B \in \mathscr{C}$ for each pair $A, B \in \mathscr{C}$, called a β-* (binary) product *of A and B, equipped with β-morphisms $\pi_1^{A,B} : A \times B \to A$ and $\pi_2^{A,B} : A \times B \to B$, called the* first *and* the second β-projections *on $A \times B$, respectively, and an assignment $\langle \_, \_ \rangle$ of a β-morphism $\langle a, b \rangle_{A,B}^C : C \to A \times B$, called the β-*pairing *of a and b, to given object $C \in \mathscr{C}$ and β-morphisms $a : C \to A$ and $b : C \to B$, that satisfies*

$$\langle a, b \rangle_{A,B}^C; \pi_1^{A,B} \simeq a \qquad \langle a, b \rangle_{A,B}^C; \pi_2^{A,B} \simeq b \qquad \langle h; \pi_1^{A,B}, h; \pi_2^{A,B} \rangle_{A,B}^C \simeq h$$

$$(a \simeq a' \wedge b \simeq b') \Rightarrow \langle a, b \rangle \simeq \langle a', b' \rangle,$$

where $h : C \to A \times B$, $a' : C \to A$ and $b' : C \to B$ are arbitrary β-morphisms;
- *There are an object $C^B \in \mathscr{C}$ and a β-morphism $ev_{B,C} : C^B \times B \to C$, called the β-*exponential *and the β-*evaluation *of B and C, respectively, for each pair $B, C \in \mathscr{C}$, equipped with an assignment $\Lambda_{A,B,C}$ (also written $\Lambda_A^{B,C}(k)$ or $\Lambda_A(k)$) of a β-morphism $\Lambda_{A,B,C}(k) : A \to C^B$, called the β-*currying *of k, to any object $A \in \mathscr{C}$ and β-morphism $k : A \times B \to C$, that satisfies*

$$\langle \pi_1^{A,B}; \Lambda_{A,B,C}(k), \pi_2^{A,B} \rangle_{C^B,B}^{A \times B}; ev_{B,C} \simeq k \qquad \Lambda_{A,B,C}(\langle \pi_1^{A,B}; l, \pi_2^{A,B} \rangle_{C^B,B}^{A \times B}; ev_{B,C}) \simeq l$$

$$k \simeq k' \Rightarrow \Lambda_{A,B,C}(k) \simeq \Lambda_{A,B,C}(k'),$$

where $l : A \to C^B$ and $k' : A \times B \to C$ are arbitrary β-morphisms. We frequently omit the sub/superscripts on $\pi_i^{A,B}$, $\langle \_, \_ \rangle_{A,B}^C$, $ev_{B,C}$ and $\Lambda_{A,B,C}$.

That is, a (resp. cartesian closed) β-category $\mathscr{C} = (\mathscr{C}, \simeq)$ is a (resp. cartesian closed) category *up to* $\simeq$ (i.e., the equation $=$ on morphisms is replaced with the equivalence relation $\simeq$ on 1-cells), where the prefix 'β-' signifies the compromise 'up to $\simeq$.' Alternatively, regarding objects and β-morphisms of $\mathscr{C}$ as 0-cells and 1-cells, respectively, and defining 2-cells by $\mathscr{C}(A, B)(d, c) := \begin{cases} \{\simeq\} & \text{if } d \simeq c; \\ \emptyset & \text{otherwise} \end{cases}$ for any objects $A, B \in \mathscr{C}$ and β-morphisms $d, c : A \to B$, where $\{\simeq\}$ denotes an arbitrary singleton set, we may identify $\mathscr{C}$ with a (resp. cartesian closed (Ouaknine, 1997)) *bicategory* whose 2-cells are only the trivial one.

We are now ready to define *β-categories of computation (BoCs)*:

Definition 2.2 (BoCs). *A β*-category of computation (BoC) *is a β-category $\mathscr{C} = (\mathscr{C}, \simeq)$ equipped with a (class) function $\mathscr{E}$ on β-morphisms of $\mathscr{C}$, called the* evaluation *(of computation), that satisfies*

- (Subject reduction). *$\mathscr{E}(f) : A \to B$ for all $A, B \in \mathscr{C}$ and $f : A \to B$;*
- (Termination). *$f \downarrow$ for all $A, B \in \mathscr{C}$ and $f : A \to B$;*
- (β-identities). *$id_A \in \mathscr{V}_{\mathscr{C}}(A, A)$ for all $A \in \mathscr{C}$;*
- (Evaluation). *$f \simeq f' \Leftrightarrow \exists v \in \mathscr{V}_{\mathscr{C}}(A, B). f \downarrow v \wedge f' \downarrow v$ for all $A, B \in \mathscr{C}$ and $f, f' : A \to B$,*

where $\mathscr{V}_{\mathscr{C}}(A, B) := \{ v \in \mathscr{C}(A, B) \mid \mathscr{E}(v) = v \}$, whose elements are called values *from A to B in $\mathscr{C}$, and we write $f \downarrow$, or specifically $f \downarrow \mathscr{E}^n(f)$, if $\mathscr{E}^n(f) \in \mathscr{V}_{\mathscr{C}}(A, B)$ for some $n \in \mathbb{N}$.*[7]

A BoC $\mathscr{C}$ is cartesian closed, *which we call a* cartesian closed BoC (CCBoC), *if so is $\mathscr{C}$ as a β-category, all the canonical β-morphisms, the β-projections and the β-evaluations of $\mathscr{C}$ are values, and both of the β-pairing and the β-currying of $\mathscr{C}$ preserve values.*

Convention. Since the equivalence $\simeq$ of a BoC $\mathscr{C}$ may be completely recovered from the evaluation $\mathscr{E}$, we usually specify the BoC by a pair $\mathscr{C} = (\mathscr{C}, \mathscr{E})$. If $f \downarrow \mathscr{E}^n(f)$ holds for some $n \in \mathbb{N}$, then we call $\mathscr{E}^n(f)$ the *value* of f and also write $\mathscr{E}^\omega(f)$ for it.

The intuition behind Definition 2.2 is as follows. In a BoC $\mathscr{C} = (\mathscr{C}, \mathscr{E})$, β-morphisms are (possibly *intensional* but not necessarily 'effective') *computations* with the domain and the codomain (objects) specified, and values are *extensional* computations such as functions (as graphs). The β-composition is 'non-normalising composition' or *concatenation* of computations, and β-identities are *unit computations* (they are just like identity functions). The execution of a computation f is achieved by *evaluating* it into a unique value $\mathscr{E}^\omega(f)$, which corresponds to *dynamics* of computation.[8] We regard $\mathscr{E}^\omega$ as the operation on β-morphisms that maps them into their values, where the superscript $(\_)^\omega$ indicates that it is equivalent to iterating $\mathscr{E}$ by ω (i.e., the least transfinite number) times. In addition, the equivalence relation $\simeq$ witnesses the *extensional equivalence* between β-morphisms modulo $\mathscr{E}^\omega$. The four axioms then should make sense from this perspective. In this way, each BoC provides a 'universe' of dynamic, intensional computations.

It is easy to see that each BoC $\mathscr{C} = (\mathscr{C}, \mathscr{E})$ induces the category $\mathscr{V}_{\mathscr{C}}$ such that

- Objects are those of $\mathscr{C}$;
- Morphisms $A \to B$ are elements in $\mathscr{V}_{\mathscr{C}}(A, B)$, i.e., values from A to B in $\mathscr{C}$;
- The composition of morphisms $u : A \to B$ and $v : B \to C$ is the value $\mathscr{E}^\omega(u; v) : A \to C$;
- Identities are β-identities in $\mathscr{C}$.

Regarding the BoC $\mathscr{C}$ as the trivial bicategory specified above, and the category $\mathscr{V}_{\mathscr{C}}$ as the trivial 2-category, the evaluation $\mathscr{E}$ induces the 2-functor $\mathscr{E}^\omega : \mathscr{C} \to \mathscr{V}_{\mathscr{C}}$ that maps $A \mapsto A$ for 0-cells A, $f \mapsto \mathscr{E}^\omega(f)$ for 1-cells f and $\simeq \mapsto =$ for (unique) 2-cells $\simeq$. Clearly, $\mathscr{V}_{\mathscr{C}}$ is cartesian closed if so is $\mathscr{C}$, where canonical morphisms into a terminal object, projections, evaluations, pairing and currying of $\mathscr{V}_{\mathscr{C}}$ come from the corresponding 'β-ones' in $\mathscr{C}$, respectively.

The point here is that we may now decompose the standard interpretation $[\![\_]\!]_{\mathscr{S}}$ of functional programming languages in a CCC $\mathscr{V}_{\mathscr{C}}$ (Jacobs, 1999; Lambek and Scott, 1988; Pitts, 2001) into a more intensional interpretation $[\![\_]\!]_{\mathscr{D}}$ in a CCBoC $\mathscr{C} = (\mathscr{C}, \mathscr{E})$ and the full evaluation $\mathscr{E}^\omega : \mathscr{C} \to \mathscr{V}_{\mathscr{C}}$, i.e., $[\![\_]\!]_{\mathscr{S}} = \mathscr{E}^\omega([\![\_]\!]_{\mathscr{D}})$, and talk about *intensional difference* between computations: Terms M and M' are interpreted to be *intensionally equal* if $[\![\mathsf{M}]\!]_{\mathscr{D}} = [\![\mathsf{M}']\!]_{\mathscr{D}}$ and *extensionally equal* if $[\![\mathsf{M}]\!]_{\mathscr{D}} \simeq [\![\mathsf{M}']\!]_{\mathscr{D}}$. Also, the *one-step* evaluation $\mathscr{E}$ is to capture the small-step operational semantics of the target language, i.e., to satisfy the DCP; see Definition 2.16 for the precise definition of the DCP specialised to our target language.

2.2 Finitary PCF

Next, let us introduce our target programming language for dynamic game semantics.

Recall first that there is a one-to-one correspondence between *PCF Böhm trees* (i.e., terms of PCF in *η-long normal form*) (Amadio and Curien, 1998) and innocent, well bracketed strategies (Abramsky and McCusker, 1999; Curien, 2006; Hyland and Ong, 2000). This technical highlight in the literature of game semantics is called *strong definability*. Naturally, we would like to exploit the strong definability result to establish the first instance of dynamic game semantics because the task would be easier than otherwise.

On the other hand, the higher-order programming language *PCF* (Plotkin, 1977; Scott, 1993) has the *natural number type* and the *fixed-point combinators*, which make PCF Böhm trees *infinitary* in width and depth, respectively. However, we would like to select, as the first target language for dynamic game semantics, the simplest one possible because then the idea and the mechanism would be most visible. For this reason, let us choose *finitary PCF*, i.e., the finite fragment of PCF that has only the *boolean type* as a ground type (or equivalently, the *simply-typed λ-calculus* (Church, 1940; Sørensen and Urzyczyn, 2006) equipped with the boolean type).

We then define small-step operational semantics (or a reduction strategy) on finitary PCF as follows. We restrict terms to those built from PCF Böhm trees via application and currying (hence, e.g., the language does not have the usual variable rule), where PCF Böhm trees are *normalised*, and application generates non-normalised terms, and define operational semantics that normalises applications occurring in such a restricted term in the order from older to newer ones. A main virtue of the resulting language is that we may exploit the strong definability result straightforwardly, so that we may achieve a tight correspondence between syntax and semantics.

Remark. Note that an execution of *linear head reduction (LHR)* (Danos and Regnier, 2004) corresponds in a step-by-step fashion to an 'internal communication' between strategies (Danos et al., 1996). Hence, one may wonder if it would be better to employ LHR as the operational semantics of finitary PCF. However, note that

- The correspondence is *not* between terms and strategies;
- LHR is executed by *linear substitution*, which makes the calculus very different from the usual λ-calculus equipped with β-reduction.

For these two points, we conjecture that it would require significantly more work than the present work to establish the game-semantic DCP with respect to LHR, and so we leave it as future work.

In the following, we give the precise definition of the resulting target programming language (viz., finitary PCF equipped with the small-step operational semantics).

Notation 2.3. *We employ the following notations:*

- *Let $\mathcal{V}$ be a countably infinite set of* variables, *written* x, y, z, *etc., for which we assume the* variable convention *(or* Barendregt's convention *Hankin* 1994[9]*);*
- *We use sans-serif letters such as* Γ, A *and* a *for syntactic objects and* $\equiv$ *for syntactic equality up to* α-equivalence, *i.e., up to renaming of bound variables.*

Definition 2.4 (FPCF). *The* finitary PCF (FPCF) *is a functional programming language defined as follows:*

- (Types). *A* type A *is an expression generated by the grammar*

$$\mathsf{A} \overset{\text{df.}}{\equiv} o \mid \mathsf{A}_1 \Rightarrow \mathsf{A}_2$$

where o *is the* boolean type, *and* $A_1 \Rightarrow A_2$ *is the* function type *from* A_1 *to* A_2 *(n.b.,* $\Rightarrow$ *is right associative). We write* A, B, C, *etc. for types. Note that each type may be written uniquely of the form* $A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_k \Rightarrow o$ $(k \in \mathbb{N})$*.*

- (Raw-terms). *A* raw-term M *is an expression generated by the grammar*

$$M \overset{\text{df.}}{\equiv} x \mid tt \mid ff \mid case(M)[M_1; M_2] \mid \lambda x^A. M \mid M_1M_2$$

where x *ranges over variables, and* A *over types. We call* tt, ff, $case(M)[M_1; M_2]$, $\lambda x^A. M$ *and* M_1M_2 *respectively* true, false, *a* case statement, *an* abstraction *and an* application. *We write* M, P, Q, R, *etc. for raw terms and often omit* A *in an abstraction* $\lambda x^A. M$*. An application is always left associative, e.g.,* $M_1M_2M_3$ *may be written informally* $(M_1M_2)M_3$*. The set* $\mathscr{FV}(M) \subseteq \mathscr{V}$ *of all* free variables *occurring in a raw-term* M *is defined by the following induction on* M*:*

$$\mathscr{FV}(x) := \{x\} \qquad \mathscr{FV}(tt) := \mathscr{FV}(ff) := \emptyset$$

$$\mathscr{FV}(case(M)[M_1; M_2]) := \mathscr{FV}(M) \cup \mathscr{FV}(M_1) \cup \mathscr{FV}(M_2)$$

$$\mathscr{FV}(\lambda x. M) := \mathscr{FV}(M) \setminus \{x\} \qquad \mathscr{FV}(M_1M_2) := \mathscr{FV}(M_1) \cup \mathscr{FV}(M_2).$$

- (Contexts). *A* context *is a finite sequence* $x_1 : A_1, x_2 : A_2, \ldots, x_k : A_k$ *of (variable : type)-pairs with* $x_i \neq x_j$ *if* $i \neq j$ $(i, j \in \overline{k})$*. We write* Γ, Δ, Θ*, etc. for contexts.*
- (Terms). *A* term *is an expression of the form* $\Gamma \vdash M : B$*, where* Γ *is a context,* M *is a raw-term, and* B *is a type, generated by the following* typing rules*:*

$$(C1)\ \frac{\begin{array}{cc}\Gamma \equiv \Delta, \Theta & \forall i \in \overline{k}.\, \Gamma \vdash V_i : A_i \wedge \sharp(V_i) = 0 \wedge x \notin \mathscr{FV}(V_i) \\ A \equiv A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_k \Rightarrow o & \forall j \in \overline{2}.\, \Gamma \vdash W_j : o \wedge \sharp(W_j) = 0 \wedge x \notin \mathscr{FV}(W_j)\end{array}}{\Delta, x : A, \Theta \vdash case(xV_1V_2 \ldots V_k)[W_1; W_2] : o}$$

$$(C2)\ \frac{\Gamma \vdash M : o \quad \forall j \in \overline{2}.\, \Gamma \vdash P_j : o}{\Gamma \vdash case(M)[P_1; P_2] : o} \qquad (L)\ \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A. M : A \Rightarrow B}$$

$$(A)\ \frac{\Gamma \vdash M_1 : A \Rightarrow B \qquad \Gamma \vdash M_2 : A}{\Gamma \vdash M_1M_2 : B} \qquad (B)\ \frac{b \in \{tt, ff\}}{\Gamma \vdash b : o}$$

where V_i $(i = 1, 2, \ldots, k)$ *and* W_j $(j = 1, 2)$ *range over* values *defined below, and* $\sharp(\Gamma \vdash M : B) \in \mathbb{N}$*, often abbreviated as* $\sharp(M)$*, is the* execution number *of each term* $\Gamma \vdash M : B$ *defined by the following induction on the derivation of* $\Gamma \vdash M : B$*:*

- $\sharp(b) := 0$ *if* $b \in \{tt, ff\}$*;*
- $\sharp(case(xV_1V_2 \ldots V_k)[W_1; W_2]) := 0$*;*
- $\sharp(case(M)[P_1; P_2]) := 0$*;*
- $\sharp(\lambda x^A.M) := \sharp(M)$*;*
- $\sharp(M_1M_2) := \max(\sharp(M_1), \sharp(M_2)) + 1$.

A subterm *of a term* $\Gamma \vdash M : B$ *is a term that occurs in the deduction of* $\Gamma \vdash M : B$*, where a deduction (tree) of each term of FPCF is clearly* unique. *Execution numbers are to assign, to each subterm of a term, the priority order of the subterm during the execution of the term, i.e., they inform the operational semantics of how to reduce the term, which is made precise below. We write* $\Gamma \vdash \{M\}_e : B$ *for the term* $\Gamma \vdash M : B$ *such that* $\sharp(M) = e$*. Also, we often omit the context and/or the type of a term if it does not bring confusion. A* programme *(resp. a* value*) is a term generated by the rules B, C1, L and A (resp. B, C1 and L), where the rule C2 is redundant here, but it is necessary to define the operational semantics given below.*

Remark. The rules C2 (above) and ϑ_4 (below) are auxiliary concepts and only necessary for 'intermediate terms' during an evaluation of a programme into a value.

- ($\beta\vartheta$-REDUCTION). *The* $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$ *on terms is the* contextual closure, *i.e., the closure with respect to the typing rules, of the union of the rules*

$$\begin{aligned}(\lambda x.\, M)P &\rightarrow_{\beta} M[P/x]\\ \mathsf{case}(\mathsf{tt})[M_1; M_2] &\rightarrow_{\vartheta_1} M_1\\ \mathsf{case}(\mathsf{ff})[M_1; M_2] &\rightarrow_{\vartheta_2} M_2\\ \mathsf{case}(\mathsf{case}(x\mathbf{V})[W_1; W_2])[M_1; M_2] &\rightarrow_{\vartheta_3} \mathsf{case}(x\mathbf{V})[\mathsf{case}(W_1)[M_1; M_2]; \mathsf{case}(W_2)[M_1; M_2]]\\ \mathsf{case}(\mathsf{case}(M)[P_1; P_2])[Q_1; Q_2] &\rightarrow_{\vartheta_4} \mathsf{case}(M)[\mathsf{case}(P_1)[Q_1; Q_2]; \mathsf{case}(P_2)[Q_1; Q_2]]\end{aligned}$$

  where $M[P/x]$ *denotes the* capture-free substitution *(Barendregt et al.,* 1984*; Hankin,* 1994*) of* P *for* x *in* M*, and* $x\mathbf{V}$ *abbreviates* $xV_1V_2 \ldots V_k$ *in the rule C1. We write* $\mathit{nf}(M)$ *for the* normal form *of each term* M *with respect to* $\rightarrow_{\beta\vartheta}$*, i.e.,* $\mathit{nf}(M)$ *is a term such that* $M \rightarrow^{*}_{\beta\vartheta} \mathit{nf}(M)$ *and* $\mathit{nf}(M) \not\rightarrow_{\beta\vartheta} M'$ *for any term* M'*, which uniquely exists by Theorems* 2.11 *and* 2.12 *given below. The* parallel $\beta\vartheta$-reduction $\rightrightarrows_{\beta\vartheta}$ *on terms is defined to evaluate each term* M *in a single-step to its normal form* $\mathit{nf}(M)$.
- (OPERATIONAL SEMANTICS). *The* (small-step) operational semantics *(or the* reduction strategy*)* $\rightarrow$ *on programmes* M *is the 'simultaneous execution' of* $\rightrightarrows_{\beta\vartheta}$ *on all subterms of* M *with the execution number* 1*, or more precisely* $\rightarrow$ *is defined by*

$$M \rightarrow \begin{cases} V & \text{if } M \equiv M_1M_2,\ \sharp(M_1M_2) = 1 \text{ and } M_1M_2 \rightrightarrows_{\beta\vartheta} V;\\ M'_1M'_2 & \text{if } M \equiv M_1M_2,\ \sharp(M_1M_2) \geqslant 2 \text{ and } M_i \rightarrow M'_i \text{ for } i = 1, 2;\\ \lambda x^A.\tilde{M}' & \text{if } M \equiv \lambda x^A.\tilde{M} \text{ and } \tilde{M} \rightarrow \tilde{M}'.\end{cases}$$

We write Eq(FPCF) for the equational theory that consists of judgements $\Gamma \vdash M = M' : B$*, where* $\Gamma \vdash M : B$ *and* $\Gamma \vdash M' : B$ *are programmes of FPCF, such that* $\mathit{nf}(M) \equiv \mathit{nf}(M')$.

Values of FPCF are PCF Böhm trees except that the *bottom* (term) $\perp$ and the *natural number type* ι are excluded. The $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$ is taken from Section 6 of Amadio and Curien (1998). Hence, as announced before, programmes of FPCF are constructed from (finitary) PCF Böhm trees via currying and application, where non-programme terms (i.e., the rules C2 and ϑ_4) are auxiliary and only necessary for the operational semantics $\rightarrow$.

Remark. Let $A \equiv A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_k \Rightarrow o$ be an arbitrary type of FPCF. Note that an expression $\Delta, x : A, \Theta \vdash x : A$ is *not* a term of FPCF, but instead there is another, $\Delta, x : A, \Theta \vdash \underline{x}^A : A$, where $\underline{x}^A \stackrel{\mathrm{df.}}{\equiv} \lambda x_1^{A_1} x_2^{A_2} \ldots x_k^{A_k}.\, \mathsf{case}(x\underline{x_1}^{A_1}\underline{x_2}^{A_2} \ldots \underline{x_k}^{A_k})[\mathsf{tt}; \mathsf{ff}]$, which *is* a programme of FPCF. We often abbreviate $\underline{x}$ as $\underline{x}^A$ if it does not bring confusion.

Given a term $\Gamma \vdash M : B$, an execution number is assigned to each subterm of M. These execution numbers represent the priority order of the subterms with respect to the operational semantics $\rightarrow$ applied to M in the sense that those with the execution number 1 are normalised in one step $M \rightarrow M'$, where positive execution numbers are subtracted 1. In this manner, the operational semantics $\rightarrow$ reduces applications occurring in a given programme in the order from older to newer ones.

As we have mentioned before, programmes of FPCF are built from (finitary) PCF Böhm trees via application and currying, where PCF Böhm trees are normalised, and only application generates non-normalised programmes. This point is reflected as follows. The typing rules B and C1, corresponding to PCF Böhm trees, are assigned the execution number 0, the rule L, corresponding to currying, preserves execution numbers, and the rule A, corresponding to application, increases execution numbers by 1. The last rule C2 is only to take care of non-programme terms generated during the execution of programmes with respect to the parallel $\beta\vartheta$-reduction $\rightrightarrows_{\beta\vartheta}$ (it is why C2 is excluded from the construction of programmes), which in turn defines the operational semantics $\rightarrow$.[10]

In summary, FPCF computes as follows. Given a programme $\Gamma \vdash \{\mathsf{M}\}_e : \mathsf{B}$, it produces a *finite* chain of *finitary* rewriting

$$\mathsf{M} \rightarrow \mathsf{M}_1 \rightarrow \mathsf{M}_2 \rightarrow \cdots \rightarrow \mathsf{M}_\mathsf{e} \tag{1}$$

where M_e is a value. Note that the programme M is constructed from values by a finite number of applications, and the computation (1) is executed in the *first-applications-first-evaluated* fashion, e.g., if $\mathsf{M} \equiv (\mathsf{V}_1\mathsf{V}_2)((\mathsf{V}_3\mathsf{V}_4)(\mathsf{V}_5\mathsf{V}_6))$, where $\mathsf{V}_1, \mathsf{V}_2, \ldots, \mathsf{V}_6$ are values, then $e = 3$, and the computation (1) would be of the form

$$(\mathsf{V}_1\mathsf{V}_2)((\mathsf{V}_3\mathsf{V}_4)(\mathsf{V}_5\mathsf{V}_6)) \rightarrow \mathsf{V}_7(\mathsf{V}_8\mathsf{V}_9) \rightarrow \mathsf{V}_7\mathsf{V}_{10} \rightarrow \mathsf{V}_{11}$$

where $\mathsf{V}_7 \equiv nf(\mathsf{V}_1\mathsf{V}_2)$, $\mathsf{V}_8 \equiv nf(\mathsf{V}_3\mathsf{V}_4)$, $\mathsf{V}_9 \equiv nf(\mathsf{V}_5\mathsf{V}_6)$, $\mathsf{V}_{10} \equiv nf(\mathsf{V}_8\mathsf{V}_9)$ and $\mathsf{V}_{11} \equiv nf(\mathsf{V}_7\mathsf{V}_{10})$.

The rest of the present section is devoted to showing that the computation (1) of FPCF in fact correctly works (Corollary 2.14).

First, by the following Proposition 2.5 and Theorem 2.9, it makes sense that the $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$ is defined *on terms* (not on raw-terms):

Proposition 2.5 (Unique typing). *If* $\Gamma \vdash \{\mathsf{M}\}_e : \mathsf{B}$ *and* $\Gamma \vdash \{\mathsf{M}\}_{e'} : \mathsf{B}'$, *then* $e = e'$ *and* $\mathsf{B} \equiv \mathsf{B}'$.

Proof. By induction on the construction of $\Gamma \vdash \mathsf{M} : \mathsf{B}$. □

Lemma 2.6 (Free variable lemma). *If* $\Gamma \vdash \mathsf{M} : \mathsf{B}$, *and* $\mathsf{x} \in \mathscr{V}$ *occurs free in* M, *then* $\mathsf{x} : \mathsf{A}$ *occurs in* Γ *for some type* A.

Proof. By induction on the construction of $\Gamma \vdash \mathsf{M} : \mathsf{B}$. □

Lemma 2.7 (EW-lemma). *If* $\mathsf{x}_1 : \mathsf{A}_1, \mathsf{x}_2 : \mathsf{A}_2, \ldots, \mathsf{x}_\mathsf{k} : \mathsf{A}_\mathsf{k} \vdash \{\mathsf{M}\}_e : \mathsf{B}$, *then*

(1) $\mathsf{x}_{\sigma(1)} : \mathsf{A}_{\sigma(1)}, \mathsf{x}_{\sigma(2)} : \mathsf{A}_{\sigma(2)}, \ldots, \mathsf{x}_{\sigma(\mathsf{k})} : \mathsf{A}_{\sigma(\mathsf{k})} \vdash \{\mathsf{M}\}_e : \mathsf{B}$ *for any permutation* σ *on* $\overline{k}$;
(2) $\mathsf{x}_1 : \mathsf{A}_1, \mathsf{x}_2 : \mathsf{A}_2, \ldots, \mathsf{x}_\mathsf{k} : \mathsf{A}_\mathsf{k}, \mathsf{x}_{\mathsf{k}+1} : \mathsf{A}_{\mathsf{k}+1} \vdash \{\mathsf{M}\}_e : \mathsf{B}$ *for any variable* $\mathsf{x}_{\mathsf{k}+1} \in \mathscr{V}$ *and type* $\mathsf{A}_{\mathsf{k}+1}$ *such that* $\mathsf{x}_{\mathsf{k}+1} \not\equiv \mathsf{x}_\mathsf{i}$ *for* $i = 1, 2, \ldots, k$.

Proof. By induction on the construction of $\mathsf{x}_1 : \mathsf{A}_1, \mathsf{x}_2 : \mathsf{A}_2, \ldots, \mathsf{x}_\mathsf{k} : \mathsf{A}_\mathsf{k} \vdash \mathsf{M} : \mathsf{B}$. □

Lemma 2.8 (Substitution lemma). *If* $\Gamma, \mathsf{x} : \mathsf{A} \vdash \{\mathsf{P}\}_e : \mathsf{B}$ *and* $\Gamma \vdash \mathsf{Q} : \mathsf{A}$, *then* $\Gamma \vdash \{\mathsf{P}[\mathsf{Q}/\mathsf{x}]\}_e : \mathsf{B}$.

Proof. By induction on the length $|\mathsf{P}|$ with the help of Lemmas 2.6 and 2.7. □

Theorem 2.9 (Subject reduction). *If* $\Gamma \vdash \mathsf{M} : \mathsf{B}$ *and* $\mathsf{M} \rightarrow_{\beta\vartheta} \mathsf{R}$, *then* $\Gamma \vdash \mathsf{R} : \mathsf{B}$.

Proof. By induction on the structure of $\mathsf{M} \rightarrow_{\beta\vartheta} \mathsf{R}$ with the help of Lemma 2.8. □

Next, we show that the parallel $\beta\vartheta$-reduction $\rightrightarrows_{\beta\vartheta}$ is well defined (Theorems 2.11 and 2.12).

Lemma 2.10 (Hindley-Rosen). *Let* R_1 *and* R_2 *be binary relations on the set* $\mathscr{T}$ *of all terms, and let us write* $\rightarrow_{R_i}$ *for the contextual closure of* R_i *for* $i = 1, 2$. *If* $\rightarrow_{R_1}$ *and* $\rightarrow_{R_2}$ *are Church–Rosser, and for any* $\mathsf{M}, \mathsf{P}, \mathsf{Q} \in \mathscr{T}$ *the conjunction of* $\mathsf{M} \rightarrow^*_{\mathsf{R}_1} \mathsf{P}$ *and* $\mathsf{M} \rightarrow^*_{\mathsf{R}_2} \mathsf{Q}$ *implies the conjunction of* $\mathsf{P} \rightarrow^*_{\mathsf{R}_2} \mathsf{R}$ *and* $\mathsf{Q} \rightarrow^*_{\mathsf{R}_2} \mathsf{R}$ *for some* $\mathsf{R} \in \mathscr{T}$, *then* $\rightarrow_{R_1 \cup R_2}$ *is Church–Rosser.*

Proof. By a simple 'diagram chase'; see Hankin (1994) for the details. □

Theorem 2.11 (Church–Rosser). *The $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$ is Church–Rosser.*

Proof. First, it is easy to see that the ϑ*-reduction* $\rightarrow_{\vartheta} := \bigcup_{i=1}^{4} \rightarrow_{\vartheta_i}$ satisfies the diamond property, and thus it is Church–Rosser.

Next, let us show that the implication

$$\mathsf{M} \rightarrow_{\beta} \mathsf{P} \wedge \mathsf{M} \rightarrow_{\vartheta} \mathsf{Q} \Rightarrow \exists \mathsf{R}.\, \mathsf{P} \rightarrow_{\vartheta}^{*} \mathsf{R} \wedge \mathsf{Q} \rightarrow_{\beta} \mathsf{R} \tag{2}$$

holds for all terms M, P and Q, where note the asymmetry between $\rightarrow_{\vartheta}$ and $\rightarrow_{\beta}$ in (2), by the following case analysis on the relation between β- and ϑ-redexes in M:

- If the β-redex is inside the ϑ-redex, then it is easy to see that (2) holds;
- If the ϑ-redex is inside the body of the function subterm of the β-redex, then it suffices to show that $\rightarrow_{\vartheta}$ commutes with substitution, but it is straightforward;
- If ϑ-redex is inside the argument of the β-redex, then it may be duplicated by a finite number n, but whatever the number n is, (2) clearly holds;
- If the β- and ϑ-redexes are disjoint, then (2) trivially holds.

It then follows from (2) that the implication

$$\mathsf{M} \rightarrow_{\beta} \mathsf{P} \wedge \mathsf{M} \rightarrow_{\vartheta}^{*} \mathsf{Q} \Rightarrow \exists \mathsf{R}.\, \mathsf{P} \rightarrow_{\vartheta}^{*} \mathsf{R} \wedge \mathsf{Q} \rightarrow_{\beta} \mathsf{R} \tag{3}$$

holds, which in turn implies that the implication

$$\mathsf{M} \rightarrow_{\beta}^{*} \mathsf{P} \wedge \mathsf{M} \rightarrow_{\vartheta}^{*} \mathsf{Q} \Rightarrow \exists \mathsf{R}.\, \mathsf{P} \rightarrow_{\vartheta}^{*} \mathsf{R} \wedge \mathsf{Q} \rightarrow_{\beta}^{*} \mathsf{R} \tag{4}$$

holds for all terms M, P and Q. Applying Lemma 2.10 to (4) (or equivalently by the well-known 'diagram chase' argument on $\rightarrow_{\beta}^{*}$ and $\rightarrow_{\vartheta}^{*}$), we may conclude that the $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta} = \rightarrow_{\beta} \cup \rightarrow_{\vartheta}$ is Church–Rosser, completing the proof. □

Now, we show *strong normalisation* of $\rightarrow_{\beta\vartheta}$, i.e., there is no infinite chain of $\rightarrow_{\beta\vartheta}$:

Theorem 2.12 (SN). *The $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$ is strongly normalising (SN).*

Proof. By a moderate, straightforward modification of the proof of strong normalisation of the simply-typed λ-calculus given in Hankin (1994). □

Therefore, it follows from Theorems 2.11 and 2.12 that the normal form $nf(\mathsf{M})$ of each term M of FPCF (with respect to the $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$) uniquely exists. Moreover, we have

Theorem 2.13 (Normal forms are values). *The normal form $nf(\mathsf{M})$ of every programme M (with respect to the $\beta\vartheta$-reduction $\rightarrow_{\beta\vartheta}$) is a value.*

Proof. The theorem has been shown in Amadio and Curien (1998) during the proof to show that PCF Böhm trees are closed under composition. □

Therefore, we have shown that the operational semantics $\rightarrow$ is well defined:

Corollary 2.14 (Correctness of operational semantics). *If $\Gamma \vdash \{\mathsf{M}\}_e : \mathsf{B}$ is a programme, and $e > 1$ (resp. $e = 1$), then there is a unique programme (resp. value) $\Gamma \vdash \{\mathsf{M}'\}_{e-1} : \mathsf{B}$ that satisfies $\mathsf{M} \rightarrow \mathsf{M}'$.*

Proof. By Theorems 2.9, 2.11, 2.12 and 2.13. □

2.3 Dynamic bicategorical semantics of finitary PCF

Now, let us present a general, categorical recipe to give semantics of FPCF in a CCBoC in such a way that satisfies the DCP, specifically in the sense of Definition 2.16.

Definition 2.15 (Structures for FPCF). *A* structure *for FPCF in a CCBoC $\mathscr{C} = (\mathscr{C}, \mathscr{E})$ is a tuple $\mathscr{S} = (\mathscr{B}, 1, \times, \pi, \Rightarrow, ev, \underline{tt}, \underline{ff}, \vartheta)$ such that*

- $\mathscr{B} \in \mathscr{C}$;
- 1, $(\times, \pi_1, \pi_2)$ *and* $(\Rightarrow, ev)$ *are, respectively, a β-terminal object, a β-product (with β-projections) and a β-exponential (with β-evaluations) in $\mathscr{C}$;*
- $\underline{tt}, \underline{ff} : 1 \to \mathscr{B}$ *and* $\vartheta : \mathscr{B} \times (\mathscr{B} \times \mathscr{B}) \to \mathscr{B}$ *are values in $\mathscr{C}$.*

The interpretation $[\![\_]\!]^{\mathscr{S}}_{\mathscr{C}}$ of FPCF induced by $\mathscr{S}$ in $\mathscr{C}$ assigns an object $[\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}} \in \mathscr{C}$ to each type A, *an object $[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}} \in \mathscr{C}$ to each context Γ and a β-morphism $[\![\mathsf{M}]\!]^{\mathscr{S}}_{\mathscr{C}} : [\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}} \to [\![\mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}$ to each term* $\Gamma \vdash \mathsf{M} : \mathsf{B}$ *inductively as follows:*

- (Types). $[\![o]\!]^{\mathscr{S}}_{\mathscr{C}} := \mathscr{B}$ *and* $[\![\mathsf{A} \Rightarrow \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}} := [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}} \Rightarrow [\![\mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}$;
- (Contexts). $[\![\boldsymbol{\varepsilon}]\!]^{\mathscr{S}}_{\mathscr{C}} := 1$ *and* $[\![\Gamma, \mathsf{x} : \mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}} := [\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}} \times [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}$;
- (Terms).

$$\begin{aligned}
[\![\Gamma \vdash \mathsf{tt} : o]\!]^{\mathscr{S}}_{\mathscr{C}} &:= \mathscr{E}^{\omega}(!_{[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}}}; \underline{tt}) \\
[\![\Gamma \vdash \mathsf{ff} : o]\!]^{\mathscr{S}}_{\mathscr{C}} &:= \mathscr{E}^{\omega}(!_{[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}}}; \underline{ff}) \\
[\![\Gamma \vdash \lambda \mathsf{x}.\, \mathsf{M} : \mathsf{A} \Rightarrow \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}} &:= \Lambda_{[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}}([\![\Gamma, \mathsf{x} : \mathsf{A} \vdash \mathsf{M} : \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}) \\
[\![\Gamma \vdash \mathsf{MN} : \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}} &:= \langle [\![\Gamma \vdash \mathsf{M} : \mathsf{A} \Rightarrow \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\Gamma \vdash \mathsf{N} : \mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}} \rangle^{[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}}}_{[\![\mathsf{A} \Rightarrow \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}}; ev_{[\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}} \\
[\![\Gamma \vdash \mathsf{case}(\mathsf{x}\mathbf{V})[\mathsf{W}_1; \mathsf{W}_2] : o]\!]^{\mathscr{S}}_{\mathscr{C}} &:= \mathscr{E}^{\omega}(\langle [\![\Gamma \vdash \mathsf{x}\mathbf{V} : o]\!]^{\mathscr{S}}_{\mathscr{C}}, \langle [\![\Gamma \vdash \mathsf{W}_1 : o]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\Gamma \vdash \mathsf{W}_2 : o]\!]^{\mathscr{S}}_{\mathscr{C}} \rangle\rangle; \vartheta) \\
[\![\Gamma \vdash \mathsf{case}(\mathsf{M})[\mathsf{P}_1; \mathsf{P}_2] : o]\!]^{\mathscr{S}}_{\mathscr{C}} &:= \mathscr{E}^{\omega}(\langle [\![\Gamma \vdash \mathsf{M} : o]\!]^{\mathscr{S}}_{\mathscr{C}}, \langle [\![\Gamma \vdash \mathsf{P}_1 : o]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\Gamma \vdash \mathsf{P}_2 : o]\!]^{\mathscr{S}}_{\mathscr{C}} \rangle\rangle; \vartheta),
\end{aligned}$$

where $[\![\Gamma \vdash \mathsf{x} : \mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}} : [\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}} \to [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}$ (n.b., $\Gamma \vdash \mathsf{x} : \mathsf{A}$ is not a term of FPCF, but we need it for the application $\mathsf{x}\mathbf{V}$) is the obvious (possibly iterated) β-projection.

Moreover, the structure $\mathscr{S}$ is standard *if it satisfies the following three axioms:*

1. *The maps $\Lambda_{A,B,C}$ and $\langle \_, \_ \rangle^{C}_{A,B}$ in $\mathscr{C}$ are bijections for each triple $A, B, C \in \mathscr{C}$;*
2. *Each β-composition that occurs as the interpretation of a term is not a value;*
3. *If $\Gamma \vdash \mathsf{L} : \mathsf{A} \Rightarrow \mathsf{B}$ and $\Gamma \vdash \mathsf{R} : \mathsf{A}$ such that $\mathsf{L} \to \mathsf{L}' \wedge \mathsf{R} \to \mathsf{R}'$, $\mathsf{L} \equiv \mathsf{L}' \wedge \mathsf{R} \to \mathsf{R}'$ or $\mathsf{L} \to \mathsf{L}' \wedge \mathsf{R} \equiv \mathsf{R}'$ in FPCF, then*

$$\langle [\![\mathsf{L}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{R}]\!]^{\mathscr{S}}_{\mathscr{C}} \rangle^{[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}}}_{[\![\mathsf{A} \Rightarrow \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}}; ev_{[\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}} \neq \langle [\![\mathsf{L}']\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{R}']\!]^{\mathscr{S}}_{\mathscr{C}} \rangle^{[\![\Gamma]\!]^{\mathscr{S}}_{\mathscr{C}}}_{[\![\mathsf{A} \Rightarrow \mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}}; ev_{[\![\mathsf{A}]\!]^{\mathscr{S}}_{\mathscr{C}}, [\![\mathsf{B}]\!]^{\mathscr{S}}_{\mathscr{C}}}. \quad (5)$$

Remark. Recall that PCF Böhm trees are *normalised*, while β-composition in a BoC is non-normalising. It is why the evaluation $\mathscr{E}^{\omega}$ is applied to the interpretation of each PCF Böhm tree of FPCF in Definition 2.15.

It is easy to see that the interpretation $[\![\_]\!]^{\mathscr{S}}_{\mathscr{C}}$ followed by $\mathscr{E}^{\omega}$, i.e., $\mathscr{E}^{\omega}([\![\_]\!]^{\mathscr{S}}_{\mathscr{C}})$, coincides with the standard categorical interpretation of the equational theory Eq(FPCF) in the CCC $\mathscr{V}_{\mathscr{C}}$ (Jacobs, 1999; Lambek and Scott, 1988; Pitts, 2001). In this sense, we have refined the standard categorical semantics of simple type theories in CCCs.

At this point, let us recall the central concept of DCPs (see Section 1), specifically tailored for the interpretation of FPCF induced by a structure in a CCBoC:

Definition 2.16 (DCP for FPCF). *The interpretation* $[\![\_]\!]^{\mathscr{S}}_{\mathscr{C}}$ *of FPCF induced by a structure* $\mathscr{S}$ *for FPCF in a CCBoC* $\mathscr{C} = (\mathscr{C}, \mathscr{E})$ *satisfies the dynamic correspondence property (DCP) if* $\mathsf{M}_1 \to \mathsf{M}_2$ *implies* $[\![\mathsf{M}_1]\!]^{\mathscr{S}}_{\mathscr{C}} \neq [\![\mathsf{M}_2]\!]^{\mathscr{S}}_{\mathscr{C}}$ *and* $\mathscr{E}([\![\mathsf{M}_1]\!]^{\mathscr{S}}_{\mathscr{C}}) = [\![\mathsf{M}_2]\!]^{\mathscr{S}}_{\mathscr{C}}$.

Now, we reduce the DCP for FPCF to the following:

Definition 2.17 (PDCP for FPCF). *The interpretation* $[\![\_]\!]^{\mathscr{S}}_{\mathscr{C}}$ *of FPCF induced by a structure* $\mathscr{S}$ *for FPCF in a CCBoC* $\mathscr{C} = (\mathscr{C}, \mathscr{E})$ *satisfies the* pointwise dynamic correspondence property (PDCP) *if for each term* $\Gamma \vdash \{\mathsf{M}\}_e : \mathsf{B}$ *it satisfies*

$$\mathscr{E}([\![\mathsf{M}]\!]^{\mathscr{S}}_{\mathscr{C}}) = \begin{cases} \Lambda \circ \mathscr{E}([\![\mathsf{P}]\!]^{\mathscr{S}}_{\mathscr{C}}) & \textit{if } \mathsf{M} \equiv \lambda \mathsf{x}.\mathsf{P}; \\ [\![\mathsf{W}]\!]^{\mathscr{S}}_{\mathscr{C}} \textit{ such that } [\![\mathsf{W}]\!]^{\mathscr{S}}_{\mathscr{C}} \neq [\![\mathsf{M}]\!]^{\mathscr{S}}_{\mathscr{C}} & \textit{if } \mathsf{M} \equiv \mathsf{UV},\ e = 1 \textit{ and } \mathsf{UV} \to \mathsf{W}; \\ \langle \mathscr{E}([\![\mathsf{L}]\!]^{\mathscr{S}}_{\mathscr{C}}), \mathscr{E}([\![\mathsf{R}]\!]^{\mathscr{S}}_{\mathscr{C}}) \rangle; ev & \textit{if } \mathsf{M} \equiv \mathsf{LR} \textit{ and } e > 1; \\ [\![\mathsf{M}]\!]^{\mathscr{S}}_{\mathscr{C}} & \textit{otherwise.} \end{cases}$$

Theorem 2.18 (Standard semantics of FPCF). *The interpretation* $[\![\_]\!]^{\mathscr{S}}_{\mathscr{C}}$ *of FPCF induced by a standard structure* $\mathscr{S}$ *for FPCF in a CCBoC* $\mathscr{C} = (\mathscr{C}, \mathscr{E})$ *satisfies the DCP if it does the PDCP.*

Proof. In the following, we abbreviate $[\![\_]\!]^{\mathscr{S}}_{\mathscr{C}}$ as $[\![\_]\!]$. Assume that $[\![\_]\!]$ satisfies the PDCP. We show that $\mathsf{M} \to \mathsf{M}'$ implies the conjunction of $[\![\mathsf{M}]\!] \neq [\![\mathsf{M}']\!]$ and $\mathscr{E}([\![\mathsf{M}]\!]) = [\![\mathsf{M}']\!]$ for any programmes $\Gamma \vdash \{\mathsf{M}\}_e : \mathsf{B}$ and $\Gamma \vdash \{\mathsf{M}'\}_{e'} : \mathsf{B}$ in FPCF by induction on the construction of the programme M:

- If $\mathsf{M} \equiv \mathsf{tt}$, $\mathsf{M} \equiv \mathsf{ff}$ or $\mathsf{M} \equiv \mathsf{case}(\mathsf{x}\mathsf{V}_1\mathsf{V}_2 \ldots \mathsf{V}_k)[\mathsf{W}_1; \mathsf{W}_2]$, then there is no M' such that $\mathsf{M} \to \mathsf{M}'$.
- If $\Gamma \vdash \mathsf{M} \equiv \lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{P} : \mathsf{A} \Rightarrow \mathsf{C}$, then

$$\begin{aligned} \mathsf{M} \to \mathsf{M}' &\Rightarrow \mathsf{M}' \equiv \lambda \mathsf{x}.\, \mathsf{P}' \wedge \mathsf{P} \to \mathsf{P}' \text{ for some programme } \mathsf{P}' \text{ and variable } \mathsf{x} \\ &\Rightarrow \mathsf{M}' \equiv \lambda \mathsf{x}.\, \mathsf{P}' \wedge [\![\mathsf{P}]\!] \neq [\![\mathsf{P}']\!] \wedge \mathscr{E}([\![\mathsf{P}]\!]) = [\![\mathsf{P}']\!] \text{ for some } \mathsf{P}' \text{ and } \mathsf{x} \\ &\quad \text{(by the induction hypothesis)} \\ &\Rightarrow [\![\mathsf{P}]\!] \neq \Lambda^{-1}([\![\mathsf{M}']\!]) \wedge \mathscr{E}([\![\mathsf{P}]\!]) = \Lambda^{-1}([\![\mathsf{M}']\!]) \\ &\Rightarrow \Lambda^{-1}([\![\mathsf{M}]\!]) \neq \Lambda^{-1}([\![\mathsf{M}']\!]) \wedge \Lambda^{-1} \circ \mathscr{E}([\![\mathsf{M}]\!]) = \mathscr{E} \circ \Lambda^{-1}([\![\mathsf{M}]\!]) = \Lambda^{-1}([\![\mathsf{M}']\!]) \\ &\Rightarrow [\![\mathsf{M}]\!] \neq [\![\mathsf{M}']\!] \wedge \mathscr{E}([\![\mathsf{M}]\!]) = [\![\mathsf{M}']\!]. \end{aligned}$$

- If $\mathsf{M} \equiv \mathsf{LR}$, $\sharp(\mathsf{L}) \geqslant 1$ and $\sharp(\mathsf{R}) \geqslant 1$, then

$$\begin{aligned} \mathsf{M} \to \mathsf{M}' &\Rightarrow \mathsf{M}' \equiv \mathsf{L}'\mathsf{R}' \wedge \mathsf{L} \to \mathsf{L}' \wedge \mathsf{R} \to \mathsf{R}' \text{ for some programmes } \mathsf{L}' \text{ and } \mathsf{R}' \\ &\Rightarrow \mathsf{M}' \equiv \mathsf{L}'\mathsf{R}' \wedge [\![\mathsf{L}]\!] \neq [\![\mathsf{L}']\!] \wedge \mathscr{E}([\![\mathsf{L}]\!]) = [\![\mathsf{L}']\!] \wedge [\![\mathsf{R}]\!] \neq [\![\mathsf{R}']\!] \wedge \mathscr{E}([\![\mathsf{R}]\!]) = [\![\mathsf{R}']\!] \\ &\quad \text{for some } \mathsf{L}' \text{ and } \mathsf{R}' \text{ (by the induction hypothesis)} \\ &\Rightarrow [\![\mathsf{M}']\!] = \langle \mathscr{E}([\![\mathsf{L}]\!]), \mathscr{E}([\![\mathsf{R}]\!]) \rangle; ev \wedge [\![\mathsf{L}]\!] \neq \mathscr{E}([\![\mathsf{L}]\!]) \wedge [\![\mathsf{R}]\!] \neq \mathscr{E}([\![\mathsf{R}]\!]) \\ &\Rightarrow [\![\mathsf{M}']\!] = \mathscr{E}([\![\mathsf{M}]\!]) \wedge \mathscr{E}([\![\mathsf{M}]\!]) = \langle \mathscr{E}([\![\mathsf{L}]\!]), \mathscr{E}([\![\mathsf{R}]\!]) \rangle; ev \neq \langle [\![\mathsf{L}]\!], [\![\mathsf{R}]\!] \rangle; ev = [\![\mathsf{M}]\!] \\ &\Rightarrow [\![\mathsf{M}]\!] \neq [\![\mathsf{M}']\!] \wedge \mathscr{E}([\![\mathsf{M}]\!]) = [\![\mathsf{M}']\!]. \end{aligned}$$

- If $\mathsf{M} \equiv \mathsf{LR}$, $\sharp(\mathsf{L}) = 0$ and $\sharp(\mathsf{R}) \geqslant 1$, then we have

$$\begin{aligned} \mathsf{M} \to \mathsf{M}' &\Rightarrow \mathsf{M}' \equiv \mathsf{LR}' \wedge \mathsf{R} \to \mathsf{R}' \text{ for some programme } \mathsf{R}' \\ &\Rightarrow \mathsf{M}' \equiv \mathsf{LR}' \wedge [\![\mathsf{R}]\!] \neq [\![\mathsf{R}']\!] \wedge \mathscr{E}([\![\mathsf{R}]\!]) = [\![\mathsf{R}']\!] \text{ for some } \mathsf{R}' \\ &\quad \text{(by the induction hypothesis)} \\ &\Rightarrow [\![\mathsf{M}']\!] = \langle [\![\mathsf{L}]\!], \mathscr{E}([\![\mathsf{R}]\!]) \rangle ; ev = \mathscr{E}([\![\mathsf{M}]\!]) \wedge [\![\mathsf{R}]\!] \neq \mathscr{E}([\![\mathsf{R}]\!]) \\ &\Rightarrow [\![\mathsf{M}']\!] = \mathscr{E}([\![\mathsf{M}]\!]) \wedge \mathscr{E}([\![\mathsf{M}]\!]) = \langle [\![\mathsf{L}]\!], \mathscr{E}([\![\mathsf{R}]\!]) \rangle ; ev \neq \langle [\![\mathsf{L}]\!], [\![\mathsf{R}]\!] \rangle ; ev = [\![\mathsf{M}]\!] \\ &\Rightarrow [\![\mathsf{M}]\!] \neq [\![\mathsf{M}']\!] \wedge \mathscr{E}([\![\mathsf{M}]\!]) = [\![\mathsf{M}']\!]. \end{aligned}$$

- If $\mathsf{M} \equiv \mathsf{LR}$, $\sharp(\mathsf{L}) \geqslant 1$ and $\sharp(\mathsf{R}) = 0$, then it is handled similarly to the above case.
- If $\mathsf{M} \equiv \mathsf{LR}$, $\sharp(\mathsf{L}) = 0$ and $\sharp(\mathsf{R}) = 0$, then, by the PDCP, we have

$$\mathsf{M} \to \mathsf{M}' \Rightarrow [\![\mathsf{M}]\!] \neq [\![\mathsf{M}']\!] \wedge \mathscr{E}([\![\mathsf{M}]\!]) = [\![\mathsf{M}']\!],$$

which completes the proof. □

To summarise the present section, we have defined bicategorical 'universes' of dynamic, intensional computations, viz., (CC)BoCs, presented the language FPCF, and given an interpretation of the latter in the former and a sufficient condition, viz., the PDCP, for the interpretation to satisfy the DCP. Consequently, our research problem (described in Section 1) has been reduced to giving a standard structure for FPCF in a game-semantic CCBoC that satisfies the PDCP.

3. Dynamic Games and Strategies

Main contributions of the present work start from this section, which introduces dynamic games and strategies, and studies their algebraic structures. The idea of dynamic games and strategies is to introduce the distinction between *internal* and *external* moves to conventional games and strategies. External moves correspond to ordinary moves in conventional games; internal moves constitute the 'internal communication' between dynamic strategies, representing *intensionality* of computation, and they are to be *a posteriori* 'hidden' by the *hiding operation*, capturing *dynamics* of computation. Conceptually, external moves are 'official' ones for the underlying dynamic game, while internal moves are supposed to be 'invisible' to Opponent because they represent how Player 'internally' computes the next external move.

Dynamic games and strategies are based on the variant given in Abramsky and McCusker (1999), which we call *static* games and strategies (more generally, to distinguish our 'dynamic concepts' from conventional ones, we add the word *static* in front of the corresponding notions given in Abramsky and McCusker (1999), e.g., static arenas, static legal positions, etc.). This choice is because their variant combines good points of the two best-known variants: *AJM-games* (Abramsky et al., 2000) and *HO-games* (Hyland and Ong, 2000): It has the *linear decomposition* of implication (Girard, 1987), and also it is *flexible* enough to model a wide range of programming features systematically (Abramsky and McCusker, 1999). We have chosen this variant with the hope that our framework is also applicable to various formal systems and programming languages.

3.1 Dynamic arenas and legal positions

First, just like static games (Abramsky and McCusker, 1999), dynamic games are based on (the 'dynamic generalisations' of) *arenas* and *legal positions*. An arena defines the basic components of a game, which in turn induces its legal positions that specify the basic rules of the game. Let us first introduce these two preliminary concepts.

Definition 3.1 (Dynamic arenas). *A* dynamic arena *is a triple $G = (M_G, \lambda_G, \vdash_G)$ such that*

- *M_G is a set, whose elements are called* moves;

- *λ_G is a function $M_G \to \{\mathsf{O},\mathsf{P}\} \times \{\mathsf{Q},\mathsf{A}\} \times \mathbb{N}$, called the* labelling function, *where* O, P, Q *and* A *are arbitrarily fixed, pairwise distinct symbols, that satisfies*

$$\mu(G) := \mathsf{Sup}(\{\, \lambda_G^{\mathbb{N}}(m) \mid m \in M_G \,\}) \in \mathbb{N};$$

- *$\vdash_G$ is a subset of $(\{\star\} \cup M_G) \times M_G$, where $\star$ is an arbitrarily fixed element such that $\star \notin M_G$, called the* enabling relation, *that satisfies*
  - (E1). *If $\star \vdash_G m$, then $\lambda_G(m) = \mathsf{OQ}0$ and $n = \star$ whenever $n \vdash_G m$;*
  - (E2). *If $m \vdash_G n$ and $\lambda_G^{\mathsf{QA}}(n) = \mathsf{A}$, then $\lambda_G^{\mathsf{QA}}(m) = \mathsf{Q}$ and $\lambda_G^{\mathbb{N}}(m) = \lambda_G^{\mathbb{N}}(n)$;*
  - (E3). *If $m \vdash_G n$ and $m \neq \star$, then $\lambda_G^{\mathsf{OP}}(m) \neq \lambda_G^{\mathsf{OP}}(n)$;*
  - (E4). *If $m \vdash_G n$, $m \neq \star$ and $\lambda_G^{\mathbb{N}}(m) \neq \lambda_G^{\mathbb{N}}(n)$, then $\lambda_G^{\mathsf{OP}}(m) = \mathsf{O}$,*

where we define $\lambda_G^{\mathsf{OP}} := \pi_1 \circ \lambda_G : M_G \to \{\mathsf{O},\mathsf{P}\}$, $\lambda_G^{\mathsf{QA}} := \pi_2 \circ \lambda_G : M_G \to \{\mathsf{Q},\mathsf{A}\}$ and $\lambda_G^{\mathbb{N}} := \pi_3 \circ \lambda_G : M_G \to \mathbb{N}$. A move $m \in M_G$ is initial *if $\star \vdash_G m$, an* O-move *(resp. a* P-move*) if $\lambda_G^{\mathsf{OP}}(m) = \mathsf{O}$ (resp. if $\lambda_G^{\mathsf{OP}}(m) = \mathsf{P}$), a* question *(resp. an* answer*) if $\lambda_G^{\mathsf{QA}}(m) = \mathsf{Q}$ (resp. if $\lambda_G^{\mathsf{QA}}(m) = \mathsf{A}$), and* internal *or* $\lambda_G^{\mathbb{N}}(m)$-internal *(resp.* external*) if $\lambda_G^{\mathbb{N}}(m) > 0$ (resp. if $\lambda_G^{\mathbb{N}}(m) = 0$). Any finite sequence $\boldsymbol{s} \in M_G^*$ of moves is* d-complete *if it ends with a move m such that $\lambda_G^{\mathbb{N}}(m) = 0$ or $\lambda_G^{\mathbb{N}}(m) > d$, where $d \in \mathbb{N} \cup \{\omega\}$, and ω is the least transfinite ordinal.*

Recall that a static arena G (Abramsky and McCusker, 1999) determines possible *moves* of a game, each of which is Opponent's/Player's question/answer, where the third parity $\lambda_G^{\mathbb{N}}$ is not included, and specifies which move n can be performed for each move m by the relation $m \vdash_G n$ (and $\star \vdash_G m$ means that m can *initiate* a play). The axioms on a static arena are the following:

- (E1). An initial move must be Opponent's question, and an initial move cannot be enabled by any move;
- (The first part of E2). An answer must be performed for a question;
- (E3). An O-move must be performed for a P-move, and vice versa.

Hence, a dynamic arena is a static arena equipped with the *priority order* $\lambda_G^{\mathbb{N}}$ on moves that satisfies additional axioms on the priority order. The priority order $\lambda_G^{\mathbb{N}}$ determines the 'priority order' of moves to be 'hidden' by the hiding operations on dynamic games (Definition 3.26) and dynamic strategies (Definition 3.66), which explains the terminology. We need all natural numbers for $\lambda_G^{\mathbb{N}}$, not only the internal/external (I/E) distinction, to define a *step-by-step* execution of the hiding operations, as we shall see. Conversely, dynamic arenas are generalised static arenas: A static arena is equivalent to a dynamic arena whose moves are all external.

The additional axioms for dynamic arenas G are intuitively natural ones:

- We require a *finite* upper bound $\mu(G)$ of the priority orders since it is conceptually natural and technically necessary for concatenation of dynamic games (Definition 3.47) to be well defined and for the hiding operation on dynamic games to terminate;
- The axiom E1 adds the equation $\lambda_G^{\mathbb{N}}(m_0) = 0$ for all $m_0 \in M_G^{\mathsf{Init}} := \{\, m \in M_G \mid \star \vdash m \,\}$ since Opponent cannot 'see' internal moves;
- The second requirement of the axiom E2 states that the priority orders between a 'QA-pair' must coincide, which is intuitively reasonable;
- The additional axiom E4 states that only Player can make a move for a previous one if they have different priority orders because internal moves are 'invisible' to Opponent (as we shall see, if $\lambda_G^{\mathbb{N}}(m_1) = k_1 < k_2 = \lambda_G^{\mathbb{N}}(m_2)$, then after the k_1-many iteration of the hiding operation, m_1 and m_2 become external and internal, respectively, i.e., the I/E-parity is *relative*, which is why E4 is not only concerned with the I/E-parity but more fine-grained priority orders).

Technically, the additional axioms are vital for Lemma 3.13, which is in turn a key step towards a main theorem: closure of dynamic games under the hiding operation (Theorem 3.28).

Convention. Henceforth, an *arena* refers to a dynamic arena by default.

Example 3.2. The *terminal arena* T is given by $T := (\emptyset, \emptyset, \emptyset)$.

Example 3.3. The *flat arena* $\mathit{flat}(S)$ on a given set S is given by $M_{\mathit{flat}(S)} := \{q\} \cup S$, where q is any element such that $q \notin S$; $\lambda_{\mathit{flat}(S)} : q \mapsto \mathsf{OQ0}, (m \in S) \mapsto \mathsf{PA0}$; $\vdash_{\mathit{flat}(S)} := \{(\star, q)\} \cup \{(q, m) \mid m \in S\}$. For instance, $N := \mathit{flat}(\mathbb{N})$ is the arena of natural numbers, and $\mathbf{2} := \mathit{flat}(\mathbb{B})$, where $\mathbb{B} := \{tt, \mathit{ff}\}$, is the arena of booleans (tt and ff are arbitrarily fixed elements for 'true' and 'false,' respectively).

As mentioned before, interactions between Opponent and Player in a (dynamic or static) game are represented by finite sequences of moves of the underlying arena, equipped with *pointers* (Definition 3.5) that specify the occurrence of a move in the sequence for which each occurrence of a non-initial move in the sequence is performed. Technically, pointers are to distinguish similar yet different computations; see Abramsky and McCusker (1999), Curien (2006) for this point.

Let us now introduce such finite sequences equipped with pointers, called *j-sequences*:

Definition 3.4 (Occurrences of moves). *Given a finite sequence $\boldsymbol{s} \in M_G^*$ of moves of an arena G, an* occurrence (of a move) *in $\boldsymbol{s}$ is a pair $(\boldsymbol{s}(i), i)$ such that $i \in \overline{|\boldsymbol{s}|}$. More specifically, we call the pair $(\boldsymbol{s}(i), i)$ an* initial occurrence *(resp. a* non-initial occurrence*) in $\boldsymbol{s}$ if $\star \vdash_G \boldsymbol{s}(i)$ (resp. otherwise).*

Definition 3.5 (J-sequences (Abramsky and McCusker, 1999; Hyland and Ong, 2000)). *A* justified (j-) sequence *in an arena G is a pair $\boldsymbol{s} = (\boldsymbol{s}, \mathcal{J}_{\boldsymbol{s}})$ of a finite sequence $\boldsymbol{s} \in M_G^*$ and a map $\mathcal{J}_{\boldsymbol{s}} : \overline{|\boldsymbol{s}|} \to \{0\} \cup \overline{|\boldsymbol{s}| - 1}$ such that, for all $i \in \overline{|\boldsymbol{s}|}$, $\star \vdash_G \boldsymbol{s}(i)$ if $\mathcal{J}_{\boldsymbol{s}}(i) = 0$, and $0 < \mathcal{J}_{\boldsymbol{s}}(i) < i$ and $\boldsymbol{s}(\mathcal{J}_{\boldsymbol{s}}(i)) \vdash_G \boldsymbol{s}(i)$ otherwise. The occurrence $(\boldsymbol{s}(\mathcal{J}_{\boldsymbol{s}}(i)), \mathcal{J}_{\boldsymbol{s}}(i))$ is called the* justifier *of a non-initial occurrence $(\boldsymbol{s}(i), i)$ in $\boldsymbol{s}$. We also say that $(\boldsymbol{s}(i), i)$ is* justified by *$(\boldsymbol{s}(\mathcal{J}_{\boldsymbol{s}}(i)), \mathcal{J}_{\boldsymbol{s}}(i))$, or there is a* pointer *from the former to the latter.*

The idea is that each non-initial occurrence in a j-sequence must be performed for a specific previous occurrence, viz., its justifier, in the j-sequence.

Convention. By abuse of notation, we usually keep the pointer structure $\mathcal{J}_{\boldsymbol{s}}$ of each j-sequence $\boldsymbol{s} = (\boldsymbol{s}, \mathcal{J}_{\boldsymbol{s}})$ implicit and often abbreviate occurrences $(\boldsymbol{s}(i), i)$ in $\boldsymbol{s}$ as $\boldsymbol{s}(i)$. Also, we usually write $\mathcal{J}_{\boldsymbol{s}}(\boldsymbol{s}(i)) = \boldsymbol{s}(j)$ if $\mathcal{J}_{\boldsymbol{s}}(i) = j$. This convention is mathematically imprecise, but it does not bring any serious confusion in practice. In fact, it has been the standard convention in the literature.

Notation 3.6. *We write $\mathbb{J}_G$ for the set of all j-sequences in an arena G. We write $\boldsymbol{s} = \boldsymbol{t}$ for any $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{J}_G$ if $\boldsymbol{s}$ and $\boldsymbol{t}$ are the same j-sequence in G, i.e., $\boldsymbol{s} = \boldsymbol{t}$ and $\mathcal{J}_{\boldsymbol{s}} = \mathcal{J}_{\boldsymbol{t}}$.*

Next, for technical convenience, we introduce a substructure relation between j-sequences. A j-sequence $\boldsymbol{t}$ is a substructure, or a *j-subsequence*, of a j-sequence $\boldsymbol{s}$ if $\boldsymbol{t}$ is a subsequence of $\boldsymbol{s}$, and its pointer is obtained from that of $\boldsymbol{s}$ by 'concatenating' it. Formally, we define

Definition 3.7 (J-subsequences). *Let G be an arena. A* justified (j-) subsequence *of a j-sequence $\boldsymbol{s} \in \mathbb{J}_G$ in G is a j-sequence $\boldsymbol{t} \in \mathbb{J}_G$ such that $\boldsymbol{t}$ is a subsequence of $\boldsymbol{s}$, and, for all $i, j \in \mathbb{N}$, $\mathcal{J}_{\boldsymbol{t}}(i) = j$ if and only if $\mathcal{J}_{\boldsymbol{s}}^n(i) = j$ for some $n \in \mathbb{N}^+$.*

Let us now consider justifiers, j-sequences and arenas from the 'external point of view':

Definition 3.8 (External justifiers). *Let G be an arena and assume $\boldsymbol{s} \in \mathbb{J}_G$ and $d \in \mathbb{N} \cup \{\omega\}$. Note that each non-initial occurrence n in $\boldsymbol{s}$ has a unique sequence of justifiers $mm_1m_2 \dots m_kn$ ($k \geqslant 0$), i.e., $\mathcal{J}_{\boldsymbol{s}}(n) = m_k$, $\mathcal{J}_{\boldsymbol{s}}(m_k) = m_{k-1}$, ..., $\mathcal{J}_{\boldsymbol{s}}(m_2) = m_1$ and $\mathcal{J}_{\boldsymbol{s}}(m_1) = m$, such that $\lambda_G^{\mathbb{N}}(m) = 0$ or $\lambda_G^{\mathbb{N}}(m) > d$, and $0 < \lambda_G^{\mathbb{N}}(m_i) \leqslant d$ for $i = 1, 2, \dots, k$. We call m the* d-external justifier *of n in $\boldsymbol{s}$.*

Notation 3.9. *We write $\mathcal{J}_{\boldsymbol{s}}^{\ominus d}(n)$ for the d-external justifier of a non-initial occurrence n in a j-sequence $\boldsymbol{s}$.*

Note that d-external justifiers are a simple generalisation of justifiers because 0-external justifiers coincide with justifiers (as there is no '0-internal' move). More generally, d-external justifiers become justifiers after the d-times iteration of the hiding operation, as we shall see shortly.

Definition 3.10 (External j-subsequences). *Let G be an arena, $\boldsymbol{s} \in \mathbb{J}_G$ and $d \in \mathbb{N} \cup \{\omega\}$. The* d-external j-subsequence *of $\boldsymbol{s}$ is the j-subsequence $\mathcal{H}_G^d(\boldsymbol{s})$ of $\boldsymbol{s}$ obtained from $\boldsymbol{s}$ by deleting occurrences of internal moves m such that $0 < \lambda_G^{\mathbb{N}}(m) \leqslant d$ and equipping the pointer $\mathcal{J}_{\mathcal{H}_G^d(\boldsymbol{s})} : n \mapsto \mathcal{J}_{\boldsymbol{s}}^{\ominus d}(n)$ (n.b., strictly speaking, $\mathcal{J}_{\mathcal{H}_G^d(\boldsymbol{s})}$ is the* restriction *of $\mathcal{J}_{\boldsymbol{s}}^{\ominus d}$).*

Definition 3.11 (External arenas). *Let G be an arena, and $d \in \mathbb{N} \cup \{\omega\}$. The* d-external arena *of G is the arena $\mathcal{H}^d(G)$ given by*

- $M_{\mathcal{H}^d(G)} := \{ m \in M_G \mid \lambda_G^{\mathbb{N}}(m) = 0 \vee \lambda_G^{\mathbb{N}}(m) > d \}$;
- $\lambda_{\mathcal{H}^d(G)} := \lambda_G^{\ominus d} \upharpoonright M_{\mathcal{H}^d(G)}$, *where*

$$\lambda_G^{\ominus d} := \langle \lambda_G^{\mathsf{OP}}, \lambda_G^{\mathsf{QA}}, n \mapsto \lambda_G^{\mathbb{N}}(n) \ominus d \rangle \qquad x \ominus d := \begin{cases} x - d & \text{if } x \geqslant d; \\ 0 & \text{otherwise} \end{cases} \quad (x \in \mathbb{N});$$

- $m \vdash_{\mathcal{H}^d(G)} n \overset{\mathrm{df.}}{\Leftrightarrow} \exists k \in \mathbb{N}, m_1, m_2, \dots, m_{2k} \in M_G \setminus M_{\mathcal{H}^d(G)}.\, m \vdash_G m_1 \wedge \forall i \in \overline{2k-1}.\, m_i \vdash_G m_{i+1} \wedge m_{2k} \vdash_G n$ ($\Leftrightarrow m \vdash_G n$ *if* $k = 0$).

That is, the d-external arena $\mathcal{H}^d(G)$ is obtained from the arena G by deleting internal moves m such that $0 < \lambda_G^{\mathbb{N}}(m) \leqslant d$, decreasing by d the priority orders of the remaining moves and 'concatenating' the enabling relation to form the 'd-external' one.

Convention. Given $d \in \mathbb{N} \cup \{\omega\}$, we regard $\mathcal{H}^d$ as an operation on dynamic arenas G, and $\mathcal{H}_G^d$ as an operation on j-sequences $\boldsymbol{s} \in \mathbb{J}_G$.

Example 3.12. Define an arena G by $M_G := \{a, b, c, d\}$, $\lambda_G : a \mapsto \mathsf{OQ0}, b \mapsto \mathsf{PQ1}, c \mapsto \mathsf{OQ1}, d \mapsto \mathsf{PQ0}$, $\vdash_G := \{(\star, a), (a, b), (b, c), (c, d)\}$, $P_G := \mathsf{Pref}(\{abcd\})$ and $\simeq_G := \{ (\boldsymbol{s}, \boldsymbol{t}) \in P_G \times P_G \mid \boldsymbol{s} = \boldsymbol{t} \}$. A j-sequence $\boldsymbol{s}$ in G consists of the sequence $abcd$ and the pointer $d \mapsto c, c \mapsto b, b \mapsto a$. The 1-external j-subsequence $\mathcal{H}_G^1(\boldsymbol{s})$ consists of the sequence ad and the pointer $d \mapsto a$. Note that $\mathcal{H}_G^1(\boldsymbol{s})$ is a j-sequence of the 1-external arena $\mathcal{H}^1(G)$. It is not a coincidence; see Lemma 3.13.

Now, let us establish the following key technical lemma towards defining the hiding operation on dynamic games:

Lemma 3.13 (External closure lemma). *If G is an arena, then, for all $d \in \mathbb{N} \cup \{\omega\}$, so is $\mathcal{H}^d(G)$, and $\mathcal{H}_G^d(\boldsymbol{s}) \in \mathbb{J}_{\mathcal{H}^d(G)}$ for all $\boldsymbol{s} \in \mathbb{J}_G$.*

Proof. The case $d = 0$ is trivial; thus, assume $d > 0$. Clearly, the set $M_{\mathcal{H}^d(G)}$ and the map $\lambda_{\mathcal{H}^d(G)}$ satisfy the required axioms. Now, let us verify the axioms for the enabling relation $\vdash_{\mathcal{H}^d(G)}$:

- (E1). We have $\star \vdash_{\mathcal{H}^d(G)} m \Leftrightarrow \star \vdash_G m$ (because $\Leftarrow$ is immediate, and $\Rightarrow$ holds by E4 on G as initial moves are all external). Thus, if $\star \vdash_{\mathcal{H}^d(G)} m$, then $\lambda_{\mathcal{H}^d(G)}(m) = \lambda_G^{\ominus d}(m) = \mathsf{OQ0}$, and $n \vdash_{\mathcal{H}^d(G)} m \Rightarrow n = \star$.
- (E2). Assume $m \vdash_{\mathcal{H}^d(G)} n$ and $\lambda^{\mathsf{QA}}_{\mathcal{H}^d(G)}(n) = \mathsf{A}$. If $m \vdash_G n$, then $\lambda^{\mathsf{QA}}_{\mathcal{H}^d(G)}(m) = \lambda^{\mathsf{QA}}_G(m) = \mathsf{Q}$ and $\lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(m) = \lambda^{\mathbb{N}}_G(m) \ominus d = \lambda^{\mathbb{N}}_G(n) \ominus d = \lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(n)$. Otherwise, i.e., there are some $k \in \mathbb{N}^+$, $m_1, m_2, \ldots, m_{2k} \in M_G \setminus M_{\mathcal{H}^d(G)}$ such that $m \vdash_G m_1$, $m_i \vdash_G m_{i+1}$ for all $i \in \overline{2k-1}$ and $m_{2k} \vdash_G n$, then in particular $m_{2k} \vdash_G n$ and $\lambda^{\mathsf{QA}}_G(n) = \mathsf{A}$, but $\lambda^{\mathbb{N}}_G(m_{2k}) \neq \lambda^{\mathbb{N}}_G(n)$, a contradiction.
- (E3). Assume $m \vdash_{\mathcal{H}^d(G)} n$ and $m \neq \star$. If $m \vdash_G n$, then
$$\lambda^{\mathsf{OP}}_{\mathcal{H}^d(G)}(m) = \lambda^{\mathsf{OP}}_G(m) \neq \lambda^{\mathsf{OP}}_G(n) = \lambda^{\mathsf{OP}}_{\mathcal{H}^d(G)}(n).$$
If there are some $k \in \mathbb{N}^+$, $m_1, m_2, \ldots, m_{2k} \in M_G \setminus M_{\mathcal{H}^d(G)}$ such that $m \vdash_G m_1$, $m_i \vdash_G m_{i+1}$ for all $i \in \overline{2k-1}$ and $m_{2k} \vdash_G n$, then
$$\lambda^{\mathsf{OP}}_{\mathcal{H}^d(G)}(m) = \lambda^{\mathsf{OP}}_G(m) = \lambda^{\mathsf{OP}}_G(m_2) = \lambda^{\mathsf{OP}}_G(m_4) = \cdots = \lambda^{\mathsf{OP}}_G(m_{2k}) \neq \lambda^{\mathsf{OP}}_G(n) = \lambda^{\mathsf{OP}}_{\mathcal{H}^d(G)}(n).$$
- (E4). Assume $m \vdash_{\mathcal{H}^d(G)} n$, $m \neq \star$ and $\lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(m) \neq \lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(n)$. Then $\lambda^{\mathbb{N}}_G(m) \neq \lambda^{\mathbb{N}}_G(n)$. If $m \vdash_G n$, then it is trivial; otherwise, i.e., there are $k \in \mathbb{N}^+$, $m_1, m_2, \ldots, m_{2k} \in M_G \setminus M_{\mathcal{H}^d(G)}$ with the same property as in the case of E3 above, we have $\lambda^{\mathsf{OP}}_{\mathcal{H}^d(G)}(m) = \lambda^{\mathsf{OP}}_G(m) = \mathsf{O}$ by E3 on G since $\lambda^{\mathbb{N}}_G(m) \neq \lambda^{\mathbb{N}}_G(m_1)$.

Hence, we have shown that the structure $\mathcal{H}^d(G)$ forms a well defined arena.

Next, let $\boldsymbol{s} \in \mathbb{J}_G$; we have to show $\mathcal{H}^d_G(\boldsymbol{s}) \in \mathbb{J}_{\mathcal{H}^d(G)}$. Assume that m is a non-initial occurrence in $\mathcal{H}^d_G(\boldsymbol{s})$. By the definition, the d-external justifier $m_0 := \mathcal{J}_{\mathcal{H}^d_G(\boldsymbol{s})}(m)$ occurs in $\mathcal{H}^d_G(\boldsymbol{s})$. If m is a P-move, then the sequence of justifiers $m_0 \vdash_G m_1 \vdash_G \cdots \vdash_G m_k \vdash m$ satisfies $\mathsf{Even}(k)$ by E3 and E4 on G, so that $m_0 \vdash_{\mathcal{H}^d(G)} m$ by the definition. If m is an O-move, then the justifier $m'_0 := \mathcal{J}_{\boldsymbol{s}}(m)$ satisfies $\lambda^{\mathbb{N}}_G(m'_0) = \lambda^{\mathbb{N}}_G(m)$ by E4 on G, and so $m'_0 \vdash_{\mathcal{H}^d(G)} m$ by the definition. Since m is arbitrary, we have shown that $\mathcal{H}^d_G(\boldsymbol{s}) \in \mathbb{J}_{\mathcal{H}^d(G)}$, completing the proof. □

Let us proceed to introduce a useful lemma:

Lemma 3.14 (Stepwise hiding on arenas). *Given an arena G, we have $\widetilde{\mathcal{H}^i}(G) = \mathcal{H}^i(G)$ for all $i \in \mathbb{N}$, where $\widetilde{\mathcal{H}^i}$ denotes the i-times iteration of $\mathcal{H}^1$.*

Proof. By induction on i. □

Thus, we may just focus on $\mathcal{H}^1$: Henceforth, we write $\mathcal{H}$ for $\mathcal{H}^1$ and call it the *hiding operation (on arenas)*; $\mathcal{H}^i$ for each $i \in \mathbb{N}$ denotes the i-times iteration of $\mathcal{H}$.

We may establish a similar inductive property for j-sequences:

Lemma 3.15 (Stepwise hiding on j-sequences). *Given a j-sequence $\boldsymbol{s} \in \mathbb{J}_G$ in an arena G, we have $\mathcal{H}^{i+1}_G(\boldsymbol{s}) = \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i_G(\boldsymbol{s}))$ for all $i \in \mathbb{N}$.*

Proof. By induction on i, where $\mathcal{H}^{i+1}_G(\boldsymbol{s}), \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i_G(\boldsymbol{s})) \in \mathbb{J}_{\mathcal{H}^{i+1}(G)}$ by Lemmas 3.13 and 3.14. □

Lemma 3.15 implies that the equation

$$\mathscr{H}_G^i(\boldsymbol{s}) = \mathscr{H}^1_{\mathscr{H}^{i-1}(G)} \circ \mathscr{H}^1_{\mathscr{H}^{i-2}(G)} \circ \cdots \circ \mathscr{H}^1_{\mathscr{H}^1(G)} \circ \mathscr{H}^1_G(\boldsymbol{s}) \tag{6}$$

holds for any arena G, $\boldsymbol{s} \in \mathbb{J}_G$ and $i \in \mathbb{N}$ (n.b., the equation (6) means $\boldsymbol{s} = \boldsymbol{s}$ if $i = 0$). Thus, we may focus on the operation $\mathscr{H}^1_G$ on j-sequences, where G ranges over all arenas. Henceforth, we write $\mathscr{H}_G$ for $\mathscr{H}^1_G$ and call it the *hiding operation on j-sequences in G*; $\mathscr{H}^i_G$ for each $i \in \mathbb{N}$ denotes the operation on the right-hand side of the equation (6).

Now, to deal with external j-subsequences in a mathematically rigorous manner, let us extend the hiding operation on j-sequences to that on j-subsequences (Definition 3.7):

Definition 3.16 (Pointwise hiding on j-sequences). *Let* $\boldsymbol{s}$ *be a j-sequence in an arena* G. *Given an occurrence* m *in* $\boldsymbol{s}$, *we define another j-sequence* $\hat{\mathscr{H}}^m_G(\boldsymbol{s})$ *in* G *by case analysis: If* m *is 1-internal, then* $\hat{\mathscr{H}}^m_G(\boldsymbol{s})$ *is the j-subsequence of* $\boldsymbol{s}$ *that consists of occurrences in* $\boldsymbol{s}$ *that differ from* m, *and it is* $\boldsymbol{s}$ *otherwise. Moreover, given a subsequence* $\boldsymbol{t} = m_1 m_2 \dots m_k$ *of (the underlying finite sequence of)* $\boldsymbol{s}$ *and a permutation* σ *on* $\overline{k}$, *we define* $\hat{\mathscr{H}}^{\boldsymbol{t},\sigma}_G(\boldsymbol{s}) := \hat{\mathscr{H}}^{m_{\sigma(k)}}_G \circ \cdots \circ \hat{\mathscr{H}}^{m_{\sigma(2)}}_G \circ \hat{\mathscr{H}}^{m_{\sigma(1)}}_G(\boldsymbol{s})$.

The point here is that the hiding operation on j-sequences can be executed in the 'move-wise' fashion in any order:

Lemma 3.17 (Move-wise lemma). *Let* G *be an arena, and* $\boldsymbol{s} \in \mathbb{J}_G$.

(1) $\hat{\mathscr{H}}^{\boldsymbol{t},\sigma_1}_G(\boldsymbol{s}) = \hat{\mathscr{H}}^{\boldsymbol{t},\sigma_2}_G(\boldsymbol{s})$ *for any subsequence* $\boldsymbol{t}$ *of* $\boldsymbol{s}$ *and permutations* σ_1 *and* σ_2 *on* $\overline{|\boldsymbol{t}|}$;
(2) $\hat{\mathscr{H}}^{\boldsymbol{s},\sigma}_G(\boldsymbol{s}) = \mathscr{H}_G(\boldsymbol{s})$ *for any permutation* σ *on* $\overline{|\boldsymbol{s}|}$.

Proof. Immediate from the definition. □

Lemma 3.17 gives a 'move-wise' procedure to execute the hiding operation $\mathscr{H}_G$ on j-sequences in a given arena G, where the order of deleting occurrences is irrelevant. Then, e.g., it follows that $\mathscr{H}_G(\boldsymbol{stuv}) = \hat{\mathscr{H}}^{\boldsymbol{v},\nu}_G \circ \hat{\mathscr{H}}^{\boldsymbol{u},\mu}_G \circ \hat{\mathscr{H}}^{\boldsymbol{t},\tau}_G \circ \hat{\mathscr{H}}^{\boldsymbol{s},\sigma}_G(\boldsymbol{stuv})$ for any arena G, $\boldsymbol{stuv} \in \mathbb{J}_G$ and permutations σ, τ, μ and ν on $\overline{|\boldsymbol{s}|}$, $\overline{|\boldsymbol{t}|}$, $\overline{|\boldsymbol{u}|}$ and $\overline{|\boldsymbol{v}|}$, respectively, which is useful in the rest of the paper.

Convention. By Lemma 3.17, we henceforth dispense with the notation $\hat{\mathscr{H}}^{\boldsymbol{s},\sigma}_G$, where G ranges over arenas, $\boldsymbol{s}$ over j-sequences in G and σ over permutations on $\overline{|\boldsymbol{s}|}$, admitting any order of 'move-wise' execution of $\mathscr{H}_G$. We also write, abusing notation, $\mathscr{H}_G(\boldsymbol{s}).\mathscr{H}_G(\boldsymbol{t}).\mathscr{H}_G(\boldsymbol{u}).\mathscr{H}_G(\boldsymbol{v})$ for $\hat{\mathscr{H}}^{\boldsymbol{v},\nu}_G \circ \hat{\mathscr{H}}^{\boldsymbol{u},\mu}_G \circ \hat{\mathscr{H}}^{\boldsymbol{t},\tau}_G \circ \hat{\mathscr{H}}^{\boldsymbol{s},\sigma}_G(\boldsymbol{stuv})$ given above, so that $\mathscr{H}_G(\boldsymbol{stuv}) = \mathscr{H}_G(\boldsymbol{s}).\mathscr{H}_G(\boldsymbol{t}).\mathscr{H}_G(\boldsymbol{u}).\mathscr{H}_G(\boldsymbol{v})$.

Next, let us recall the notion of 'relevant part' of previous moves, called *views*:

Definition 3.18 (Views (Abramsky and McCusker, 1999)). *Given a j-sequence* $\boldsymbol{s}$ *in an arena* G, *the* Player (P-) view $\lceil \boldsymbol{s} \rceil_G$ *and the* Opponent (O-) view $\lfloor \boldsymbol{s} \rfloor_G$ *(we often omit the subscript* G*) are given by the following induction on* $|\boldsymbol{s}|$:

- $\lceil \boldsymbol{\varepsilon} \rceil_G := \boldsymbol{\varepsilon}$;
- $\lceil \boldsymbol{s}m \rceil_G := \lceil \boldsymbol{s} \rceil_G.m$ *if* m *is a P-move;*
- $\lceil \boldsymbol{s}m \rceil_G := m$ *if* m *is initial;*
- $\lceil \boldsymbol{s}m\boldsymbol{t}n \rceil_G := \lceil \boldsymbol{s} \rceil_G.mn$ *if* n *is an O-move with* $\mathcal{J}_{\boldsymbol{s}m\boldsymbol{t}n}(n) = m$;
- $\lfloor \boldsymbol{\varepsilon} \rfloor_G := \boldsymbol{\varepsilon}$;
- $\lfloor \boldsymbol{s}m \rfloor_G := \lfloor \boldsymbol{s} \rfloor_G.m$ *if* m *is an O-move;*
- $\lfloor \boldsymbol{s}m\boldsymbol{t}n \rfloor_G := \lfloor \boldsymbol{s} \rfloor_G.mn$ *if* n *is a P-move with* $\mathcal{J}_{\boldsymbol{s}m\boldsymbol{t}n}(n) = m$,

where the justifiers of the remaining occurrences in $\lceil \boldsymbol{s} \rceil$ (resp. $\lfloor \boldsymbol{s} \rfloor$) are unchanged if they occur in $\lceil \boldsymbol{s} \rceil$ (resp. $\lfloor \boldsymbol{s} \rfloor$), and undefined otherwise. A view *is a P- or O-view.*

The idea behind Definition 3.18 is as follows. For a j-sequence $\boldsymbol{t}m$ in an arena G such that m is a P-move (resp. an O-move), the P-view $\lceil \boldsymbol{t} \rceil$ (resp. the O-view $\lfloor \boldsymbol{t} \rfloor$) is intended to be the currently 'relevant part' of $\boldsymbol{t}$ for Player (resp. Opponent). That is, Player (resp. Opponent) is concerned only with the last O-move (resp. P-move), its justifier and that justifier's P-view (resp. O-view), which then recursively proceeds.

We are now ready to introduce a 'dynamic generalisation' of static legal positions:

Definition 3.19 (Dynamic legal positions). *Given an arena G, a* dynamic legal position *in G is a j-sequence $\boldsymbol{s} \in \mathbb{J}_G$ that satisfies*

- (Alternation). *If $\boldsymbol{s} = \boldsymbol{s_1} mn \boldsymbol{s_2}$, then $\lambda_G^{\mathsf{OP}}(m) \neq \lambda_G^{\mathsf{OP}}(n)$;*
- (Generalised visibility). *If $\boldsymbol{s} = \boldsymbol{t}m\boldsymbol{u}$ with m non-initial, and $d \in \mathbb{N} \cup \{\omega\}$ satisfy $\lambda_G^{\mathbb{N}}(m) = 0$ or $\lambda_G^{\mathbb{N}}(m) > d$, then $\mathcal{J}_{\boldsymbol{s}}^{\ominus d}(m)$ occurs in $\lceil \mathscr{H}_G^d(\boldsymbol{t}) \rceil_{\mathscr{H}^d(G)}$ if m is a P-move, and it occurs in $\lfloor \mathscr{H}_G^d(\boldsymbol{t}) \rfloor_{\mathscr{H}^d(G)}$ if m is an O-move;*
- (IE-switch). *If $\boldsymbol{s} = \boldsymbol{s_1} mn \boldsymbol{s_2}$ with $\lambda_G^{\mathbb{N}}(m) \neq \lambda_G^{\mathbb{N}}(n)$, then m is an O-move.*

Notation 3.20. *$\mathscr{L}_G$ denotes the set of all dynamic legal positions in a dynamic arena G.*

Recall that a static legal position (Abramsky and McCusker, 1999) in a static arena is a j-sequence in the arena that satisfies alternation and *visibility*, i.e., generalised visibility only for $d = 0$. It specifies the basic rules of a static game in the sense that every 'development' or *(valid) position* in the game must be a legal position in the underlying arena (but the converse does not necessarily hold):

- In a position in the static game, Opponent always makes the first move by a question, and then Player and Opponent alternately play (by alternation), in which every non-initial move must be made for a specific previous move (viz., its justifier);
- The justifier of each non-initial move occurring in the position must belong to the 'relevant part' of previous moves occurring in the position (by visibility).

The additional axioms on dynamic legal positions are conceptually natural ones:

- Generalised visibility is a generalisation of visibility, which requires that visibility must hold after any number of iterations of the hiding operation on j-sequences;
- IE-switch states that only Player can change a priority order during a play as internal moves are 'invisible' to Opponent, where the same remark as the one for the axiom E4 is applied to the distinction of priority orders in IE-switch which is finer than I/E-parity.

Note that a dynamic legal position in a static arena, seen as a dynamic arena whose moves are all external, is clearly a static legal position, and vice versa. Hence, dynamic legal positions are in fact a generalisation of static legal positions.

Convention. Henceforth, a *legal position* refers to a dynamic legal position by default.

3.2 Dynamic games

We are now ready to define the central notion of *dynamic games*:

Definition 3.21 (Dynamic games). *A* dynamic game *is a tuple* $G = (M_G, \lambda_G, \vdash_G, P_G, \simeq_G)$ *such that*

- *The triple* $(M_G, \lambda_G, \vdash_G)$ *forms an arena (Definition 3.1);*
- P_G *is a subset of* $\mathscr{L}_G$*, whose elements are called* (valid) positions *in G, that satisfies*
  - (P1). P_G *is nonempty and prefix-closed;*
  - (DP2). *Given* $\boldsymbol{tr}, \boldsymbol{t'}r' \in P_G^{\mathsf{Odd}}$ *and* $i \in \mathbb{N}$ *such that* $i < \lambda_G^{\mathbb{N}}(r) = \lambda_G^{\mathbb{N}}(r')$*, if* $\mathscr{H}_G^i(\boldsymbol{t}) = \mathscr{H}_G^i(\boldsymbol{t'})$*, then* $\mathscr{H}_G^i(\boldsymbol{t}r) = \mathscr{H}_G^i(\boldsymbol{t'}r')$*;*
- $\simeq_G$ *is an equivalence relation on* P_G*, called the* identification *of positions in G, that satisfies*
  - (I1). $\boldsymbol{s} \simeq_G \boldsymbol{t} \Rightarrow |\boldsymbol{s}| = |\boldsymbol{t}|$*;*
  - (I2). $\boldsymbol{s}m \simeq_G \boldsymbol{t}n \Rightarrow \boldsymbol{s} \simeq_G \boldsymbol{t} \wedge \lambda_G(m) = \lambda_G(n) \wedge \mathcal{J}_{\boldsymbol{s}m}(|\boldsymbol{s}m|) = \mathcal{J}_{\boldsymbol{t}n}(|\boldsymbol{t}n|)$*;*
  - (DI3). $\forall d \in \mathbb{N} \cup \{\omega\}.\ \boldsymbol{s} \simeq_G^d \boldsymbol{t} \wedge \boldsymbol{s}m \in P_G \Rightarrow \exists \boldsymbol{t}n \in P_G.\ \boldsymbol{s}m \simeq_G^d \boldsymbol{t}n$*, where*

$$\boldsymbol{u} \simeq_G^d \boldsymbol{v} \overset{\text{df.}}{\Leftrightarrow} \exists \boldsymbol{u'}, \boldsymbol{v'} \in P_G.\ \boldsymbol{u'} \simeq_G \boldsymbol{v'} \wedge \mathscr{H}_G^d(\boldsymbol{u'}) = \mathscr{H}_G^d(\boldsymbol{u}) \wedge \mathscr{H}_G^d(\boldsymbol{v'}) = \mathscr{H}_G^d(\boldsymbol{v})$$

  for all $\boldsymbol{u}, \boldsymbol{v} \in P_G$.

A play *in G is an finitely or infinitely increasing sequence of positions* $(\boldsymbol{\varepsilon}, m_1, m_1 m_2, \ldots)$ *in G. A dynamic game whose moves are all external is said to be* normalised.

Recall that a static game (McCusker, 1998) is a tuple similar to a dynamic game except that the underlying arena is static, and it only satisfies the axioms P1, I1, I2 and *I3*, i.e., DI3 only for $d = 0$. The axiom P1 captures the phenomenon that a nonempty 'moment' or position in a game must have a previous 'moment.' Identifications of positions are originally introduced in Abramsky et al. (2000) and also employed in Section 3.6 of McCusker (1998). They are meant to identify positions modulo inessential details of 'tags' for disjoint union, particularly for *exponential* ! (Definition 3.41): Each position $\boldsymbol{s} \in P_G$ in a game G is a representative of the equivalence class $[\boldsymbol{s}] := \{\, \boldsymbol{t} \in P_G \mid \boldsymbol{t} \simeq_G \boldsymbol{s} \,\} \in P_G/\simeq_G$ which we take as primary. For this underlying idea, the three axioms I1, I2 and I3 should make sense.

The additional axiom DP2 is to enable Player to 'play alone,' i.e., Opponent does not have to choose odd-length positions, for the internal part of a play since conceptually Opponent cannot 'see' internal moves. Technically, the axiom DP2 is for *external consistency* of dynamic strategies: A dynamic strategy behaves always in the same manner from the viewpoint of Opponent, i.e., the external part of a play by a dynamic strategy does not depend on the internal part (Theorem 3.62). Note that the axiom DP2 is slightly involved to be preserved under the hiding operation (Theorem 3.28); it is necessary to generalise the axiom I3 to DI3 for the same reason.

Remark. It is certainly simpler to dispense with the identification $\simeq_G$ of positions for each game G by adopting a simpler variant of exponential ! as in McCusker (1998). However, it would be mathematically *ad hoc* since the cartesian closed structure of games and strategies would not arise via the standard *Girard translation* (Girard, 1987). Recall that the aim of the present work is to establish *mathematics* for dynamics and intensionality of logic and computation, where 'good' mathematics should be robust and general, not ad hoc. Also, it is interesting as future work to extend the present work to *linear* logic and computation. For these reasons, we retain $\simeq_G$ as a structure of each game G. Moreover, we shall establish various reasonable properties on identifications of positions, which adds credibility to the notions of dynamic games and strategies.

Convention. Henceforth, a *game* refers to a dynamic game by default.

Example 3.22. The *terminal game* $T := (\emptyset, \emptyset, \emptyset, \{\boldsymbol{\varepsilon}\}, \{(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})\})$ is the simplest game.

Example 3.23. The *flat game* $\mathit{flat}(S)$ on a given set S is defined as follows. The triple $\mathit{flat}(S) = (M_{\mathit{flat}(S)}, \lambda_{\mathit{flat}(S)}, \vdash_{\mathit{flat}(S)})$ is the flat arena given in Example 3.3, $P_{\mathit{flat}(S)} := \{\boldsymbol{\varepsilon}, q\} \cup \{ qm \mid m \in S \}$, and $\simeq_{\mathit{flat}(S)} := \{(\boldsymbol{s}, \boldsymbol{s}) \mid \boldsymbol{s} \in P_{\mathit{flat}(S)}\}$. For instance, $N := \mathit{flat}(\mathbb{N})$ is the game of natural numbers given in the introduction, $\mathbf{2} := \mathit{flat}(\mathbb{B})$ is the game of booleans, and $\mathbf{0} := \mathit{flat}(\emptyset)$ is the *empty game*.

Also, let us define a substructure relation between games:

Definition 3.24 (Subgames). *A game H is a* (dynamic) subgame *of a game G, written $H \trianglelefteq G$, if $M_H \subseteq M_G$, $\lambda_H = \lambda_G \upharpoonright M_H$, $\vdash_H \subseteq \vdash_G \cap ((\{\star\} \cup M_H) \times M_H)$, $P_H \subseteq P_G$, $\simeq_H^d = \simeq_G^d \cap (P_H \times P_H)$ for all $d \in \mathbb{N} \cup \{\omega\}$ and $\mu(H) = \mu(G)$.*

For $H \trianglelefteq G$, the condition on the identifications of positions is required for *all* $d \in \mathbb{N} \cup \{\omega\}$, so that the relation $\trianglelefteq$ is preserved under the hiding operation (Theorem 3.28). The last equation $\mu(H) = \mu(G)$ is to preserve the relation $\trianglelefteq$ under *concatenation* on dynamic games (Definition 3.47).

We shall later focus on *well founded* games:

Definition 3.25 (Well founded games (Claimrambault and Harmer, 2010)). *A game G is* well founded *if $\vdash_G$ is well founded* downwards, *i.e., there is no countably infinite sequence $(m_i)_{i \in \mathbb{N}}$ of moves $m_i \in M_G$ such that $\star \vdash_G m_0$ and $m_i \vdash_G m_{i+1}$ for all $i \in \mathbb{N}$.*

Now, let us define the *hiding operation* on games:

Definition 3.26 (Hiding operation on games). *Given $d \in \mathbb{N} \cup \{\omega\}$, the d-*hiding operation (on games) *maps each game G to its d-*external game *$\mathscr{H}^d(G)$ defined by*

- *The triple $(M_{\mathscr{H}^d(G)}, \lambda_{\mathscr{H}^d(G)}, \vdash_{\mathscr{H}^d(G)})$ is the d-external arena $\mathscr{H}^d(G)$ of the underlying arena G (Definition 3.11);*
- *$P_{\mathscr{H}^d(G)} := \{ \mathscr{H}_G^d(\boldsymbol{s}) \mid \boldsymbol{s} \in P_G \}$;*
- *$\mathscr{H}_G^d(\boldsymbol{s}) \simeq_{\mathscr{H}^d(G)} \mathscr{H}_G^d(\boldsymbol{t}) \overset{\mathrm{df.}}{\Leftrightarrow} \boldsymbol{s} \simeq_G^d \boldsymbol{t}$ (Definition 3.21).*

Example 3.27. Let us revisit the game $N_{[0]} \multimap N_{[3]}$ of linear maps on natural numbers sketched in the introduction, where we replace the 'tag' $(\_)_{[1]}$ with $(\_)_{[3]}$ for convenience. Formally, the game consists of the set $M_{N_{[0]} \multimap N_{[3]}} := \{ m_{[i]} \mid m \in M_N, i \in \{0, 3\} \}$ of moves, the labelling function $\lambda_{N_{[0]} \multimap N_{[3]}}$ that maps $q_{[0]} \mapsto \mathsf{PQ0}$, $q_{[3]} \mapsto \mathsf{OQ0}$, $n_{[0]} \mapsto \mathsf{OA0}$ and $n'_{[3]} \mapsto \mathsf{PA0}$ $(n, n' \in \mathbb{N})$, the enabling relation $\vdash_{N_{[0]} \multimap N_{[3]}} := \{(\star, q_{[3]})\} \cup \{(q_{[3]}, q_{[0]})\} \cup \{ (m_{[0]}, n_{[0]}) \mid m \vdash_N n \} \cup \{ (m'_{[3]}, n'_{[3]}) \mid m' \vdash_N n'\}$, the set $P_{N_{[0]} \multimap N_{[3]}} := \mathsf{Pref}(\{ q_{[3]} n_{[3]} \mid n \in \mathbb{N}' \} \cup \{ q_{[3]} q_{[0]} n_{[0]} n'_{[3]} \mid n, n' \in \mathbb{N} \})$ of positions and the identification $\simeq_{N_{[0]} \multimap N_{[3]}} := \{ (\boldsymbol{s}, \boldsymbol{t}) \in P_{N_{[0]} \multimap N_{[3]}} \times P_{N_{[0]} \multimap N_{[3]}} \mid \boldsymbol{s} = \boldsymbol{t} \}$ of positions.

The game underlying the composition of *succ* and *double* 'without hiding' given in the introduction, which we write G, consists of the set $M_G := \{ m_{[i]} \mid m \in M_N, i \in \{0\} \cup \overline{3} \}$ of moves, the labelling function λ_G that maps $m_{[i]} \mapsto \lambda_{N_{[0]} \multimap N_{[3]}}(m_{[i]})$ $(i = 0, 3)$, $q_{[2]} \mapsto \mathsf{PQ1}$, $n_{[2]} \mapsto \mathsf{OA1}$, $q_{[1]} \mapsto \mathsf{OQ1}$ and $n'_{[1]} \mapsto \mathsf{PA1}$ $(n, n' \in \mathbb{N})$, the enabling relation $\vdash_G := \{(\star, q_{[3]})\} \cup \{(q_{[3]}, q_{[2]})\} \cup \{(q_{[2]}, q_{[1]})\} \cup \{(q_{[1]}, q_{[0]})\} \cup \{ (m_{[i]}, n_{[i]}) \mid m \vdash_N n, i \in \{0\} \cup \overline{3} \}$, the set $P_G := \mathsf{Pref}(\{ q_{[3]} n_{[3]} \mid n \in \mathbb{N} \} \cup \{ q_{[3]} q_{[2]} q_{[1]} q_{[0]} n_{[0]} m_{[1]} m_{[2]} n'_{[3]} \mid n, m, n' \in \mathbb{N} \})$ of positions and the identification $\simeq_G := \{ (\boldsymbol{s}, \boldsymbol{t}) \in P_G \times P_G \mid \boldsymbol{s} = \boldsymbol{t} \}$ of positions. It is then easy to see that the equation $\mathscr{H}(G) = N_{[0]} \multimap N_{[3]}$ holds as in the introduction. We shall introduce general constructions on games that include this example in the next section.

Now, we give the first main theorem of the present work, which shows that the hiding operation on games is in fact well defined:

Theorem 3.28 (External closure of games). *Given $d \in \mathbb{N} \cup \{\omega\}$, games (resp. well founded ones) are closed under the operation $\mathcal{H}^d$ (Definition 3.26), and $H \trianglelefteq G$ implies $\mathcal{H}^d(H) \trianglelefteq \mathcal{H}^d(G)$.*

Proof. Let G be a game and assume $d \in \mathbb{N} \cup \{\omega\}$; we have to show that $\mathcal{H}^d(G)$ is a game. By Lemma 3.13, it suffices to show that j-sequences in $P_{\mathcal{H}^d(G)}$ are legal positions in the arena $\mathcal{H}^d(G)$, the set $P_{\mathcal{H}^d(G)}$ satisfies P1 and DP2, and the relation $\simeq_{\mathcal{H}^d(G)}$ is an equivalence relation on $P_{\mathcal{H}^d(G)}$ that satisfies I1, I2 and DI3. Since $\mu(G) \in \mathbb{N}$, we assume $d \in \mathbb{N}$ without loss of generality.

For alternation, let $\boldsymbol{s_1} mn\boldsymbol{s_2} \in P_{\mathcal{H}^d(G)}$; we have to show $\lambda^{\mathrm{OP}}_{\mathcal{H}^d(G)}(m) \neq \lambda^{\mathrm{OP}}_{\mathcal{H}^d(G)}(n)$. We have $\mathcal{H}^d_G(\boldsymbol{t_1} mm_1m_2 \dots m_kn\boldsymbol{t_2}) = \boldsymbol{s_1}mn\boldsymbol{s_2}$ for some $\boldsymbol{t_1}mm_1m_2 \dots m_kn\boldsymbol{t_2} \in P_G$, where $\mathcal{H}^d_G(\boldsymbol{t_1}) = \boldsymbol{s_1}$, $\mathcal{H}^d_G(\boldsymbol{t_2}) = \boldsymbol{s_2}$ and $\mathcal{H}^d_G(m_1m_2 \dots m_k) = \boldsymbol{\varepsilon}$. Note that $\lambda^{\mathbb{N}}_G(m) = 0 \vee \lambda^{\mathbb{N}}_G(m) > d$, $\lambda^{\mathbb{N}}_G(n) = 0 \vee \lambda^{\mathbb{N}}_G(n) > d$ and $0 < \lambda^{\mathbb{N}}_G(m_i) \leqslant d$ for $i = 1, 2, \dots, k$. By E3 and E4 on G, k must be even, and thus $\lambda^{\mathrm{OP}}_{\mathcal{H}^d(G)}(m) = \lambda^{\mathrm{OP}}_G(m) = \lambda^{\mathrm{OP}}_G(m_2) = \lambda^{\mathrm{OP}}_G(m_4) = \dots = \lambda^{\mathrm{OP}}_G(m_k) \neq \lambda^{\mathrm{OP}}_G(n) = \lambda^{\mathrm{OP}}_{\mathcal{H}^d(G)}(n)$.

For generalised visibility, let $\boldsymbol{tmu} \in P_{\mathcal{H}^d(G)}$ with m non-initial. We have to show, for each $e \in \mathbb{N} \cup \{\omega\}$, that if $\boldsymbol{t}m$ is e-complete, then

- if m is a P-move, then the justifier $(\mathcal{J}^{\ominus d}_{\boldsymbol{s}})^{\ominus e}(m)$ occurs in $\ulcorner \mathcal{H}^e_{\mathcal{H}^d(G)}(\boldsymbol{t}) \urcorner_{\mathcal{H}^e(\mathcal{H}^d(G))}$;
- if m is an O-move, then the justifier $(\mathcal{J}^{\ominus d}_{\boldsymbol{s}})^{\ominus e}(m)$ occurs in $\llcorner \mathcal{H}^e_{\mathcal{H}^d(G)}(\boldsymbol{t}) \lrcorner_{\mathcal{H}^e(\mathcal{H}^d(G))}$.

Again, for $\mu(G) \in \mathbb{N}$, we may assume without loss of generality that $e \in \mathbb{N}$. Note that the two conditions just given are then equivalent to

- if m is a P-move, then the justifier $\mathcal{J}^{\ominus(d+e)}_{\boldsymbol{s}}(m)$ occurs in $\ulcorner \mathcal{H}^{d+e}_G(\boldsymbol{t}') \urcorner_{\mathcal{H}^{d+e}(G)}$;
- if m is an O-move, then the justifier $\mathcal{J}^{\ominus(d+e)}_{\boldsymbol{s}}(m)$ occurs in $\llcorner \mathcal{H}^{d+e}_G(\boldsymbol{t}') \lrcorner_{\mathcal{H}^{d+e}(G)}$,

where $\boldsymbol{t}'m \in P_G$ such that $\mathcal{H}^d_G(\boldsymbol{t}'m) = \boldsymbol{t}m$. They hold by generalised visibility on G.

For IE-switch, let $\boldsymbol{s_1}mn\boldsymbol{s_2} \in P_{\mathcal{H}^d(G)}$ such that $\lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(m) \neq \lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(n)$. Then, there is some $\boldsymbol{t_1}m\boldsymbol{u}n\boldsymbol{t_2} \in P_G$ such that $\mathcal{H}^d_G(\boldsymbol{t_1}m\boldsymbol{u}n\boldsymbol{t_2}) = \boldsymbol{s_1}mn\boldsymbol{s_2}$, where note that $\lambda^{\mathbb{N}}_G(m) \neq \lambda^{\mathbb{N}}_G(n)$. Therefore, if $\boldsymbol{u} = \boldsymbol{\varepsilon}$, then we clearly have $\lambda^{\mathrm{OP}}_{\mathcal{H}^d(G)}(m) = \mathsf{O}$ by IE-switch on G; otherwise, i.e., $\boldsymbol{u} = l\boldsymbol{u}'$, then we have the same conclusion because $\lambda^{\mathbb{N}}_G(m) \neq \lambda^{\mathbb{N}}_G(l)$.

We have established $P_{\mathcal{H}^d(G)} \subseteq \mathscr{L}_{\mathcal{H}^d(G)}$. Let us proceed to verify P1 and DP2 on $\mathcal{H}^d(G)$:

- (P1). Because $\boldsymbol{\varepsilon} \in P_G$, we have $\boldsymbol{\varepsilon} = \mathcal{H}^d_G(\boldsymbol{\varepsilon}) \in P_{\mathcal{H}^d(G)}$; thus, $P_{\mathcal{H}^d(G)}$ is nonempty. For prefix-closure, let $\boldsymbol{s}m \in P_{\mathcal{H}^d(G)}$; we have to show $\boldsymbol{s} \in P_{\mathcal{H}^d(G)}$. There must be some $\boldsymbol{t}m \in P_G$ such that $\boldsymbol{s}m = \mathcal{H}^d_G(\boldsymbol{t}m) = \mathcal{H}^d_G(\boldsymbol{t}).m$. Hence, $\boldsymbol{s} = \mathcal{H}^d_G(\boldsymbol{t}) \in P_{\mathcal{H}^d(G)}$.
- (DP2). Assume $\boldsymbol{t}r, \boldsymbol{t}'r' \in P^{\mathrm{Odd}}_{\mathcal{H}^d(G)}$, $i \in \mathbb{N}$, $i < \lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(r) = \lambda^{\mathbb{N}}_{\mathcal{H}^d(G)}(r')$ and $\mathcal{H}^i_{\mathcal{H}^d(G)}(\boldsymbol{t}) = \mathcal{H}^i_{\mathcal{H}^d(G)}(\boldsymbol{t}')$. We have some $\boldsymbol{u}r, \boldsymbol{u}'r' \in P_G$ with $\mathcal{H}^d_G(\boldsymbol{u}) = \boldsymbol{t}$ and $\mathcal{H}^d_G(\boldsymbol{u}') = \boldsymbol{t}'$. Then,
$$\mathcal{H}^{d+i}_G(\boldsymbol{u}) = \mathcal{H}^i_{\mathcal{H}^d(G)}(\mathcal{H}^d_G(\boldsymbol{u})) = \mathcal{H}^i_{\mathcal{H}^d(G)}(\boldsymbol{t}) = \mathcal{H}^i_{\mathcal{H}^d(G)}(\boldsymbol{t}') = \mathcal{H}^i_{\mathcal{H}^d(G)}(\mathcal{H}^d_G(\boldsymbol{u}')) = \mathcal{H}^{d+i}_G(\boldsymbol{u}').$$
Thus, by DP2 on G, we have $r = r'$ and $\mathcal{J}^{\ominus i}_{\boldsymbol{t}r}(r) = \mathcal{J}^{\ominus(d+i)}_{\boldsymbol{u}r}(r) = \mathcal{J}^{\ominus(d+i)}_{\boldsymbol{u}'r'}(r') = \mathcal{J}^{\ominus i}_{\boldsymbol{t}'r'}(r')$, establishing DP2 on $\mathcal{H}^d(G)$.

Next, $\simeq_{\mathcal{H}^d(G)}$ is a well-defined relation on $P_{\mathcal{H}^d(G)}$ since $\mathcal{H}^d_G(\boldsymbol{s}) \simeq_{\mathcal{H}^d(G)} \mathcal{H}^d_G(\boldsymbol{t})$ does not depend on the choice of representatives $\boldsymbol{s}, \boldsymbol{t} \in P_G$. Also, it is straightforward to see that $\simeq_{\mathcal{H}^d(G)}$ is an equivalence relation. Now, let us show that $\simeq_{\mathcal{H}^d(G)}$ satisfies I1, I2 and DI3. Note that I1 and I2 on $\simeq_{\mathcal{H}^d(G)}$ follow from those on $\simeq_G$. For DI3 on $\simeq_{\mathcal{H}^d(G)}$, if $\mathcal{H}^d_G(\boldsymbol{s}) \simeq^e_{\mathcal{H}^d(G)} \mathcal{H}^d_G(\boldsymbol{t})$ and

$\mathcal{H}^d_G(\boldsymbol{s}).m \in P_{\mathcal{H}^d(G)}$, where we may assume $e \neq \omega$, then $\exists \boldsymbol{s}'m \in P_G. \mathcal{H}^d_G(\boldsymbol{s}'m) = \mathcal{H}^d_G(\boldsymbol{s}).m$, and so $\mathcal{H}^{d+e}_G(\boldsymbol{s}') = \mathcal{H}^{d+e}_G(\boldsymbol{s}) \simeq_{\mathcal{H}^{d+e}(G)} \mathcal{H}^{d+e}_G(\boldsymbol{t})$. By DI3 on $\simeq_G$, we have $\exists \boldsymbol{t}n \in P_G. \boldsymbol{s}'m \simeq^{d+e}_G \boldsymbol{t}n$, whence $\mathcal{H}^d_G(\boldsymbol{t}).n \in P_{\mathcal{H}^d(G)}$ such that $\mathcal{H}^d_G(\boldsymbol{s}).m = \mathcal{H}^d_G(\boldsymbol{s}'m) \simeq^e_{\mathcal{H}^d(G)} \mathcal{H}^d_G(\boldsymbol{t}n) = \mathcal{H}^d_G(\boldsymbol{t}).n$.

Finally, the preservations of well founded games and the dynamic subgame relation $\trianglelefteq$ under the operation $\mathcal{H}^d$ are clear from the definitions, completing the proof. □

Corollary 3.29 (Stepwise hiding on games). *For any game G and natural number $i \in \mathbb{N}$, we have*

$$\mathcal{H}^1(\mathcal{H}^i(G)) = \mathcal{H}^{i+1}(G).$$

Proof. By Lemmas 3.14 and 3.15, it suffices to show the equation $\simeq_{\mathcal{H}^1(\mathcal{H}^i(G))} = \simeq_{\mathcal{H}^{i+1}(G)}$. Then, given $\boldsymbol{s}, \boldsymbol{t} \in P_G$, we have

$$\begin{aligned}
&\mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i_G(\boldsymbol{s})) \simeq_{\mathcal{H}^1(\mathcal{H}^i(G))} \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i_G(\boldsymbol{t}))\\
\Leftrightarrow\ &\exists \mathcal{H}^i(\boldsymbol{s}'), \mathcal{H}^i(\boldsymbol{t}') \in P_{\mathcal{H}^i(G)}. \mathcal{H}^i(\boldsymbol{s}') \simeq_{\mathcal{H}^i(G)} \mathcal{H}^i(\boldsymbol{t}') \wedge \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i(\boldsymbol{s}')) = \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i(\boldsymbol{s}))\\
&\wedge \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i(\boldsymbol{t}')) = \mathcal{H}^1_{\mathcal{H}^i(G)}(\mathcal{H}^i(\boldsymbol{t}))\\
\Leftrightarrow\ &\exists \boldsymbol{s}'', \boldsymbol{t}'' \in P_G. \boldsymbol{s}'' \simeq_G \boldsymbol{t}'' \wedge \mathcal{H}^{i+1}_G(\boldsymbol{s}'') = \mathcal{H}^{i+1}_G(\boldsymbol{s}) \wedge \mathcal{H}^{i+1}_G(\boldsymbol{t}'') = \mathcal{H}^{i+1}_G(\boldsymbol{t})\\
\Leftrightarrow\ &\mathcal{H}^{i+1}_G(\boldsymbol{s}) \simeq_{\mathcal{H}^{i+1}(G)} \mathcal{H}^{i+1}_G(\boldsymbol{t}),
\end{aligned}$$

which establishes the required equation. □

By the corollary just given, we may focus on $\mathcal{H}^1$:

Convention. We write $\mathcal{H}$ for $\mathcal{H}^1$ and call it the *hiding operation (on games)*; $\mathcal{H}^i$ denotes the i-times iteration of $\mathcal{H}$ for all $i \in \mathbb{N}$.

Corollary 3.30 (Hiding operation on legal positions). *Given an arena G and a number $d \in \mathbb{N} \cup \{\omega\}$, we have $\{\, \mathcal{H}^d_G(\boldsymbol{s}) \mid \boldsymbol{s} \in \mathcal{L}_G \,\} = \mathcal{L}_{\mathcal{H}^d(G)}$.*

Proof. Since there is an upper bound $\mu(G) \in \mathbb{N}$, it suffices to consider the case $d \in \mathbb{N}$. Then, by Lemmas 3.14 and 3.15, we may just focus on the case $d = 1$.

The inclusion $\{\mathcal{H}_G(\boldsymbol{s}) \mid \boldsymbol{s} \in \mathcal{L}_G\} \subseteq \mathcal{L}_{\mathcal{H}(G)}$ is immediate by Theorem 3.28. To show the other inclusion, let $\boldsymbol{t} \in \mathcal{L}_{\mathcal{H}(G)}$; we shall find some $\boldsymbol{s} \in \mathcal{L}_G$ such that

(1) $\mathcal{H}_G(\boldsymbol{s}) = \boldsymbol{t}$;
(2) 1-internal moves in $\boldsymbol{s}$ occur as even length consecutive segments $m_1 m_2 \ldots m_{2k}$ ($k \in \mathbb{N}$), where m_i justifies m_{i+1} for $i = 1, 2, \ldots, 2k-1$;
(3) $\boldsymbol{s}$ is 1-complete.

We proceed by induction on the length $|\boldsymbol{t}|$. The base case $\boldsymbol{t} = \boldsymbol{\varepsilon}$ is trivial. For the inductive step, let $\boldsymbol{t}m \in \mathcal{L}_{\mathcal{H}(G)}$. Then, $\boldsymbol{t} \in \mathcal{L}_{\mathcal{H}(G)}$, and by the induction hypothesis there is some $\boldsymbol{s} \in \mathcal{L}_G$ that satisfies the three conditions (n.b., the first one is for $\boldsymbol{t}$).

If m is initial, then $\boldsymbol{s}m \in \mathcal{L}_G$, and $\boldsymbol{s}m$ satisfies the three conditions. Hence, assume that m is non-initial; we may then write $\boldsymbol{t}m = \boldsymbol{t}_1 n \boldsymbol{t}_2 m$, where m is justified by n.

We then need the following case analysis:

- Assume $n \vdash_G m$. We take $\boldsymbol{s}m$, where m points to n. Then, $\boldsymbol{s}m \in \mathcal{L}_G$ because
  - (Justification). It is immediate from $n \vdash_G m$.
  - (Alternation). By the condition 3 on $\boldsymbol{s}$, the last moves of $\boldsymbol{s}$ and $\boldsymbol{t}$ just coincide. Thus, the alternation condition holds for $\boldsymbol{s}m$.

  - (Generalised visibility). It suffices to establish the visibility on $\boldsymbol{sm}$ since the other cases are included as the generalised visibility on $\boldsymbol{tm}$. It is straightforward to see that, by the condition 2 on $\boldsymbol{s}$, if the view of $\boldsymbol{t}$ contains n, then so does the view of $\boldsymbol{s}$. And since $\boldsymbol{tm} \in \mathscr{L}_{\mathscr{H}(G)}$, the view of $\boldsymbol{t}$ contains n. Hence, the view of $\boldsymbol{s}$ contains n as well.
  - (IE-switch). Again, the last moves of $\boldsymbol{s}$ and $\boldsymbol{t}$ coincide by the condition 3 on $\boldsymbol{s}$; thus, IE-switch on $\boldsymbol{tm}$ can be directly applied.

  Moreover, it is easy to see that $\boldsymbol{sm}$ satisfies the three conditions.
- Assume $n \neq \star$, and there are some $k \in \mathbb{N}^+$ and $m_1, m_2, \ldots, m_{2k} \in M_G \setminus M_{\mathscr{H}(G)}$ such that $n \vdash_G m_1$, $m_i \vdash_G m_{i+1}$ for all $i \in \overline{2k-1}$ and $m_{2k} \vdash_G m$. We then take $\boldsymbol{s}m_1m_2 \ldots m_{2k}m$, in which m_1 points to n, m_i points to m_{i-1} for $i = 2, 3, \ldots, 2k$, and m points to m_{2k}. Then, $\boldsymbol{s}m_1m_2 \ldots m_{2k}m \in \mathscr{L}_G$ because
  - (Justification). Obvious.
  - (Alternation). By the condition 3 on $\boldsymbol{s}$, the last moves of $\boldsymbol{s}$ and $\boldsymbol{t}$ just coincide. Thus, the alternation condition holds for $\boldsymbol{s}m_1m_2 \ldots m_{2k}m$.
  - (Generalised visibility). By the same argument as the above case.
  - (IE-switch). It clearly holds by the axiom E4.

  Finally, it is easy to see that $\boldsymbol{s}m_1m_2 \ldots m_{2k}m$ satisfies the three conditions.

We have completed the case analysis. □

3.3 Constructions on dynamic games

In this section, we show that dynamic games accommodate all the standard constructions on static games (Abramsky and McCusker, 1999), i.e., they preserve the additional axioms for dynamic games, as well as some new constructions. This result implies that the notion of dynamic games (Definition 3.21) is in some sense 'correct.'

Convention. We omit 'tags' for disjoint union of sets. For instance, we write $x \in A + B$ if and only if $x \in A$ or $x \in B$ (not both); also, given relations $R_A \subseteq A \times A$ and $R_B \subseteq B \times B$, we write $R_A + R_B$ for the relation on the disjoint union $A + B$ such that $(x, y) \in R_A + R_B \overset{\mathrm{df.}}{\Leftrightarrow} (x, y) \in R_A \vee (x, y) \in R_B$.

Let us begin with *tensor (product)* $\otimes$. Roughly, a position of the tensor $A \otimes B$ of games A and B is an interleaving mixture of a position in A and that in B, in which an AB-parity change is made always by Opponent. Formally:

Definition 3.31 (Tensor of games (Abramsky and McCusker, 1999)). *Given games A and B, the* tensor (product) *$A \otimes B$ of A and B is defined by*

- $M_{A \otimes B} := M_A + M_B$;
- $\lambda_{A \otimes B} := [\lambda_A, \lambda_B]$;
- $\vdash_{A \otimes B} := \vdash_A + \vdash_B$;
- $P_{A \otimes B} := \{ \boldsymbol{s} \in \mathscr{L}_{A \otimes B} \mid \boldsymbol{s} \upharpoonright A \in P_A, \boldsymbol{s} \upharpoonright B \in P_B \}$;
- $\boldsymbol{s} \simeq_{A \otimes B} \boldsymbol{t} \overset{\mathrm{df.}}{\Leftrightarrow} \boldsymbol{s} \upharpoonright A \simeq_A \boldsymbol{t} \upharpoonright A \wedge \boldsymbol{s} \upharpoonright B \simeq_B \boldsymbol{t} \upharpoonright B \wedge \forall i \in \mathbb{N}.\, \boldsymbol{s}(i) \in M_A \Leftrightarrow \boldsymbol{t}(i) \in M_A$,

where $\boldsymbol{s} \upharpoonright A$ (resp. $\boldsymbol{s} \upharpoonright B$) denotes the j-subsequence of $\boldsymbol{s}$ that consists of occurrences of moves of A (resp. B).

In fact, as shown in Abramsky et al. (1997), only Opponent can switch between the component games A and B (by alternation) in a position in a tensor $A \otimes B$.

$$\begin{tikzcd}
(\mathsf{P}^{\mathsf{I}}, \mathsf{O}^{\mathsf{E}}) \arrow[d, leftrightarrow, "A"] & & (\mathsf{O}^{\mathsf{E}}, \mathsf{O}^{\mathsf{E}}) \arrow[ll, "A"'] \arrow[rr, "B"] \arrow[dl, leftrightarrow, "A"'] \arrow[dr, leftrightarrow, "B"] & & (\mathsf{O}^{\mathsf{E}}, \mathsf{P}^{\mathsf{I}}) \arrow[d, leftrightarrow, "B"'] \\
(\mathsf{O}^{\mathsf{I}}, \mathsf{O}^{\mathsf{E}}) \arrow[r, "A"] & (\mathsf{P}^{\mathsf{E}}, \mathsf{O}^{\mathsf{E}}) & & (\mathsf{O}^{\mathsf{E}}, \mathsf{P}^{\mathsf{E}}) & (\mathsf{O}^{\mathsf{E}}, \mathsf{O}^{\mathsf{I}}) \arrow[l, "B"']
\end{tikzcd}$$

Figure 1. The double parity diagram for the tensor $A \otimes B$.

Example 3.32. Consider the tensor $N \otimes N$ of the natural number game N with itself, whose maximal position is either of the following forms:

$$\begin{tikzcd}
N_{[0]} & \otimes & N_{[1]} \\
q_{[0]} & & \\
n_{[0]} \arrow[u, bend left] & & \\
& & q_{[1]} \\
& & m_{[1]} \arrow[u, bend left]
\end{tikzcd}
\qquad
\begin{tikzcd}
N_{[0]} & \otimes & N_{[1]} \\
& & q_{[1]} \\
& & m_{[1]} \arrow[u, bend left] \\
q_{[0]} & & \\
n_{[0]} \arrow[u, bend left] & &
\end{tikzcd}$$

where $n, m \in \mathbb{N}$, and $(\_)_{[i]}$ $(i = 0, 1)$ are again arbitrary, unspecified 'tags' such that $[0] \neq [1]$ just to distinguish the two copies of N, and the arrows represent pointers. Henceforth, however, we usually omit the 'tags' $(\_)_{[i]}$ unless it is strictly necessary.

Theorem 3.33 (Well defined tensor of games). *Games (resp. well founded ones) are closed under tensor $\otimes$.*

Proof. Since static games are closed under tensor $\otimes$ (Abramsky and McCusker, 1999), it suffices to show that $\otimes$ preserves the condition on labeling function and the axioms E1, E2, E4, DP2 and DI3 (n.b., $\otimes$ clearly preserves well foundedness of games). However, nontrivial ones are just DP2 and DI3; hence, we just focus on these two axioms.

Let A and B be any games. To verify DP2 on $A \otimes B$, let $\boldsymbol{slmn}, \boldsymbol{s'}l'm'n' \in P_{A\otimes B}^{\mathsf{Odd}}$ and $i \in \mathbb{N}$ such that $\mathscr{H}^i_{A\otimes B}(\boldsymbol{s}lm) = \mathscr{H}^i_{A\otimes B}(\boldsymbol{s'}l'm')$ and $i < \lambda^{\mathbb{N}}_{A\otimes B}(n) = \lambda^{\mathbb{N}}_{A\otimes B}(n')$. Note that $\lambda^{\mathbb{N}}_{A\otimes B}(m) = \lambda^{\mathbb{N}}_{A\otimes B}(n) = \lambda^{\mathbb{N}}_{A\otimes B}(n') = \lambda^{\mathbb{N}}_{A\otimes B}(m')$ by IE-switch. At a first glance, it seems that $A \otimes B$ does not satisfy DP2 as Opponent may play in A or B at will. It is, however, not the case for internal moves since $\boldsymbol{s}lmn \in P^{\mathsf{Odd}}_{A\otimes B}$ with m internal implies $m, n \in M_A$ or $m, n \in M_B$. This property immediately follows from Figure 1, which shows all possible transitions of OP- and IE-parities for a play in $A \otimes B$, where a state (X^Y, Z^W) means that the next move of A (resp. B) has the OP-parity X (resp. Z) and the IE-parity Y (resp. W). Note that $m = m'$ and $\mathscr{J}^{\ominus i}_{\boldsymbol{s}lm}(m) = \mathscr{J}^{\ominus i}_{\boldsymbol{s'}l'm'}(m')$ as $\mathscr{H}^i_{A\otimes B}(\boldsymbol{s}l).m = \mathscr{H}^i_{A\otimes B}(\boldsymbol{s}lm) = \mathscr{H}^i_{A\otimes B}(\boldsymbol{s'}l'm') = \mathscr{H}^i_{A\otimes B}(\boldsymbol{s'}l').m'$. Thus, m, n, m' and n' belong to the same component game. If $m, n, m', n' \in M_A$, then $(\boldsymbol{s}l \upharpoonright A).mn, (\boldsymbol{s'}l' \upharpoonright A).m'n' \in P^{\mathsf{Odd}}_A$, $\mathscr{H}^i_A((\boldsymbol{s}l \upharpoonright A).m) = \mathscr{H}^i_{A\otimes B}(\boldsymbol{s}lm) \upharpoonright \mathscr{H}^i(A) = \mathscr{H}^i_{A\otimes B}(\boldsymbol{s'}l'm') \upharpoonright \mathscr{H}^i(A) = \mathscr{H}^i_A((\boldsymbol{s'}l' \upharpoonright A).m')$ and $i < \lambda^{\mathbb{N}}_A(n) = \lambda^{\mathbb{N}}_A(n')$; thus, by DP2 on A, we conclude that $n = n'$ and $\mathscr{J}^{\ominus i}_{\boldsymbol{s}lmn}(n) = \mathscr{J}^{\ominus i}_{(\boldsymbol{s}l\upharpoonright A).mn}(n) = \mathscr{J}^{\ominus i}_{(\boldsymbol{s'}l'\upharpoonright A).m'n'}(n') = \mathscr{J}^{\ominus i}_{\boldsymbol{s'}l'm'n'}(n')$ hold. The other case is completely analogous, showing that $A \otimes B$ satisfies DP2.

Finally, to show that $A \otimes B$ satisfies DI3, assume $\boldsymbol{s} \simeq^d_{A\otimes B} \boldsymbol{t}$ and $\boldsymbol{s}m \in P_{A\otimes B}$ for some $d \in \mathbb{N}$; we have to find some $\boldsymbol{t}n \in P_{A\otimes B}$ such that $\boldsymbol{s}m \simeq^d_{A\otimes B} \boldsymbol{t}n$. Assume $m \in M_A$ since the other case is symmetric. Because $\boldsymbol{s} \simeq^d_{A\otimes B} \boldsymbol{t}$, we have some $\boldsymbol{s'} \simeq_{A\otimes B} \boldsymbol{t'}$ such that $\mathscr{H}^d_{A\otimes B}(\boldsymbol{s'}) = \mathscr{H}^d_{A\otimes B}(\boldsymbol{s})$ and $\mathscr{H}^d_{A\otimes B}(\boldsymbol{t'}) = \mathscr{H}^d_{A\otimes B}(\boldsymbol{t})$. Thus, $\boldsymbol{s'} \upharpoonright A \simeq_A \boldsymbol{t'} \upharpoonright A$, $\mathscr{H}^d_A(\boldsymbol{s} \upharpoonright A) = \mathscr{H}^d_{A\otimes B}(\boldsymbol{s}) \upharpoonright \mathscr{H}^d(A) = \mathscr{H}^d_{A\otimes B}(\boldsymbol{s'}) \upharpoonright \mathscr{H}^d(A) = \mathscr{H}^d_A(\boldsymbol{s'} \upharpoonright A)$ and $\mathscr{H}^d_A(\boldsymbol{t} \upharpoonright A) = \mathscr{H}^d_{A\otimes B}(\boldsymbol{t}) \upharpoonright \mathscr{H}^d(A) = \mathscr{H}^d_{A\otimes B}(\boldsymbol{t'}) \upharpoonright \mathscr{H}^d(A) = \mathscr{H}^d_A(\boldsymbol{t'} \upharpoonright A)$, whence $\boldsymbol{s} \upharpoonright A \simeq^d_A \boldsymbol{t} \upharpoonright A$. Similarly, $\boldsymbol{s} \upharpoonright B \simeq^d_B \boldsymbol{t} \upharpoonright B$ with $\boldsymbol{s'} \upharpoonright B \simeq_B \boldsymbol{t'} \upharpoonright B$, $\mathscr{H}^d_B(\boldsymbol{s} \upharpoonright B) = \mathscr{H}^d_B(\boldsymbol{s'} \upharpoonright B)$ and

$\mathscr{H}^d_B(\boldsymbol{t} \upharpoonright B) = \mathscr{H}^d_B(\boldsymbol{t}' \upharpoonright B)$. Now, since $(\boldsymbol{s} \upharpoonright A).m = \boldsymbol{s}m \upharpoonright A \in P_A$, we have some $(\boldsymbol{t} \upharpoonright A).n \in P_A$ such that $(\boldsymbol{s} \upharpoonright A).m \simeq^d_A (\boldsymbol{t} \upharpoonright A).n$, i.e., some $\boldsymbol{u} \simeq_A \boldsymbol{v}$ such that $\mathscr{H}^d_A(\boldsymbol{u}) = \mathscr{H}^d_A((\boldsymbol{s} \upharpoonright A).m)$ and $\mathscr{H}^d_A(\boldsymbol{v}) = \mathscr{H}^d_A((\boldsymbol{t} \upharpoonright A).n)$. By Figure 1, we may obtain a unique $\tilde{\boldsymbol{s}} \in P_{A \otimes B}$ from $\boldsymbol{u}$ and $\boldsymbol{s}' \upharpoonright B$ and a unique $\tilde{\boldsymbol{t}} \in P_{A \otimes B}$ from $\boldsymbol{v}$ and $\boldsymbol{t}' \upharpoonright B$ such that $\tilde{\boldsymbol{s}} \simeq_{A \otimes B} \tilde{\boldsymbol{t}}$, $\mathscr{H}^d_{A \otimes B}(\tilde{\boldsymbol{s}}) = \mathscr{H}^d_{A \otimes B}(\boldsymbol{s}m)$ and $\mathscr{H}^d_{A \otimes B}(\tilde{\boldsymbol{t}}) = \mathscr{H}^d_{A \otimes B}(\boldsymbol{t}n)$, establishing $\boldsymbol{s}m \simeq^d_{A \otimes B} \boldsymbol{t}n$. □

Next, recall *linear implication* $\multimap$, which has been illustrated by examples in the introduction. The linear implication $A \multimap B$ is meant to be the 'space' of *linear maps* from A to B in the sense of linear logic (Girard, 1987), i.e., they consume exactly one input in A to produce an output in B. Strictly, a play in $A \multimap B$ consumes *at most one* input to produce an output since it is possible that no moves of A are performed at all during a play in $A \multimap B$. That is, $A \multimap B$ is actually the 'space' of *affine maps* from A to B. Nevertheless, we follow the convention to call it linear implication.

One additional point for *dynamic* games is that we need to apply the ω-hiding operation $\mathscr{H}^\omega$ to the domain A of each linear implication $A \multimap B$ since otherwise the linear implication $A \multimap B$ may not satisfy the axiom DP2. It conceptually makes sense too because the roles of Player and Opponent in A are exchanged, and thus Player should not be able to 'see' internal moves of A.

Definition 3.34 (Linear implication between games (Abramsky and McCusker, 1999)). *The* linear implication $A \multimap B$ *from a game A to another B is defined by*

- $M_{A \multimap B} := M_{\mathscr{H}^\omega(A)} + M_B$;
- $\lambda_{A \multimap B} := [\overline{\lambda_{\mathscr{H}^\omega(A)}}, \lambda_B]$, *where* $\overline{\lambda_{\mathscr{H}^\omega(A)}} := \langle \overline{\lambda^{\mathsf{OP}}_{\mathscr{H}^\omega(A)}}, \lambda^{\mathsf{QA}}_{\mathscr{H}^\omega(A)}, \lambda^{\mathbb{N}}_{\mathscr{H}^\omega(A)} \rangle$ *and for any game G*

$$\overline{\lambda^{\mathsf{OP}}_G}(m) := \begin{cases} \mathsf{P} & \text{if } \lambda^{\mathsf{OP}}_G(m) = \mathsf{O}; \\ \mathsf{O} & \text{otherwise;} \end{cases}$$

- $\star \vdash_{A \multimap B} m \overset{\text{df.}}{\Leftrightarrow} \star \vdash_B m$;
- $m \vdash_{A \multimap B} n \ (m \neq \star) \overset{\text{df.}}{\Leftrightarrow} (m \vdash_{\mathscr{H}^\omega(A)} n) \vee (m \vdash_B n) \vee (\star \vdash_B m \wedge \star \vdash_{\mathscr{H}^\omega(A)} n)$;
- $P_{A \multimap B} := \{ \boldsymbol{s} \in \mathscr{L}_{\mathscr{H}^\omega(A) \multimap B} \mid \boldsymbol{s} \upharpoonright \mathscr{H}^\omega(A) \in P_{\mathscr{H}^\omega(A)}, \boldsymbol{s} \upharpoonright B \in P_B \}$;
- $\boldsymbol{s} \simeq_{A \multimap B} \boldsymbol{t} \overset{\text{df.}}{\Leftrightarrow} \boldsymbol{s} \upharpoonright \mathscr{H}^\omega(A) \simeq_{\mathscr{H}^\omega(A)} \boldsymbol{t} \upharpoonright \mathscr{H}^\omega(A) \wedge \boldsymbol{s} \upharpoonright B \simeq_B \boldsymbol{t} \upharpoonright B \wedge \forall i \in \mathbb{N}.\, \boldsymbol{s}(i) \in M_{\mathscr{H}^\omega(A)} \Leftrightarrow \boldsymbol{t}(i) \in M_{\mathscr{H}^\omega(A)}$,

where pointers from an initial occurrence of $\mathscr{H}^\omega(A)$ to that of B in $\boldsymbol{s}$ are deleted.

Dually to $A \otimes B$, it is easy to see that during a play in $A \multimap B$ only Player may switch between $\mathscr{H}^\omega(A)$ and B (again by alternation); see Abramsky et al. (1997) for the details.

Example 3.35. The examples of linear implication in Section 1 illustrate how Definition 3.34 works.

Theorem 3.36 (Well defined linear implication between games). *Games (resp. well founded ones) are closed under linear implication $\multimap$.*

Proof. Again, it suffices to show preservation of the additional conditions on the labelling function and the axioms E1, E2, E4, DP2 and DI3. For brevity, assume that A is normalised and consider $A \multimap B$. Again, nontrivial conditions are just DP2 and DI3, but DI3 may be shown in a way similar to the case of tensor $\otimes$ and thus is omitted.

To verify DP2, let $i \in \mathbb{N}$ and $\boldsymbol{s}lmn, \boldsymbol{s}'l'm'n' \in P^{\mathsf{Odd}}_{A \multimap B}$ such that $\mathscr{H}^i_{A \multimap B}(\boldsymbol{s}lm) = \mathscr{H}^i_{A \multimap B}(\boldsymbol{s}'l'm')$ and $i < \lambda^{\mathbb{N}}_{A \multimap B}(n) = \lambda^{\mathbb{N}}_{A \multimap B}(n')$. Again, m and m' are both internal, and so m, n, m' and n' all belong to B. Thus, $(sl \upharpoonright B).mn, (s'l' \upharpoonright B).m'n' \in P_B^{\mathsf{Odd}}$ such that $\mathscr{H}_B^i((sl \upharpoonright B).m) = \mathscr{H}^i_{A \multimap B}(slm) \upharpoonright \mathscr{H}^i(B) = \mathscr{H}^i_{A \multimap B}(s'l'm') \upharpoonright \mathscr{H}^i(B) = \mathscr{H}^i_B((s'l' \upharpoonright B).m')$ and $i < \lambda_B^{\mathbb{N}}(n) = \lambda_B^{\mathbb{N}}(n')$. Then, by DP2 on B, we may conclude that $n = n'$ and $\mathcal{J}^{\ominus i}_{slmn}(n) = \mathcal{J}^{\ominus i}_{(sl \upharpoonright B).mn}(n) = \mathcal{J}^{\ominus i}_{(sl \upharpoonright B).mn}(n') = \mathcal{J}^{\ominus i}_{slmn}(n')$. □

Next, *product* & forms product in the categories of static games and strategies (Abramsky and McCusker, 1999). A position in the product $A \& B$ is simply a position in A or B:

Definition 3.37 (Product of games (Abramsky and McCusker, 1999)). *Given games A and B, the* product $A \& B$ *of A and B is defined by*

- $M_{A\&B} := M_A + M_B$;
- $\lambda_{A\&B} := [\lambda_A, \lambda_B]$;
- $\vdash_{A\&B} := \vdash_A + \vdash_B$;
- $P_{A\&B} := \{ \boldsymbol{s} \in \mathscr{L}_{A\&B} \mid (\boldsymbol{s} \upharpoonright A \in P_A \wedge \boldsymbol{s} \upharpoonright B = \boldsymbol{\varepsilon}) \vee (\boldsymbol{s} \upharpoonright A = \boldsymbol{\varepsilon} \wedge \boldsymbol{s} \upharpoonright B \in P_B) \}$;
- $\boldsymbol{s} \simeq_{A\&B} \boldsymbol{t} \overset{\mathrm{df.}}{\Leftrightarrow} \boldsymbol{s} \simeq_A \boldsymbol{t} \vee \boldsymbol{s} \simeq_B \boldsymbol{t}$.

Example 3.38. A maximal position in the product $\mathbf{2} \& N$ is either of the following forms:

$$\begin{tikzcd} \mathbf{2} & \& & N \\ q & & \\ b \arrow[u, bend right] & & \end{tikzcd} \qquad \begin{tikzcd} \mathbf{2} & \& & N \\ & & q \\ & & n \arrow[u, bend right] \end{tikzcd}$$

where $b \in \mathbb{B}$ and $n \in \mathbb{N}$.

Now, for our game-semantic CCBoC (given in Section 4), let us generalise product as follows:

Definition 3.39 (Pairing of games). *The* pairing $\langle L, R \rangle$ *of games L and R that satisfy $\mathscr{H}^\omega(L) \trianglelefteq C \multimap A$ and $\mathscr{H}^\omega(R) \trianglelefteq C \multimap B$ for some normalised games A, B and C is defined by*

- $M_{\langle L,R \rangle} := M_C + (M_L \setminus M_C) + (M_R \setminus M_C)$, *where 'tags' for the disjoint union is chosen in such a way that $\mathscr{H}^\omega(\langle L, R \rangle) \trianglelefteq C \multimap A \& B$ holds;*
- $\lambda_{\langle L,R \rangle} := [\overline{\lambda_C}, \lambda_L \downharpoonright M_C, \lambda_R \downharpoonright M_C]$;
- $m \vdash_{\langle L,R \rangle} n \overset{\mathrm{df.}}{\Leftrightarrow} (att_{\langle L,R \rangle}(m) = att_{\langle L,R \rangle}(n) \vee att_{\langle L,R \rangle}(m) = C \vee att_{\langle L,R \rangle}(n) = C) \wedge (peel_{\langle L,R \rangle}(m) \vdash_L peel_{\langle L,R \rangle}(n) \vee peel_{\langle L,R \rangle}(m) \vdash_R peel_{\langle L,R \rangle}(n))$;
- $P_{\langle L,R \rangle} := \{ \boldsymbol{s} \in \mathscr{L}_{L\&R} \mid (\boldsymbol{s} \upharpoonright L \in P_L \wedge \boldsymbol{s} \upharpoonright B = \boldsymbol{\varepsilon}) \vee (\boldsymbol{s} \upharpoonright A = \boldsymbol{\varepsilon} \wedge \boldsymbol{s} \upharpoonright R \in P_R) \}$;
- $\boldsymbol{s} \simeq_{\langle L,R \rangle} \boldsymbol{t} \overset{\mathrm{df.}}{\Leftrightarrow} (\boldsymbol{s} \upharpoonright A = \boldsymbol{\varepsilon} \Leftrightarrow \boldsymbol{t} \upharpoonright A = \boldsymbol{\varepsilon}) \wedge \boldsymbol{s} \upharpoonright L \simeq_L \boldsymbol{t} \upharpoonright L \wedge \boldsymbol{s} \upharpoonright R \simeq_R \boldsymbol{t} \upharpoonright R$,

where the map $peel_{\langle L,R \rangle} : M_{\langle L,R \rangle} \to M_L \cup M_R$ is the obvious left inverse of the 'tagging' for $M_{\langle L,R \rangle}$, $\boldsymbol{s} \upharpoonright L$ (resp. $\boldsymbol{s} \upharpoonright R$) is the j-subsequence of $\boldsymbol{s}$ that consists of moves x such that $peel_{\langle L,R \rangle}(x) \in M_L$ (resp. $peel_{\langle L,R \rangle}(x) \in M_R$) yet changed into $peel_{\langle L,R \rangle}(x)$, and the map $att_{\langle L,R \rangle} : M_{\langle L,R \rangle} \to \{L, R, C\}$ is given by

$$att_{\langle L,R \rangle}(m) := \begin{cases} L & \text{if } peel_{\langle L,R \rangle}(m) \in M_L \setminus M_C; \\ R & \text{if } peel_{\langle L,R \rangle}(m) \in M_R \setminus M_C; \\ C & \text{otherwise (i.e., if } peel_{\langle L,R \rangle}(m) \in M_C). \end{cases}$$

Pairing of games is indeed a generalisation of product because we have $\langle T \multimap A, T \multimap B \rangle = T \multimap A \& B$ for any games A and B, where note that each game G coincides with the linear implication $T \multimap G$ up to 'tags.' The point is that the *(generalised) pairing $\langle \sigma, \tau \rangle$ of strategies* $\sigma : L$ and $\tau : R$ forms a strategy on the pairing $\langle L, R \rangle$ (Definition 3.91).

Theorem 3.40 (Well defined pairing of games). *If games (resp. well founded ones) L and R satisfy $\mathscr{H}^\omega(L) \trianglelefteq C \multimap A$ and $\mathscr{H}^\omega(R) \trianglelefteq C \multimap B$ for normalised games A, B and C, then the pairing $\langle L, R\rangle$ is a game (resp. well founded one) that satisfies $\mathscr{H}^\omega(\langle L, R\rangle) \trianglelefteq C \multimap A\&B$.*

Proof. Similar to and simpler than the case of tensor $\otimes$. □

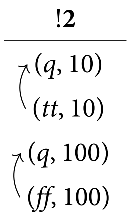
Now, let us recall *exponential* !, which is essentially the countably infinite iteration of tensor $\otimes$, i.e., $!A$ and $A \otimes A \otimes \dots$ coincide up to 'tags.' More precisely, it is defined as follows:

Definition 3.41 (Exponential of games (Abramsky et al. 2000; McCusker, 1998)). *Given a game A, the* exponential $!A$ *of A is defined by*

- $M_{!A} := M_A \times \mathbb{N}$;
- $\lambda_{!A} : (a, i) \mapsto \lambda_A(a)$;
- $\star \vdash_{!A} (a, i) \overset{\text{df.}}{\Leftrightarrow} \star \vdash_A a$;
- $(a, i) \vdash_{!A} (a', i') \overset{\text{df.}}{\Leftrightarrow} i = i' \wedge a \vdash_A a'$;
- $P_{!A} := \{ \boldsymbol{s} \in \mathscr{L}_{!A} \mid \forall i \in \mathbb{N}. \boldsymbol{s} \upharpoonright i \in P_A \}$;
- $\boldsymbol{s} \simeq_{!A} \boldsymbol{t} \overset{\text{df.}}{\Leftrightarrow} \exists \varphi \in \mathscr{P}(\mathbb{N}). \forall i \in \mathbb{N}. \boldsymbol{s} \upharpoonright \varphi(i) \simeq_A \boldsymbol{t} \upharpoonright i \wedge \pi_2^*(\boldsymbol{s}) = (\varphi \circ \pi_2)^*(\boldsymbol{t})$,

where $\boldsymbol{s} \upharpoonright i$ is the j-subsequence of $\boldsymbol{s}$ that consists of occurrences of moves of the form (a, i) yet changed into a, and $\mathscr{P}(\mathbb{N})$ is the set of all permutations of natural numbers.

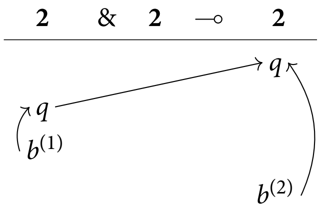
Example 3.42. A typical position in the exponential !**2** is as follows:

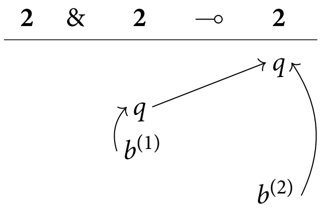


It is now clear, from the definition of $\simeq_{!A}$, why we equip each game with an identification of positions: A particular choice of the 'tag' $(\_, i)$ for an exponential $!A$ should not matter; since the identification may occur *locally* in games in a *nested* form, e.g., $!(!A \otimes B)$, $!!A \multimap !B$, etc., it gives a neat solution to define a *tailored* identification $\simeq_G$ of positions as part of the structure of each game G. Identifications of positions are originally introduced by Abramsky et al. (2000) and later employed in McCusker (1998, Section 3.6).

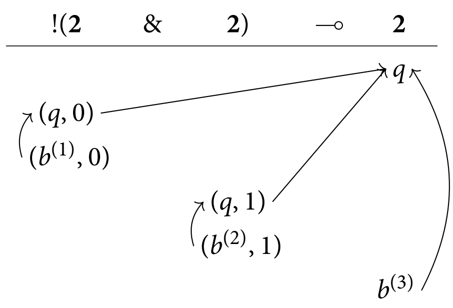
Exponential ! enables us, via *Girard's translation* (Girard, 1987) $A \Rightarrow B := !A \multimap B$, to model the construction $\Rightarrow$ of the usual *implication* (or the *function space*).

Example 3.43. In the linear implication $\mathbf{2}\&\mathbf{2} \multimap \mathbf{2}$, Player may play in at most only one copy of **2** out of the domain **2**&**2**:

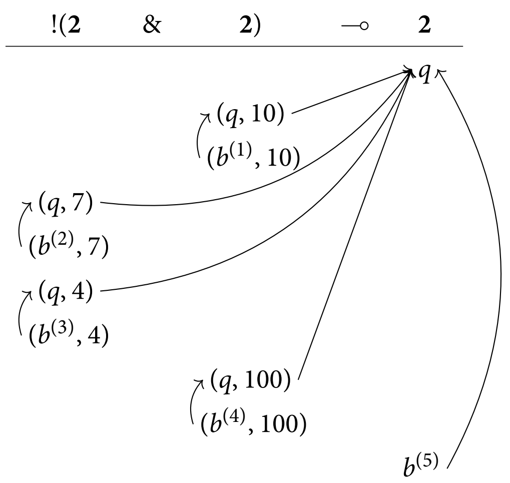




where $b^{(1)}, b^{(2)} \in \mathbb{B}$. On the other hand, positions in the implication $\mathbf{2}\&\mathbf{2} \Rightarrow \mathbf{2} = !(\mathbf{2}\&\mathbf{2}) \multimap \mathbf{2}$ are of the expected form; for instance:



$!(\mathbf{2} \quad \& \quad \mathbf{2}) \quad \multimap \quad \mathbf{2}$
q
$(q, 0)$
$(b^{(1)}, 0)$
$(q, 1)$
$(b^{(2)}, 1)$
$b^{(3)}$



where $b^{(1)}, b^{(2)}, b^{(3)}, b^{(4)}, b^{(5)} \in \mathbb{B}$. Hence, e.g., Player may play as conjunction $\wedge : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ or disjunction $\vee : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ on the implication $\mathbf{2}\&\mathbf{2} \Rightarrow \mathbf{2}$ in the obvious manner, but not on the linear implication $\mathbf{2}\&\mathbf{2} \multimap \mathbf{2}$. This example illustrates why the standard notion of functions corresponds in game semantics to implication $\Rightarrow$, not linear one $\multimap$.

For the game-semantic CCBoC, let us generalise exponential of games as follows:

Definition 3.44 (Promotion of games). *Fix once and for all an arbitrary bijection* $\langle \_, \_ \rangle : \mathbb{N} \times \mathbb{N} \xrightarrow{\sim} \mathbb{N}$. *Given a game* G *such that* $\mathscr{H}^{\omega}(G) \trianglelefteq\, !A \multimap B$ *for some normalised games* A *and* B, *the promotion* $G^{\dagger}$ *of* G *is defined by*

- $M_{G^\dagger} := ((M_G \setminus M_{!A}) \times \mathbb{N}) + M_{!A}$;
- $\lambda_{G^\dagger} : ((m, i) \in (M_G \setminus M_{!A}) \times \mathbb{N}) \mapsto \lambda_G(m), ((a, j) \in M_{!A}) \mapsto \lambda_G(a, j)$;
- $\star \vdash_{G^\dagger} (m, i) \overset{\text{df.}}{\Leftrightarrow} \star \vdash_G m$ *for all* $i \in \mathbb{N}$;
- $m \vdash_{G^\dagger} n \overset{\text{df.}}{\Leftrightarrow} att_{G^\dagger}(m) = att_{G^\dagger}(n) \wedge peel_{G^\dagger} \vdash_G peel_{G^\dagger}$, *where* $att_{G^\dagger}$ *(resp.* $peel_{G^\dagger}$*) is a function* $M_{G^\dagger} \to \mathbb{N}$ *(resp.* $M_{G^\dagger} \to M_G$*) that maps*

$$att_{G^\dagger} : (a, \langle i, j \rangle) \in M_{!A} \mapsto i \in \mathbb{N}, \quad (x, i) \in (M_{G^\dagger} \setminus M_{!A}) \times \mathbb{N} \mapsto i \in \mathbb{N}$$

$$peel_{G^\dagger} : (a, \langle i, j \rangle) \in M_{!A} \mapsto (a, j) \in M_{!A}, \quad (x, i) \in (M_{G^\dagger} \setminus M_{!A}) \times \mathbb{N} \mapsto x \in M_G \setminus M_{!A};$$

- $P_{G^\dagger} := \{ \boldsymbol{s} \in \mathscr{L}_{G^\dagger} \mid \forall i \in \mathbb{N}. \boldsymbol{s} \upharpoonright i \in P_G \}$;
- $\boldsymbol{s} \simeq_{G^\dagger} \boldsymbol{t} \overset{\text{df.}}{\Leftrightarrow} \exists \varphi \in \mathscr{P}(\mathbb{N}). \forall i \in \mathbb{N}. \boldsymbol{s} \upharpoonright \varphi(i) \simeq_G \boldsymbol{t} \upharpoonright i \wedge \pi_2^*(\boldsymbol{s}) = (\varphi \circ \pi_2)^*(\boldsymbol{t})$,

where $\boldsymbol{s} \upharpoonright i$ *is the j-subsequence of* $\boldsymbol{s}$ *that consists of moves* (m, i) *with* $m \in M_G \setminus M_{!A}$, *or* $(a, \langle i, j \rangle)$ *with* $a \in M_A \wedge j \in \mathbb{N}$, *yet changed into* m *or* (a, j), *respectively.*

Note that the equation $(!T \multimap A)^\dagger = !T \multimap !A$ holds for any game A, and therefore promotion $(\_)^\dagger$ is indeed a generalisation of exponential !. The point is that the *(generalised) promotion* $\phi^\dagger$ *of a strategy* $\phi : G$ forms a strategy on the promotion $G^\dagger$ (Definition 3.93).

Example 3.45. Let us consider the promotion $(!A \multimap B)^\dagger$, where A and B are arbitrary normalised games. If there is the position

$$\begin{tikzcd}
!A & \multimap & B \\
 & & b^{(1)} \\
(a^{(1)}, i) \arrow[urr] & & \\
(a^{(2)}, i) \arrow[u, bend left] & & \\
 & & b^{(2)} \arrow[uuu, bend right]
\end{tikzcd}$$

in $!A \multimap B$, then there is the position

$$\begin{tikzcd}
!A & \multimap & !B \\
 & & (b^{(1)}, j) \\
(a^{(1)}, \langle i, j \rangle) \arrow[urr] & & \\
(a^{(2)}, \langle i, j \rangle) \arrow[u, bend left] & & \\
 & & (b^{(2)}, j) \arrow[uuu, bend right] \\
 & & (b^{(1)}, j') \\
(a^{(1)}, \langle i, j' \rangle) \arrow[urr] & & \\
(a^{(2)}, \langle i, j' \rangle) \arrow[u, bend left] & & \\
 & & (b^{(2)}, j') \arrow[uuu, bend right]
\end{tikzcd}$$

in the promotion $(!A \multimap B)^\dagger$, where note that the 'tags' $j, j' \in \mathbb{N}$ are chosen by Opponent.

Theorem 3.46 (Well defined promotion of games). *If a (well founded) game G satisfies $\mathcal{H}^\omega(G) \trianglelefteq !A \multimap B$ for some normalised games A and B, then $G^\dagger$ is a (well founded) game that satisfies $\mathcal{H}^\omega(G)^\dagger \trianglelefteq !A \multimap !B$.*

Proof. Similar to the case of tensor $\otimes$. □

Now, let us introduce a new construction on games, which formalises the construction for 'internal communication' between strategies sketched in the introduction:

Definition 3.47 (Concatenation and composition of games). *Given games J and K such that $\mathcal{H}^\omega(J) \trianglelefteq A \multimap B$ and $\mathcal{H}^\omega(K) \trianglelefteq B \multimap C$ for some normalised games A, B and C, the concatenation $J \ddagger K$ of J and K is defined by*

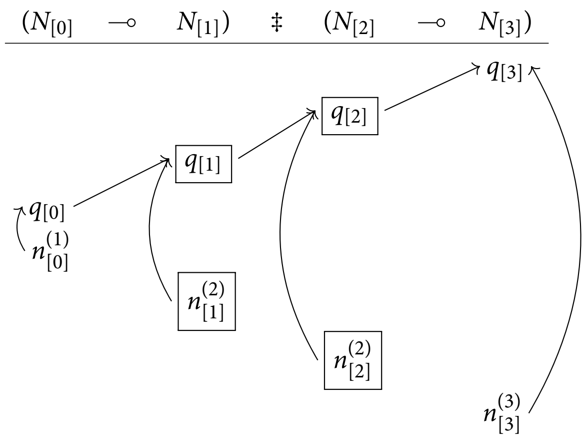
- *$M_{J \ddagger K} := M_J + M_K$, where 'tags' for the disjoint union is chosen in such a way that the subgame relation $\mathcal{H}^\omega(J \ddagger K) \trianglelefteq A \multimap C$ holds;*
- *$\lambda_{J \ddagger K} := [\lambda_J \downharpoonright M_{B_{[1]}}, \lambda_J^{+\mu} \upharpoonright M_{B_{[1]}}, \lambda_K^{+\mu} \upharpoonright M_{B_{[2]}}, \lambda_K \downharpoonright M_{B_{[2]}}]$, where $B_{[1]}$ and $B_{[2]}$ are the copies of B that belong to J and K, respectively, $\lambda_G^{+\mu} := \langle \lambda_G^{\mathsf{OP}}, \lambda_G^{\mathsf{QA}}, n \mapsto \lambda_G^{\mathbb{N}}(n) + \mu \rangle$ (G is J or K), and $\mu := \max(\mu(J), \mu(K)) + 1$ (see Definition 3.1 for $\mu(J)$ and $\mu(K)$);*
- *$\star \vdash_{J \ddagger K} m \overset{\mathrm{df.}}{\Leftrightarrow} \star \vdash_K m$;*
- *$m \vdash_{J \ddagger K} n \ (m \neq \star) \overset{\mathrm{df.}}{\Leftrightarrow} m \vdash_J n \vee m \vdash_K n \vee (\star \vdash_{B_{[2]}} m \wedge \star \vdash_{B_{[1]}} n)$;*
- *$P_{J \ddagger K} := \{ \boldsymbol{s} \in \mathbb{J}_{J \ddagger K} \mid \boldsymbol{s} \upharpoonright J \in P_J, \boldsymbol{s} \upharpoonright K \in P_K, \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \in pr_B \}$;*
- *$\boldsymbol{s} \simeq_{J \ddagger K} \boldsymbol{t} \overset{\mathrm{df.}}{\Leftrightarrow} (\forall i \in \mathbb{N}. \boldsymbol{s}(i) \in M_J \Leftrightarrow \boldsymbol{t}(i) \in M_J) \wedge \boldsymbol{s} \upharpoonright J \simeq_J \boldsymbol{t} \upharpoonright J \wedge \boldsymbol{s} \upharpoonright K \simeq_K \boldsymbol{t} \upharpoonright K,$*

where $pr_B := \{ \boldsymbol{s} \in P_{B_{[1]} \multimap B_{[2]}} \mid \forall \boldsymbol{t} \preceq \boldsymbol{s}. \, \mathsf{Even}(\boldsymbol{t}) \Rightarrow \boldsymbol{t} \upharpoonright B_{[1]} = \boldsymbol{t} \upharpoonright B_{[2]} \}$.

Moreover, the composition $J; K$ *(also written* $K \circ J$*) of* J *and* K *is defined by*

$$J; K := \mathscr{H}^{\omega}(J \ddagger K).$$

Example 3.48. A typical position in the concatenation $(N \multimap N) \ddagger (N \multimap N)$ is



where $n^{(1)}, n^{(2)}, n^{(3)} \in \mathbb{N}$. We have marked internal moves by a square box just for clarity. Note that this game coincides with the game given in Example 3.27.

We shall see that the 'non-hiding composition' or *concatenation* $\iota \ddagger \kappa$ of dynamic strategies $\iota : J$ and $\kappa : K$ is a dynamic strategy on the concatenation $J \ddagger K$. It generalises the particular case such that $J = A \multimap B$ and $K = B \multimap C$, so that $\iota; \kappa = \mathscr{H}^{\omega}(\iota \ddagger \kappa) : \mathscr{H}^{\omega}(J \ddagger K) = J; K = A \multimap C$ holds (as we shall establish), which reformulates the conventional composition of static strategies as *concatenation plus hiding* of dynamic strategies.

Theorem 3.49 (Well defined concatenation and composition of games). *Games (resp. well founded ones) are closed under concatenation and composition.*

Proof. By Theorem 3.28, it suffices to focus on concatenation, where well foundedness of games is clearly preserved under concatenation. We first show that the arena $J \ddagger K$ is well defined. The set $M_{J \ddagger K}$ and the function $\lambda_{J \ddagger K}$ are clearly well defined, where the *finite* upper bounds $\mu(J)$ and $\mu(K)$ are crucial. For the relation $\vdash_{J \ddagger K}$, E1 and E3 clearly hold. For E2, if $m \vdash_{J \ddagger K} n$ and $\lambda^{\mathrm{QA}}_{J \ddagger K}(n) = \mathsf{A}$, then $m, n \in M_K \setminus M_{B_{[2]}}$, $m, n \in M_{B_{[2]}}$, $m, n \in M_{B_{[1]}}$ or $m, n \in M_J \setminus M_{B_{[1]}}$. In either case, we have $\lambda^{\mathrm{QA}}_{J \ddagger K}(m) = \mathsf{Q}$ and $\lambda^{\mathbb{N}}_{J \ddagger K}(m) = \lambda^{\mathbb{N}}_{J \ddagger K}(n)$.

For E4, let $m \vdash_{J \ddagger K} n$, $m \neq \star$ and $\lambda^{\mathbb{N}}_{J \ddagger K}(m) \neq \lambda^{\mathbb{N}}_{J \ddagger K}(n)$. We proceed by the following case analysis. If $(m \vdash_K n) \wedge (m, n \in M_K \setminus M_{B_{[2]}} \vee m, n \in M_{B_{[2]}})$, then we may just apply E4 on K. It is similar if $(m \vdash_J n) \wedge (m, n \in M_J \setminus M_{B_{[1]}} \vee m, n \in M_{B_{[1]}})$. Note that the case $\star \vdash_{B_{[2]}} m \wedge \star \vdash_{B_{[1]}} n$ cannot happen. Now, consider the case $m \vdash_K n \wedge m \in M_K \setminus M_{B_{[2]}} \wedge n \in M_{B_{[2]}}$. If m is external, then $m \in M_C$, and so E4 on $J \ddagger K$ is satisfied by the definition of $B \multimap C$; if m is internal, then we may just apply E4 on K. The case $m \vdash_K n \wedge n \in M_K \setminus M_{B_{[2]}} \wedge m \in M_{B_{[2]}}$ is simpler as m must be internal. The remaining cases $m \vdash_J n \wedge m \in M_J \setminus M_{B_{[1]}} \wedge n \in M_{B_{[1]}}$ and $m \vdash_J n \wedge n \in M_J \setminus M_{B_{[1]}} \wedge m \in M_{B_{[1]}}$ are analogous. Hence, we have shown that the arena $J \ddagger K$ is well defined.

Next, we show that $P_{J \ddagger K} \subseteq \mathscr{L}_{J \ddagger K}$ holds. For justification, let $\boldsymbol{s}m \in P_{J \ddagger K}$ with m non-initial. The nontrivial case is when m is initial in J. But in this case, m is initial in $B_{[1]}$, and so it has a justifier in $B_{[2]}$. For alternation and IE-switch, similarly to Figure 1 for tensor $\otimes$, we have Figure 2 for $J \ddagger K$,

$$\begin{tikzcd}
 & (\mathsf{O}^{\mathsf{E}},\mathsf{O}^{\mathsf{E}}) \arrow[r, "C"] \arrow[d, shift right, "C"'] & (\mathsf{O}^{\mathsf{E}},\mathsf{P}^{\mathsf{I}}) \arrow[d, shift right, "K"'] \\
(\mathsf{P}^{\mathsf{I}},\mathsf{O}^{\mathsf{E}}) \arrow[d, shift right, "J"'] & (\mathsf{O}^{\mathsf{E}},\mathsf{P}^{\mathsf{E}}) \arrow[l, dotted, "B_{[1]}B_{[2]}"'] \arrow[u, shift right, "C"'] \arrow[d, dotted, shift right, "\substack{B_{[2]}\\ B_{[1]}}"'] & (\mathsf{O}^{\mathsf{E}},\mathsf{O}^{\mathsf{I}}) \arrow[l, "K"'] \arrow[u, shift right, "K"'] \arrow[d, shift right, "K"'] \\
(\mathsf{O}^{\mathsf{I}},\mathsf{O}^{\mathsf{E}}) \arrow[r, "J"] \arrow[u, shift right, "J"'] \arrow[d, shift right, "J"'] & (\mathsf{P}^{\mathsf{E}},\mathsf{O}^{\mathsf{E}}) \arrow[r, dotted, "B_{[1]}B_{[2]}"] \arrow[u, dotted, shift right, "\substack{B_{[2]}\\ B_{[1]}}"'] \arrow[d, shift right, "A"'] & (\mathsf{O}^{\mathsf{E}},\mathsf{P}^{\mathsf{I}}) \arrow[u, shift right, "K"'] \\
(\mathsf{P}^{\mathsf{I}},\mathsf{O}^{\mathsf{E}}) \arrow[u, shift right, "J"'] & (\mathsf{O}^{\mathsf{E}},\mathsf{O}^{\mathsf{E}}) \arrow[l, "A"'] \arrow[u, shift right, "A"'] &
\end{tikzcd}$$

Figure 2. The double parity diagram for the concatenation $J \ddagger K$.

in which the first (resp. the second) component of each state is about the OP- and IE-parities of the next move of J (resp. K). For readability, some states are written twice, and the dotted arrow indicates two necessarily consecutive moves of B. Then, alternation and IE-switch on $J \ddagger K$ immediately follow from this diagram and the corresponding axioms on J and K.

For generalised visibility, let $\boldsymbol{s}m \in P_{J \ddagger K}$ with m non-initial and $d \in \mathbb{N} \cup \{\omega\}$ such that $\boldsymbol{s}m$ is d-complete. Without loss of generality, we may assume $d \in \mathbb{N}$ since $\boldsymbol{s}$ is finite. It is not hard to see that $\mathscr{H}^d_{J\ddagger K}(\boldsymbol{s}m) \in P_{\mathscr{H}^d(J) \ddagger \mathscr{H}^d(K)}$ holds if $\mathscr{H}^d(J \ddagger K)$ is not normalised; thus, this case is reduced to the (usual) visibility on $\mathscr{H}^d(J) \ddagger \mathscr{H}^d(K)$. Otherwise, it is no harm to select the *least* $d \in \mathbb{N}^+$ such that $\mathscr{H}^d(J \ddagger K)$ is normalised; then $\mathscr{H}^{d-1}_{J\ddagger K}(\boldsymbol{s}m) \in P_{(A \multimap B_{[1]}) \ddagger (B_{[2]} \multimap C)}$ holds, and thus the visibility of $\mathscr{H}^d_{J\ddagger K}(\boldsymbol{s}m) = \mathscr{H}_{\mathscr{H}^{d-1}(J \ddagger K)}(\mathscr{H}^{d-1}_{J\ddagger K}(\boldsymbol{s}m))$ can be shown completely in the same way as the proof that shows the composition of static strategies is well defined (in particular it satisfies visibility) (Harmer, 2004; McCusker, 1998). As a consequence, it suffices to consider the case $d = 0$, i.e., to show the (usual) visibility.

For this, we need the following:

Lemma 3.50 (Visibility lemma). *Assume that $\boldsymbol{t} \in P_{J \ddagger K}$ and $\boldsymbol{t} \neq \boldsymbol{\epsilon}$.*

(1) *If the last move of $\boldsymbol{t}$ is of $M_J \setminus M_{B_{[1]}}$, then $\lceil \boldsymbol{t} \upharpoonright J \rceil_J \preceq \lceil \boldsymbol{t} \rceil_{J \ddagger K} \upharpoonright J$ and $\lfloor \boldsymbol{t} \upharpoonright J \rfloor_J \preceq \lfloor \boldsymbol{t} \rfloor_{J \ddagger K} \upharpoonright J$;*
(2) *If the last move of $\boldsymbol{t}$ is of $M_K \setminus M_{B_{[2]}}$, then $\lceil \boldsymbol{t} \upharpoonright K \rceil_K \preceq \lceil \boldsymbol{t} \rceil_{J \ddagger K} \upharpoonright K$ and $\lfloor \boldsymbol{t} \upharpoonright K \rfloor_K \preceq \lfloor \boldsymbol{t} \rfloor_{J \ddagger K} \upharpoonright K$;*
(3) *If the last move of $\boldsymbol{t}$ is an O-move of $M_{B_{[1]}} \cup M_{B_{[2]}}$, then $\lceil \boldsymbol{t} \upharpoonright B_{[1]}, B_{[2]} \rceil_{B_{[1]} \multimap B_{[2]}} \preceq \lfloor \boldsymbol{t} \rfloor_{J \ddagger K} \upharpoonright B_{[1]}, B_{[2]}$ and $\lfloor \boldsymbol{t} \upharpoonright B_{[1]}, B_{[2]} \rfloor_{B_{[1]} \multimap B_{[2]}} \preceq \lceil \boldsymbol{t} \rceil_{J \ddagger K} \upharpoonright B_{[1]}, B_{[2]}$.*

Proof of the lemma. By induction on $|\boldsymbol{t}|$ with case analysis on the last move of $\boldsymbol{t}$. □

Note that we may write $\boldsymbol{s}m = \boldsymbol{s_1}n\boldsymbol{s_2}m$, where n justifies m. If $\boldsymbol{s_2} = \boldsymbol{\epsilon}$, then it is trivial; so assume $\boldsymbol{s_2} = \boldsymbol{s'_2}r$. We then proceed by the following case analysis on m:

- Assume $m \in M_J \setminus M_{B_{[1]}}$. By Figure 2, $n \in M_J$ and $r \in M_J$. By Lemma 3.50, $\lceil \boldsymbol{s} \upharpoonright J \rceil \preceq \lceil \boldsymbol{s} \rceil \upharpoonright J$ and $\lfloor \boldsymbol{s} \upharpoonright J \rfloor \preceq \lfloor \boldsymbol{s} \rfloor \upharpoonright J$. Also, by $(\boldsymbol{s} \upharpoonright J).m \in P_J$ and visibility on J, n occurs in $\lceil \boldsymbol{s} \upharpoonright J \rceil$ (resp. $\lfloor \boldsymbol{s} \upharpoonright J \rfloor$) if m is a P- (resp. O-) move. Thus, n occurs in $\lceil \boldsymbol{s} \rceil$ (resp. $\lfloor \boldsymbol{s} \rfloor$) if m is a P- (resp. O-) move.
- Assume $m \in M_K \setminus M_{B_{[2]}}$. This case can be handled in a completely analogous way to the above case.
- Assume $m \in M_{B_{[1]}}$. If m is a P-move, then $n, r \in M_J$ and so it can be handled in the same way as the case $m \in M_J \setminus M_{B_{[1]}}$; thus, assume that m is an O-move. Then, $r \in M_{B_{[2]}}$, and it is a 'copy' of m. Since r is an O-move in $B_{[1]} \multimap B_{[2]}$, by Lemma 3.50, $\lceil \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \rceil \preceq$

$\lfloor \boldsymbol{s} \rfloor \upharpoonright B_{[1]}, B_{[2]}$. Note that n is a move of $B_{[1]}$ or an initial move of $B_{[2]}$. In either case, $(\boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]}).m \in P_{B_{[1]} \multimap B_{[2]}}$; thus, n occurs in $\lceil \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \rceil$. Hence, n occurs in $\lfloor \boldsymbol{s} \rfloor$.

- Assume $m \in M_{B_{[2]}}$. If m is a P-move, then $n, r \in M_K$; so it can be dealt with in the same way as the case $m \in M_K \setminus M_{B_{[2]}}$. Thus, assume m is an O-move. By Figure 2, we have $r \in M_{B_{[1]}}$, and it is an O-move in $B_{[1]} \multimap B_{[2]}$. Thus, by Lemma 3.50, $\lceil \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \rceil \preceq \lfloor \boldsymbol{s} \rfloor \upharpoonright B_{[1]}, B_{[2]}$. Then again, $(\boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]}).m \in P_{B_{[1]} \multimap B_{[2]}}$; thus, n occurs in $\lceil \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \rceil$, and so n occurs in $\lfloor \boldsymbol{s} \rfloor$.

Next, we verify P1 and DP2. For P1, $\boldsymbol{\epsilon} \in P_{J \ddagger K}$ is clear; for prefix-closure, let $\boldsymbol{s}m \in P_{J \ddagger K}$. If $m \in M_J \setminus M_{B_{[1]}}$, then $(\boldsymbol{s} \upharpoonright J).m = \boldsymbol{s}m \upharpoonright J \in P_J$; thus, $\boldsymbol{s} \upharpoonright J \in P_J$, $\boldsymbol{s} \upharpoonright K = \boldsymbol{s}m \upharpoonright K \in P_K$ and $\boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} = \boldsymbol{s}m \upharpoonright B_{[1]}, B_{[2]} \in pr_B$, whence $\boldsymbol{s} \in P_{J \ddagger K}$. The other cases may be handled similarly.

For DP2, let $i \in \mathbb{N}$ and $\boldsymbol{s}m, \boldsymbol{s}'m' \in P_{J \ddagger K}^{\mathsf{Odd}}$ such that $i < \lambda_{J \ddagger K}^{\mathbb{N}}(m) = \lambda_{J \ddagger K}^{\mathbb{N}}(m')$ and $\mathscr{H}_{J \ddagger K}^{i}(\boldsymbol{s}) = \mathscr{H}_{J \ddagger K}^{i}(\boldsymbol{s}')$. Without loss of generality, we may assume $i = 0$ and $\lambda_{J \ddagger K}^{\mathbb{N}}(m) = 1 = \lambda_{J \ddagger K}^{\mathbb{N}}(m')$ because if $\lambda_{J \ddagger K}^{\mathbb{N}}(m) = \lambda_{J \ddagger K}^{\mathbb{N}}(m') = j > 1$, then we may consider $\mathscr{H}_{J \ddagger K}^{j-1}(\boldsymbol{s}m), \mathscr{H}_{J \ddagger K}^{j-1}(\boldsymbol{s}'m') \in P_{\mathscr{H}^{j-1}(J) \ddagger \mathscr{H}^{j-1}(K)}$ (n.b., the justifiers of m and m' have the same priority order). Thus, $\boldsymbol{s} = \boldsymbol{s}'$ and $m, m' \in M_J \vee m, m' \in M_K$. If $m, m' \in M_J$ (resp. $m, m' \in M_K$), then $(\boldsymbol{s} \upharpoonright J).m, (\boldsymbol{s}' \upharpoonright J).m' \in P_J^{\mathsf{Odd}}$ (resp. $(\boldsymbol{s} \upharpoonright K).m, (\boldsymbol{s}' \upharpoonright K).m' \in P_K^{\mathsf{Odd}}$), and so we may just apply DP2 on J (resp. K).

Finally, I1, I2 and DI3 on $\simeq_{J \ddagger K}$ can be verified similarly to the case of tensor $\otimes$, completing the proof. $\square$

For completeness, let us explicitly define the rather trivial *currying* of games:

Definition 3.51 (Currying of games). *Given a game G such that $\mathscr{H}^{\omega}(G) \trianglelefteq A \otimes B \multimap C$ for some normalised games A, B and C, the* currying *$\Lambda(G)$ of G is G up to 'tags' that satisfies $\mathscr{H}^{\omega}(\Lambda(G)) \trianglelefteq A \multimap (B \multimap C)$.*

Trivially, games (resp. well founded ones) are closed under currying.

Next, we show that these constructions as well as the hiding operation preserve the subgame relation $\trianglelefteq$ (Definition 3.24):

Notation 3.52. *We write $\clubsuit_{i \in I}$, where I is $\{1\}$ or $\{1, 2\}$, for any of the constructions on games introduced so far, i.e., $\clubsuit_{i \in I}$ is either $\otimes$, $\multimap$, $\langle \_, \_ \rangle$, $(\_)^{\dagger}$, $\ddagger$ or Λ.*

Lemma 3.53 (Preservation of subgames). *Let $\clubsuit_{i \in I}$ be a construction on games and assume $H_i \trianglelefteq G_i$ for all $i \in I$. Then, $\clubsuit_{i \in I} H_i \trianglelefteq \clubsuit_{i \in I} G_i$.*

Proof. Let us first consider tensor $\otimes$. It is trivial to check the conditions on the sets of moves and the labelling functions, and so we omit them. For the enabling relations, we calculate

$$\begin{aligned}
\vdash_{H_1 \otimes H_2} &= \vdash_{H_1} + \vdash_{H_2} \\
&\subseteq (\vdash_{G_1} \cap ((\{\star\} \cup M_{H_1}) \times M_{H_1})) + (\vdash_{G_2} \cap ((\{\star\} \cup M_{H_2}) \times M_{H_2})) \\
&= (\vdash_{G_1} \cap ((\{\star\} \cup M_{H_1 \otimes H_2}) \times M_{H_1 \otimes H_2})) + (\vdash_{G_2} \cap ((\{\star\} \cup M_{H_1 \otimes H_2}) \times M_{H_1 \otimes H_2})) \\
&= (\vdash_{G_1} + \vdash_{G_2}) \cap ((\{\star\} \cup M_{H_1 \otimes H_2}) \times M_{H_1 \otimes H_2}) \\
&= \vdash_{G_1 \otimes G_2} \cap ((\{\star\} \cup M_{H_1 \otimes H_2}) \times M_{H_1 \otimes H_2}).
\end{aligned}$$

For the positions, we calculate

$$\begin{aligned}
P_{H_1 \otimes H_2} &= \{\, \boldsymbol{s} \in \mathscr{L}_{H_1 \otimes H_2} \mid \forall i \in \{1, 2\}.\, \boldsymbol{s} \upharpoonright H_i \in P_{H_i} \,\} \\
&\subseteq \{\, \boldsymbol{s} \in \mathscr{L}_{G_1 \otimes G_2} \mid \forall i \in \{1, 2\}.\, \boldsymbol{s} \upharpoonright G_i \in P_{G_i} \,\} \\
&= P_{G_1 \otimes G_2}.
\end{aligned}$$

For the identifications of positions, given $d \in \mathbb{N} \cup \{\omega\}$, we calculate

$$\begin{aligned}
\boldsymbol{s} \simeq^{d}_{H_1 \otimes H_2} \boldsymbol{t} &\Leftrightarrow \exists \boldsymbol{s}', \boldsymbol{t}' \in P_{H_1 \otimes H_2}.\, \boldsymbol{s}' \simeq_{H_1 \otimes H_2} \boldsymbol{t}' \wedge \mathscr{H}^d_{H_1 \otimes H_2}(\boldsymbol{s}') = \mathscr{H}^d_{H_1 \otimes H_2}(\boldsymbol{s}) \wedge \mathscr{H}^d_{H_1 \otimes H_2}(\boldsymbol{t}') = \mathscr{H}^d_{H_1 \otimes H_2}(\boldsymbol{t}) \\
&\Leftrightarrow \forall j \in \{1, 2\}.\, \exists \boldsymbol{s}'_j, \boldsymbol{t}'_j \in P_{H_j}.\, \boldsymbol{s}'_j \simeq_{H_j} \boldsymbol{t}'_j \wedge \mathscr{H}^d_{H_j}(\boldsymbol{s}'_j) = \mathscr{H}^d_{H_j}(\boldsymbol{s} \upharpoonright H_j) \\
&\wedge \mathscr{H}^d_{H_j}(\boldsymbol{t}'_j) = \mathscr{H}^d_{H_j}(\boldsymbol{t} \upharpoonright H_j) \wedge \forall k \in \mathbb{N}.\, s_k \in M_{H_1} \Leftrightarrow t_k \in M_{H_1} \\
&\Leftrightarrow \forall j \in \{1, 2\}.\, \boldsymbol{s} \upharpoonright H_j \simeq^d_{H_j} \boldsymbol{t} \upharpoonright H_j \wedge \forall k \in \mathbb{N}.\, s_k \in M_{H_1} \Leftrightarrow t_k \in M_{H_1} \\
&\Leftrightarrow \forall j \in \{1, 2\}.\, \boldsymbol{s} \upharpoonright G_j, \boldsymbol{t} \upharpoonright G_j \in P_{H_j} \wedge \boldsymbol{s} \upharpoonright G_j \simeq^d_{G_j} \boldsymbol{t} \upharpoonright G_j \wedge \forall k \in \mathbb{N}.\, s_k \in M_{G_1} \Leftrightarrow t_k \in M_{G_1} \\
&\Leftrightarrow \boldsymbol{s}, \boldsymbol{t} \in P_{H_1 \otimes H_2} \wedge \boldsymbol{s} \simeq^d_{G_1 \otimes G_2} \boldsymbol{t}.
\end{aligned}$$

Finally, we have $\mu(H_1 \otimes H_2) = \max(\mu(H_1), \mu(H_2)) = \max(\mu(G_1), \mu(G_2)) = \mu(G_1 \otimes G_2)$, showing that $H_1 \otimes H_2 \trianglelefteq G_1 \otimes G_2$.

Linear implication $\multimap$ and promotion $(\_)^\dagger$ are similar, and pairing $\langle \_, \_ \rangle$ and currying Λ are even simpler; thus, we omit them. Next, let us consider concatenation $\ddagger$. Assume that $\mathscr{H}^\omega(H_1) \trianglelefteq A \multimap B$, $\mathscr{H}^\omega(H_2) \trianglelefteq B \multimap C$, $\mathscr{H}^\omega(G_1) \trianglelefteq D \multimap E$, $\mathscr{H}^\omega(G_2) \trianglelefteq E \multimap F$ for some normalised games A, B, C, D, E and F; without loss of generality, we assume that these normalised games are the least ones with respect to $\trianglelefteq$. By Theorem 3.28, $\mathscr{H}^\omega(H_1) \trianglelefteq \mathscr{H}^\omega(G_1) \trianglelefteq D \multimap E$ and $\mathscr{H}^\omega(H_2) \trianglelefteq \mathscr{H}^\omega(G_2) \trianglelefteq E \multimap F$, which in turn implies $A \trianglelefteq D$, $B \trianglelefteq E$ and $C \trianglelefteq F$.

First, we clearly have $M_{H_1 \ddagger H_2} \subseteq M_{G_1 \ddagger G_2}$ and $\lambda_{G_1 \ddagger G_2} \upharpoonright M_{H_1 \ddagger H_2} = \lambda_{H_1 \ddagger H_2}$, where $\mu(H_i) = \mu(G_i)$ for $i = 1, 2$ ensures that the priority orders of moves of B coincide.

Next, for the enabling relations, we have

$$\star \vdash_{H_1 \ddagger H_2} m \Leftrightarrow \star \vdash_{H_2} m \Leftrightarrow \star \vdash_C m \Rightarrow \star \vdash_F m \Leftrightarrow \star \vdash_{G_1 \ddagger G_2} m$$

as well as

$$\begin{aligned}
m \vdash_{H_1 \ddagger H_2} n &\Leftrightarrow m \vdash_{H_1} n \vee m \vdash_{H_2} n \vee (\star \vdash_{B_{[2]}} m \wedge \star \vdash_{B_{[1]}} n) \\
&\Rightarrow m \vdash_{G_1} n \vee m \vdash_{G_2} n \vee (\star \vdash_{E_{[2]}} m \wedge \star \vdash_{E_{[1]}} n) \\
&\Leftrightarrow m \vdash_{G_1 \ddagger G_2} n
\end{aligned}$$

for any $m, n \in M_{H_1 \ddagger H_2}$. For the positions, we have

$$\begin{aligned}
P_{H_1 \ddagger H_2} &= \{ \boldsymbol{s} \in \mathbb{J}_{H_1 \ddagger H_2} \mid \boldsymbol{s} \upharpoonright H_1 \in P_{H_1}, \boldsymbol{s} \upharpoonright H_2 \in P_{H_2}, \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \in pr_B \} \\
&\subseteq \{ \boldsymbol{s} \in \mathbb{J}_{G_1 \ddagger G_2} \mid \boldsymbol{s} \upharpoonright G_1 \in P_{G_1}, \boldsymbol{s} \upharpoonright G_2 \in P_{G_2}, \boldsymbol{s} \upharpoonright E_{[1]}, E_{[2]} \in pr_E \} \\
&= P_{G_1 \ddagger G_2}.
\end{aligned}$$

Finally, we may show, in the same manner as in the case of tensor $\otimes$, the required condition on the identifications of positions, completing the proof. □

At the end of the present section, we establish the following useful lemma:

Lemma 3.54 (Hiding lemma on games). *Let $\clubsuit_{i \in I}$ be a construction on games, and G_i a game for each $i \in I$. Given $d \in \mathbb{N} \cup \{\omega\}$, we have*

(1) $\mathscr{H}^d(\clubsuit_{i \in I} G_i) = \clubsuit_{i \in I} \mathscr{H}^d(G_i)$ *if* $\clubsuit_{i \in I} \neq \ddagger$;
(2) $\mathscr{H}^d((G_1) \ddagger (G_2)) \trianglelefteq A \multimap C$ *if* $\mathscr{H}^d(G_1 \ddagger G_2)$ *is normalised, where A, B and C are normalised games such that* $\mathscr{H}^\omega(G_1) \trianglelefteq A \multimap B$ *and* $\mathscr{H}^\omega(G_2) \trianglelefteq B \multimap C$, *and* $(A \multimap B); (B \multimap C) = A \multimap C$;
(3) $\mathscr{H}^d(G_1 \ddagger G_2) = \mathscr{H}^d(G_1) \ddagger \mathscr{H}^d(G_2)$ *otherwise.*

Proof. Since there is an upper bound of the priority orders of each game, it suffices to consider the case $d \in \mathbb{N}$. But then, because $\mathscr{H}^{i+1} = \mathscr{H} \circ \mathscr{H}^i$ for all $i \in \mathbb{N}$, we may focus on $d = 1$. Let us focus on tensor $\otimes$ since the other constructions may be handled similarly.

We have to show $\mathscr{H}(G_1 \otimes G_2) = \mathscr{H}(G_1) \otimes \mathscr{H}(G_2)$. Clearly, their sets of moves and labelling functions coincide. For the enabling relations, we calculate

$$\begin{aligned}\star \vdash_{\mathscr{H}(G_1 \otimes G_2)} m &\Leftrightarrow \star \vdash_{G_1 \otimes G_2} m \Leftrightarrow \star \vdash_{G_1} m \vee \star \vdash_{G_2} m \\ &\Leftrightarrow \star \vdash_{\mathscr{H}(G_1)} m \vee \star \vdash_{\mathscr{H}(G_2)} m \\ &\Leftrightarrow \star \vdash_{\mathscr{H}(G_1) \otimes \mathscr{H}(G_2)} m\end{aligned}$$

as well as

$$\begin{aligned}&m \vdash_{\mathscr{H}(G_1 \otimes G_2)} n \ (m \neq \star) \\ &\quad \Leftrightarrow \ (m \vdash_{G_1 \otimes G_2} n) \vee \exists k \in \mathbb{N}^+, m_1, m_2, \ldots, m_{2k} \in M_{G_1 \otimes G_2} \setminus M_{\mathscr{H}(G_1 \otimes G_2)}.\, m \vdash_{G_1 \otimes G_2} m_1 \\ &\qquad \wedge \forall i \in \overline{2k-1}.\, m_i \vdash_{G_1 \otimes G_2} m_{i+1} \wedge m_{2k} \vdash_{G_1 \otimes G_2} n \\ &\quad \Leftrightarrow \ (m \vdash_{G_1} n \vee m \vdash_{G_2} n) \vee \exists i \in \{1,2\}, k \in \mathbb{N}^+, m_1, m_2, \ldots, m_{2k} \in M_{G_i} \setminus M_{\mathscr{H}(G_i)}.\, m \vdash_{G_i} m_1 \\ &\qquad \wedge \forall j \in \overline{2k-1}.\, m_j \vdash_{G_i} m_{j+1} \wedge m_{2k} \vdash_{G_i} n \\ &\quad \Leftrightarrow \ \exists i \in \{1,2\}.\, m \vdash_{G_i} n \vee \exists k \in \mathbb{N}^+, m_1, m_2, \ldots, m_{2k} \in M_{G_i} \setminus M_{\mathscr{H}(G_i)}.\, m \vdash_{G_i} m_1 \\ &\qquad \wedge \forall j \in \overline{2k-1}.\, m_j \vdash_{G_i} m_{j+1} \wedge m_{2k} \vdash_{G_i} n \\ &\quad \Leftrightarrow \ m \vdash_{\mathscr{H}(G_1) \otimes \mathscr{H}(G_2)} n.\end{aligned}$$

We have shown that the arenas $\mathscr{H}(G_1 \otimes G_2)$ and $\mathscr{H}(G_1) \otimes \mathscr{H}(G_2)$ coincide.

For the positions, we have

$$\begin{aligned}&\boldsymbol{s} \in P_{\mathscr{H}(G_1 \otimes G_2)} \\ &\Leftrightarrow \exists \boldsymbol{t} \in \mathscr{L}_{G_1 \otimes G_2}.\, \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) = \boldsymbol{s} \wedge \forall i \in \{1,2\}.\, \boldsymbol{t} \upharpoonright G_i \in P_{G_i} \\ &\Leftrightarrow \exists \boldsymbol{t} \in \mathscr{L}_{G_1 \otimes G_2}.\, \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) = \boldsymbol{s} \wedge \forall i \in \{1,2\}.\, \mathscr{H}_{G_i}(\boldsymbol{t} \upharpoonright G_i) \in P_{\mathscr{H}(G_i)} \\ &\Leftrightarrow \exists \boldsymbol{t} \in \mathscr{L}_{G_1 \otimes G_2}.\, \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) = \boldsymbol{s} \wedge \forall i \in \{1,2\}.\, \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) \upharpoonright \mathscr{H}(G_i) \in P_{\mathscr{H}(G_i)} \\ &\quad (\Leftarrow \text{ is by induction on } |\boldsymbol{t}|) \\ &\Leftrightarrow \boldsymbol{s} \in \mathscr{L}_{\mathscr{H}(G_1 \otimes G_2)} = \mathscr{L}_{\mathscr{H}(G_1) \otimes \mathscr{H}(G_2)} \wedge \forall i \in \{1,2\}.\, \boldsymbol{s} \upharpoonright \mathscr{H}(G_i) \in P_{\mathscr{H}(G_i)} \\ &\Leftrightarrow \boldsymbol{s} \in P_{\mathscr{H}(G_1) \otimes \mathscr{H}(G_2)}.\end{aligned}$$

Finally, for the identifications of positions, given $d \in \mathbb{N} \cup \{\omega\}$, we have

$$\begin{aligned}&\mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{s}) \simeq^d_{\mathscr{H}(G_1 \otimes G_2)} \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) \\ &\Leftrightarrow \exists \boldsymbol{s}', \boldsymbol{t}' \in P_{G_1 \otimes G_2}.\, \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{s}') \simeq_{\mathscr{H}(G_1 \otimes G_2)} \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}') \wedge \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{s}') = \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{s}) \\ &\quad \wedge \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{t}') = \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{t}) \\ &\Leftrightarrow \forall j \in \{1,2\}.\, \exists \boldsymbol{s}'_j, \boldsymbol{t}'_j \in P_{G_j}. \mathscr{H}_{G_j}(\boldsymbol{s}'_j) \simeq_{\mathscr{H}(G_j)} \mathscr{H}_{G_j}(\boldsymbol{t}'_j) \wedge \mathscr{H}^{d+1}_{G_j}(\boldsymbol{s}'_j) = \mathscr{H}^{d+1}_{G_j}(\boldsymbol{s} \upharpoonright G_j) \\ &\quad \wedge \mathscr{H}^{d+1}_{G_j}(\boldsymbol{t}'_j) = \mathscr{H}^{d+1}_{G_j}(\boldsymbol{t} \upharpoonright G_j) \wedge \forall k \in \mathbb{N}.\, \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{s}(k)) \in M_{\mathscr{H}^{d+1}(G_1)} \\ &\quad \Leftrightarrow \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{t}(k)) \in M_{\mathscr{H}^{d+1}(G_1)} \\ &\Leftrightarrow \forall j \in \{1,2\}.\, \boldsymbol{s} \upharpoonright G_j \simeq^{d+1}_{G_j} \boldsymbol{t} \upharpoonright G_j \wedge \forall k \in \mathbb{N}. \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{s}(k)) \in M_{\mathscr{H}^{d+1}(G_1)} \\ &\quad \Leftrightarrow \mathscr{H}^{d+1}_{G_1 \otimes G_2}(\boldsymbol{t}(k)) \in M_{\mathscr{H}^{d+1}(G_1)} \\ &\Leftrightarrow \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{s}) \simeq^d_{\mathscr{H}(G_1) \otimes \mathscr{H}(G_2)} \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) \wedge \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{s}), \mathscr{H}_{G_1 \otimes G_2}(\boldsymbol{t}) \in P_{\mathscr{H}(G_1 \otimes G_2)},\end{aligned}$$

which completes the proof. □

Example 3.55. For instance, as Lemma 3.54 states, we have

$$\begin{aligned}\mathscr{H}(((N \multimap N) \ddagger (N \multimap N)) \ddagger (N \multimap N)) &= \mathscr{H}((N \multimap N) \ddagger (N \multimap N)) \ddagger \mathscr{H}(N \multimap N)\\ &= (N \multimap N) \ddagger (N \multimap N).\end{aligned}$$

3.4 Dynamic strategies

Dynamic strategies, another central notion of the present work, are just static strategies (Abramsky and McCusker, 1999) *on dynamic games*:

Definition 3.56 (Dynamic strategies). *A* dynamic strategy *on a (dynamic) game G is a subset $\sigma \subseteq P_G^{\mathsf{Even}}$, written $\sigma : G$, that is*

- (S1). *Nonempty and* even-prefix-closed *(i.e., $\boldsymbol{s}mn \in \sigma \Rightarrow \boldsymbol{s} \in \sigma$);*
- (S2). Deterministic *(i.e., $\boldsymbol{s}mn, \boldsymbol{s}'m'n' \in \sigma \wedge \boldsymbol{s}m = \boldsymbol{s}'m' \Rightarrow \boldsymbol{s}mn = \boldsymbol{s}'m'n'$).*

A dynamic strategy $\sigma : G$ is said to be normalised *if $\forall \boldsymbol{s} \in \sigma, \forall i \in \overline{|\boldsymbol{s}|}.\ \lambda_G^{\mathbb{N}}(\boldsymbol{s}(i)) = 0$.*

Therefore, a dynamic strategy is normalised if and only if it consists of external moves only. Note that a dynamic strategy on a normalised dynamic game is necessarily normalised, and it is equivalent to a static strategy.

Convention. Henceforth, a *strategy* refers to a dynamic strategy by default.

Example 3.57. The normalised strategies *succ*, *double* : $N_{[0]} \multimap N_{[1]}$ sketched in the introduction are given formally by

$$succ := \{\, q_{[1]}q_{[0]}n_{[0]}(n+1)_{[1]} \mid n \in \mathbb{N} \,\} \qquad double := \{\, q_{[1]}q_{[0]}n'_{[0]}2n'_{[1]} \mid n' \in \mathbb{N} \,\}.$$

The non-normalised strategy obtained from *succ* and *double* by 'non-hiding composition,' which we write $succ \ddagger double : (N_{[0]} \multimap N_{[1]}) \ddagger (N_{[2]} \multimap N_{[3]})$, is given by

$$succ \ddagger double := \{\, q_{[3]}q_{[2]}q_{[1]}q_{[0]}n''_{[0]}(n''+1)_{[1]}(n''+1)_{[2]}2(n''+1)_{[3]} \mid n'' \in \mathbb{N} \,\}.$$

Since positions of a game G are identified modulo $\simeq_G$, we must identify strategies on G if they behave in the same manner modulo $\simeq_G$, leading to

Definition 3.58 (Identification of strategies (Abramsky et al. 2000)). *The* identification of strategies *on a game G, written $\simeq_G$, is the relation between strategies $\sigma, \tau : G$ defined by*

$$\begin{aligned}\sigma \simeq_G \tau \overset{\mathrm{df.}}{\Leftrightarrow} \forall \boldsymbol{s} \in \sigma, \boldsymbol{t} \in \tau, \boldsymbol{s}m, \boldsymbol{t}l \in P_G.\ \boldsymbol{s}m \simeq_G \boldsymbol{t}l \Rightarrow (&\forall \boldsymbol{s}mn \in \sigma.\ \exists \boldsymbol{t}lr \in \tau.\ \boldsymbol{s}mn \simeq_G \boldsymbol{t}lr)\\ &\wedge (\forall \boldsymbol{t}lr \in \tau.\ \exists \boldsymbol{s}mn \in \sigma.\ \boldsymbol{t}lr \simeq_G \boldsymbol{s}mn).\end{aligned}$$

We are particularly concerned with strategies identified with themselves:

Definition 3.59 (Validity of strategies). *A strategy $\sigma : G$ is* valid *if $\sigma \simeq_G \sigma$.*

Example 3.60. The normalised strategies $succ_0, succ_1 : N \Rightarrow N$ given by:

$$succ_0 := \{\, q(q,0)(n,0)(n+1) \mid n \in \mathbb{N} \,\} \qquad succ_1 := \{\, q(q,1)(n',1)(n'+1) \mid n' \in \mathbb{N} \,\}$$

are both valid and identified with each other by the identification $\simeq_{N \Rightarrow N}$.

On the other hand, the normalised strategies $succ_t, double_t : !N$ given by:

$$succ_t := \{ (q,i)(i+1,i) \mid i \in \mathbb{N} \} \qquad double_t := \{ (q,j)(2j,j) \mid j \in \mathbb{N} \}$$

are clearly not valid, and they are not identified by the identification $\simeq_{!N}$.

Since internal moves are 'invisible' to Opponent, a strategy $\sigma : G$ must be *externally consistent*: If $\boldsymbol{smn}, \boldsymbol{s'm'n'} \in \sigma$, $\lambda_G^{\mathbb{N}}(n) = \lambda_G^{\mathbb{N}}(n') = 0$ and $\mathscr{H}_G^{\omega}(\boldsymbol{sm}) = \mathscr{H}_G^{\omega}(\boldsymbol{s'm'})$, then $n = n'$ and $\mathcal{J}_{\boldsymbol{smn}}^{\ominus\omega}(n) = \mathcal{J}_{\boldsymbol{s'm'n'}}^{\ominus\omega}(n')$. Moreover, external consistency of strategies should hold with respect to identification of positions as well. In fact, we now proceed to establish a stronger property (Theorem 3.62).

Lemma 3.61 (O-determinacy). *Let $\sigma, \tau : G$ such that $\sigma \simeq_G \tau$, and $d \in \mathbb{N} \cup \{\omega\}$.*

(1) *If $\boldsymbol{sm}, \boldsymbol{s'm'} \in P_G$ are d-complete, $\boldsymbol{s}, \boldsymbol{s'} \in \sigma$, and $\mathscr{H}_G^d(\boldsymbol{sm}) = \mathscr{H}_G^d(\boldsymbol{s'm'})$, then $\boldsymbol{sm} = \boldsymbol{s'm'}$;*
(2) *If $\boldsymbol{sm}, \boldsymbol{tl} \in P_G$ are d-complete, $\boldsymbol{s} \in \sigma$, $\boldsymbol{t} \in \tau$, and $\mathscr{H}_G^d(\boldsymbol{sm}) \simeq_{\mathscr{H}^d(G)} \mathscr{H}_G^d(\boldsymbol{tl})$, then $\boldsymbol{sm} \simeq_G \boldsymbol{tl}$.*

Proof. Let us focus on the first clause because the second one can be proved similarly. We proceed by induction on $|\boldsymbol{s}|$. The base case $\boldsymbol{s} = \boldsymbol{\epsilon}$ is trivial: For any $d \in \mathbb{N} \cup \{\omega\}$, if $\mathscr{H}_G^d(\boldsymbol{sm}) = \mathscr{H}_G^d(\boldsymbol{s'm'})$, then $\mathscr{H}_G^d(\boldsymbol{s'm'}) = \mathscr{H}_G^d(\boldsymbol{sm}) = m$, and so $\boldsymbol{s'm'} = m = \boldsymbol{sm}$.

For the induction step, let $d \in \mathbb{N} \cup \{\omega\}$ be fixed and assume $\mathscr{H}_G^d(\boldsymbol{sm}) = \mathscr{H}_G^d(\boldsymbol{s'm'})$. We may suppose that $\boldsymbol{sm} = \boldsymbol{tlrum}$, where l is the rightmost O-move occurring on the left of m in $\boldsymbol{s}$ such that $\lambda_G^{\mathbb{N}}(l) = 0 \vee \lambda_G^{\mathbb{N}}(l) > d$. Then, $\mathscr{H}_G^d(\boldsymbol{s'm'}) = \mathscr{H}_G^d(\boldsymbol{sm}) = \mathscr{H}_G^d(\boldsymbol{t}).l.\mathscr{H}_G^d(\boldsymbol{ru}).m$, and so we may write $\boldsymbol{s'm'} = \boldsymbol{t'_1}.l.\boldsymbol{t'_2}.m$. Now, $\boldsymbol{t}, \boldsymbol{t'_1} \in \sigma$, $\boldsymbol{tl}, \boldsymbol{t'_1}l \in P_G$, $\mathscr{H}_G^d(\boldsymbol{tl}) = \mathscr{H}_G^d(\boldsymbol{t'_1}l)$, and $\boldsymbol{tl}$ and $\boldsymbol{t'}l'$ are both d-complete. Hence, by the induction hypothesis, $\boldsymbol{tl} = \boldsymbol{t'_1}l$. Therefore, $\boldsymbol{t'_2}$ is of the form $r\boldsymbol{t''_2}$ by the determinacy of σ. Thus, $\boldsymbol{sm} = \boldsymbol{tlrm}$ and $\boldsymbol{s'm'} = \boldsymbol{tlrt''_2m}$. Finally, if r is external, then so is m by IE-switch, and so $\boldsymbol{s'm'} = \boldsymbol{sm}$ (n.b., $\boldsymbol{t''_2} = \boldsymbol{\epsilon}$ in this case since otherwise $\mathscr{H}_G^d(\boldsymbol{sm}) \neq \mathscr{H}_G^d(\boldsymbol{s'm'})$ by IE-switch); if r is j-internal ($j > d$), then so is m, and we apply the axiom DP2 for $i = j - 1$ to $\boldsymbol{s}$ and $\boldsymbol{s'}$, concluding $\boldsymbol{sm} = \boldsymbol{s'm'}$. □

Theorem 3.62 (External consistency). *Let $\sigma, \tau : G$ such that $\sigma \simeq_G \tau$, and $d \in \mathbb{N} \cup \{\omega\}$.*

(1) *If $\boldsymbol{smn}, \boldsymbol{s'm'n'} \in \sigma$ are d-complete, and $\mathscr{H}_G^d(\boldsymbol{sm}) = \mathscr{H}_G^d(\boldsymbol{s'm'})$, then $\boldsymbol{smn} = \boldsymbol{s'm'n'}$;*
(2) *If $\boldsymbol{smn} \in \sigma$, $\boldsymbol{tlr} \in \tau$ are d-complete, and $\mathscr{H}_G^d(\boldsymbol{sm}) \simeq_{\mathscr{H}^d(G)} \mathscr{H}_G^d(\boldsymbol{tl})$, then $\boldsymbol{smn} \simeq_G \boldsymbol{tlr}$.*

Proof. Let us first prove the first clause. Let $\sigma : G$ be a strategy, $\boldsymbol{smn}, \boldsymbol{s'm'n'} \in \sigma$ and $d \in \mathbb{N} \cup \{\omega\}$, and assume that $\boldsymbol{smn}, \boldsymbol{s'm'n'}$ are both d-complete and $\mathscr{H}_G^d(\boldsymbol{sm}) = \mathscr{H}_G^d(\boldsymbol{s'm'})$. By the first clause of Lemma 3.61, we have $\boldsymbol{sm} = \boldsymbol{s'm'}$. Therefore, by the axiom S2 on σ, we have $n = n'$ and $\mathcal{J}_{\boldsymbol{smn}}(n) = \mathcal{J}_{\boldsymbol{s'm'n'}}(n')$, whence $\mathcal{J}_{\boldsymbol{smn}}^{\ominus d}(n) = \mathcal{J}_{\boldsymbol{s'm'n'}}^{\ominus d}(n')$.

Similarly, the second clause of the theorem is proved by the second clause of Lemma 3.61. □

Let us also establish the following technical lemma:

Corollary 3.63 (Stepwise identification of strategies). *Any strategies $\sigma, \tau : G$ such that $\sigma \simeq_G \tau$ satisfy $\sigma \simeq_G^d \tau$ for all $d \in \mathbb{N} \cup \{\omega\}$, where*

$$\sigma \simeq_G^d \tau \overset{\text{df.}}{\Leftrightarrow} \forall \boldsymbol{s} \in \sigma, \boldsymbol{t} \in \tau, \boldsymbol{sm}, \boldsymbol{tl} \in P_G.\, \boldsymbol{sm} \simeq_G^d \boldsymbol{tl} \Rightarrow (\forall \boldsymbol{smn} \in \sigma.\, \exists \boldsymbol{tlr} \in \tau.\, \boldsymbol{smn} \simeq_G^d \boldsymbol{tlr})$$
$$\wedge (\forall \boldsymbol{tlr} \in \tau.\, \exists \boldsymbol{smn} \in \sigma.\, \boldsymbol{tlr} \simeq_G^d \boldsymbol{smn}).$$

Proof. Immediate from Theorem 3.62. □

Hence, given strategies $\sigma, \tau : G$, we have

$$\sigma \simeq_G \tau \Leftrightarrow \forall d \in \mathbb{N} \cup \{\omega\}. \sigma \simeq_G^d \tau,$$

which will be useful later in the present article.

Let us proceed to show that the relation $\simeq_G$ on strategies on any game G is a PER.

Lemma 3.64 (PER lemma). *Given $\sigma, \tau : G$ such that $\sigma \simeq_G \tau$, we have*

$$(\forall \boldsymbol{s} \in \sigma . \exists \boldsymbol{t} \in \tau . \boldsymbol{s} \simeq_G \boldsymbol{t}) \wedge (\forall \boldsymbol{t} \in \tau . \exists \boldsymbol{s} \in \sigma . \boldsymbol{t} \simeq_G \boldsymbol{s}).$$

Proof. By symmetry, it suffices to show $\forall \boldsymbol{s} \in \sigma . \exists \boldsymbol{t} \in \tau . \boldsymbol{s} \simeq_G \boldsymbol{t}$. We prove it by induction on $|\boldsymbol{s}|$. The base case is trivial; for the inductive step, let $\boldsymbol{s}mn \in \sigma$. By the induction hypothesis, there is some $\boldsymbol{t} \in \tau$ such that $\boldsymbol{s} \simeq_G \boldsymbol{t}$. Then, by DI3 on $\simeq_G$, there is some $\boldsymbol{t}l \in \tau$ such that $\boldsymbol{s}m \simeq_G \boldsymbol{t}l$. Finally, since $\sigma \simeq_G \tau$, there is some $\boldsymbol{t}lr \in \tau$ such that $\boldsymbol{s}mn \simeq_G \boldsymbol{t}lr$, completing the proof. □

Proposition 3.65 (PERs on strategies). *Given a game G, the identification $\simeq_G$ of strategies on G is a PER, i.e., a symmetric, transitive relation.*

Proof. We just show the transitivity as the symmetry is obvious. Let $\sigma, \tau, \mu : G$ such that $\sigma \simeq_G \tau$ and $\tau \simeq_G \mu$. Assume that $\boldsymbol{s}mn \in \sigma$, $\boldsymbol{u} \in \mu$ and $\boldsymbol{s}m \simeq_G \boldsymbol{u}p$. By Lemma 3.64, there is some $\boldsymbol{t} \in \tau$ such that $\boldsymbol{s} \simeq_G \boldsymbol{t}$. By DI3 on $\simeq_G$, there is some $\boldsymbol{t}l \in P_G$ such that $\boldsymbol{s}m \simeq_G \boldsymbol{t}l$, whence $\boldsymbol{t}l \simeq_G \boldsymbol{u}p$. Also, since $\sigma \simeq_G \tau$, there is some $\boldsymbol{t}lr \in \tau$ such that $\boldsymbol{s}mn \simeq_G \boldsymbol{t}lr$. Finally, since $\tau \simeq_G \mu$, there is some $\boldsymbol{u}pq \in \mu$ such that $\boldsymbol{t}lr \simeq_G \boldsymbol{u}pq$, whence $\boldsymbol{s}mn \simeq_G \boldsymbol{u}pq$, completing the proof. □

Hence, given a game G, we may take the equivalence classes $[\sigma] := \{\tau : G \mid \sigma \simeq_G \tau\}$ of valid strategies $\sigma : G$. These equivalence classes, rather than strategies themselves, have interpreted proofs and programmes in Abramsky et al. (2000), McCusker (1998).

At this point, let us remark that even-length positions are not necessarily preserved under the hiding operation on j-sequences (Definition 3.10). For instance, let $\boldsymbol{s}m\boldsymbol{n}t$ be an even-length position of a game G such that $\boldsymbol{s}m$ (resp. $\boldsymbol{n}t$) consists of external (resp. internal) moves only. By IE-switch on G, m is an O-move, and so $\mathcal{H}_G^\omega(\boldsymbol{s}m\boldsymbol{n}t) = \boldsymbol{s}m$ is of odd-length.

Taking into account this fact, we introduce

Definition 3.66 (Hiding operation on strategies). *Let G be a game, and $d \in \mathbb{N} \cup \{\omega\}$. Given $\boldsymbol{s} \in P_G$, we define*

$$\boldsymbol{s} \natural \mathcal{H}_G^d := \begin{cases} \mathcal{H}_G^d(\boldsymbol{s}) & \text{if } \boldsymbol{s} \text{ is } d\text{-complete (Definition 3.1);} \\ \boldsymbol{t} & \text{otherwise, where } \mathcal{H}_G^d(\boldsymbol{s}) = \boldsymbol{t}m. \end{cases}$$

*The d-*hiding operation $\mathcal{H}^d$ (on strategies) *is then given by*

$$\mathcal{H}^d : (\sigma : G) \mapsto \{\boldsymbol{s} \natural \mathcal{H}_G^d \mid \boldsymbol{s} \in \sigma\}.$$

Let us then proceed to establish a beautiful fact: If $\sigma : G$, then $\mathcal{H}^d(\sigma) : \mathcal{H}^d(G)$ for all $d \in \mathbb{N} \cup \{\omega\}$. For this task, we need the following lemma:

Lemma 3.67 (Asymmetry lemma). *Let $\sigma : G$ be a strategy, and $d \in \mathbb{N} \cup \{\omega\}$. Assume $\boldsymbol{s}mn \in \mathcal{H}^d(\sigma)$, where $\boldsymbol{s}mn = \boldsymbol{t}mu\boldsymbol{n}v \natural \mathcal{H}_G^d$ with $\boldsymbol{t}mu\boldsymbol{n}v \in \sigma$ not d-complete. Then, we have $\boldsymbol{s}mn = \mathcal{H}^d(\boldsymbol{t}mu\boldsymbol{n}) = \mathcal{H}^d(\boldsymbol{t}).mn$.*

Proof. Since $\boldsymbol{tmunv} \in \sigma$ is not d-complete, we may write $\boldsymbol{v} = \boldsymbol{v}_1 l \boldsymbol{v}_2 r$ with $\lambda_G^{\mathbb{N}}(l) = 0 \vee \lambda_G^{\mathbb{N}}(l) > d$, $0 < \lambda_G^{\mathbb{N}}(r) \leqslant d$ and $0 < \lambda_G^{\mathbb{N}}(x) \leqslant d$ for all moves x in $\boldsymbol{v}_1$ or $\boldsymbol{v}_2$. Then, we have

$$\boldsymbol{smn} = \boldsymbol{tmunv}_1 l \boldsymbol{v}_2 r \natural \mathscr{H}_G^d = \mathscr{H}_G^d(\boldsymbol{t}) m \mathscr{H}_G^d(\boldsymbol{u}) n = \mathscr{H}_G^d(\boldsymbol{t}) mn,$$

which completes the proof. □

We are now ready to establish

Theorem 3.68 (Hiding theorem). *If $\sigma : G$, then $\mathscr{H}^d(\sigma) : \mathscr{H}^d(G)$ for all $d \in \mathbb{N} \cup \{\omega\}$.*

Proof. We first show $\mathscr{H}^d(\sigma) \subseteq P_{\mathscr{H}^d(G)}^{\mathsf{Even}}$. Let $\boldsymbol{s} \in \mathscr{H}^d(\sigma)$, i.e., $\boldsymbol{s} = \boldsymbol{t} \natural \mathscr{H}_G^d$ for some $\boldsymbol{t} \in \sigma$. Let us write $\boldsymbol{t} = \boldsymbol{t}' m$ as the case $\boldsymbol{t} = \boldsymbol{\varepsilon}$ is trivial.

- If $\boldsymbol{t}$ is d-complete, then $\boldsymbol{s} = \boldsymbol{t} \natural \mathscr{H}_G^d = \mathscr{H}_G^d(\boldsymbol{t}) \in P_{\mathscr{H}^d(G)}$. Also, since $\boldsymbol{s} = \mathscr{H}_G^d(\boldsymbol{t}') m$ and m is a P-move, $\boldsymbol{s}$ must be of even length by alternation on $\mathscr{H}^d(G)$.
- If $\boldsymbol{t}$ is not d-complete, then we may write $\boldsymbol{t} = \boldsymbol{t}'' m_0 m_1 \dots m_k$, where $m_k = m$, $\boldsymbol{t}'' m_0$ is d-complete, and $0 < \lambda_G^{\mathbb{N}}(m_i) \leqslant d$ for $i = 1, 2, \dots, k$. By IE-switch, m_0 is an O-move, and thus $\boldsymbol{s} = \mathscr{H}_G^d(\boldsymbol{t}'') \in P_{\mathscr{H}^d(G)}$ is of even length.

It remains to verify S1 and S2. For S1, $\mathscr{H}^d(\sigma)$ is nonempty since $\boldsymbol{\varepsilon} \in \mathscr{H}^d(\sigma)$. For the even-prefix-closure, let $\boldsymbol{smn} \in \mathscr{H}^d(\sigma)$; we have to show $\boldsymbol{s} \in \mathscr{H}^d(\sigma)$. We have some $\boldsymbol{tmunv} \in \sigma$ such that $\boldsymbol{tmunv} \natural \mathscr{H}_G^d = \boldsymbol{smn}$. By Lemma 3.67, $\boldsymbol{smn} = \mathscr{H}_G^d(\boldsymbol{t}) mn$, whence $\boldsymbol{s} = \mathscr{H}_G^d(\boldsymbol{t})$. Since $\boldsymbol{tm}$ is d-complete, so is $\boldsymbol{t}$ by IE-switch. Therefore, $\boldsymbol{s} = \mathscr{H}_G^d(\boldsymbol{t}) = \boldsymbol{t} \natural \mathscr{H}_G^d \in \mathscr{H}^d(\sigma)$.

Finally for S2, let $\boldsymbol{smn}, \boldsymbol{smn}' \in \mathscr{H}^d(\sigma)$; we have to show $n = n'$ and $\mathcal{J}_{\boldsymbol{sm}}^{\ominus d}(n) = \mathcal{J}_{\boldsymbol{sm}}^{\ominus d}(n')$. Clearly, $\boldsymbol{smn} = \boldsymbol{tmunv} \natural \mathscr{H}_G^d$, $\boldsymbol{smn}' = \boldsymbol{t}' m \boldsymbol{u}' n' \boldsymbol{v}' \natural \mathscr{H}_G^d$ for some $\boldsymbol{tmunv}, \boldsymbol{t}' m \boldsymbol{u}' n' \boldsymbol{v}' \in \sigma$. Then, by Lemma 3.67, $\boldsymbol{smn} = \mathscr{H}_G^d(\boldsymbol{tmu}) n$ and $\boldsymbol{smn}' = \mathscr{H}_G^d(\boldsymbol{t}' m \boldsymbol{u}') n'$. Therefore, by Theorem 3.62, $n = n'$ and $\mathcal{J}_{\boldsymbol{smn}}^{\ominus d}(n) = \mathcal{J}_{\boldsymbol{smn}'}^{\ominus d}(n')$, completing the proof. □

Next, let us review standard constraints on strategies. Recall that one of the highlights of *HO-games* (Hyland and Ong, 2000) is to establish a one-to-one correspondence between PCF Böhm trees and *innocent*, *well-bracketed* static strategies (on static games modelling types of PCF). That is, the two constraints narrow down the hom-sets of the codomain of the interpretation functor, i.e., the category of HO-games, so that the interpretation becomes *full*. Roughly, a strategy is innocent if its computation depends only on P-views, and well bracketed if every 'question-answering' in P-views by the strategy is achieved in the 'last-question-first-answered' fashion. Formally:

Definition 3.69 (Innocence of strategies (Hyland and Ong, 2000)). *A strategy $\sigma : G$ is* innocent *if $\forall \boldsymbol{smn}, \boldsymbol{t} \in \sigma, \boldsymbol{tm} \in P_G.\, \lceil \boldsymbol{tm} \rceil = \lceil \boldsymbol{sm} \rceil \Rightarrow \boldsymbol{tmn} \in \sigma \wedge \lceil \boldsymbol{tmn} \rceil = \lceil \boldsymbol{smn} \rceil$.*

Definition 3.70 (Well bracketing of strategies (Hyland and Ong, 2000)). *A strategy $\sigma : G$ is* well bracketed (wb) *if, given $\boldsymbol{sqta} \in \sigma$, $\lambda_G^{\mathsf{QA}}(q) = \mathsf{Q}$, $\lambda_G^{\mathsf{QA}}(a) = \mathsf{A}$ and $\mathcal{J}_{\boldsymbol{sqta}}(a) = q$, each occurrence of a question in $\boldsymbol{t}'$, defined by $\lceil \boldsymbol{sqt} \rceil_G = \lceil \boldsymbol{sq} \rceil_G . \boldsymbol{t}'$, justifies an occurrence of an answer in $\boldsymbol{t}'$.*

Next, recall that a programming language is *total* if its computation always terminates in a finite period of time. This programming concept is interpreted in game semantics by *totality* of strategies in a sense similar to totality of partial functions:

Definition 3.71 (Totality of strategies (Abramsky et al. 1997)). *A strategy $\sigma : G$ is* total *if it satisfies $\forall \boldsymbol{s} \in \sigma, \boldsymbol{sm} \in P_G.\, \exists \boldsymbol{smn} \in \sigma$.*

Nevertheless, it is well known that totality of strategies is *not* preserved under composition due to the problem of 'infinite chattering' (Abramsky et al., 1997; Clairambault and Harmer, 2010). For this point, one usually imposes a condition on strategies stronger than totality, e.g., *winning* (Abramsky et al., 1997), that is preserved under composition. We may certainly just apply the winning condition of Abramsky et al. (1997), but it requires an additional structure on games, which may be criticised as *extrinsic* and/or *ad hoc*; thus, we prefer another, simpler solution. A natural idea is then to require that strategies should not contain any strictly increasing (with respect to $\preceq$) infinite sequence of positions. However, we have to relax this constraint: The dereliction der_G (Definition 3.89), the β-identity on a game G in the game-semantic CCBoC given in Section 4, may not satisfy it, e.g., when G is an implication $!A \multimap B$ or a β-exponential B^A.

Therefore, instead, we apply the same idea to *P-views*, arriving at

Definition 3.72 (Noetherianity of strategies (Clairambault and Harmer, 2010) strategy). *A strategy $\sigma : G$ is* noetherian *if it does not contain any strictly increasing infinite sequence of P-views in G.*

It has been shown in Clairambault and Harmer (2010) that total, noetherian, innocent static strategies are closed under composition.

Now, let us show that the standard constraints on strategies except totality are all preserved under the hiding operation, which implies that dynamic strategies are a reasonable generalisation of static strategies in a certain sense.

Corollary 3.73 (Preservation of constraints on strategies under hiding). *If a strategy $\sigma : G$ is valid, innocent, wb or noetherian, then so is $\mathscr{H}^d(\sigma) : \mathscr{H}^d(G)$, and if another $\tau : G$ satisfies $\sigma \simeq_G \tau$, then $\mathscr{H}^d(\sigma) \simeq_{\mathscr{H}^d(G)} \mathscr{H}^d(\tau)$, for all $d \in \mathbb{N} \cup \{\omega\}$.*

Proof. Let $d \in \mathbb{N} \cup \{\omega\}$ be arbitrarily fixed. We have $\mathscr{H}^d(\sigma) : \mathscr{H}^d(G)$ by Theorem 3.68.

- Preservation of validity is by Lemma 3.61, Corollary 3.63 and the axiom DI3 on $\simeq_G$;
- Preservation of innocence and noetherianity holds since $\lceil \mathscr{H}^d_G(sm) \rceil_{\mathscr{H}^d(G)}$ is a j-subsequence in $\mathscr{H}^d_G(\lceil sm \rceil_G)$ for any $sm \in P_G^{\mathsf{Odd}}$;
- Well bracketing is preserved under the d-hiding operation $\mathscr{H}^d$ because *both* of the question and the answer of each 'QA-pair' are either deleted or retained.

Finally, preservation of identification of strategies is proved similarly to that of validity. □

Totality of strategies is, however, *not* preserved under the d-hiding operation $\mathscr{H}^d$ on strategies for some $d \in \mathbb{N} \cup \{\omega\}$. For instance, consider any total strategy that always performs a 1-internal P-move, which is no longer total when the 1-hiding operation $\mathscr{H}^1$ is applied. Clearly, even the conjunction of totality and noetherianity of strategies is not preserved under $\mathscr{H}^1$ for the same reason (n.b., it is preserved only if we take the *composition* of total, noetherian, innocent strategies). That is, recalling that composition coincides with *concatenation plus hiding* as sketched in the introduction, the conjunction of totality and noetherianity is stable only under this particular type of hiding, and *not under hiding in general.*

For this problem, let us introduce a new constraint on strategies:

Definition 3.74 (Strong totality of strategies). *A strategy $\sigma : G$ is* strongly total *if the strategy $\mathscr{H}^d(\sigma) : \mathscr{H}^d(G)$ is total for all $d \in \mathbb{N} \cup \{\omega\}$.*

Strong totality strengthens totality in the evident sense, and simply by the definition it is preserved under the d-hiding operation $\mathscr{H}^d$ for all $d \in \mathbb{N} \cup \{\omega\}$ (see Lemma 3.79).

We then introduce our notion of *winning* strategies, which is preserved under hiding:

Definition 3.75 (Winning of strategies). *A strategy is* winning *if it is strongly total, innocent and noetherian.*

Corollary 3.76 (Closure of winning strategies under hiding). *If a strategy $\sigma : G$ is winning, then so is the strategy $\mathscr{H}^d(\sigma) : \mathscr{H}^d(G)$ for all $d \in \mathbb{N} \cup \{\omega\}$.*

Proof. Immediate from Corollary 3.73 (and Lemma 3.79 below). □

Conceptually, winning strategies in the sense of Definition 3.75 can be regarded as 'strategies for proofs' as follows. First, a proof should not get 'stuck,' and thus 'strategies for proofs' must be (strongly) total. In addition, since logic is concerned with the truth of formulas, which are invariant to 'passage of time,' proofs should not depended on 'states of arguments.' Hence, it makes sense to impose innocence on 'strategies for proofs.' Finally, noetherianity is imposed because if a play by an innocent, noetherian strategy keeps growing infinitely, then it cannot be Player's 'intention,' and so it should result in a 'win' for Player.

Technically, we need winning and wb on strategies because they play an essential role for our *full completeness* result (Corollary 4.7).

At the end of the present section, we establish an inductive property of the d-hiding operation on strategies for each $d \in \mathbb{N} \cup \{\omega\}$:

Notation 3.77. *If $\sigma : G$ and $d \in \mathbb{N} \cup \{\omega\}$, then $\sigma^d_{\downarrow} := \{ \boldsymbol{s} \in \sigma \mid \boldsymbol{s}$ is d-complete $\}$ and $\sigma^d_{\uparrow} := \sigma \setminus \sigma^d_{\downarrow}$.*

Lemma 3.78 (Hiding and complete positions). *Let $\sigma : G$. Given $i, d \in \mathbb{N}$ such that $i \geqslant d$, we have $\mathscr{H}^i(\sigma) = \mathscr{H}^i(\sigma^d_{\downarrow}) := \{ \boldsymbol{s} \natural \mathscr{H}^i_G \mid \boldsymbol{s} \in \sigma^d_{\downarrow} \}$.*

Proof. The inclusion $\mathscr{H}^i(\sigma^d_{\downarrow}) \subseteq \mathscr{H}^i(\sigma)$ is obvious. For the opposite inclusion, let $\boldsymbol{s} \in \mathscr{H}^i(\sigma)$, i.e., $\boldsymbol{s} = \boldsymbol{t} \natural \mathscr{H}^i_G$ for some $\boldsymbol{t} \in \sigma$; we have to show $\boldsymbol{s} \in \mathscr{H}^i(\sigma^d_{\downarrow})$. If $\boldsymbol{t} \in \sigma^d_{\downarrow}$, then we are done; thus, assume otherwise. If there is no external or j-internal move with $j > i$ other than the first move m_0 in $\boldsymbol{t}$, then $\boldsymbol{s} = \boldsymbol{\varepsilon} \in \mathscr{H}^i(\sigma^d_{\downarrow})$; so assume otherwise. As a result, we may write $\boldsymbol{t} = m_0 \boldsymbol{t_1} m n \boldsymbol{t_2} r$, where $\boldsymbol{t_2} r$ consists only of j-internal moves with $0 < j \leqslant i$, and m and n are P- and O-moves, respectively, such that $\lambda^{\mathbb{N}}_G(m) = \lambda^{\mathbb{N}}_G(n) = 0 \vee \lambda^{\mathbb{N}}_G(m) = \lambda^{\mathbb{N}}_G(n) > i$. Take $m_0 \boldsymbol{t_1} m \in \sigma^d_{\downarrow}$ such that $m_0 \boldsymbol{t_1} m \natural \mathscr{H}^i_G = m_0 \mathscr{H}^i_G(\boldsymbol{t_1}) m = \boldsymbol{t} \natural \mathscr{H}^i_G = \boldsymbol{s}$, whence $\boldsymbol{s} \in \mathscr{H}^i(\sigma^d_{\downarrow})$. □

We are now ready to show

Lemma 3.79 (Stepwise hiding on strategies). *Given $\sigma : G$, $\mathscr{H}^{i+1}(\sigma) = \mathscr{H}^1(\mathscr{H}^i(\sigma))$ for all $i \in \mathbb{N}$.*

Proof. We first show the inclusion $\mathscr{H}^{i+1}(\sigma) \subseteq \mathscr{H}^1(\mathscr{H}^i(\sigma))$. By Lemma 3.78, we may write any element of the set $\mathscr{H}^{i+1}(\sigma)$ as $\boldsymbol{s} \natural \mathscr{H}^{i+1}_G$ for some $\boldsymbol{s} \in \sigma^{i+1}_{\downarrow}$. Then, observe that

$$\boldsymbol{s} \natural \mathscr{H}^{i+1}_G = \mathscr{H}^{i+1}_G(\boldsymbol{s}) = \mathscr{H}_{\mathscr{H}^i(G)}(\mathscr{H}^i_G(\boldsymbol{s})) = (\boldsymbol{s} \natural \mathscr{H}^i_G) \natural \mathscr{H}^1_{\mathscr{H}^i(G)} \in \mathscr{H}^1(\mathscr{H}^i(\sigma)).$$

For the opposite inclusion $\mathscr{H}^1(\mathscr{H}^i(\sigma)) \subseteq \mathscr{H}^{i+1}(\sigma)$, again by Lemma 3.78, we may write any element of $\mathscr{H}^1(\mathscr{H}^i(\sigma))$ as $(\boldsymbol{s} \natural \mathscr{H}^i_G) \natural \mathscr{H}^1_{\mathscr{H}^i(G)}$ for some $\boldsymbol{s} \in \sigma^i_{\downarrow}$. We have to show that $(\boldsymbol{s} \natural \mathscr{H}^i_G) \natural \mathscr{H}^1_{\mathscr{H}^i(G)} \in \mathscr{H}^{i+1}(\sigma)$. If $\boldsymbol{s} \in \sigma^{i+1}_{\downarrow}$, then it is completely analogous to the above argument; so assume otherwise. Also, if an external or j-internal move with $j > i+1$ in $\boldsymbol{s}$ is only the first move m_0, then $(\boldsymbol{s} \natural \mathscr{H}^i_G) \natural \mathscr{H}^1_{\mathscr{H}^i(G)} = \boldsymbol{\varepsilon} \in \mathscr{H}^{i+1}(\sigma)$; thus assume otherwise. Now, we may write

$$\boldsymbol{s} = \boldsymbol{s}' m n m_1 m_2 \dots m_{2k} r,$$

where $\lambda_G^{\mathbb{N}}(r) = i+1$, $m_1, m_2, \ldots, m_{2k}$ are j-internal with $0 < j \leqslant i+1$, and m and n are external or j-internal P- and O-moves with $j > i+1$, respectively. Then, we calculate

$$\begin{aligned}(\boldsymbol{s} \natural \mathscr{H}_G^i) \natural \mathscr{H}^1_{\mathscr{H}^i(G)} &= \mathscr{H}_G^i(\boldsymbol{s}) \natural \mathscr{H}^1_{\mathscr{H}^i(G)} \\ &= \mathscr{H}_{\mathscr{H}^i(G)}(\mathscr{H}_G^i(\boldsymbol{s}')).m \\ &= \mathscr{H}_G^{i+1}(\boldsymbol{s}').m \quad \text{(by Lemma 3.15)} \\ &= \boldsymbol{s} \natural \mathscr{H}_G^{i+1} \in \mathscr{H}^{i+1}(\sigma),\end{aligned}$$

which completes the proof. □

Consequently, as in the case of games, we may focus on the operation $\mathscr{H}^1$:

Convention. Henceforth, we write $\mathscr{H}$ for $\mathscr{H}^1$ and call it the *hiding operation (on strategies)*; $\mathscr{H}^i$ denotes the i-times iteration of $\mathscr{H}$ for all $i \in \mathbb{N}$.

3.5 Constructions on dynamic strategies

Next, let us recall standard constructions on strategies (Abramsky and McCusker, 1999). Note that since (dynamic) strategies are simply 'static strategies on (dynamic) games,' they are clearly closed under all the constructions on static strategies.

Nevertheless, the CCBoC of games and strategies given in Section 4 has normalised games as 0-cells and strategies $\phi : G$ such that $\mathscr{H}^\omega(G) \trianglelefteq A \Rightarrow B$ as 1-cells $A \to B$, and therefore we need to generalise *pairing* and *promotion* of static strategies. In fact, we have generalised product and exponential of static games, respectively, to pairing and promotion of dynamic games for this purpose. Also, we shall decompose and generalise composition of static strategies into *concatenation plus hiding* of dynamic strategies, for which we have introduced concatenation of dynamic games.

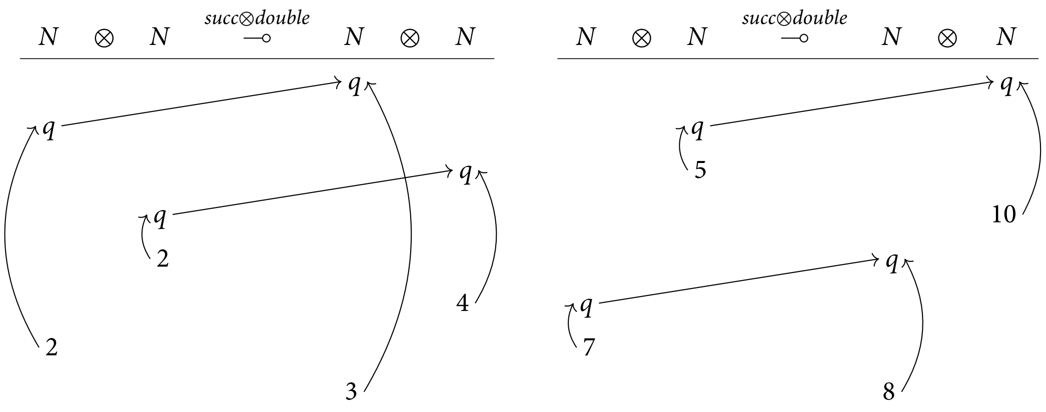
We postpone verifying the preservation of winning under these constructions on dynamic strategies (Corollary 3.103) to the latter half of this section as it needs Lemma 3.102.

We begin with *tensor* $\otimes$ on strategies. Roughly, the tensor $\phi \otimes \psi : A \otimes B \multimap C \otimes D$ of strategies $\phi : A \multimap C$ and $\psi : B \multimap D$ plays by ϕ if the last O-move is of A or C, and by ψ otherwise. Formally:

Definition 3.80 (Tensor of strategies (Abramsky and McCusker, 1999)). *Given games A, B, C and D, and strategies $\phi : A \multimap C$ and $\psi : B \multimap D$, the* tensor (product) *$\phi \otimes \psi$ of ϕ and ψ is given by*

$$\phi \otimes \psi := \{\, \boldsymbol{s} \in \mathscr{L}_{A \otimes B \multimap C \otimes D} \mid \boldsymbol{s} \upharpoonright A, C \in \phi, \boldsymbol{s} \upharpoonright B, D \in \psi \,\}.$$

Example 3.81. The tensor $succ \otimes double : N \otimes N \multimap N \otimes N$, where $succ, double : N \multimap N$ are given in Section 1, plays, e.g., as follows:



Lemma 3.82 (Well-defined tensor of strategies). *Given games A, B, C and D, and strategies $\phi : A \multimap C$ and $\psi : B \multimap D$, the tensor $\phi \otimes \psi$ is a strategy on the game $A \otimes B \multimap C \otimes D$. If ϕ and ψ are innocent (resp. wb, total, noetherian), then so is $\phi \otimes \psi$. Given $\phi' : A \multimap C$ and $\psi' : B \multimap D$ with $\phi \simeq_{A \multimap C} \phi'$ and $\psi \simeq_{B \multimap D} \psi'$, we have $\phi \otimes \psi \simeq_{A \otimes B \multimap C \otimes D} \phi' \otimes \psi'$.*

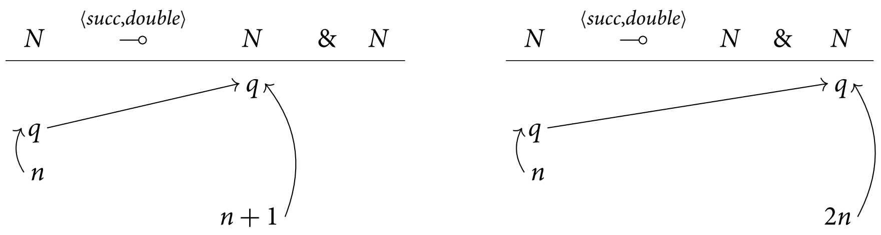
Proof. Straightforward; see McCusker (1998), Abramsky et al. (2000). □

We next recall *pairing* of strategies. Intuitively, the pairing $\langle \phi, \psi \rangle : C \multimap A \& B$ of strategies $\phi : C \multimap A$ and $\psi : C \multimap B$ plays by ϕ if the play is in $C \multimap A$, and by ψ otherwise. Formally:

Definition 3.83 (Pairing of strategies Abramsky and McCusker, 1999). *Given games A, B and C, and strategies $\phi : C \multimap A$ and $\psi : C \multimap B$, the pairing $\langle \phi, \psi \rangle$ of ϕ and ψ is defined by*

$$\langle \phi, \psi \rangle := \{ \boldsymbol{s} \in \mathscr{L}_{C \multimap A \& B} \mid (\boldsymbol{s} \upharpoonright C, A \in \phi \wedge \boldsymbol{s} \upharpoonright B = \boldsymbol{\varepsilon}) \vee (\boldsymbol{s} \upharpoonright C, B \in \psi \wedge \boldsymbol{s} \upharpoonright A = \boldsymbol{\varepsilon}) \}.$$

Example 3.84. The pairing $\langle succ, double \rangle : N \multimap N \& N$ plays as either of the following diagrams:


$N \overset{\langle succ,double \rangle}{\multimap} N \& N$
q
q
n
$n+1$
$N \overset{\langle succ,double \rangle}{\multimap} N \& N$
q
q
n
$2n$

where $n \in \mathbb{N}$, depending on the first O-move.

Lemma 3.85 (Well-defined pairing of strategies). *Given games A, B and C, and strategies $\phi : C \multimap A$ and $\psi : C \multimap B$, the pairing $\langle \phi, \psi \rangle$ is a strategy on the game $C \multimap A \& B$. If ϕ and ψ are innocent (resp. wb, total, noetherian), then so is $\langle \phi, \psi \rangle$. Given $\phi' : C \multimap A$ and $\psi' : C \multimap B$ with $\phi \simeq_{C \multimap A} \phi'$ and $\psi \simeq_{C \multimap B} \psi'$, we have $\langle \phi, \psi \rangle \simeq_{C \multimap A \& B} \langle \phi', \psi' \rangle$.*

Proof. Straightforward; see McCusker (1998), Abramsky et al. (2000). □

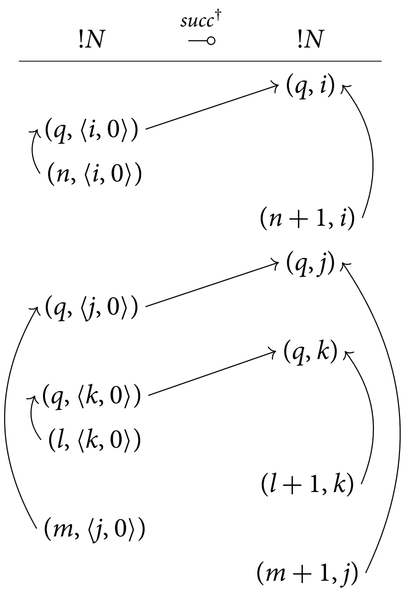
Next, let us recall *promotion* of strategies. Intuitively, the promotion $\varphi^\dagger : !A \multimap !B$ of a strategy $\varphi : A \Rightarrow B$ plays, during a play $\boldsymbol{s}$ in $!A \multimap !B$, as φ for each j-subsequence $\boldsymbol{s} \upharpoonright i$, called a *thread* (Abramsky and McCusker, 1999; McCusker, 1998). Formally:

Definition 3.86 (Promotion of strategies McCusker, 1998). *Given games A and B, and a strategy $\varphi : !A \multimap B$, the* promotion *$\varphi^\dagger$ of φ is defined by*

$$\varphi^\dagger := \{ \boldsymbol{s} \in \mathscr{L}_{!A \multimap !B} \mid \forall i \in \mathbb{N}. \boldsymbol{s} \upharpoonright i \in \varphi \}.$$

We could have defined noetherianity of strategies in terms of positions, but then it would not be preserved under promotion. It is another reason why we have defined it in terms of P-views (Definition 3.72).

Example 3.87. Let $succ : N \Rightarrow N$ be the successor strategy (n.b., it is on the implication $\Rightarrow$, not the linear implication $\multimap$), which specifically selects, say, the 'tag' $(\_, 0)$ in the domain $!N$. Then, the promotion $succ^\dagger : !N \multimap !N$ plays, e.g., as follows:



where $i, j, k, n, m, l \in \mathbb{N}$ such that $i \neq j$, $i \neq k$ and $j \neq k$, and they are all selected by Opponent. Note that $succ^{\dagger}$ consistently plays as $succ$ for each thread.

Lemma 3.88 (Well defined promotion of strategies). *Given games A and B, and a strategy $\varphi : !A \multimap B$, the promotion $\varphi^{\dagger}$ is a strategy on the game $!A \multimap !B$. If φ is innocent (resp. wb, total, noetherian), then so is $\varphi^{\dagger}$. Given $\tilde{\varphi} : !A \multimap B$ with $\varphi \simeq_{!A \multimap B} \tilde{\varphi}$, we have $\varphi^{\dagger} \simeq_{!A \multimap !B} \tilde{\varphi}^{\dagger}$.*

Proof. Straightforward; see McCusker (1998), Abramsky et al. (2000). □

We proceed to recall a class of simple strategies, which are β-identities in our game-semantic CCBoC introduced in Section 4:

Definition 3.89 (Derelictions Abramsky et al. 2000; McCusker, 1998). *The* dereliction der_A : $!A \multimap A$ *on a normalised game A is defined by*

$$der_A := \{ \boldsymbol{s} \in P^{\mathsf{Even}}_{!A \multimap A} \mid \forall \boldsymbol{t} \preceq \boldsymbol{s}.\, \mathsf{Even}(\boldsymbol{t}) \Rightarrow (\boldsymbol{t} \upharpoonright !A) \upharpoonright 0 = \boldsymbol{t} \upharpoonright A \}.$$

Note that any ‘tag’ $(\_, i)$ such that $i \in \mathbb{N}$ would work; our choice $(\_, 0)$ does not matter.

Lemma 3.90 (Well defined derelictions). *The dereliction der_A on a normalised game A is a valid, innocent, wb, strongly total strategy on the game $!A \multimap A$. It is noetherian if A is well founded.*

Proof. We just show that der_A is noetherian if A is well founded as the other points are trivial, e.g., validity of der_A is immediate from the definition of $\simeq_{!A \multimap A}$. Given $\boldsymbol{smn} \in der_A$, it is easy to see by induction on $|\boldsymbol{s}|$ that the P-view $\lceil \boldsymbol{sm} \rceil$ is of the form $m_1 m_1 m_2 m_2 \ldots m_k m_k m$, and thus there is a sequence $\star \vdash_A m_1 \vdash_A m_2 \cdots \vdash_A m_k \vdash_A m$ of enabling pairs. Therefore, if A is well founded, then der_A must be noetherian. □

Let us proceed to introduce some generalisations of existing constructions. Note that tensor $\otimes$, pairing $\langle \_, \_ \rangle$ and promotion $(\_)^{\dagger}$ of static strategies have been already generalised slightly because they allow non-normalised dynamic games and strategies. Nevertheless, for the game-semantic CCBoC introduced in Section 4, we need their further generalisations:

Definition 3.91 (Generalised pairing of strategies). *Given strategies $\phi : L$ and $\psi : R$ such that $\mathscr{H}^\omega(L) \trianglelefteq C \multimap A$ and $\mathscr{H}^\omega(R) \trianglelefteq C \multimap B$ for some normalised games A, B and C, the* (generalised) pairing $\langle \phi, \psi \rangle$ *of ϕ and ψ is defined by*

$$\langle \phi, \psi \rangle := \{ \boldsymbol{s} \in \mathscr{L}_{\langle L,R \rangle} \mid (\boldsymbol{s} \upharpoonright L \in \phi \wedge \boldsymbol{s} \upharpoonright R = \boldsymbol{\varepsilon}) \vee (\boldsymbol{s} \upharpoonright R \in \psi \wedge \boldsymbol{s} \upharpoonright L = \boldsymbol{\varepsilon}) \}.$$

Theorem 3.92 (Well defined generalised pairing of strategies). *Given strategies $\phi : L$ and ψ : R such that $\mathscr{H}^\omega(L) \trianglelefteq C \multimap A$ and $\mathscr{H}^\omega(R) \trianglelefteq C \multimap B$ for some normalised games A, B and C, the generalised pairing $\langle \phi, \psi \rangle$ is a strategy on the game $\langle L, R \rangle$. If ϕ and ψ are innocent (resp. wb, total, noetherian), then so is $\langle \phi, \psi \rangle$. Given $\phi' : L$ and $\psi' : R$ such that $\phi \simeq_L \phi'$ and $\psi \simeq_R \psi'$, we have $\langle \phi, \psi \rangle \simeq_{\langle L,R \rangle} \langle \phi', \psi' \rangle$.*

Proof. Straightforward. □

Convention. Henceforth, *pairing of strategies* refers to the generalised one.

Definition 3.93 (Generalised promotion of strategies). *Given a strategy $\varphi : G$ such that $\mathscr{H}^\omega(G) \trianglelefteq\, !A \multimap B$ for some normalised games A and B, the* (generalised) promotion $\varphi^\dagger$ *of φ is defined by*

$$\varphi^\dagger := \{ \boldsymbol{s} \in \mathscr{L}_{G^\dagger} \mid \forall i \in \mathbb{N}.\, \boldsymbol{s} \upharpoonright i \in \varphi \}.$$

Theorem 3.94 (Well defined generalised promotion on strategies). *Given a strategy $\varphi : G$ such that $\mathscr{H}^\omega(G) \trianglelefteq\, !A \multimap B$ for some normalised games A and B, the generalised promotion $\varphi^\dagger$ is a strategy on the game $G^\dagger$. If φ is innocent (resp. wb, total, noetherian), then so is $\varphi^\dagger$. Given $\tilde{\varphi} : G$ such that $\varphi \simeq_G \tilde{\varphi}$, we have $\varphi^\dagger \simeq_{G^\dagger} \tilde{\varphi}^\dagger$.*

Proof. Straightforward. □

Convention. Henceforth, *promotion of strategies* refers to the generalised one.

Next, let us introduce a new construction on strategies, which plays a fundamental role in the present work:

Definition 3.95 (Concatenation of strategies). *Let $\iota : J$ and $\kappa : K$ be strategies such that $\mathscr{H}^\omega(J) \trianglelefteq A \multimap B$ and $\mathscr{H}^\omega(K) \trianglelefteq B \multimap C$ for some normalised games A, B and C. The* concatenation $\iota \ddagger \kappa$ *of ι and κ is defined by*

$$\iota \ddagger \kappa := \{ \boldsymbol{s} \in \mathbb{J}_{J \ddagger K} \mid \boldsymbol{s} \upharpoonright J \in \iota, \boldsymbol{s} \upharpoonright K \in \kappa, \boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \in pr_B \}.$$

Example 3.96. The 'non-hiding composition' between *succ*, *double* : $N \multimap N$ in the introduction is formalised by the concatenation *succ* ‡ *double* : $(N \multimap N) \ddagger (N \multimap N)$.

Theorem 3.97 (Well defined concatenation of strategies). *Let $\iota : J$ and $\kappa : K$ be strategies such that $\mathscr{H}^\omega(J) \trianglelefteq A \multimap B$ and $\mathscr{H}^\omega(K) \trianglelefteq B \multimap C$, where A, B and C are normalised games. Then, we have $\iota \ddagger \kappa : J \ddagger K$ and $\mathscr{H}^\omega(\iota); \mathscr{H}^\omega(\kappa) = \mathscr{H}^\omega(\iota \ddagger \kappa) : A \multimap C$, where $\mathscr{H}^\omega(\iota); \mathscr{H}^\omega(\kappa)$ is the* compo*sition of $\mathscr{H}^\omega(\iota) : A \multimap B$ and $\mathscr{H}^\omega(\kappa) : B \multimap C$ (Abramsky and McCusker, 1999). If ι and κ are innocent (resp. wb, noetherian, total), then so is $\iota \ddagger \kappa$. Given $\iota' : J$ and $\kappa' : K$ with $\iota \simeq_J \iota'$ and $\kappa \simeq_K \kappa'$, we have $\iota \ddagger \kappa \simeq_{J \ddagger K} \iota' \ddagger \kappa'$.*

Proof. We just show the first statement since the other ones are straightforward. It then suffices to prove $\iota \ddagger \kappa : J \ddagger K$ and $\mathscr{H}^\omega(\iota \ddagger \kappa) = \iota; \kappa$ since it implies $\iota; \kappa = \mathscr{H}^\omega(\iota \ddagger \kappa) : \mathscr{H}^\omega(J \ddagger K) \trianglelefteq$

$A \multimap C$ by Lemmas 3.54 and 3.68. However, $\mathscr{H}^{\omega}(\iota \ddagger \kappa) = \iota ; \kappa$ is immediate from the definition of concatenation; thus, we focus on $\iota \ddagger \kappa : J \ddagger K$.

First, we have the inclusion $\iota \ddagger \kappa \subseteq P_{J \ddagger K}$ because any $\boldsymbol{s} \in \iota \ddagger \kappa$ satisfies $\boldsymbol{s} \in \mathbb{J}_{J \ddagger K}$, $\boldsymbol{s} \upharpoonright J \in \iota \subseteq P_J$, $\boldsymbol{s} \upharpoonright K \in \kappa \subseteq P_K$ and $\boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \in pr_B$. It is also immediate that such $\boldsymbol{s}$ is of even length. It remains to verify the axioms S1 and S2. For this, we need the following claim:

($\diamondsuit$) Each $\boldsymbol{s} \in \iota \ddagger \kappa$ consists of adjacent pairs mn such that $m, n \in M_J$ or $m, n \in M_K$.

Proof of the claim $\diamondsuit$. By induction on $|\boldsymbol{s}|$. The base case is trivial. For the inductive step, let $\boldsymbol{s}mn \in \iota \ddagger \kappa$. If $m \in M_J$, then $(\boldsymbol{s} \upharpoonright J).m.(n \upharpoonright J) \in \sigma$, where $\boldsymbol{s} \upharpoonright J$ is of even length by the induction hypothesis. Thus, we must have $n \in M_J$. If $m \in M_K$, then $n \in M_K$ by the same argument. □

- (S1). Since $\boldsymbol{\epsilon} \in \iota \ddagger \kappa$, we have $\iota \ddagger \kappa \neq \emptyset$. For even-prefix-closure, assume $\boldsymbol{s}mn \in \iota \ddagger \kappa$. By the claim $\diamondsuit$, either $m, n \in M_J$ or $m, n \in M_K$. In either case, it is straightforward to see that $\boldsymbol{s} \in P_{J \ddagger K}$, $\boldsymbol{s} \upharpoonright J \in \iota$, $\boldsymbol{s} \upharpoonright K \in \kappa$ and $\boldsymbol{s} \upharpoonright B_{[1]}, B_{[2]} \in pr_B$, i.e., $\boldsymbol{s} \in \iota \ddagger \kappa$.
- (S2). Assume $\boldsymbol{s}mn, \boldsymbol{s}mn' \in \iota \ddagger \kappa$. By the claim $\diamondsuit$, either $m, n, n' \in M_J$ or $m, n, n' \in M_K$. In the former case, $(\boldsymbol{s} \upharpoonright J).mn, (\boldsymbol{s} \upharpoonright J).mn' \in \iota$. Thus, $n = n'$ and $\mathcal{J}_{\boldsymbol{s}mn}(n) = \mathcal{J}_{(\boldsymbol{s} \upharpoonright J).mn}(n) = \mathcal{J}_{(\boldsymbol{s} \upharpoonright J).mn'}(n') = \mathcal{J}_{\boldsymbol{s}mn'}(n')$ by S2 on ι, where note that n and n' are both P-moves and thus non-initial in J. The latter case may be handled similarly.

Therefore, we have shown that the relation $\iota \ddagger \kappa : J \ddagger K$ holds. □

At this point, let us note that totality of (dynamic) strategies is *not* preserved under composition, but it is preserved under concatenation. This phenomenon is essentially because totality is not preserved under the hiding operation as already remarked before.

For completeness, let us explicitly define the rather trivial *currying* of strategies:

Definition 3.98 (Currying of strategies). *Given a strategy $\sigma : G$ with $\mathscr{H}^{\omega}(G) \trianglelefteq A \otimes B \multimap C$ for some normalised games A, B and C, the* currying *$\Lambda(\sigma) : \Lambda(G)$ of σ is σ up to 'tags.'*

Lemma 3.99 (Well-defined currying of strategies). *Strategies are closed under currying, and currying preserves totality, innocence, well-bracketing, noetherianity and identification of strategies.*

Proof. Obvious. □

Now, as in the case of games, let us proceed to establish the *hiding lemma* on strategies (Lemma 3.102). We first need the following:

Lemma 3.100 (Hiding on legal positions in the second form). *For any arena G and number $d \in \mathbb{N} \cup \{\omega\}$, we have $\mathscr{L}_{\mathscr{H}^d(G)} = \{\, \boldsymbol{s} \natural \mathscr{H}^d_G \mid \boldsymbol{s} \in \mathscr{L}_G \,\}$.*

Proof. Observe that

$$\begin{aligned} \{\, \boldsymbol{s} \natural \mathscr{H}^d_G \mid \boldsymbol{s} \in \mathscr{L}_G \,\} &= \{\, \boldsymbol{s} \natural \mathscr{H}^d_G \mid \boldsymbol{s} \in \mathscr{L}_G, \boldsymbol{s} \text{ is } d\text{-complete} \,\} \\ &= \{\, \mathscr{H}^d_G(\boldsymbol{s}) \mid \boldsymbol{s} \in \mathscr{L}_G, \boldsymbol{s} \text{ is } d\text{-complete} \,\} \\ &= \{\, \mathscr{H}^d_G(\boldsymbol{s}) \mid \boldsymbol{s} \in \mathscr{L}_G \,\} \quad \text{(by the same argument as above)} \\ &= \mathscr{L}_{\mathscr{H}^d(G)} \quad \text{(by Corollary 3.30),} \end{aligned}$$

completing the proof. □

Notation 3.101. *We write $\spadesuit_{i \in I}$, where I is $\{1\}$ or $\{1, 2\}$, for any of the constructions on strategies introduced so far, i.e., $\spadesuit_{i \in I}$ is either $\otimes$, $(\_)^{\dagger}$, $\langle \_, \_ \rangle$, $\ddagger$, ; or Λ.*

Lemma 3.102 (Hiding lemma on strategies). *Let $\spadesuit_{i\in I}$ be a construction on strategies, and $\sigma_i : G_i$ for each $i \in I$. Then, for all $d \in \mathbb{N} \cup \{\omega\}$, we have*

(1) $\mathcal{H}^d(\spadesuit_{i\in I}\sigma_i) = \spadesuit_{i\in I}\mathcal{H}^d(\sigma_i)$ *if* $\spadesuit_{i\in I}$ *is* $\otimes$, $(\_)^\dagger$, $\langle\_,\_\rangle$ *or* Λ;
(2) $\mathcal{H}^d(\sigma_1 \ddagger \sigma_2) = \mathcal{H}^d(\sigma_1) \ddagger \mathcal{H}^d(\sigma_2)$ *if* $\mathcal{H}^d(\sigma_1 \ddagger \sigma_2)$ *is not normalised;*
(3) $\mathcal{H}^d(\sigma_1 \ddagger \sigma_2) = \mathcal{H}^d(\sigma_1); \mathcal{H}^d(\sigma_2)$ *otherwise.*

Proof. As in the case of games, it suffices to assume $d = 1$. Here, we just focus on pairing $\langle\_,\_\rangle$ since the other constructions may be handled analogously.

Let $\sigma_i : G_i$ $(i = 1, 2)$ be strategies such that $\mathcal{H}^\omega(G_1) \trianglelefteq C \multimap A$, $\mathcal{H}^\omega(G_2) \trianglelefteq C \multimap B$ for some normalised games A, B and C. For $\mathcal{H}(\langle\sigma_1, \sigma_2\rangle) \subseteq \langle\mathcal{H}(\sigma_1), \mathcal{H}(\sigma_2)\rangle$, observe that

$$\begin{aligned}
\boldsymbol{s} \in \mathcal{H}(\langle\sigma_1, \sigma_2\rangle) &\Rightarrow \exists \boldsymbol{t} \in \langle\sigma_1, \sigma_2\rangle .\, \boldsymbol{t}\natural\mathcal{H}^1_{\langle G_1,G_2\rangle} = \boldsymbol{s} \\
&\Rightarrow \exists \boldsymbol{t} \in \mathcal{L}_{\langle G_1,G_2\rangle} .\, \boldsymbol{t}\natural\mathcal{H}^1_{\langle G_1,G_2\rangle} = \boldsymbol{s} \wedge ((\boldsymbol{t} \upharpoonright G_1 \in \sigma_1 \wedge \boldsymbol{t} \upharpoonright G_2 = \boldsymbol{\varepsilon}) \vee (\boldsymbol{t} \upharpoonright G_2 \in \sigma_2 \\
&\qquad \wedge \boldsymbol{t} \upharpoonright G_1 = \boldsymbol{\varepsilon})) \\
&\Rightarrow \boldsymbol{s} \in \mathcal{L}_{\mathcal{H}(\langle G_1,G_2\rangle)} \wedge (\boldsymbol{s} \upharpoonright \mathcal{H}(G_1) \in \mathcal{H}(\sigma_1) \wedge \boldsymbol{s} \upharpoonright \mathcal{H}(G_2) = \boldsymbol{\varepsilon}) \\
&\qquad \vee (\boldsymbol{s} \upharpoonright \mathcal{H}(G_2) \in \mathcal{H}(\sigma_2) \wedge \boldsymbol{s} \upharpoonright \mathcal{H}(G_1) = \boldsymbol{\varepsilon})) \\
&\Rightarrow \boldsymbol{s} \in \langle\mathcal{H}(\sigma_1), \mathcal{H}(\sigma_2)\rangle,
\end{aligned}$$

where the third implication is by Lemma 3.100. Next, we show the converse by

$$\begin{aligned}
\boldsymbol{s} \in \langle\mathcal{H}(\sigma_1), \mathcal{H}(\sigma_2)\rangle &\Rightarrow \boldsymbol{s} \in \mathcal{L}_{\mathcal{H}(\langle G_1,G_2\rangle)} \wedge (\boldsymbol{s} \upharpoonright \mathcal{H}(G_1) \in \mathcal{H}(\sigma_1) \wedge \boldsymbol{s} \upharpoonright \mathcal{H}(G_2) = \boldsymbol{\varepsilon}) \\
&\qquad \vee (\boldsymbol{s} \upharpoonright \mathcal{H}(G_2) \in \mathcal{H}(\sigma_2) \wedge \boldsymbol{s} \upharpoonright \mathcal{H}(G_1) = \boldsymbol{\varepsilon})) \\
&\Rightarrow (\exists \boldsymbol{u} \in \sigma_1 .\, \boldsymbol{u}\natural\mathcal{H}^1_{G_1} = \boldsymbol{s} \upharpoonright \mathcal{H}(G_1) \wedge \boldsymbol{u} \upharpoonright G_2 = \boldsymbol{\varepsilon}) \\
&\qquad \vee (\exists \boldsymbol{v} \in \sigma_2 .\, \boldsymbol{v}\natural\mathcal{H}^1_{G_2} = \boldsymbol{s} \upharpoonright \mathcal{H}(G_2) \wedge \boldsymbol{v} \upharpoonright \mathcal{H}(G_1) = \boldsymbol{\varepsilon}) \\
&\Rightarrow \exists \boldsymbol{w} \in \langle\sigma_1, \sigma_2\rangle .\, \boldsymbol{w}\natural\mathcal{H}^1_{\langle G_1,G_2\rangle} = \boldsymbol{s} \\
&\Rightarrow \boldsymbol{s} \in \mathcal{H}(\langle\sigma_1, \sigma_2\rangle),
\end{aligned}$$

which completes the proof. □

A particularly important consequence of the hiding lemma is the following:

Corollary 3.103 (Closure of winning strategies). *Winning strategies are closed under tensor $\otimes$, pairing $\langle\_,\_\rangle$, promotion $(\_)^\dagger$, concatenation $\ddagger$, composition ; and currying Λ.*

Proof. Thanks to Lemmas 3.82 and 3.99 and Theorems 3.92, 3.94 and 3.97, it suffices to show that the conjunction of strong totality and noetherianity of strategies is preserved under the constructions. However, it simply follows from Lemma 3.102. □

Finally, as a technical preparation for the next section, let us define the *dereliction game* $\varXi_G$ on each game G, which is, roughly, the subgame of $G \Rightarrow G$ in which only plays by the dereliction der_G are possible. Formally:

Definition 3.104 (Dereliction games). *The* dereliction game *on a game G is the subgame $\varXi_G \trianglelefteq G \Rightarrow G$ defined by*

$$M_{\varXi_G} := M_{G\Rightarrow G} \qquad \lambda_{\varXi_G} := \lambda_{G\Rightarrow G} \qquad \vdash_{\varXi_G} := \vdash_{G\Rightarrow G}$$

$$P_{\varXi_G} := \{\, \boldsymbol{s} \in P_{G_{[0]}\Rightarrow G_{[1]}} \mid \forall \boldsymbol{t} \preceq \boldsymbol{s}.\, \mathsf{Even}(\boldsymbol{t}) \Rightarrow \boldsymbol{t} \upharpoonright G_{[0]} = \boldsymbol{t} \upharpoonright G_{[1]} \,\} \qquad \simeq_{\varXi_G} := \simeq_{G\Rightarrow G} \upharpoonright P_{\varXi_G} \times P_{\varXi_G}.$$

Further, given normalised games A and B, we define

- $\Pi_1^{A,B} \trianglelefteq A\&B \Rightarrow A$ *to be* Ξ_A *up to 'tags,' where we often abbreviate it as* Π_1*;*
- $\Pi_2^{A,B} \trianglelefteq A\&B \Rightarrow B$ *to be* Ξ_B *up to 'tags,' where we often abbreviate it as* Π_2*;*
- $\Upsilon_{A,B} \trianglelefteq B^A\&A \Rightarrow B$ *to be* $\Xi_{A\Rightarrow B}$ *up to 'tags,' where we often abbreviate it as* Υ*.*

Lemma 3.105 (D-lemma). *Given normalised games* A, B, C, $L \trianglelefteq C \Rightarrow A$, $R \trianglelefteq C \Rightarrow B$, $P \trianglelefteq C \Rightarrow A\&B$, $U \trianglelefteq A\&B \Rightarrow C$ *and* $V \trianglelefteq A \Rightarrow C^B$*, we have the equations*

$$\langle L, R\rangle^\dagger ; \Pi_1^{A,B} = L \qquad \langle L, R\rangle^\dagger ; \Pi_2^{A,B} = R \qquad \langle P^\dagger ; \Pi_1^{A,B}, P^\dagger ; \Pi_2^{A,B}\rangle = P$$

$$\langle (\Pi_1^{A,B})^\dagger ; \Lambda(U), \Pi_2^{A,B}\rangle^\dagger ; \Upsilon_{B,C} = U \qquad \Lambda(\langle (\Pi_1^{A,B})^\dagger ; V, \Pi_2^{A,B}\rangle^\dagger ; \Upsilon_{B,C}) = V.$$

Proof. Straightforward. □

4. Dynamic Game Semantics of Finitary PCF

This last main section is the climax of the present work. We define a game-semantic CCBoC $\mathbb{LDG}$ (Definition 4.1) and a standard structure $\mathscr{S}_\mathscr{G}$ for FPCF in $\mathbb{LDG}$ (Definition 4.3) in Section 4.1. Then, in Section 4.2, we show that the induced interpretation $[\![\_]\!]_{\mathbb{LDG}}^{\mathscr{S}_\mathscr{G}}$ meets the PDCP (Theorem 4.5), thus the DCP by Theorem 2.18, giving the first instance of dynamic game semantics.

4.1 Dynamic game semantics of finitary PCF

Let us now establish the CCBoC $\mathbb{LDG}$ of dynamic games and strategies:

Definition 4.1 (The CCBoC $\mathbb{LDG}$). *The CCBoC* $\mathbb{LDG} = (\mathbb{LDG}, \mathscr{H})$ *is defined as follows:*

- *Objects are normalised, well founded games;*
- *A* β*-morphism* $A \to B$ *is a pair* $(J, [\phi]_W)$ *of a game* J *that satisfies* $\mathscr{H}^\omega(J) \trianglelefteq A \Rightarrow B$ *and the equivalence class* $[\phi]_W := \{\psi : J \mid \psi \text{ is wb and winning}, \psi \simeq_J \phi\}$ *of a valid, wb, winning strategy* $\phi : J$*;*
- *The* β*-composition* $A \overset{(J,[\phi]_W)}{\to} B \overset{(K,[\psi]_W)}{\to} C$ *is the pair* $(J^\dagger \ddagger K, [\phi^\dagger \ddagger \psi]_W)$*;*
- *The* β*-identity* $id_A : A \to A$ *on each object* A *is the pair* $(\Xi_A, [der_A]_W)$*;*
- *The evaluation* $\mathscr{H}$ *maps morphisms* $(J, [\phi]_W) : A \to B$ *to the pair* $(\mathscr{H}(J), [\mathscr{H}(\phi)]_W)$*;*
- *The* β*-terminal object is the terminal game* T *(Example 3.22);*
- β*-product and* β*-exponential are, respectively, given by* $A \times B := A\&B$ *and* $B^A := A \Rightarrow B = {!A} \multimap B$ *for all objects* $A, B \in \mathbb{LDG}$*;*
- β*-pairing is given by* $\langle (L, [\alpha]_W), (R, [\beta]_W)\rangle := (\langle L, R\rangle, [\langle \alpha, \beta\rangle]_W) : C \to A\&B$ *for all objects* $A, B, C \in \mathbb{LDG}$*, and morphisms* $(L, [\alpha]_W) : C \to A$ *and* $(R, [\beta]_W) : C \to B$*;*
- *The* β*-projections* $\pi_1 : A\&B \to A$ *and* $\pi_2 : A\&B \to B$ *are the pairs* $(\Pi_1^{A,B}, [\varpi_1^{A,B}]_W)$ *and* $(\Pi_2^{A,B}, [\varpi_2^{A,B}]_W)$ *for all objects* $A, B \in \mathbb{LDG}$*, respectively, where* $\varpi_1^{A,B} : \Pi_1^{A,B}$ *and* $\varpi_2^{A,B} : \Pi_2^{A,B}$ *are, respectively, the derelictions* der_A *and* der_B *up to 'tags';*
- β*-currying is given by* $\Lambda(G, [\varphi]_W) := (\Lambda(G), [\Lambda(\varphi)]_W) : A \to (B \Rightarrow C)$ *for all objects* $A, B, C \in \mathbb{LDG}$*, and morphism* $(G, [\varphi]_W) : A\&B \to C$*;*
- *The* β*-evaluation* $ev_{B,C} : C^B\&B \to C$ *for all objects* $B, C \in \mathbb{LDG}$ *is the pair* $(\Upsilon_{B,C}, [\upsilon_{B,C}]_W)$*, where* $\upsilon_{B,C} : \Upsilon_{B,C}$ *is the dereliction* $der_{B\Rightarrow C}$ *up to 'tags.'*

Note that we have made the *underlying game* of each β-morphism in $\mathbb{LDG}$ explicit in order to take the equivalence class of strategies. Also, we have focused on *well founded* games and *wb, winning* strategies for the full completeness result (Corollary 4.7), where note that games must be well founded for derelictions to be noetherian (Lemma 3.90).

Theorem 4.2 (Well defined $\mathbb{LDG}$). *$\mathbb{LDG}$ forms a CCBoC.*

Proof. First, for β-composition, let $A, B, C \in \mathbb{LDG}$, $(J, [\phi]_{\mathrm{W}}) : A \to B$ and $(K, [\psi]_{\mathrm{W}}) : B \to C$ in $\mathbb{LDG}$. Then, $\phi^\dagger : J^\dagger$ by Theorem 3.94, and $\mathscr{H}^\omega(J^\dagger) \trianglelefteq\, !A \multimap\, !B$ by Theorem 3.46. Therefore, we have $\phi^\dagger \ddagger \psi : J^\dagger \ddagger K$ such that $\mathscr{H}^\omega(J^\dagger \ddagger K) \trianglelefteq A \Rightarrow C$ by Theorem 3.97. Also, promotion and concatenation preserve validity, wb and winning of strategies (by Theorems 3.94 and 3.97, and Corollary 3.103). Hence, the pair $(J^\dagger \ddagger K, [\phi^\dagger \ddagger \psi]_{\mathrm{W}})$ is a β-morphism $A \to C$ in $\mathbb{LDG}$. Note that the composition does not depend on the representatives ϕ and ψ.

Next, β-composition preserves the equivalence (i.e., the trivial 2-cells) $\simeq$: For any $A, B, C \in \mathbb{LDG}$, $(J, [\iota]_{\mathrm{W}}), (\tilde{J}, [\tilde{\iota}]_{\mathrm{W}}) : A \to B$ and $(K, [\kappa]_{\mathrm{W}}), (\tilde{K}, [\tilde{\kappa}]_{\mathrm{W}}) : B \to C$ in $\mathbb{LDG}$, if $\mathscr{H}^\omega(J, [\iota]_{\mathrm{W}}) = \mathscr{H}^\omega(\tilde{J}, [\tilde{\iota}]_{\mathrm{W}})$ and $\mathscr{H}^\omega(K, [\kappa]_{\mathrm{W}}) = \mathscr{H}^\omega(\tilde{K}, [\tilde{\kappa}]_{\mathrm{W}})$, then $\mathscr{H}^\omega(J^\dagger \ddagger K) = \mathscr{H}^\omega(J)^\dagger; \mathscr{H}^\omega(K) = \mathscr{H}^\omega(\tilde{J})^\dagger; \mathscr{H}^\omega(\tilde{K}) = \mathscr{H}^\omega(\tilde{J}^\dagger \ddagger \tilde{K})$ by Lemma 3.54, and $\mathscr{H}^\omega(\iota^\dagger \ddagger \kappa) \simeq_{\mathscr{H}^\omega(J^\dagger \ddagger K)} \mathscr{H}^\omega(\tilde{\iota}^\dagger \ddagger \tilde{\kappa})$ by Corollary 3.73, whence $\mathscr{H}^\omega(J^\dagger \ddagger K, [\iota^\dagger \ddagger \kappa]_{\mathrm{W}}) = \mathscr{H}^\omega(\tilde{J}^\dagger \ddagger \tilde{K}, [\tilde{\iota}^\dagger \ddagger \tilde{\kappa}]_{\mathrm{W}})$.

Clearly, the associativity of β-composition modulo the equivalence $\simeq$ holds: Given $D \in \mathbb{LDG}$, and $(G, [\varphi]) : C \to D$ in $\mathbb{LDG}$, by Lemma 3.54, we have

$$\begin{aligned}
\mathscr{H}^\omega((J^\dagger \ddagger K)^\dagger \ddagger G) &= (\mathscr{H}^\omega(J^\dagger); \mathscr{H}^\omega(K))^\dagger; \mathscr{H}^\omega(G) \\
&= (\mathscr{H}^\omega(J)^\dagger; \mathscr{H}^\omega(K)^\dagger); \mathscr{H}^\omega(G) \\
&= \mathscr{H}^\omega(J)^\dagger; (\mathscr{H}^\omega(K)^\dagger; \mathscr{H}^\omega(G)) \\
&= \mathscr{H}^\omega(J^\dagger); (\mathscr{H}^\omega(K^\dagger); \mathscr{H}^\omega(G)) \\
&= \mathscr{H}^\omega(J^\dagger \ddagger (K^\dagger \ddagger G))
\end{aligned}$$

as well as by Lemma 3.102

$$\begin{aligned}
\mathscr{H}^\omega((\phi^\dagger \ddagger \psi)^\dagger \ddagger \varphi) &= (\mathscr{H}^\omega(\phi^\dagger); \mathscr{H}^\omega(\psi))^\dagger; \mathscr{H}^\omega(\varphi) \\
&= (\mathscr{H}^\omega(\phi^\dagger); \mathscr{H}^\omega(\psi)^\dagger); \mathscr{H}^\omega(\varphi) \\
&= (\mathscr{H}^\omega(\phi^\dagger); (\mathscr{H}^\omega(\psi^\dagger); \mathscr{H}^\omega(\varphi)) \\
&= \mathscr{H}^\omega(\phi^\dagger \ddagger (\psi^\dagger \ddagger \varphi)),
\end{aligned}$$

whence $((J, [\phi]_{\mathrm{W}}); (K, [\psi]_{\mathrm{W}})); (G, [\varphi]_{\mathrm{W}}) \simeq (J, [\phi]_{\mathrm{W}}); ((K, [\psi]_{\mathrm{W}}); (G, [\varphi]_{\mathrm{W}}))$.

Similarly, the unit law modulo the equivalence $\simeq$ holds; we leave the details to the reader. Also, $\mathscr{H}$ clearly satisfies the four axioms of BoC (Definition 2.2), having shown that $\mathbb{LDG}$ is a BoC. It remains to verify its cartesian closed structure modulo the equivalence $\simeq$.

The universal property of the β-terminal game T modulo the equivalence $\simeq$ is obvious, where we define $!_A := (A \Rightarrow T, [\{\boldsymbol{\varepsilon}\}]_{\mathrm{W}}) : A \to T$ for each $A \in \mathbb{LDG}$. The β-projections are clearly values in $\mathbb{LDG}$. Given β-morphisms $(L, [\alpha]_{\mathrm{W}}) : C \to A$ and $(R, [\beta]_{\mathrm{W}}) : C \to B$ in $\mathbb{LDG}$, i.e., $\alpha : L$, $\beta : R$, $\mathscr{H}^\omega(L) \trianglelefteq C \Rightarrow A$ and $\mathscr{H}^\omega(R) \trianglelefteq C \Rightarrow B$, we obtain the valid, wb, winning pairing $\langle \alpha, \beta \rangle : \langle L, R \rangle$ such that $\mathscr{H}^\omega(\langle L, R \rangle) \trianglelefteq C \Rightarrow A \& B$ by Theorem 3.40 and Corollary 3.103. Hence, the pair $(\langle L, R \rangle, [\langle \alpha, \beta \rangle]_{\mathrm{W}})$ is a β-morphism $C \to A \& B$ in $\mathbb{LDG}$, which does not depend on the representatives α and β. Also, the β-pairing clearly preserves values in $\mathbb{LDG}$.

Also, by Lemmas 3.54 and 3.105, we have

$$\mathscr{H}^\omega(\langle L, R \rangle^\dagger \ddagger \Pi_1^{A,B}) = \langle \mathscr{H}^\omega(L), \mathscr{H}^\omega(R) \rangle^\dagger; \Pi_1^{A,B} = \mathscr{H}^\omega(L)$$

as well as by Lemma 3.102 we have

$$\mathscr{H}^\omega(\langle \alpha, \beta \rangle^\dagger \ddagger \varpi_1^{A,B}) = \langle \mathscr{H}^\omega(\alpha)^\dagger, \mathscr{H}^\omega(\beta)^\dagger \rangle; \varpi_1^{A,B} = \mathscr{H}^\omega(\alpha).$$

Similarly, we have $\mathscr{H}^\omega(\langle L, R \rangle^\dagger \ddagger \Pi_2^{A,B}) = \mathscr{H}^\omega(R)$ and $\mathscr{H}^\omega(\langle \alpha, \beta \rangle^\dagger \ddagger \varpi_2^{A,B}) = \mathscr{H}^\omega(\beta)$. Hence, $\langle (L, [\alpha]_{\mathrm{W}}), (R, [\beta]_{\mathrm{W}}) \rangle; \pi_1 \simeq (L, [\alpha]_{\mathrm{W}})$ and $\langle (L, [\alpha]_{\mathrm{W}}), (R, [\beta]_{\mathrm{W}}) \rangle; \pi_2 = (R, [\beta]_{\mathrm{W}})$ hold.

Next, given any β-morphism $(P, [\rho]_{\mathrm{W}}) : C \to A\&B$ in $\mathbb{LDG}$, we have

$$\mathscr{H}^\omega(\langle P^\dagger \ddagger \Pi_1^{A,B}, P^\dagger \ddagger \Pi_2^{A,B} \rangle) = \langle \mathscr{H}^\omega(P)^\dagger ; \Pi_1^{A,B}, \mathscr{H}^\omega(P)^\dagger ; \Pi_2^{A,B} \rangle = \mathscr{H}^\omega(P)$$

again by Lemmas 3.54 and 3.105, as well as by Lemma 3.102, we have

$$\mathscr{H}^\omega(\langle \rho^\dagger \ddagger \varpi_1^{A,B}, \rho^\dagger \ddagger \varpi_2^{A,B} \rangle) = \langle \mathscr{H}^\omega(\rho)^\dagger ; \varpi_1^{A,B}, \mathscr{H}^\omega(\rho)^\dagger ; \varpi_2^{A,B} \rangle = \mathscr{H}^\omega(\rho).$$

Hence, $\langle (P, [\rho]); \pi_1, (P, [\rho]); \pi_2 \rangle \simeq (P, [\rho])$ holds.

It is also straightforward to check that β-pairing in $\mathbb{LDG}$ preserves the equivalence $\simeq$: Given any β-morphisms $(L, [\alpha]_{\mathrm{W}}), (\tilde{L}, [\tilde{\alpha}]_{\mathrm{W}}) : C \to A$ and $(R, [\beta]_{\mathrm{W}}), (\tilde{R}, [\tilde{\beta}]_{\mathrm{W}}) : C \to B$ in $\mathbb{LDG}$ such that $\mathscr{H}^\omega(L, [\alpha]_{\mathrm{W}}) = \mathscr{H}^\omega(\tilde{L}, [\tilde{\alpha}]_{\mathrm{W}})$ and $\mathscr{H}^\omega(R, [\beta]_{\mathrm{W}}) = \mathscr{H}^\omega(\tilde{R}, [\tilde{\beta}]_{\mathrm{W}})$, we have

$$\begin{aligned}
\mathscr{H}^\omega(\langle (L, [\alpha]_{\mathrm{W}}), (R, [\beta]_{\mathrm{W}}) \rangle) &= (\mathscr{H}^\omega(\langle L, R \rangle), [\mathscr{H}^\omega(\langle \alpha, \beta \rangle)]_{\mathrm{W}}) \\
&= (\langle \mathscr{H}^\omega(L), \mathscr{H}^\omega(R) \rangle, [\langle \mathscr{H}^\omega(\alpha), \mathscr{H}^\omega(\beta) \rangle]_{\mathrm{W}}) \\
&= (\langle \mathscr{H}^\omega(\tilde{L}), \mathscr{H}^\omega(\tilde{R}) \rangle, [\langle \mathscr{H}^\omega(\tilde{\alpha}), \mathscr{H}^\omega(\tilde{\beta}) \rangle]_{\mathrm{W}}) \\
&= (\mathscr{H}^\omega(\langle \tilde{L}, \tilde{R} \rangle), [\mathscr{H}^\omega(\langle \tilde{\alpha}, \tilde{\beta} \rangle)]_{\mathrm{W}}) \\
&= \mathscr{H}^\omega(\langle (\tilde{L}, [\tilde{\alpha}]_{\mathrm{W}}), (\tilde{R}, [\tilde{\beta}]_{\mathrm{W}}) \rangle).
\end{aligned}$$

Finally, the requirements for β-exponentials, β-currying and β-evaluations are proved similarly to the cases of β-products, β-pairing and β-projections; we leave them to the reader. □

We proceed to give a standard structure (Definition 2.15) for FPCF in $\mathbb{LDG}$:

Definition 4.3 (Standard structure in $\mathbb{LDG}$). *The standard structure*

$$\mathscr{S}_{\mathscr{G}} = (\mathbf{2}, T, \&, \pi, \Rightarrow, ev, \underline{tt}, \underline{ff}, \vartheta)$$

of games and strategies for FPCF in $\mathbb{LDG}$ is defined as follows:

- *$\mathbf{2}$ is the game of booleans (Example 3.23), and T is the terminal game (Example 3.22);*
- *$\&$ is product on games, and $\pi_i^{A,B} := (\Pi_i^{A,B}, [\varpi_i^{A,B}]_{\mathrm{W}})$ $(i = 1, 2)$ for any $A, B \in \mathbb{LDG}$;*
- *$\Rightarrow$ is function space on games, and $ev_{A,B} := (\Upsilon_{A,B}, [\upsilon_{A,B}]_{\mathrm{W}})$ for any $A, B \in \mathbb{LDG}$;*
- *$\underline{tt} := (T \Rightarrow \mathbf{2}, [\mathsf{Pref}(\{q.tt\})^{\mathsf{Even}}]_{\mathrm{W}}), \underline{ff} := (T \Rightarrow \mathbf{2}, [\mathsf{Pref}(\{q.ff\})^{\mathsf{Even}}]_{\mathrm{W}}) : T \to \mathbf{2}$;*
- *$\vartheta := (\mathbf{2}\&(\mathbf{2}\&\mathbf{2}) \Rightarrow \mathbf{2}, [case]_{\mathrm{W}}) : \mathbf{2}\&(\mathbf{2}\&\mathbf{2}) \to \mathbf{2}$, where $case : \mathbf{2}\&(\mathbf{2}\&\mathbf{2}) \Rightarrow \mathbf{2}$, is the standard game semantics of the* case*-construction (Abramsky and McCusker, 1999) modified to a normalised (dynamic) strategy in the obvious manner.*

Lemma 4.4 (Standardness of $\mathscr{S}_{\mathscr{G}}$). *The structure $\mathscr{S}_{\mathscr{G}}$ for FPCF in $\mathbb{LDG}$ is standard in the sense defined in Definition 2.15.*

Proof. Straightforward, where note that the underlying game J of each morphism $(J, [\phi]_{\mathrm{W}})$ in $\mathbb{LDG}$ is essential for the structure $\mathscr{S}_{\mathscr{G}}$ to satisfy the inequality (5). □

4.2 Game-semantic dynamic correspondence property for FPCF

At last, we are now ready to prove that our game semantics satisfies the DCP (Definition 2.16):

Theorem 4.5 (PDCP-theorem). *The interpretation $[\![\_]\!]_{\mathbb{LDG}}^{\mathscr{S}_{\mathscr{G}}}$ of FPCF (Definitions 2.15 and 4.1) satisfies the PDCP (Definition 2.17).*

Proof. For the PDCP, the only nontrivial case is to show for any reduction of FPCF of the form $(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{V})\mathsf{W} \to \mathsf{U}$, where V, W and U are values, the equation $\mathscr{H}([\![(\lambda \mathsf{x}^{\mathsf{A}}.\,\mathsf{V})\mathsf{W}]\!]_{\mathbb{LDG}}^{\mathscr{S}_{\mathscr{G}}}) = [\![\mathsf{U}]\!]_{\mathbb{LDG}}^{\mathscr{S}_{\mathscr{G}}}$ (n.b., the inequality $[\![(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{V})\mathsf{W}]\!]^{\mathscr{S}\!\mathscr{G}}_{\mathbb{LDG}} \neq [\![\mathsf{U}]\!]^{\mathscr{S}\!\mathscr{G}}_{\mathbb{LDG}}$ is immediate from the first component of each β-morphism in $\mathbb{LDG}$ and the second axiom on standardness of $\mathscr{S}\!\mathscr{G}$); the other conditions for the PDCP follow from Lemmas 3.54 and 3.102. Let us focus on the nontrivial case, for which we define the *height* $Ht(\mathsf{B}) \in \mathbb{N}$ of each type B by $Ht(o) := 0$ and $Ht(\mathsf{B}_1 \Rightarrow \mathsf{B}_2) := \max(Ht(\mathsf{B}_1)+1, Ht(\mathsf{B}_2))$. We proceed by induction on the height of the type A of W.

In the following, given β-morphisms $(H, [\tau]_{\mathrm{W}}) : C \to (A \Rightarrow B)$ and $(G, [\sigma]_{\mathrm{W}}) : C \to A$ in $\mathbb{LDG}$, we define the β-morphism $(H, [\tau]_{\mathrm{W}})\lfloor(G, [\sigma]_{\mathrm{W}})\rfloor := (\langle G, H\rangle^{\dagger} \ddagger \Upsilon_{A,B}, [\langle\tau, \sigma\rangle^{\dagger} \ddagger \upsilon_{A,B}]_{\mathrm{W}}) : C \to B$ in $\mathbb{LDG}$. More generally, if $(H, [\tau]_{\mathrm{W}}) : C \to (A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_k \Rightarrow B)$ and $(G_i, [\sigma_i]_{\mathrm{W}}) : C \to A_i$ for $i = 1, 2, \dots, k$, then we write $(H, [\tau]_{\mathrm{W}})\lfloor(G_1, [\sigma_1]_{\mathrm{W}}), (G_2, [\sigma_2]_{\mathrm{W}}), \dots, (G_k, [\sigma_k]_{\mathrm{W}})\rfloor$ for $(H, [\tau]_{\mathrm{W}})\lfloor(G_1, [\sigma_1]_{\mathrm{W}})\rfloor\lfloor(G_2, [\sigma_2]_{\mathrm{W}})\rfloor \dots \lfloor(G_k, [\sigma_k]_{\mathrm{W}})\rfloor : C \to B$. We abbreviate $[\![\_]\!]^{\mathscr{S}\!\mathscr{G}}_{\mathbb{LDG}}$ as $[\![\_]\!]$. Let Γ be the context of $(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{V})\mathsf{W}$ (and U). We abbreviate each β-morphism $(G, [\sigma]_{\mathrm{W}})$ in $\mathbb{LDG}$ as $[\sigma]$ for brevity, and focus on the second components (i.e., the equivalence classes of strategies); the corresponding equations on the first components (i.e., games) may be obtained, thanks to Lemmas 3.54 and 3.105, similarly to the ways for the first components shown below.

For the base case, assume $Ht(\mathsf{A}) = 0$, i.e., $\mathsf{A} \equiv o$. By induction on $|\mathsf{V}|$, we have

- If $\mathsf{V} \equiv \mathsf{tt}$, then $(\lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{tt})\mathsf{W} \to \mathsf{tt}$, and clearly $\mathscr{H}([\![(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{tt})\mathsf{W}]\!]) = \langle \Lambda([\![\mathsf{tt}]\!]), [\![\mathsf{W}]\!]\rangle^{\dagger}; [\upsilon] = [\![\mathsf{tt}]\!]$. The case of $\mathsf{V} \equiv \mathsf{ff}$ is analogous.
- If $\mathsf{V} \equiv \lambda \mathsf{y}^{\mathsf{C}}.\, \mathsf{V}'$, then $(\lambda \mathsf{x}^{\mathsf{A}}\mathsf{y}^{\mathsf{C}}.\, \mathsf{V}')\mathsf{W} \to \lambda \mathsf{y}^{\mathsf{C}}.\mathsf{U}'$ with $(\lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{V}')\mathsf{W} \to \mathsf{U}'$ (since $\mathit{nf}((\lambda \mathsf{x}^{\mathsf{A}}\mathsf{y}^{\mathsf{C}}.\, \mathsf{V}')\mathsf{W}) \equiv \mathit{nf}(\lambda \mathsf{y}^{\mathsf{C}}.\, \mathsf{V}'[\mathsf{W}/\mathsf{x}]) \equiv \lambda \mathsf{y}^{\mathsf{C}}.\, \mathit{nf}(\mathsf{V}'[\mathsf{W}/\mathsf{x}]) \equiv \lambda \mathsf{y}^{\mathsf{C}}.\, \mathit{nf}((\lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{V}')\mathsf{W}))$. By the induction hypothesis, $\mathscr{H}([\![(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{V}')\mathsf{W}]\!]) = [\![\mathsf{U}']\!]$. Hence, we have

$$\begin{aligned}
\mathscr{H}([\![\mathsf{VW}]\!]) &= \mathscr{H}(\langle \Lambda_{[\![\mathsf{A}]\!]}(\Lambda_{[\![\mathsf{C}]\!]}([\![\mathsf{V}']\!])), [\![\mathsf{W}]\!]\rangle^{\dagger} \ddagger [\upsilon]) \\
&= \langle \Lambda_{[\![\mathsf{A}]\!]}(\Lambda_{[\![\mathsf{C}]\!]}([\![\mathsf{V}']\!])), [\![\mathsf{W}]\!]\rangle^{\dagger}; [\upsilon] \quad \text{(by Lemma 3.102)} \\
&= \Lambda_{[\![\mathsf{C}]\!]}(\langle \Lambda_{[\![\mathsf{A}]\!]}([\![\mathsf{V}']\!]), [\![\mathsf{W}]\!]\rangle^{\dagger}; [\upsilon]) \\
&= \Lambda_{[\![\mathsf{C}]\!]}(\mathscr{H}(\langle \Lambda_{[\![\mathsf{A}]\!]}([\![\mathsf{V}']\!]), [\![\mathsf{W}]\!]\rangle^{\dagger} \ddagger [\upsilon])) \\
&= \Lambda_{[\![\mathsf{C}]\!]}(\mathscr{H}([\![(\lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{V}')\mathsf{W}]\!])) \\
&= \Lambda_{[\![\mathsf{C}]\!]}([\![\mathsf{U}']\!]) \\
&= [\![\lambda \mathsf{y}^{\mathsf{C}}.\, \mathsf{U}']\!].
\end{aligned}$$

- If $\mathsf{V} \equiv \mathsf{case}(\mathsf{y}\mathsf{V}_1 \dots \mathsf{V}_k)[\tilde{\mathsf{V}}_1; \tilde{\mathsf{V}}_2]$ with $\mathsf{x} \neq \mathsf{y}$, then $(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{V})\mathsf{W} \to \mathsf{U}$, where

$$\mathsf{U} \equiv \mathsf{case}(\mathsf{y}\,\mathit{nf}(\mathsf{V}_1[\mathsf{W}/\mathsf{x}]) \dots \mathit{nf}(\mathsf{V}_k[\mathsf{W}/\mathsf{x}]))[\mathit{nf}(\tilde{\mathsf{V}}_1[\mathsf{W}/\mathsf{x}]); \mathit{nf}(\tilde{\mathsf{V}}_1[\mathsf{W}/\mathsf{x}])].$$

By the induction hypothesis and the interpretation of the variable y, we have

$$\begin{aligned}
&\mathscr{H}([\![(\lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{V})\mathsf{W}]\!]) \\
&= \mathscr{H}^{\omega}(\Lambda_{[\![\mathsf{A}]\!]}(\langle [\![\mathsf{y}]\!]\lfloor [\![\mathsf{V}_1]\!], \dots, [\![\mathsf{V}_k]\!]\rfloor, \langle [\![\tilde{\mathsf{V}}_1]\!], [\![\tilde{\mathsf{V}}_2]\!]\rangle\rangle^{\dagger} \ddagger [\mathit{case}])\lfloor [\![\mathsf{W}]\!]\rfloor) \\
&= \mathscr{H}^{\omega}(\langle \Lambda_{[\![\mathsf{A}]\!]}([\![\mathsf{y}]\!])\lfloor [\![\mathsf{W}]\!]\rfloor \lfloor \Lambda_{[\![\mathsf{A}]\!]}([\![\mathsf{V}_1]\!])\lfloor [\![\mathsf{W}]\!]\rfloor, \dots, \Lambda_{[\![\mathsf{A}]\!]}([\![\mathsf{V}_k]\!])\lfloor [\![\mathsf{W}]\!]\rfloor\rfloor, \langle \Lambda_{[\![\mathsf{A}]\!]}([\![\tilde{\mathsf{V}}_1]\!])\lfloor [\![\mathsf{W}]\!]\rfloor, \\
&\qquad \Lambda_{[\![\mathsf{A}]\!]}([\![\tilde{\mathsf{V}}_2]\!])\lfloor [\![\mathsf{W}]\!]\rfloor\rangle\rangle^{\dagger} \ddagger [\mathit{case}]) \\
&= \mathscr{H}^{\omega}(\langle [\![(\lambda \mathsf{x}.\, \mathsf{y})\mathsf{W}]\!]\lfloor [\![(\lambda \mathsf{x}.\, \mathsf{V}_1)\mathsf{W}]\!], \dots, [\![(\lambda \mathsf{x}.\, \mathsf{V}_k)\mathsf{W}]\!]\rfloor, \langle [\![(\lambda \mathsf{x}.\, \tilde{\mathsf{V}}_1)\mathsf{W}]\!], [\![(\lambda \mathsf{x}.\, \tilde{\mathsf{V}}_2)\mathsf{W}]\!]\rangle\rangle^{\dagger} \ddagger [\mathit{case}]) \\
&= \mathscr{H}^{\omega}(\langle [\![\mathsf{y}]\!]\lfloor [\![\mathit{nf}(\mathsf{V}_1[\mathsf{W}/\mathsf{x}])]\!], \dots, [\![\mathit{nf}(\mathsf{V}_k[\mathsf{W}/\mathsf{x}])]\!]\rfloor, \langle [\![\mathit{nf}(\tilde{\mathsf{V}}_1[\mathsf{W}/\mathsf{x}])]\!], [\![\mathit{nf}(\tilde{\mathsf{V}}_2[\mathsf{W}/\mathsf{x}])]\!]\rangle\rangle^{\dagger} \ddagger [\mathit{case}]) \\
&= [\![\mathsf{U}]\!].
\end{aligned}$$

- If $\mathsf{V} \equiv \mathsf{case}(\mathsf{x})[\tilde{\mathsf{V}}_1; \tilde{\mathsf{V}}_2]$, then $(\lambda \mathsf{x}^{\mathsf{A}}.\mathsf{V})\mathsf{W} \to \mathsf{U}$, where

$$\mathsf{U} \equiv \mathsf{case}(\mathsf{W})[\mathit{nf}(\tilde{\mathsf{V}}_1[\mathsf{W}/\mathsf{x}]); \mathit{nf}(\tilde{\mathsf{V}}_2[\mathsf{W}/\mathsf{x}])].$$

By the same reasoning as the above case, we get $\mathscr{H}([\![(\lambda \mathsf{x}^{\mathsf{A}}.\, \mathsf{V})\mathsf{W}]\!]) = [\![\mathsf{U}]\!]$.

Next, for the inductive step, assume $Ht(\mathsf{A}) = h+1$. We proceed in the same way as the base case, i.e., by induction on $|\mathsf{V}|$, except that the last case is generalised to $\mathsf{V} \equiv \mathsf{case}(\mathsf{x}\mathsf{V}_1 \dots \mathsf{V}_k)[\tilde{\mathsf{V}}_1; \tilde{\mathsf{V}}_2]$, where $\mathsf{A} \equiv \mathsf{A}_1 \Rightarrow \mathsf{A}_2 \Rightarrow \dots \Rightarrow \mathsf{A}_k \Rightarrow o$ $(k \geqslant 0)$. We have to consider the additional case of $k \geqslant 1$; then we have $(\lambda \mathsf{x}^{\mathsf{A}}. \mathsf{V})\mathsf{W} \to \mathsf{U}$, where

$$\mathsf{U} \equiv \mathsf{case}(nf(\mathsf{W}(\mathsf{V}_1[\mathsf{W}/\mathsf{x}]) \dots (\mathsf{V}_k[\mathsf{W}/\mathsf{x}])))[nf(\tilde{\mathsf{V}}_1[\mathsf{W}/\mathsf{x}]); nf(\tilde{\mathsf{V}}_2[\mathsf{W}/\mathsf{x}])].$$

We then have the following chain of equations:

$$\begin{aligned}
&\mathscr{H}[\![(\lambda \mathsf{x}.\, \mathsf{V})\mathsf{W}]\!] \\
&= \mathscr{H}(\Lambda(\mathscr{H}^{\omega}(\langle [\![\mathsf{x}]\!] \lfloor [\![\mathsf{V}_1]\!], \dots, [\![\mathsf{V}_k]\!] \rfloor, \langle [\![\tilde{\mathsf{V}}_1]\!], [\![\tilde{\mathsf{V}}_2]\!] \rangle \rangle^{\dagger} \ddagger [case]))\lfloor [\![\mathsf{W}]\!] \rfloor) \\
&= \mathscr{H}^{\omega}(\langle \Lambda([\![\mathsf{x}]\!])\lfloor [\![\mathsf{W}]\!] \rfloor \lfloor \Lambda([\![\mathsf{V}_1]\!])\lfloor [\![\mathsf{W}]\!] \rfloor, \dots, \Lambda([\![\mathsf{V}_k]\!])\lfloor [\![\mathsf{W}]\!] \rfloor \rfloor, \langle \Lambda([\![\tilde{\mathsf{V}}_1]\!])\lfloor [\![\mathsf{W}]\!] \rfloor, \\
&\qquad \Lambda([\![\tilde{\mathsf{V}}_2]\!])\lfloor [\![\mathsf{W}]\!] \rfloor \rangle \rangle^{\dagger} \ddagger [case]) \\
&= \mathscr{H}^{\omega}(\langle [\![(\lambda \mathsf{x}.\, \mathsf{x})\mathsf{W}]\!] \lfloor [\![(\lambda \mathsf{x}.\, \mathsf{V}_1)\mathsf{W}]\!], \dots, [\![(\lambda \mathsf{x}.\, \mathsf{V}_k)\mathsf{W}]\!] \rfloor, \langle [\![(\lambda \mathsf{x}.\, \tilde{\mathsf{V}}_1)\mathsf{W}]\!], [\![(\lambda \mathsf{x}.\, \tilde{\mathsf{V}}_2)\mathsf{W}]\!] \rangle \rangle^{\dagger} \ddagger [case]) \\
&= \mathscr{H}^{\omega}(\langle [\![\mathsf{W}]\!] \lfloor [\![nf(\mathsf{V}_1[\mathsf{W}/\mathsf{x}])]\!], \dots, [\![nf(\mathsf{V}_k[\mathsf{W}/\mathsf{x}])]\!] \rfloor, \langle [\![nf(\mathsf{V}_1[\tilde{\mathsf{W}}/\mathsf{x}])]\!], [\![nf(\mathsf{V}_2[\tilde{\mathsf{W}}/\mathsf{x}])]\!] \rangle \rangle^{\dagger} \ddagger [case]) \\
&\quad \text{(by the induction hypothesis with respect to } |\mathsf{V}|) \\
&= \mathscr{H}^{\omega}(\langle [\![nf(\mathsf{W}(\mathsf{V}_1[\mathsf{W}/\mathsf{x}]) \dots (\mathsf{V}_k[\mathsf{W}/\mathsf{x}]))]\!], \langle [\![nf(\mathsf{V}_1[\tilde{\mathsf{W}}/\mathsf{x}])]\!], [\![nf(\mathsf{V}_2[\tilde{\mathsf{W}}/\mathsf{x}])]\!] \rangle \rangle^{\dagger} \ddagger [case]) \\
&\quad \text{(by the induction hypothesis (applied } k\text{-times) with respect to the hight of types } \mathsf{A}_i \ (i \in \overline{k})) \\
&= [\![\mathsf{case}(nf(\mathsf{W}(\mathsf{V}_1[\mathsf{W}/\mathsf{x}]) \dots (\mathsf{V}_k[\mathsf{W}/\mathsf{x}])))[nf(\mathsf{V}_1[\tilde{\mathsf{W}}/\mathsf{x}]); nf(\mathsf{V}_2[\tilde{\mathsf{W}}/\mathsf{x}])]]\!] \\
&= [\![\mathsf{U}]\!]
\end{aligned}$$

which completes the proof. □

Corollary 4.6 (Dynamic game semantics of FPCF). *The interpretation $[\![\_]\!]^{\mathscr{S}\mathscr{G}}_{\mathbb{LDG}}$ of FPCF and the hiding operation $\mathscr{H}$ satisfy the DCP in the sense of Definition 2.16.*

Proof. By Lemma 4.4 and Theorems 2.18, 4.2 and 4.5. □

The relation between the syntax and the semantics is actually tighter than Corollary 4.6: Exploiting the *strong definability* result (Amadio and Curien, 1998; Hyland and Ong, 2000), FPCF can be seen as a *formal calculus* for computations in the CCBoC $\mathbb{LDG}$. In addition, FPCF represents every computation in $\mathbb{LDG}$ by the following *full completeness* result (Curien, 2007): Any strategy on a game that interprets a type of FPCF is the denotation of some term of FPCF.

Corollary 4.7 (Dynamic full completeness). *Let G be a game such that for some strategy $\sigma : G$ the pair $(G, [\sigma]_W)$ is the interpretation $[\![\Gamma \vdash \mathsf{M} : \mathsf{B}]\!]^{\mathscr{S}\mathscr{G}}_{\mathbb{LDG}}$ of a programme $\Gamma \vdash \mathsf{M} : \mathsf{B}$ in FPCF. Then, for any strategy $\tilde{\sigma} : G$, there is a programme $\Gamma \vdash \tilde{\mathsf{M}} : \mathsf{B}$ in FPCF such that $[\![\Gamma \vdash \tilde{\mathsf{M}} : \mathsf{B}]\!]^{\mathscr{S}\mathscr{G}}_{\mathbb{LDG}} = (G, [\tilde{\sigma}]_W)$.*

Proof. Note that the game G is constructed along the construction of type B in FPCF. We proceed by induction on the construction of G (or B). First, since values of FPCF are PCF Böhm trees except that the natural number type ι is replaced with the boolean type o, and the bottom term $\perp$ is deleted, the conventional full completeness and the strong definability hold for values of FPCF in the same way as that of the wb, innocent game semantics of PCF (Abramsky and McCusker, 1999; Hyland and Ong, 2000), where winning of strategies excludes the denotation of the bottom term $\perp$; see Abramsky and McCusker (1999), Curien (2006) for the details.

It remains to consider the rule A for applications (Definition 2.4), i.e., the case where G is of the form $\langle U, V \rangle^{\dagger} \ddagger \Upsilon$. But then, note that only plays by the dereliction (up to 'tags') are possible in Υ (Definition 3.104), and therefore we may just apply the induction hypothesis. □

5. Conclusion and Future Work

We have presented a *mathematical* (and *syntax-independent*) formulation of dynamics and intensionality of computation in terms of bicategories and game semantics. From the opposite angle, we have developed bicategorical and game-semantic frameworks for dynamic, intensional computation with a convenient formal calculus.

Let us emphasise that the dynamic, intensional nature of our semantics stands in sharp contrast to the static, extensional nature of conventional (categorical or game) semantics. In particular, our semantics satisfies the highly nontrivial DCP with respect to FPCF (Definition 2.16).

The present work refines and generalises standard categorical and game semantics of type theories. For instance, composition of static strategies is decomposed and generalised into *concatenation plus hiding* on dynamic strategies. Also, standard constructions and constraints on static games and strategies are naturally accommodated in dynamic games and strategies. From the category-theoretic point, the present work refines the standard CCC-interpretation of type theories by the CCBoC-interpretation. In this sense, our approach is natural and general, achieving *mathematics* of dynamics and intensionality of computation as promised in Section 1.

Let us remark that our result does not contradict the standard result by Danos et al. (1996), i.e., the correspondence between the execution of LHR and the step-by-step 'internal communication' between conventional strategies. In fact, LHR is a finer reduction strategy than the operational semantics of FPCF (Definition 2.4), and the work by Danos et al. implies that LHR corresponds in conventional game semantics what should be called a 'move-wise' execution of the hiding operation. On the other hand, our operational semantics is executed in a much coarser, 'type-wise' fashion, and thus it may be seen as executing at a time a certain 'chunk' of LHR in a specific order. Our dynamic game semantics captures such a coarser dynamics of computation, and thus it does not contradict Danos et al. (1996).

Of course, it is highly interesting to refine the present work to capture LHR or another, finer reduction strategy such as *explicit substitution* (Rose, 1996) and the *differential λ-calculus* (Ehrhard and Regnier, 2003), which we leave as future work.

More generally, the most immediate future work is to apply the present framework of dynamic game semantics to various other logics and computations as in the case of static game semantics (Abramsky and McCusker, 1999). Also, it would be interesting to see how accurately our game-semantic approach can measure the computational complexity of (higher-order) programmes.

Finally, the notion of (CC)BoCs can be a concept of interest in its own right. For instance, it might be fruitful to develop it further to accommodate various models of computations in the same spirit of Longley and Normann (2015) but on *computation*, not computability. Also, it might be interesting to consider their relation with *computations as monads* (Moggi, 1991).

Acknowledgements. The first author acknowledges the financial support from Funai Overseas Scholarship, and Luke Ong and Sam Staton for fruitful discussions. The second author acknowledges support from the EPSRC grant EP/K015478/1 on Quantum Mathematics and Computation and U.S. AFOSR FA9550-12-1-0136.

Notes

1 For simplicity, here we focus on *closed* terms, i.e., ones with the *empty context*.

2 The diagram is depicted as above only to clarify which component game each move belongs to; it should be read just as a finite sequence, namely, $q_{[1]}q_{[0]}n_{[0]}m_{[1]}$, equipped with the pointers represented by the arrows. (N.b., the arrows represent pointers, not edges of the tree, unlike the diagram of N.)

3 Composition of strategies is *associative*; see Abramsky et al. (1997), Hyland (1997), Abramsky and McCusker (1999) for the details. Therefore, the order of applying composition on strategies does not matter.

4 N.b., for the present work, it suffices to know that a CCB is a generalised CCC in the sense that the equational axioms of CCCs are required to hold only *up to 2-cell isomorphisms*.

5 N.b., the unit law *on the nose* does not hold if the composition is a non-normalising one.

6 N.b., there is no rewriting between 1-cells $(f; g); h$ and $f; (g; h)$ if the composition is non-normalising.

7 Note that if $\mathscr{E}^{n_1}(f), \mathscr{E}^{n_2}(f) \in \mathscr{V}_{\mathscr{C}}(A, B)$ for any $n_1, n_2 \in \mathbb{N}$, then clearly $\mathscr{E}^{n_1}(f) = \mathscr{E}^{n_2}(f)$, where $\mathscr{E}^n$ denotes the n-times iteration of $\mathscr{E}$ for each $n \in \mathbb{N}$.
8 In the present work, every dynamic strategy (or β-morphism) becomes a value by a *finite* iteration of the hiding operation (or evaluation) due to the axiom on labelling functions (Definition 3.1), and thus the axiom Termination (Definition 2.2) makes sense. Of course, if we consider another, in particular finer, evaluation of computations (which is left as future work), then this point may no longer hold.
9 I.e., we assume that in any term of concern every bound variable is chosen to be different from any free variable occurring in that mathematical context.
10 One may say that the operational semantics $\rightarrow$ executes the parallel $\beta\vartheta$-reduction $\rightrightarrows_{\beta\vartheta}$ in a controlled manner that does not produce non-programme terms at all.

References

Abramsky, S. (1997). Semantics of interaction: An introduction to game semantics. In: *Semantics and Logics of Computation*, Publications of the Newton Institute, 1–31.
Abramsky, S., Jagadeesan, R. and Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation* **163** (2) 409–470.
Abramsky, S. and Jung, A. (1994). Domain theory. In: *Handbook of Logic in Computer Science*, Oxford University Press, New York.
Abramsky, S. and McCusker, G. (1999). Game semantics. In: *Computational Logic*, Springer, Berlin, Heidelberg, 1–55.
Abramsky, S. and Melliès, P.-A. (1999). Concurrent games and full completeness. In: *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science*, IEEE Computer Society, 431.
Amadio, R. M. and Curien, P.-L. (1998). *Domains and Lambda-Calculi*, vol. 46, Cambridge, Cambridge University Press.
Barendregt, H. P. (1984). *The Lambda Calculus*, vol. 3, Amsterdam, North-Holland.
Church, A. (1940). A formulation of the simple theory of types. *The Journal of Symbolic Logic* **5** (02) 56–68.
Clairambault, P. and Harmer, R. (2010). Totality in arena games. *Annals of Pure and Applied Logic* **161** (5) 673–689.
Curien, P.-L. (2006). Notes on game semantics. From the author's web page.
Curien, P.-L. (2007). Definability and full abstraction. *Electronic Notes in Theoretical Computer Science* **172** 301–310.
Danos, V., Herbelin, H. and Regnier, L. (1996). Game semantics and abstract machines. In: *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, IEEE Computer Society, 394.
Danos, V. and Regnier, L. (2004). Head linear reduction. Unpublished.
Dimovski, A., Ghica, D. R. and Lazić, R. (2005). Data-abstraction refinement: A game semantic approach. In: *International Static Analysis Symposium*, Springer, 102–117.
Ehrhard, T. and Regnier, L. (2003). The differential lambda-calculus. *Theoretical Computer Science* **309** (1) 1–41.
Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove, M. and Scott, D. S. (2003). *Continuous Lattices and Domains*, (Encyclopedia of Mathematics and its Applications, pp. I-Iv). vol. 93, Cambridge, Cambridge University Press.
Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science* **50** (1) 1–101.
Girard, J.-Y. (1989). Geometry of interaction I: Interpretation of system F. *Studies in Logic and the Foundations of Mathematics* **127** 221–260.
Girard, J.-Y. (1990). Geometry of interaction II: Deadlock-free algorithms. In: *COLOG-88*, Springer, 76–93.
Girard, J.-Y. (1995). Geometry of interaction III : Accommodating the additives. In: Girard, Lafont and Regnier, (eds.) *Advances in Linear Logic*, 329–389, Cambridge, Cambridge University Press.
Girard, J.-Y. (2003). Geometry of interaction IV: The feedback equation. In: *Logic Colloquium*, vol. 3, Citeseer, 76–117.
Girard, J.-Y. (2011). Geometry of interaction V: Logic in the hyperfinite factor. *Theoretical Computer Science* **412** (20) 1860–1883.
Girard, J.-Y. (2013). Geometry of interaction VI: A blueprint for transcendental syntax. preprint.
Girard, J.-Y., Taylor, P. and Lafont, Y. (1989). *Proofs and Types*, vol. 7, Cambridge, Cambridge University Press.
Greenland, W. E. (2005). *Game Semantics for Region Analysis*. Phd thesis, University of Oxford.
Gunter, C. A. (1992). *Semantics of Programming Languages: Structures and Techniques,* Cambridge, MA, MIT press.
Hankin, C. (1994). *Lambda Calculi: A Guide for computer scientists* (Vol. 3), USA, Oxford University Press.
Harmer, R. (2004). Innocent game semantics. In: *Lecture Notes*, 2007.
Hilken, B. P. (1996). Towards a proof theory of rewriting: The simply typed 2λ-calculus. *Theoretical Computer Science* **170** (1–2) 407–444.
Hyland, J. M. E. and Ong, C.-H. (2000). On full abstraction for PCF: I, II, and III. *Information and Computation* **163** (2) 285–408.
Hyland, M. (1997). Game semantics. In: *Semantics and Logics of Computation*, vol. 14, New York, Cambridge University Press, 131.
Jacobs, B. (1999). *Categorical Logic and Type Theory*, vol. 141, Amsterdam, Elsevier.
Lambek, J. and Scott, P. J. (1988). *Introduction to Higher-order Categorical Logic*, vol. 7, USA, Cambridge University Press.

Laurent, O. (2004). Polarized games. *Annals of Pure and Applied Logic* **130** (1–3) 79–123.

Longley, J. and Normann, D. (2015). *Higher-Order Computability*, Heidelberg, Springer.

McCusker, G. (1998). *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*, London, Springer Science & Business Media.

Mellies, P.-A. (2005). Axiomatic rewriting theory I: A diagrammatic standardization theorem. In: *Processes, Terms and Cycles: Steps on the Road to Infinity*, Springer, 554–638.

Moggi, E. (1991). Notions of computation and monads. *Information and Computation* **93** (1) 55–92.

Ong, C.-H. (2006). On model-checking trees generated by higher-order recursion schemes. In: *21st Annual IEEE Symposium on Logic in Computer Science (LICS'06)*, IEEE, 81–90.

Ouaknine, J. (1997). *A Two-Dimensional Extension of Lambek's Categorical Proof Theory*. Phd thesis, McGill University, Montréal.

Pitts, A. M. (2001). Categorical Logic. In: *Handbook of Logic in Computer Science*, New York, Oxford University Press, 39–123.

Plotkin, G. D. (1977). LCF considered as a programming language. *Theoretical Computer Science* **5** (3) 223–255.

Rose, K. H. (1996). *Explicit Substitution: Tutorial & Survey*. Computer Science Department.

Scott, D. (1976). Data types as lattices. *Siam Journal on Computing* **5** (3) 522–587.

Scott, D. S. (1993). A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science* **121** (1) 411–440.

Seely, R. A. (1987). Modelling computations: A 2-categorical framework. In: *Proceedings of the 2nd Annual IEEE Symposium on Logic in Computer Science*, IEEE Computer Society, 65–71.

Sørensen, M. H. and Urzyczyn, P. (2006). *Lectures on the Curry-Howard Isomorphism*, vol. 149, Amsterdam, Elsevier.

Winskel, G. (1993). *The Formal Semantics of Programming Languages: An Introduction*, Cambridge, MA, MIT Press.

Cite this article: Yamada N and Abramsky S (2020). Dynamic game semantics. *Mathematical Structures in Computer Science* **30**, 892–951. https://doi.org/10.1017/S0960129520000250