# Two-Level Game Semantics, Intersection Types, and Recursion Schemes

C.-H. Luke Ong[1] and Takeshi Tsukada[2,3]

[1] Department of Computer Science, University of Oxford
[2] Graduate School of Information Science, Tohoku University
[3] JSPS Research Fellow

**Abstract.** We introduce a new cartesian closed category of *two-level* arenas and innocent strategies to model intersection types that are refinements of simple types. Intuitively a property (respectively computation) on the upper level refines that on the lower level. We prove *Subject Expansion*—any lower-level computation is closely and canonically tracked by the upper-level computation that lies over it—which is a measure of the robustness of the two-level semantics. The game semantics of the type system is *fully complete*: every winning strategy is the denotation of some derivation. To demonstrate the relevance of the game model, we use it to construct new semantic proofs of non-trivial algorithmic results in higher-order model checking.

## 1 Introduction

The recent development of *higher-order model checking*—the model checking of trees generated by higher-order recursion schemes (HORS) against (alternating parity) tree automata—has benefitted much from ideas and methods in semantics. Ong's proof [1] of the decidability of the monadic second-order (MSO) theories of trees generated by HORS was based on game semantics [2]. Using HORS as an intermediate model of higher-order computation, Kobayashi [3] showed that safety properties of functional programs can be verified by reduction to the model checking of HORS against trivial automata (i.e. Büchi tree automata with a trivial acceptance condition). His model checking algorithm is based on an intersection-type-theoretic characterisation of the trivial automata acceptance problem of trees generated by HORS.[4] This type-theoretic approach was subsequently refined and extended to characterise alternating parity tree automata [5], thus yielding a new proof of Ong's MSO decidability result. (Several other proofs of the result are now known. Hague et al. [6] developed a new hierarchy of *collapsible pushdown automata* and proved that they are equi-expressive with HORS for generating trees. Salvati and Walukiewicz's proof [7] uses a Krivine machine formulation of the operational semantics of HORS.)

This paper was motivated by a desire to understand the connexions between the game-semantic proof [1] and the type-based proof [3,5] of the MSO decidability result.

[4] Independently, Salvati [4] has proposed essentially the same intersection type system for the simply-typed $\lambda$-calculus without recursion from a different perspective.

As a first step in clarifying their relationship, we construct a *two-level game semantics* to model intersection types that are refinements of simple types. Given a set $Q$ of colours (modelling the states of an automaton), we introduce a cartesian closed category whose objects are triples $(A, U, K)$ called *two-level arenas*, where $A$ is a *$Q$-coloured arena* (modelling intersection types), $K$ is a standard arena (modelling simple types), and $U$ is a colour-forgetting function from $A$-moves to $K$-moves which preserves the justification relation. A map of the category from $(A, U, K)$ to $(A', U', K')$ is a pair of innocent and colour-reflecting strategies, $\sigma : A \longrightarrow A'$ and $\bar{\sigma} : K \longrightarrow K'$, such that the induced colour-forgetting function maps plays of $\sigma$ to plays of $\bar{\sigma}$. This captures the intuition that the upper-level computation represented by $\sigma$ refines (or is more constrained than) the lower-level computation represented by $\bar{\sigma}$, a semantic framework reminiscent of two-level denotational semantics in abstract interpretation as studied by Nielson [8]. Given triples $\mathcal{A}_1 = (A_1, U_1, K)$ and $\mathcal{A}_2 = (A_2, U_2, K)$ that have the same base arena $K$, their *intersection* $\mathcal{A}_1 \wedge \mathcal{A}_2$ is $(A_1 \times A_2, [U_1, U_2], K)$. Building on the two-level game semantics, we make the following contributions.

(i) How good is the two-level game semantics? Our answer is *Subject Expansion* (Theorem 3), which says intuitively that any computation (reduction) on the lower level can be closely and canonically tracked by the higher-level computation that lies over it. Subject Expansion clarifies the relationship between the two levels; we think it is an important measure of the robustness (and, as we shall see, the reason for the usefulness) of the game semantics.

(ii) We put the two-level game model to use by modelling Kobayashi's intersection type system [3]. Derivations of intersection-type judgements, which we represent by the terms of a new proof calculus, are interpreted by *winning strategies* i.e. compact and total (in addition to innocent and colour-reflecting). We prove that the interpretation is *fully complete* (Theorem 5): every winning strategy is the denotation of some derivation.

(iii) Finally, to demonstrate the usefulness and relevance of the two-level game semantics, we apply it to construct new semantic proofs of three non-trivial *algorithmic* results in higher-order model checking: (a) characterisation of trivial automata acceptance (existence of an accepting run-tree) by a notion of typability [3], (b) minimality of the type environment induced by traversal tree [1], and (c) completeness of GTRecS, a game-semantics based practical algorithm for model checking HORS against trivial automata [9].

*Outline* We introduce (coloured) arenas, innocent strategies and related game-semantic notions in Section 3. In Section 4 we present two-level games, culminating in the Subject Expansion Theorem. In Section 5 we construct a fully complete two-level game model of Kobayashi's intersection type system. Finally, Section 6 applies the game model to reason about algorithmic problems in higher-order model checking.

## 2 Two Structures of Intersection Type System

This section presents the intuitions behind the two levels. We explain that two different structures are naturally extracted from a derivation in an intersection type system. Here we use term representation for explanation. Two-level game semantics will be developed in the following sections based on this idea.

$$\dfrac{\dfrac{\dfrac{g:\tau}{g : p_1\wedge p_3 \to q_1} \quad \dfrac{\dfrac{x:\sigma}{x:p_1}\quad\dfrac{x:\sigma}{x:p_3}}{x:p_1\wedge p_3}}{g\;x:q_1} \qquad \dfrac{\dfrac{g:\tau}{g:p_2\to q_2}\quad \dfrac{x:\sigma}{x:p_2}}{g\;x:q_2}}{g\;x : q_1\wedge q_2}$$

**Fig. 1.** A type derivation of the intersection type system. Here type environment $\mathit{\Gamma} = \{g : ((p_1 \wedge p_3) \to q_1) \wedge (p_2 \to q_2),\ x : p_1 \wedge p_2 \wedge p_3\}$ is omitted.

$$\dfrac{\dfrac{\dfrac{g:\tau'}{\mathsf{p}_1(g) : p_1\times p_3 \to q_1} \quad \dfrac{\dfrac{x:\sigma'}{\mathsf{p}_1(x):p_1}\quad\dfrac{x:\sigma}{\mathsf{p}_3(x):p_3}}{\langle \mathsf{p}_1(x), \mathsf{p}_3(x)\rangle : p_1\times p_3}}{\mathsf{p}_1(g)\;\langle \mathsf{p}_1(x), \mathsf{p}_3(x)\rangle:q_1} \qquad \dfrac{\dfrac{g:\tau}{\mathsf{p}_2(g):p_2\to q_2}\quad \dfrac{x:\sigma}{\mathsf{p}_2(x):p_2}}{\mathsf{p}_2(g)\;\mathsf{p}_2(x):q_2}}{\langle \mathsf{p}_1(g)\;\langle \mathsf{p}_1(x), \mathsf{p}_3(x)\rangle,\ \mathsf{p}_2(g)\;\mathsf{p}_2(x)\rangle : q_1\times q_2}$$

**Fig. 2.** A type derivation of the product type system, which corresponds to Fig. 1. Here $\mathit{\Gamma}' = \{g : ((p_1 \times p_3) \to q_1) \times (p_2 \to q_2),\ x : p_1 \times p_2 \times p_3\}$ is omitted.

The intersection type constructor $\wedge$ of an intersection type system is characterised by the following typing rules.[5]

$$\dfrac{\mathit{\Gamma} \vdash t : \tau_1 \qquad \mathit{\Gamma} \vdash t : \tau_2}{\mathit{\Gamma} \vdash t : \tau_1 \wedge \tau_2} \qquad \dfrac{\mathit{\Gamma} \vdash t : \tau_1 \wedge \tau_2}{\mathit{\Gamma} \vdash t : \tau_1} \qquad \dfrac{\mathit{\Gamma} \vdash t : \tau_1 \wedge \tau_2}{\mathit{\Gamma} \vdash t : \tau_2}$$

At first glance, they resemble the rules for products. Let $\langle t_1, t_2\rangle$ be a pair of $t_1$ and $t_2$ and $\mathsf{p}_i$ be the projection to the $i$th element (for $i \in \{1, 2\}$).

$$\dfrac{\mathit{\Gamma} \vdash t_1 : \tau_1 \qquad \mathit{\Gamma} \vdash t_2 : \tau_2}{\mathit{\Gamma} \vdash \langle t_1, t_2\rangle : \tau_1 \times \tau_2} \qquad \dfrac{\mathit{\Gamma} \vdash t : \tau_1 \times \tau_2}{\mathit{\Gamma} \vdash \mathsf{p}_1(t) : \tau_1} \qquad \dfrac{\mathit{\Gamma} \vdash t : \tau_1 \times \tau_2}{\mathit{\Gamma} \vdash \mathsf{p}_2(t) : \tau_2}$$

When we ignore terms and replace $\times$ by $\wedge$, the rules in the two groups coincide. In fact, they are so similar that a derivation of the intersection type system can be transformed to a derivation of the product type system by replacing $\wedge$ by $\times$ and adjusting terms to the rules for product. See Figures 1 and 2 for example. This is the first structure behind an intersection-type derivation, which we call the *upper-level structure.*

However the upper-level structure alone does not capture all features of the intersection type system: specifically some derivations of the product type system have no corresponding derivation in the intersection type system. For example, while the type judgement $x : p_1, y : p_2 \vdash \langle x, y\rangle : p_1 \times p_2$ is derivable, no term inhabits the judgement $x : p_1, y : p_2 \vdash ? : p_1 \wedge p_2$.

Terms in the rules explain this gap. We call them *lower-level structures.* To construct a term of type $\tau_1 \times \tau_2$, it suffices to find *any* two terms $t_1$ of type $\tau_1$ and $t_2$ of type $\tau_2$. However to construct a term of type $\tau_1 \wedge \tau_2$, we need to find a term $t$ that has both type $\tau_1$ and type $\tau_2$. Thus a product type derivation has a corresponding intersection

[5] In the type system in Section 5, these rules are no longer to be independent rules, but a similar argument stands.

type derivation only if for all pairs $\langle t_1, t_2\rangle$ appearing at the derivation, the respective structures of $t_1$ and $t_2$ are "coherent".

For example, let us examine the derivation in Figure 2, which contains two pair constructors. One appears at $\langle \mathsf{p}_1(x), \mathsf{p}_3(x)\rangle : p_1 \times p_3$. Here the left argument $\mathsf{p}_1(x) : p_1$ and the right argument $\mathsf{p}_3(x) : p_3$ are "coherent" in the sense that they are the same except for details such as types and indexes of projections. In other words, by forgetting such details, $\mathsf{p}_1(x) : p_1$ and $\mathsf{p}_3(x) : p_3$ become the same term $x$. The other pair appears at the root and the "forgetful" map maps both the left and right arguments to $g\ x$.

This interpretation decomposes an intersection type derivation into three components: a derivation in the simple type system with product (the upper-level structure), a term (the lower-level structure) and a "forgetful" map from the upper-level structure to the lower-level structure. Since recursion schemes are simply typed, we can assume a term to also be simply typed for our purpose. Hence the resulting two-level structure consists of two derivations in the simple type system with a map on nodes from one to the other.

## 3 Coloured Arena Games

This section defines coloured arenas, innocent strategies and related notions. We first introduce some basic notions in game semastics [2]. For sets $A$ and $B$, we write $A + B$ for the disjoint union and $A \times B$ for the Cartesian product.

**Definition 1 (Coloured Arena).** For a set $Q$ of symbols, a *$Q$-coloured arena $A$* is a quadruple $(M_A, \vdash_A, \lambda_A, c_A)$, where

- $M_A$ is a set of *moves*,
- $\vdash_A \subseteq M_A + (M_A \times M_A)$ is a *justification relation*,
- $\lambda_A : M_A \to \{P, O\}$, and
- $c_A : M_A \to Q$ is a *colouring*.

We write $\vdash_A m$ for $m \in (\vdash_A)$ and $m \vdash_A m'$ for $(m, m') \in (\vdash_A)$. The justification relation must satisfy the following conditions:

- For each $m \in M_A$, either $\vdash_A m$ or $m' \vdash_A m$ for a unique move $m' \in M_A$.
- If $\vdash_A m$, then $\lambda_A(m) = O$. If $m \vdash_A m'$, then $\lambda_A(m) \neq \lambda(m')$.

Fer a $Q$-coloured arena $A$, the set $\mathit{Init}_A \subseteq M_A$ of *initial moves* of $A$ is $\{m \in M_A \mid \vdash_A m\}$. A move $m \in M_A$ is called an ***O-move*** if $\lambda_A(m) = O$ and a ***P-move*** if $\lambda_A(m) = P$.

A *justified sequence* of a $Q$-coloured arena $A$ is a sequence of moves such that each element except the first is equipped with a pointer to some previous move. We call the pointer a *justification pointer*. For a justified sequence $s$ and moves $m$ and $m'$ in $s$, we say $m'$ is *hereditary justified* by $m$ if there exists a sequence of moves $m_0, m_1, \dots, m_n$ in $s$ that starting from $m$ and ending with $m'$ such that $m_i$ is justified by $m_{i-1}$ ($1 \leq i \leq n$).

A *well-formed sequence* over $A$ is a justified sequence $s = m_0 \cdot m_1 \cdot \cdots \cdot m_n$ that has the following properties:

**Well-openness** $\vdash_A m_0$,
**Alternation** For all $i < n$, $\lambda_A(m_i) \neq \lambda_A(m_{i+1})$, and
**Justification** If $m_i$ points $m_j$ ($j < i$), then $m_j \vdash_A m_i$.

For well-formed sequences $s$ and $s'$, we say $s$ is a *prefix* of $s'$ if the underlying sequence of moves of $s$ is a prefix of that of $s'$ and their justification pointers coincide.

For a well-formed sequence $s$, its *P-view* $\ulcorner s \urcorner$ and O-view $\llcorner s \lrcorner$ are defined inductively as follows:

$$\begin{aligned}
\ulcorner \epsilon \urcorner &= \epsilon \\
\ulcorner m \urcorner &= m \\
\ulcorner s \cdot m \urcorner &= \ulcorner s \urcorner \cdot m \qquad (\text{if } \lambda(m) = P) \\
\ulcorner s \cdot m \cdot s' \cdot m' \urcorner &= \ulcorner s \urcorner \cdot m \cdot m' \qquad (\text{if } \lambda(m) = O \text{ and } m' \text{ is justified by } m)
\end{aligned}$$

$$\begin{aligned}
\llcorner \epsilon \lrcorner &= \epsilon \\
\llcorner m \lrcorner &= m \\
\llcorner s \cdot m \lrcorner &= \llcorner s \lrcorner \cdot m \qquad (\text{if } \lambda(m) = O) \\
\llcorner s \cdot m \cdot s' \cdot m' \lrcorner &= \llcorner s \lrcorner \cdot m \cdot m' \qquad (\text{if } \lambda(m) = P \text{ and } m' \text{ is justified by } m)
\end{aligned}$$

A *play* of an arena $A$ is a well-formed sequence $s$ satisfying the following conditions:

**Visibility** For every prefix $s' \cdot m \preceq s$ ending with a P-move (resp. an O-move that is not initial), $m$ is justified by a move in $\ulcorner s' \urcorner$ (resp. $\llcorner s' \lrcorner$).

A *P-strategy* (or a *strategy*) $\sigma$ of an arena $A$ is a prefix-closed subset of plays of $A$ satisfying the following conditions:

**Determinacy** If $s \cdot m \in \sigma$ and $s \cdot m' \in \sigma$ for P-moves $m$ and $m'$ then $s \cdot m = s \cdot m'$.
**Contingent Completeness** If $s \in \sigma$, $m$ is an O-move and $s \cdot m$ is a justified sequence, then $s \cdot m \in \sigma$.
**Colour Reflecting** Only the opponent can change the colour, i.e. for every P-move $m_1^P$ and O-move $m_2^O$, if $s \cdot m_1^O \cdot m_2^P \in \sigma$, then $c(m_1^O) = c(m_2^P)$.

For arenas $A_1$, $A_2$ and $A_3$, an *interaction sequence* is a play of $(A_1 \Rightarrow A_2) \Rightarrow A_3$. We write $\mathrm{Int}(A_1, A_2, A_3)$ for the set of all interaction sequences. For an interaction sequence $s \in \mathrm{Int}(A_1, A_2, A_3)$, a component of $s$ is either $(A_2, A_3)$ or $(A_1, A_2, b)$ where $b$ is an initial move occurring in $s$. The projection $s \upharpoonright_X$ of an interaction sequence $s$ into a component $X$ is defined by:

- $s \upharpoonright_{(A_2, A_3)}$ is a subsequence of $s$ consisting of all $A_2$ moves and $A_3$ moves in $s$.
- $s \upharpoonright_{(A_1, A_2, b)}$ is a subsequence of $s$ consisting of all moves that are hereditary justified by $b$.

The projection into $(A_1, A_3)$ is defined by a similar way: $s \upharpoonright_{(A_1, A_3)}$ is a subsequence of $s$ consisting of all $A_1$ moves and $A_3$ moves, in which initial $A_1$ moves are justified by a (unique) initial $A_3$ move occurring in $s$. For an interaction sequence $s \in$

$\mathrm{Int}(A_1, A_2, A_3)$, $s \upharpoonright_{(A_2,A_3)}$ is a play of $A_2 \Rightarrow A_3$, $s \upharpoonright_{(A_1,A_2,b)}$ is of $A_1 \Rightarrow A_2$ (for every initial $A_2$ move $b$ occurring in $s$) and $s \upharpoonright_{(A_1,A_3)}$ is of $A_1 \Rightarrow A_3$.

For strategies (or just sets of plays) $\sigma_1 : A_1 \Rightarrow A_2$ and $\sigma_2 : A_2 \Rightarrow A_3$, the set $\mathbf{Int}(\sigma_1, \sigma_2) \subseteq \mathrm{Int}(A_1, A_2, A_3)$ of interaction sequences that are consistent with $\sigma_1$ and $\sigma_2$ is give by:

$$\begin{aligned}\mathbf{Int}(\sigma_1, \sigma_2) = \{s \in \mathrm{Int}(A_1, A_2, A_3) \mid\ & s \upharpoonright_{(A_2,A_3)} \in \sigma_2 \text{ and} \\ & \text{for every initial } A_2 \text{ move } b,\ s \upharpoonright_{(A_1,A_2,b)} \in \sigma_1\}.\end{aligned}$$

The *composition* $(\sigma_1; \sigma_2) : A_1 \Rightarrow A_3$ is defined as $\{s \upharpoonright_{(A_1,A_3)} \mid s \in \mathbf{Int}(\sigma_1, \sigma_2)\}$. For each $s \in (\sigma_1; \sigma_2)$, the *uncovering* of $s$ is the *minimum* interaction sequence $u \in \mathbf{Int}(\sigma_1, \sigma_2)$ (with respect to the prefix ordering) such that $s = u \upharpoonright_{(A_1,A_3)}$.[6]

A strategy $\sigma$ is *innocent* if for every pair of plays $s \cdot m,\ s' \cdot m' \in \sigma$ ending with P-moves $m$ and $m'$, $\ulcorner s \urcorner = \ulcorner s' \urcorner$ implies $\ulcorner s \cdot m \urcorner = \ulcorner s' \cdot m' \urcorner$. For an innocent strategy $\sigma$ of $A$, the *view-function* $f_\sigma$ of $\sigma$ is the partial function on P-views of $A$, which maps a P-view $p \in \sigma$ that ends with an O-move to a unique P-view $p \cdot m \in \sigma$.

We say an innocent strategy $\sigma$ is *winning* just if the following holds:

**Compact** The domain $\mathrm{dom}(f_\sigma)$ of the view function of $\sigma$ is a finite set.
**Total** If $s \cdot m \in \sigma$ for an O-move $m$, then $s \cdot m \cdot m' \in \sigma$ for some P-move $m'$.

We define three constructions of arenas: a binary product, an indexed product and a function space.

**Product** For $Q$-coloured arenas $A$ and $B$, we define $A \times B$ by:

- $M_{A \times B} = M_A + M_B$,
- $\vdash_{A \times B} m \iff \vdash_A m$ or $\vdash_B m$,
- $m \vdash_{A \times B} m' \iff m \vdash_A m'$ or $m \vdash_B m'$,
- $\lambda_{A \times B}(m) = \begin{cases} \lambda_A(m) & (\text{if } m \in M_A) \\ \lambda_B(m) & (\text{if } m \in M_B), \end{cases}$
- $c_{A \times B}(m) = \begin{cases} c_A(m) & (\text{if } m \in M_A) \\ c_B(m) & (\text{if } m \in M_B). \end{cases}$

For an indexed set $\{A_i\}_{i \in I}$ of $Q$-coloured arenas, their product $\prod_{i \in I} A_i$ is defined similarly.

**Function Space** For $Q$-coloured arenas $A$ and $B$, we define $A \Rightarrow B$ by:

- $M_{A \Rightarrow B} = M_A \times \mathit{Init}_B + M_B$,
- $\vdash_{A \Rightarrow B} m \iff \vdash_B m$,
- $m \vdash_{A \Rightarrow B} m' \iff$
  - $m \vdash_B m'$, or
  - $\vdash_B m$ and $m' = (m'_A, m)$ and $\vdash_A m_A$, or
  - $m = (m_A, m_B)$ and $m' = (m'_A, m_B)$ and $m_A \vdash_A m'_A$,

[6] The definition here differs from the one in [2]: in [2], the uncovering is the *maximum* interaction sequence.

– $\lambda_{A\Rightarrow B}(m) = \begin{cases} \lambda_A(m_A) & \text{(if } m = (m_A, m_B) \in M_A \times \mathit{Init}_B\text{)} \\ \lambda_B(m) & \text{(if } m \in M_B\text{),} \end{cases}$
– $c_{A\times B}(m) = \begin{cases} c_A(m) & \text{(if } m = (m_A, m_B) \in M_A \times \mathit{Init}_B\text{)} \\ c_B(m) & \text{(if } m \in M_B\text{).} \end{cases}$

We define a category whose objects are $Q$-coloured arenas; maps from $A$ to $B$ are innocent strategies of the arena $A \Rightarrow B$. The category is cartesian closed, and is thus a model of the simply-typed lambda calculus with (indexed) products.

**Theorem 1.** *For every set $Q$, the category of $Q$-coloured arenas and innocent strategies is certesian closed with the product $A \times B$ and function space $A \Rightarrow B$.*

# 4 Two-level Game Semantics

## 4.1 Category of Two-Level Arenas and Innocent Strategies

**Definition 2 (Two-Level Arenas).** An *two-level arena* based on $Q$ is a triple $\mathcal{A} = (A, U, K)$, where $A$ is a $Q$-colored arena, $K$ is a $\{o\}$-colored arena (i.e. an ordinary arena, which we call the *base arena* of $\mathcal{A}$) and $U$ is a map from $M_A$ to $M_K$ that satisfies: (i) $\lambda_A(m) = \lambda_K(U(m))$ (ii) If $m \vdash_A m'$ then $U(m) \vdash_K U(m')$; and if $\vdash_A m$ then $\vdash_K U(m)$.

For a justified sequence $s = m_1 \cdot m_2 \cdots m_k$, we write $U(s)$ to mean the justified sequence $U(m_1) \cdot U(m_2) \cdots U(m_k)$ whose justification pointers are induced by those of $s$.

**Lemma 1.** *Let $\mathcal{A} = (A, U, K)$ be a two-level arena and $s$ be a play of $A$. Then $U(s)$ is a play of $K$.*

*Proof.* Easy. □

For a strategy $\sigma$ of $A$, $U(\sigma) := \{U(s) \mid s \in \sigma\}$ is a set of plays of $K$, which is not necessarily a strategy, since $U(s)$ may not satisfy determinacy. (Recall that some upper-level structure has no corresponding lower-level structure.)

**Definition 3 (Strategy of Two-Level Arena Games).** A *strategy* of a two-level arena $(A, U, K)$ is a pair $(\sigma, \bar{\sigma})$ of strategies of $A$ and $K$ respectively such that $U(\sigma) \subseteq \bar{\sigma}$.

**Lemma 2.** *Let $\mathcal{A}_i = (A_i, U_i, K_i)$ be a two-level arena for $i = 1, 2, 3$. Then for every interaction sequence $s$ of $(A_1, A_2, A_3)$*

*(i)* $U(\ulcorner s \urcorner) = \ulcorner U(s) \urcorner$
*(ii) for any component $C$ of $s$, $U(s \upharpoonright_C) = U(s) \upharpoonright_{U(C)}$ (here subscripts of $U$ should be chosen appropriately).*

*In the above, the forgetful function $U$ (whose definition we omit) has a natural extension to a forgetful function on components.*

*Proof.* Because $U$ does not change the structure of justification pointers. □

**Definition 4 (Innocent Strategies).** A strategy $(\sigma, \bar{\sigma})$ of $\mathcal{A} = (A, U, K)$ is *innocent* just if $\sigma$ and $\bar{\sigma}$ are innocent as strategies of $A$ and $K$ respectively.

Let $\mathcal{A}_i = (A_i, U_i, K_i)$ where $i = 1, 2$ be two-level arenas. We define product, function space and intersection constructions as follows.

**Product** $\mathcal{A}_1 \times \mathcal{A}_2 := (A_1 \times A_2, U, K_1 \times K_2)$, where $U : (M_{A_1} + M_{A_2}) \to (M_{K_1} + M_{K_2})$ is defined as $U_1 + U_2$.

**Function Space** $\mathcal{A}_1 \Rightarrow \mathcal{A}_2 := (A_1 \Rightarrow A_2, U, K_1 \Rightarrow K_2)$, where $U : ((M_{A_1} \times \mathit{Init}_{A_2}) + M_{A_2}) \to ((M_{K_1} \times \mathit{Init}_{K_2}) + M_{K_2})$ is defined as $U_1 \times U_2 + U_2$.

We can now define a category whose objects are two-level arenas, and maps $\mathcal{A}_1 \to \mathcal{A}_2$ are innocent strategies of $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$. The composite of $(\sigma_1, \bar{\sigma}_1) : \mathcal{A}_1 \Rightarrow \mathcal{A}_2$ and $(\sigma_2, \bar{\sigma}_2) : \mathcal{A}_2 \Rightarrow \mathcal{A}_3$ is defined as $(\sigma_1; \sigma_2,\ \bar{\sigma}_1; \bar{\sigma}_2) : \mathcal{A}_1 \Rightarrow \mathcal{A}_3$.

We first show that the composition of strategies is well-defined (Lemma 4).

**Lemma 3.** *Let $\mathcal{A}_i = (A_i, U_i, K_i)$ be a two-level arena for $i = 1, 2, 3$, $\sigma_1 : A_1 \Rightarrow A_2$ and $\sigma_2 : A_2 \Rightarrow A_3$ be strategies of coloured arenas. Then $U_{\mathcal{A}_1 \Rightarrow \mathcal{A}_3}(\sigma_1; \sigma_2) \subseteq U_{\mathcal{A}_1 \Rightarrow \mathcal{A}_2}(\sigma_1); U_{\mathcal{A}_2 \Rightarrow \mathcal{A}_3}(\sigma_2)$.*

*Proof.* Let $\bar{s} \in U(\sigma_1; \sigma_2)$. By definition, we have $s \in (\sigma_1; \sigma_2)$ such that $U(s) = \bar{s}$. Let $u \in \mathbf{Int}(\sigma_1, \sigma_2)$ be the uncovering of $s$, i.e. an interaction sequence that satisfies the following properties.

(i) $u \upharpoonright_{(A_1, A_3)} = s$.
(ii) $u \upharpoonright_{(A_2, A_3)} \in \sigma_2$.
(iii) For any initial $A_2$ move $b$ in $u$, $u_A \upharpoonright_{(A_1, A_2, b)} \in \sigma_1$.

Let $U_{(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)}$ be the forgetful map on interaction sequences and $\bar{u} = U_{(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)}(u)$. The following argument shows that $\bar{u} \in \mathbf{Int}(U(\sigma_1), U(\sigma_2))$ (here we use Lemma 2. Subscripts of $U$ should be chosen appropriately).

(i) $\bar{u} \upharpoonright_{(K_1, K_3)} = \bar{s}$, since

$$\bar{u} \upharpoonright_{(K_1, K_3)} = U(u) \upharpoonright_{U(\,(A_1, A_3)\,)} = U(u \upharpoonright_{(A_1, A_3)}) = U(s) = \bar{s}.$$

(ii) $\bar{u} \upharpoonright_{(K_2, K_3)} \in U\sigma_2$, since

$$\bar{u} \upharpoonright_{(K_2, K_3)} = U(u) \upharpoonright_{U(\,(A_2, A_3)\,)} = U(u \upharpoonright_{(A_2, A_3)}) \in U(\sigma_2).$$

(iii) Let $(K_1, K_2, \bar{b})$ be a component of $\bar{u}$. There is a component $(A_1, A_2, b)$ such that $U(\,(A_1, A_2, b)\,) = (K_1, K_2, \bar{b})$. Then

$$\bar{u} \upharpoonright_{(K_1, K_2, \bar{b})} = U(u) \upharpoonright_{U(\,(A_1, A_2, b)\,)} = U(u \upharpoonright_{(A_1, A_2, b)}) \in U(\sigma_1).$$

Therefore $\bar{s} = \bar{u} \upharpoonright_{(K_1, K_3)} \in (U(\sigma_1); U(\sigma_2))$. □

**Lemma 4.** *Let $\mathcal{A}_i = (A_i, U_i, K_i)$ be a two-level arena for $i = 1, 2, 3$, $(\sigma_1, \bar{\sigma_1}) : \mathcal{A}_1 \Rightarrow \mathcal{A}_2$ and $(\sigma_2, \bar{\sigma_2}) : \mathcal{A}_2 \Rightarrow \mathcal{A}_3$ be strategies. Then $(\sigma_1, \bar{\sigma}_1); (\sigma_2, \bar{\sigma}_2) = (\sigma_1; \sigma_2,\ \bar{\sigma}_1; \bar{\sigma}_2)$ is a strategy of $\mathcal{A}_1 \Rightarrow \mathcal{A}_3$.*

*Proof.* Obviously, $\sigma_1;\sigma_2$ is a strategy of $A_1 \Rightarrow A_3$ and $\bar{\sigma}_1;\bar{\sigma}_2$ be a strategy of $K_1 \Rightarrow K_3$. So it suffices to show that $U_{\mathcal{A}_1\Rightarrow\mathcal{A}_3}(\sigma_1;\sigma_2) \subseteq (\bar{\sigma}_1;\bar{\sigma}_2)$. Since $(\sigma_1, \bar{\sigma}_1)$ is a strategy of $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$, we have $U_{\mathcal{A}_1\Rightarrow\mathcal{A}_2}(\sigma_1) \subseteq \bar{\sigma}_1$. Similarly, $U_{\mathcal{A}_2\Rightarrow\mathcal{A}_3}(\sigma_2) \subseteq \bar{\sigma}_2$. By Lemma 3 and monotonicity of composition, we have

$$U_{\mathcal{A}_1\Rightarrow\mathcal{A}_3}(\sigma_1;\sigma_2) \subseteq (U_{\mathcal{A}_1\Rightarrow\mathcal{A}_2}(\sigma_1); U_{\mathcal{A}_2\Rightarrow\mathcal{A}_3}(\sigma_2)) \subseteq (\bar{\sigma}_1;\bar{\sigma}_2)$$

as required. □

It is easy to see that innocence is preserved by composition, since innocence and composition are defined component-wise. Therefore composition of the category of two-level arenas and innocent strategies is well-defined.

**Theorem 2.** *The category of two-level arenas and innocent strategies is cartesian closed.*

*Proof.* Trivial. □

If two two-level arenas share the same base arena, then we can construct their *intersection*.

**Definition 5 (Intersection of Two-Level Arenas).** Let $\mathcal{A}_i = (A_i, U_i, K)$ for $i = 1, 2$ be two-level arenas that share the same base arena $K$. Their *intersection* $\mathcal{A}_1 \wedge \mathcal{A}_2$ is defined as $(A_1 \times A_2, U, K)$, where $U : (M_{A_1} + M_{A_2}) \to M_K$ is defined as $[U_1, U_2]$.

For every base arena $K$, we define $\top_K$ as the two-level arena $(\top, \emptyset, K)$, where $\top$ is the empty arena, which is the terminal object in the category of $Q$-coloured arenas.

For a $Q$-coloured arena $A$, we write $!_A$ for the unique strategy of $A \Rightarrow \top$. For a two-level arena $\mathcal{A} = (A, U, K)$, we define $!_{\mathcal{A}} : \mathcal{A} \Rightarrow \top_K$ as $(!_A, \mathrm{id}_K)$.

**Lemma 5.** *Let $\mathcal{A}_1 = (A_1, U_1, K)$ and $\mathcal{A}_2 = (A_2, U_2, K)$ be two-level arenas that share the same base arena $K$. The arena $\mathcal{A}_1 \wedge \mathcal{A}_2$ is the pullback of $\mathcal{A}_1$ and $\mathcal{A}_2$, i.e. there are innocent strategies $p_1$ and $p_2$ of two-level arenas that make the following diagram a pullback square.*

$$\begin{tikzcd}
\mathcal{A}_1 \wedge \mathcal{A}_2 \arrow[r, "p_1"] \arrow[d, "p_2"'] & \mathcal{A}_1 \arrow[d, "!_{\mathcal{A}_1}"] \\
\mathcal{A}_2 \arrow[r, "!_{\mathcal{A}_2}"'] & \top_K
\end{tikzcd}$$

*Proof.* Taking $p_1 = (\pi_1, \mathrm{id}_K)$ and $p_2 = (\pi_2, \mathrm{id}_K)$. □

### 4.2 Subject Expansion

**Theorem 3 (Subject Expansion).** *Let $\mathcal{A}_i = (A_i, U_i, K_i)$ be a two-level arena for $i = 1, 2$ and $K$ be a base arena. If*

$$\begin{tikzcd}
\mathcal{A}_1 \arrow[r, "(\sigma,\bar{\sigma})"] & \mathcal{A}_2
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
K_1 \arrow[rr, "\bar{\sigma}"] \arrow[dr, "\bar{\sigma}_1"'] & \circlearrowleft & K_2 \\
& K \arrow[ur, "\bar{\sigma}_2"'] &
\end{tikzcd}$$

*(in two-level arenas)* *(in base arenas)*

*then there are a two-level arena $\mathcal{A}$ whose underlying kind arena is $K$ and strategies $\sigma_1 : A_1 \to A$ and $\sigma_2 : A \to A_2$ such that*

$$\begin{tikzcd}
\mathcal{A}_1 \arrow[rr, "(\sigma,\bar{\sigma})"] \arrow[dr, "(\sigma_1,\bar{\sigma}_1)"'] & {} \arrow[d, phantom, "\circlearrowleft"] & \mathcal{A}_2 \\
& \mathcal{A} \arrow[ur, "(\sigma_2,\bar{\sigma}_2)"'] &
\end{tikzcd}$$

*Moreover, there is a canonical triple $(\sigma_1, \mathcal{A}, \sigma_2)$: for every triple $(\sigma_1', \mathcal{A}', \sigma_2')$ that satisfies $\sigma_1'; \sigma_2' = \sigma$, there exists a mapping $\varphi$ from moves of $\mathcal{A}$ to moves of $\mathcal{A}'$ such that $[\mathrm{id}_{\mathcal{A}_1}, \varphi](\sigma_1) \subseteq \sigma_1'$ and $[\varphi, \mathrm{id}_{\mathcal{A}_2}](\sigma_2) \subseteq \sigma_2'$.*

The key observation of the proof is that *innocent* strategies can (mostly) be reconstructed from their interaction sequences. Let $\sigma_1$ and $\sigma_2$ be innocent strategies of $A \Rightarrow B$ and $B \Rightarrow C$. Observe that since $\sigma_2$ is innocent, it is determined by the set of P-views in $\sigma_2$. Using $\mathbf{Int}(\sigma_1, \sigma_2)$ we define a set of P-views by $\varphi_2' = \{\ulcorner s \upharpoonright_{B \Rightarrow C} \urcorner \mid s \in \mathbf{Int}(\sigma_1, \sigma_2)\}$. Then $\varphi_2'$ can be regarded as a view function, which determines an innocent strategy $\sigma_2'$. Similarly, we can construct a view function $\varphi_1'$ and an innocent strategy $\sigma_1'$. Then the resulting strategies $\sigma_1'$ and $\sigma_2'$ are respective *under-approximations* of $\sigma_1$ and $\sigma_2$ i.e. $\sigma_1' \subseteq \sigma_1$ and $\sigma_2' \subseteq \sigma_2$ and $\sigma_1'; \sigma_2' = \sigma_1; \sigma_2$.

Now the goal is to construct "interaction sequences" of $\sigma_1$ and $\sigma_2$. There are two conditions that the set $\mathbf{Int}(\sigma_1, \sigma_2)$ of all interaction sequences must satisfy.

- $u \in \mathbf{Int}(\sigma_1, \sigma_2)$ implies $U(u) \in \mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2)$.
- $u \in \mathbf{Int}(\sigma_1, \sigma_2)$ implies $u \upharpoonright_{\mathcal{A}_1 \Rightarrow \mathcal{A}_2} \in \sigma$.

These requirements give basic patterns of interaction sequences. Let $s \in \sigma$ and $\bar{u} \in \mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2)$ and $\bar{u} = \bar{m}_1 \cdot \bar{m}_2 \cdots \bar{m}_k$, such that $U(s) = \bar{u} \upharpoonright_{K_1 \Rightarrow K_2}$. Then a justified sequence of pairs of moves of base arenas and $Q$-coloured arenas

$$\begin{bmatrix} \bar{m}_1 \\ m_1 \end{bmatrix} \cdot \begin{bmatrix} \bar{m}_2 \\ m_2 \end{bmatrix} \cdots \begin{bmatrix} \bar{m}_k \\ m_k \end{bmatrix}$$

is called *annotated interaction sequences* generated by $s$ and $u$ if (i) $\bar{m}_i \in K_1 \cup K_2$ implies $U(m_i) = \bar{m}_i$, (ii) $(m_1 \cdot m_2 \ldots m_k) \upharpoonright_{A_1 \Rightarrow A_2} = s$, (iii) $\bar{m}_i \in K$ implies $m_i = \star$. An interaction sequence over $\sigma_1$ and $\sigma_2$ can be constructed by replacing $\star$ with appropriate moves.

*Example 1.* Let $\bar{\sigma}_1$ and $\bar{\sigma}_2$ be strategies defined by

$$\begin{aligned}
\bar{\sigma}_1 &= [\![ c : o^1, a : o^2 \to o^3 \;\vdash\; (\lambda x.a(a(x)),\, c) : (o^4 \to o^5) \times o^6 ]\!] \\
\bar{\sigma}_2 &= [\![ f : o^4 \to o^5, x : o^6 \;\vdash\; f(f(x)) : o^7 ]\!],
\end{aligned}$$

where $[\![\cdot]\!]$ is the standard interpretation of the simply-typed lambda calculus. Their composition is equivalent to

$$\begin{aligned}
&[\![ c : o^1, a : o^2 \to o^3 \;\vdash\; f(f(x))[(\lambda x.a(a(x)))/f, c/x] : o^7 ]\!] \\
= &[\![ c : o^1, a : o^2 \to o^3 \;\vdash\; a(a(a(a(c)))) : o^7 ]\!],
\end{aligned}$$

which have a derivation of a judgement

$$[\![c : q_0^P, a : (q_3^O \to q_4^P) \wedge (q_2^O \to q_3^P) \wedge (q_1^O \to q_2^P) \wedge (q_0^O \to q_1^P) \ \vdash \ a(a(a(a(c)))) : q_4^O]\!].$$

(Here $o^1$ and $o^2$ are different occurrences of the same kind, $q_1$ and $q_2$ are different types and $q_1^P$ and $q_1^O$ are different occurrences of the same type $q_1$.) Let $\sigma$ be the strategy corresponding to the derivation. Then $\sigma$ contains a play

$$s = q_4^O \cdot q_4^P \cdot q_3^O \cdot q_3^P \cdot q_2^O \cdot q_2^P \cdot q_1^O \cdot q_1^P \cdot q_0^O \cdot q_0^P$$

that is mapped by $U$ to $U(s) = \bar{s} = o^7 \cdot o^3 \cdot o^2 \cdot o^3 \cdot o^2 \cdot o^3 \cdot o^2 \cdot o^3 \cdot o^2 \cdot o^1$; and $\mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2)$ contains $\bar{u} = o^7 \cdot o^5 \cdot o^3 \cdot o^2 \cdot o^3 \cdot o^2 \cdot o^4 \cdot o^5 \cdot o^3 \cdot o^2 \cdot o^3 \cdot o^2 \cdot o^4 \cdot o^6 \cdot o^1$. Note that $U(s) = \bar{u} \upharpoonright_{K_1 \Rightarrow K_2}$. The annotated interaction sequence generated by $s$ and $\bar{u}$ is

$$\begin{bmatrix} o^7 \\ q_4^O \end{bmatrix} \cdot \begin{bmatrix} o^5 \\ \star \end{bmatrix} \cdot \begin{bmatrix} o^3 \\ q_4^P \end{bmatrix} \cdot \begin{bmatrix} o^2 \\ q_3^O \end{bmatrix} \cdot \begin{bmatrix} o^3 \\ q_3^P \end{bmatrix} \cdot \begin{bmatrix} o^2 \\ q_2^O \end{bmatrix} \cdot \begin{bmatrix} o^4 \\ \star \end{bmatrix} \cdot \begin{bmatrix} o^5 \\ \star \end{bmatrix} \cdot \begin{bmatrix} o^3 \\ q_2^P \end{bmatrix} \cdots \begin{bmatrix} o^1 \\ q_0^P \end{bmatrix}.$$

The set of moves with which $\star$ is replaced should satisfy competing requirements. Occurrences of $\star$ should be distinguished as much as possible in order to fulfil the universal property, but distinguishing them too much makes $\sigma_1$ and $\sigma_2$ non-innocent strategies. A coloured arena $A = (M, \vdash, \lambda, c)$ is defined as follows.

- $M = \{\binom{\bar{u}}{p} \mid \bar{u} \in \mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2), \bar{u}$ ends with $K$-move$, p \in \sigma, \ulcorner p \urcorner = p, U(p) = \bar{u} \upharpoonright_{K_1 \Rightarrow K_2}\}$
- $\vdash \binom{\bar{u}}{p}$ iff the last move of $\bar{u}$ is an initial move of $K$; $\binom{\bar{u}}{p} \vdash \binom{\bar{u}'}{p'}$ iff (i) $p$ is a prefix of $p'$, and (ii) The last move of $\bar{u}'$ is justified by the last move of $\bar{u}$.
- $\lambda(\binom{\bar{u}}{p}) = \lambda_K(\bar{m})$ where $\bar{m}$ is the last move of $\bar{u}$; and $c(\binom{\bar{u}}{p}) = c_{A_1 \Rightarrow A_2}(m)$ where $m$ is the last move of $p$.

The two-level arena $\mathcal{A}$ is defined as $(A, U, K)$ where $U(\binom{\bar{u}}{p}) = \bar{m}$ (here $\bar{m}$ is the last move of $\bar{u}$).

**Definitions of $\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma_2}$** For each pair $(p, \bar{u}) \in \sigma \times \mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2)$ such that $p$ is a P-view and $U(p) = \bar{u} \upharpoonright_{K_1 \Rightarrow K_2}$, we construct an interaction sequence of $\mathrm{Int}(\mathcal{A}_1, \mathcal{A}, \mathcal{A}_2)$, written $\langle \bar{u}, p \rangle$. Basically, $\langle \bar{u}, p \rangle$ is generated by replacing $\star$ in the annotated interaction sequence with appropriate moves of $\mathcal{A}$. $\langle \bar{u}, p \rangle$ is defined by induction on $\bar{u}$ as follows:

$$\langle \bar{u} \cdot \bar{m}, p \rangle = \langle \bar{u}, p \rangle \cdot \begin{pmatrix} \bar{u} \cdot \bar{m} \\ p \end{pmatrix} \qquad (\text{if } \bar{m} \in K)$$

$$\langle \bar{u} \cdot \bar{m}, p \cdot m \rangle = \langle \bar{u}, p \rangle \cdot m \qquad (\text{if } \bar{m} \in K_1 \Rightarrow K_2)$$

where justification pointers are induced from $\bar{u} \cdot \bar{m}$.

**Lemma 6.** *Let $p$ be a P-view of $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$ and $\bar{u} \in \mathrm{Int}(K_1, K, K_2)$ be an interaction sequence and assume that $U(p) = \bar{u} \upharpoonright_{K_1 \Rightarrow K_2}$. Then $\langle \bar{u}, p \rangle \in \mathrm{Int}(\mathcal{A}_1, \mathcal{A}, \mathcal{A}_2)$.*

*Proof.* By induction on the length of $\bar{u}$. □

We define $I \subseteq \mathrm{Int}(\mathcal{A}_1, \mathcal{A}, \mathcal{A}_2)$ by

$$I = \{\, \langle \bar{u}, p\rangle \mid \bar{u} \in \mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2),\ p \in \sigma,\ \ulcorner p \urcorner = p,\ U(p) = u\restriction_{K_1 \Rightarrow K_2} \}\,.$$

Now we define strategies. Let $\varphi_1$ be an view function of an arena $A_1 \Rightarrow A$ determined by a set of P-views $\{\ulcorner s\restriction_{\mathcal{A}_1 \Rightarrow \mathcal{A}} \urcorner \mid s \in I\}$ and $\varphi_2$ be a view function of an arena $A \Rightarrow A_2$ determined by $\{\ulcorner s\restriction_{\mathcal{A} \Rightarrow \mathcal{A}_2} \urcorner \mid s \in I\}$. The strategy $\sigma_1$ is induced from $\varphi_1$ and $\sigma_2$ from $\varphi_2$.

We show that $\sigma_1$ and $\sigma_2$ are well-defined. We need an auxiliary lemma.

**Lemma 7.** *Let $p$ be a P-view of $\mathcal{A}_1 \Rightarrow \mathcal{A}_2$ and $\bar{u} \in \mathrm{Int}(K_1, K, K_2)$ be an interaction sequence and assume that $U(p) = \bar{u}\restriction_{K_1 \Rightarrow K_2}$.*

$(i)$ *If $\bar{u}$ ends with an O-move of $K_1 \Rightarrow K$, then $\langle \bar{u}, p\rangle$ can be determined by $\ulcorner \langle \bar{u}, p\rangle\restriction_{A_1 \Rightarrow A} \urcorner$.*
$(ii)$ *If $\bar{u}$ ends with an O-move of $K \Rightarrow K_2$, $\langle \bar{u}, p\rangle$ can be determined by $\ulcorner \langle \bar{u}, p\rangle\restriction_{A \Rightarrow A_2} \urcorner$.*

*Proof.* We prove $(i)$. $(ii)$ is shown by a similar way.

If $\bar{u}$ ends with a move of $\mathcal{A}$, then the last move contains as much information as the pair $(\bar{u}, p)$. Assume that $\bar{u}$ ends with a move of $K_1$. Then there are moves $\bar{m}_1^P$ and $\bar{m}_2^O$ and some justified sequence $\bar{v}$ such that $\bar{u} = \bar{u}' \cdot \bar{m}_1^P \cdot \bar{v} \cdot \bar{m}_2^O$, where $\bar{m}_1^P$ justifies $\bar{m}_2^O$. Note that $K_1 \Rightarrow K_2$ component of $\bar{u}$ is a P-view by the assumption. Thus $\bar{v}$ contains no moves in $K_1 \Rightarrow K_2$. Since $U(p) = \bar{u}\restriction_{K_1 \Rightarrow K_2} = \bar{u}' \cdot \bar{m}_1^P \cdot \bar{v} \cdot \bar{m}_2^O\restriction_{K_1 \Rightarrow K_2}$, there are some moves $m_1^P, m_2^O \in A_1 \Rightarrow A_2$ and a P-view $p'$ of $A_1 \Rightarrow A_2$ such that $p = p' \cdot m_1^P \cdot m_2^O$ and $U(m_1^P) = \bar{m}_1^P$ and $U(m_2^O) = \bar{m}_2^O$. Therefore we have

$$\langle \bar{u}, p\rangle = \langle \bar{u}', p'\rangle \cdot m_1^P \cdot m_2^O.$$

Since $m_2^O$ is an O-move of $A_1$,

$$\ulcorner \langle \bar{u}, p\rangle \urcorner = \ulcorner \langle \bar{u}', p'\rangle \cdot m_1^P \cdot m_2^O \urcorner = \ulcorner \langle \bar{u}', p'\rangle \urcorner \cdot m_1^P \cdot m_2^O.$$

By induction hypothesis, we can compute $\langle \bar{u}', p'\rangle$ from $\ulcorner \langle \bar{u}', p'\rangle \urcorner$. Thus $(i)$ holds. □

**Lemma 8.** *$\varphi_1$ and $\varphi_2$ are well-defined view functions.*

*Proof.* We prove that $\varphi_1$ is well-defined. Well-definedness of $\varphi_2$ is shown by the same way.

Let $s \in \varphi_1$ be an play ending with an O-move. What we should show are:

$(i)$ $s \cdot m \in \varphi_1$ for some $m$ that has the same colour as the last move of $s$.
$(ii)$ If $s \cdot m \in \varphi_1$ and $s \cdot m' \in \varphi_1$, then $s \cdot m = s \cdot m'$.

$(i)$ is easy to show because for every $u \in I$ ending with an O-move of $A_1 \Rightarrow A$, we have $u{\cdot}m \in I$ for some move $m$ of $A_1 \Rightarrow A$. We prove $(ii)$. Assume that $s{\cdot}m \in \varphi_1$ and $s{\cdot}m' \in \varphi_1$. By definition of $\varphi_1$, we have $u{\cdot}m,\ u'{\cdot}m' \in I$ such that $\ulcorner (u \cdot m)\restriction_{A_1 \Rightarrow A} \urcorner = s \cdot m$ and $\ulcorner (u' \cdot m')\restriction_{A_1 \Rightarrow A} \urcorner = s \cdot m'$. Since $s$ ends with an O-move of $A_1 \Rightarrow A$, by Lemma 7, $s$ completely determines $u$ and $u'$. Thus $u = u'$ and $u \cdot m' \in \varphi_1$. By determinacy of $\sigma$, $\bar{\sigma}_1$ and $\bar{\sigma}_2$, we have $u \cdot m = u \cdot m'$ as required. □

Thanks to Lemma 8, $\sigma_1$ and $\sigma_2$ are well-defined innocent strategies. Trivially, $U(\sigma_1) \subseteq \bar{\sigma}_1$ and $U(\sigma_2) \subseteq \bar{\sigma}_2$.

**Lemma 9.** $(\sigma_1; \sigma_2) = \sigma$.

*Proof.* We first prove that $\sigma \subseteq (\sigma_1; \sigma_2)$. Since $\sigma$ is innocent, it suffices to show that $\sigma_1; \sigma_2$ contains every P-view $p \in \sigma$. Let $p \in \sigma$ be a P-view. Then $U(p) \in \bar{\sigma} = (\bar{\sigma}_1; \bar{\sigma}_2)$. Thus there is $\bar{u} \in \mathbf{Int}(\bar{\sigma}_1, \bar{\sigma}_2)$ such that $U(p) = \bar{u} \upharpoonright_{K_1 \Rightarrow K_2}$. By definition, $\langle \bar{u}, p \rangle \in I$. We can prove $\langle \bar{u}, p \rangle \in \mathbf{Int}(\sigma_1, \sigma_2)$ by induction on the length of $\bar{u}$. So $p = \langle \bar{u}, p \rangle \upharpoonright_{A_1 \Rightarrow A_2} \in (\sigma_1; \sigma_2)$.

Second we prove that $(\sigma_1; \sigma_2) \subseteq \sigma$. It suffices to show that $p \in \sigma$ for every P-view $p \in (\sigma_1; \sigma_2)$. Since $p \in (\sigma_1; \sigma_2)$, we have its uncovering $u \in \mathbf{Int}(\sigma_1, \sigma_2)$ (so $p = u \upharpoonright_{A_1 \Rightarrow A_2}$). Then by induction on the length of $u$, we can prove that $u \upharpoonright_{A_1 \Rightarrow A_2} \in \sigma$. $\square$

Thanks to Lemma 9, we have finished to construct an object $\mathcal{A}$ and strategies $\sigma_1$ and $\sigma_2$, required by Theorem 3.

**Canonicity of $\mathcal{A}$** Let $(\sigma'_1, \mathcal{A}', \sigma'_2)$ be another triple that satisfies the requirement of subject expansion except for canonicity. So $\sigma'_1 : \mathcal{A}_1 \to \mathcal{A}'$, $\sigma'_2 : \mathcal{A}' \to \mathcal{A}_2$, $U(\sigma'_i) \subseteq \bar{\sigma}_i$ (for $i = 1, 2$) and $\sigma = (\sigma'_1; \sigma'_2)$.

What we should do is construction of a function from moves of $A$ to moves of $A'$. Let $\binom{\bar{u}}{p}$ be a move of $A$. Note that by definition $p$ ends with an O-move of $A_1 \Rightarrow A_2$. Let $m$ be the move such that $p \cdot m \in \sigma$. Since $p \cdot m \in \sigma = (\sigma'_1; \sigma'_2)$, we have the uncovering of $p \cdot m$ over $\sigma'_1$ and $\sigma'_2$, say $u'$. Since $U(\sigma'_1) \subseteq \bar{\sigma}_1$ and $U(\sigma'_2) \subseteq \bar{\sigma}_2$, we have $U(u') \in (\bar{\sigma}_1; \bar{\sigma}_2)$. So $\langle \bar{u'}, p \cdot m \rangle$ is an postfix of $\langle \bar{u}, p \rangle$. Thus $\langle \bar{u'}, p \cdot m \rangle$ contains the move $\binom{\bar{u}}{p}$, say, as the $k$th move. Let $m'$ be the $k$th move of $u'$. We map $\binom{\bar{u}}{p}$ to $m'$.

Let $\varphi$ be the mapping defined below. It is easy to prove that $\varphi$ is well-defined. $[\mathrm{id}_{K_1}, \varphi](\sigma_1) \subseteq \sigma'_1$ (resp. $[\varphi, \mathrm{id}_{K_2}](\sigma_2) \subseteq \sigma'_2$) can be show by induction on the length of plays in $\sigma_1$ (resp. $\sigma_2$).

## 5 Interpretation of Intersection Types

In this section, we interpret Kobayashi's intersection type system [3] in the two-level game model, and show that the interpretation is *fully complete* i.e. every winning strategy is the denotation of some derivation.

### 5.1 An Intersection Type System

We consider the standard Church-style simply-typed lambda calculus defined by the following grammar:

$$\begin{array}{lrcl} \textit{Sorts} & \kappa & ::= & o \mid \kappa_1 \to \kappa_2 \\ \textit{Terms} & t & ::= & x \mid \lambda x^{\kappa}.t \mid t_1 t_2 \end{array}$$

We refer to simple types as *kinds* to avoid confusion with intersection types. Let $\Delta$ be a *kind environment* i.e. a set of variable-kind bindings, $x : \kappa$. We write $\Delta \vdash t :: \kappa$ to mean $t$ has kind $\kappa$ under the environment $\Delta$. Fix a set $Q$ of symbols, ranged over by $q$. The set of *intersection pre-types* is defined by the following grammar where $n \geq 0$:

$$\textit{Intersection Pre-Types} \qquad \tau, \sigma \quad ::= \quad q \ \mid \ \tau \to \sigma \ \mid \ \bigwedge_{i \in I} \tau_i$$

The *well-kindedness relation* $\tau :: \kappa$ is defined by the following rules.

$$\frac{}{q :: o}$$

$$\frac{\tau_i :: \kappa \quad (\text{for all } i \in I) \qquad \sigma :: \kappa'}{(\bigwedge_{i \in I} \tau_i) \to \sigma :: \kappa \to \kappa'}$$

An *intersection type* is an intersection pre-type $\tau$ such that $\tau :: \kappa$ for some $\kappa$.

An *(intersection) type environment* $\Gamma$ is a set of variable-type bindings, $x : \bigwedge_{i \in I} \tau_i$. We write $\Gamma :: \Delta$ just if $x : \bigwedge_{i \in I} \tau_i \in \Gamma$ implies that for some $\kappa$, $x : \kappa \in \Delta$ and $\tau_i :: \kappa$ for all $i \in I$. *Valid typing sequents* are defined by induction over the following rules.

$$\frac{}{\Gamma, x : \bigwedge_{i \in I} \tau_i \vdash x : \tau_i}$$

$$\frac{\Gamma \vdash t_1 : (\bigwedge_{i \in I} \tau_i) \to \sigma \qquad \Gamma \vdash t_2 : \tau_i \quad (\text{for all } i \in I)}{\Gamma \vdash t_1 \ t_2 : \sigma}$$

$$\frac{\Gamma, x : \bigwedge_{i \in I} \tau_i \vdash t : \sigma \qquad \tau_i :: \kappa \quad (\text{for all } i \in I)}{\Gamma \vdash \lambda x^{\kappa}.t : (\bigwedge_{i \in I} \tau_i) \to \sigma}$$

**Lemma 10.** *If $\Delta \vdash t :: \kappa$ and $\Gamma :: \Delta$ and $\Gamma \vdash t : \tau$, then $\tau :: \kappa$.*

*Proof.* Easy induction on the structure of $\Delta \vdash t :: \kappa$. □

### 5.2 Representing Derivations by Proof Terms

For notational convenience, we use a Church-style simply-kinded lambda calculus with (indexed) product as a term representation of derivations. The raw terms are defined as follows.

$$M \ ::= \ \mathsf{p}_i(x) \ \mid \ \lambda x^{\bigwedge_{i \in I} \tau_i}.M \ \mid \ M_1 \, M_2 \ \mid \ \prod_{i \in I} M_i$$

where $I$ is a finite indexing set. We omit $I$ and simply write $\lambda x^{\bigwedge_i \tau_i}$ and so on if $I$ is clear from the context or unimportant. We say a term $M$ is *well-formed* just if for every application subterm $M_1 \, M_2$ of $M$, $M_2$ has the form $\prod_{i \in I} N_i$. We consider only well-formed terms. By abuse of notation, we write $\top$ for $\prod \emptyset$.

We give a type system for terms of the calculus, which ressemble the intersection type system, but is syntax directed, i.e., a term completely determines the structure of a derivation.

$$\frac{}{\Gamma, x : \bigwedge_{i\in I} \tau_i \Vdash \mathsf{p}_i(x) : \tau_i}$$

$$\frac{\Gamma, x : \bigwedge_{i\in I} \tau \Vdash M : \sigma}{\Gamma \Vdash \lambda x^{\bigwedge_{i\in I}\tau_i}.M : (\bigwedge_{i\in I} \tau_i) \to \sigma}$$

$$\frac{\Gamma \Vdash M_1 : (\bigwedge_i \tau_i) \to \sigma \qquad \Gamma \Vdash M_2 : \bigwedge_i \tau_i}{\Gamma \Vdash M_1\ M_2 : \sigma}$$

$$\frac{\Gamma \Vdash M_i : \tau_i \qquad \tau_i :: \kappa \qquad \text{(for all } i)}{\Gamma \Vdash \bigsqcap_i M_i : \bigwedge_i \tau_i}$$

We call a term-in-context $\Gamma \Vdash M : \tau$ a *proof term*. Observe that a proof term is essentially a typed lambda term with (indexed) product. Here an intersection type $\tau_1 \wedge \cdots \wedge \tau_n$ is interpreted as a product type $\tau_1 \times \cdots \times \tau_n$ and a proof term $M_1 \sqcap \cdots \sqcap M_n$ is a tuple $\langle M_1, \ldots, M_n \rangle$. Then all variables are bound to tuples and a proof term $\mathsf{p}_i(x)$ is a projection into the $i$th element.

Unfortunately, not all the proof terms correspond to a derivation of the intersection type system. For example, $\lambda f^{(q_1 \wedge q_2)\to p}.\lambda x^{q_1}.\lambda y^{q_2}.f(\mathsf{p}(x) \sqcap \mathsf{p}(y))$ is a proof term of the type $((q_1 \wedge q_2) \to p) \to q_1 \to q_2 \to p$, but there is no inhabitant of that type. In the intersection type system, $t : \tau \wedge \sigma$ only if $t : \tau$ and $t : \sigma$ for the same term $t$, but the proof term $\mathsf{p}(x) \sqcap \mathsf{p}(y)$ violates the requirement.

We introduce a judgement $M :: t$ that means the structure of $M$ coincides with the structure of $t$.

$$\begin{array}{rll}
\mathsf{p}_i(x) :: x & & \\
\lambda x^{\bigwedge_i \tau_i}.M :: \lambda x^{\kappa}.t & \text{iff} & M :: t \text{ and } \tau_i :: \kappa \text{ for all } i \\
M_1\ M_2 :: t_1\ t_2 & \text{iff} & M_1 :: t_1 \text{ and } M_2 :: t_2 \\
\bigsqcap_i M_i :: t & \text{iff} & M_i :: t \text{ for all } i
\end{array}$$

By definition, $\top :: t$ for every term $t$.

**Lemma 11 (Coincidence).** *(i) For every derivations $\mathcal{D}$ whose conclusion is $\Gamma \vdash t : \tau$, there exists a proof term* $\mathrm{Term}(\mathcal{D})$ *such that* $\Gamma \Vdash \mathrm{Term}(\mathcal{D}) : \tau$ *and* $\mathrm{Term}(\mathcal{D}) :: t$. *(ii) If $\Gamma \Vdash M : \tau$ and $M :: t$, there exists a unique derivation $\mathcal{D}$ of $\Gamma \vdash t : \tau$ such that* $\mathrm{Term}(\mathcal{D}) = M$.

*Proof.* Easy induction on the structure of $\mathcal{D}$ and of $M$, respectively.

We write $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$ just if $\Gamma :: \Delta$, $M :: t$, $\tau :: \kappa$, $\Delta \vdash t{::}\kappa$ and $\Gamma \Vdash M : \tau$. Let $t$ be a term such that $\Delta \vdash t :: \kappa$. The previous lemma says that there is a one-to-one correspondence between a derivation of $\Gamma \vdash t : \tau$ and a proof term $M$ such that $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$.

*Example 2.* Let $Q = \{ q_1, q_2 \}$ and take $\theta \to (q_1 \wedge q_2) \to q_1 :: (o \to o) \to o \to o$ where $\theta = (q_1 \to q_1) \wedge (q_2 \to q_1) \wedge (q_1 \wedge q_2 \to q_1)$ and terminal $f : q_1 \to q_2$. Set $M := \lambda x^{\theta}\, y^{q_1 \wedge q_2}.\mathsf{p}_2(x)(f^{q_1 \to q_2}(\mathsf{p}_1(x)(\mathsf{p}_3(x)(\mathsf{p}_1(y) \sqcap \mathsf{p}_2(y)))))$. Then we have $M :: \lambda xy.x(f(x(x\, y))$.

**Lemma 12.** *It is decidable, given $\Gamma \Vdash M : \tau$, whether $M :: t$ for some $t$. Hence, thanks to Lemma 11, it is decidable whether a proof term represents a derivation.*

*Proof.* The definition of $M :: t$ itself gives a simple decision procedure.

We define an operational semantics for terms and proof terms. The reduction relation is the least congruence defined by the following $\beta$-reduction and $\eta$-expansion redex rules:

$$(\lambda x^{\kappa}.s)\ t \longrightarrow_\beta [t/x]\, s$$
$$t^{\kappa_1 \to \kappa_2} \longrightarrow_\eta \lambda x^{\kappa_1}.(t^{\kappa_1 \to \kappa_2}\ x^{\kappa_1})^{\kappa_2} \qquad (x \text{ is fresh})$$

Here $[t'/x]$ is the standard capture-avoiding substitution of $t'$ for $x$. We write $\longrightarrow$ for $\longrightarrow_\beta \cup \longrightarrow_\eta$, $\longrightarrow^*$ for reflexive and transitive closure of $\longrightarrow$, and $=_{\beta\eta}$ for reflexive, transitive and symmetric closure of $\longrightarrow$.

The reduction relation of proof terms is defined similarly:

$$(\lambda x^{\bigwedge_i \tau_i}.M)\ (\prod_i N_i) \longrightarrow_\beta [\prod_i N_i/x]\ M$$
$$M^{\bigwedge_i \tau_i \to \sigma} \longrightarrow_\eta \lambda x^{\bigwedge_i \tau_i}.M\ (\prod_i \mathsf{p}_i(x)) \quad (x \text{ is fresh})$$

where (the base case of) the substitution is given by

$$[\prod_i N_i/x]\ (\mathsf{p}_i(x)) = N_i$$
$$[\prod_i N_i/x]\ (\mathsf{p}_i(y)) = \mathsf{p}_i(y) \quad (\text{if } x \neq y).$$

We write $[M :: t] \longrightarrow [M' :: t']$ if $t \longrightarrow t'$ and $M \longrightarrow^* M'$. It is easy to see that if $M :: t$ and $t \longrightarrow t'$, then there exists a unique $M'$ such that $M \longrightarrow^* M'$ and $M' :: t'$ ($M \longrightarrow^* M'$ reduces all the redexes at the positions similar to the redex of $t \longrightarrow t'$).

### 5.3 Game Semantics of Intersection Types

A two-level arena represents a proof of well-kindedness, $\tau :: \kappa$. The interpretation is straightforward since we have arena constructors $\Rightarrow$ and $\wedge$:

$$\begin{aligned} [\![q :: o]\!] &:= ([\![q]\!], U, [\![o]\!]) \\ [\![(\bigwedge_{i \in I} \tau_i) \to \sigma :: \kappa \to \kappa']\!] &:= (\bigwedge_{i \in I} [\![\tau_i :: \kappa]\!]) \Rightarrow [\![\sigma :: \kappa']\!], \end{aligned}$$

where $[\![q]\!]$ is a $Q$-coloured arena with a single move of the colour $q$, $[\![o]\!]$ is a $\{o\}$-coloured arena with a single move, and $U$ maps the unique move of $[\![q]\!]$ to the unique move of $[\![o]\!]$. Let $\Gamma$ be a type environment with $\Gamma :: \Delta$. Suppose

$$\begin{aligned} \Gamma &= x_1 : \textstyle\bigwedge_{i \in I_1} \tau_i^1, \ldots, x_n : \bigwedge_{i \in I_n} \tau_i^n \\ \Delta &= x_1 : \kappa_1, \qquad\quad \ldots, x_n : \kappa_n \end{aligned}$$

Then $[\![\Gamma :: \Delta]\!] := \prod_{j \leq n} (\bigwedge_{i \in I_j} [\![\tau_i^j :: \kappa_i]\!])$ .

A proof $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$, which is equivalent to a derivation of $\Gamma \vdash t : \tau$ (Lemma 11), is interpreted as a strategy of the two-level arena $[\![\Gamma :: \Delta]\!] \Rightarrow [\![\tau :: \kappa]\!]$, defined by the following rules (for simplicity, we write $[\![M :: t]\!]$ instead of $[\![[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]]\!]$):

$$\begin{aligned}
[\![\mathsf{p}_i(x) :: x]\!] &:= \pi_x ; \mathsf{p}_i \\
[\![\prod_i M_i :: t]\!] &:= \prod_i [\![M_i :: t]\!] \\
[\![M_1\ M_2 :: t_1\ t_2]\!] &:= \langle [\![M_1 :: t_1]\!], [\![M_2 :: t_2]\!] \rangle ; \mathbf{eval} \\
[\![\lambda x.M :: \lambda x.t]\!] &:= \Lambda([\![M :: t]\!])
\end{aligned}$$

where $\pi_x$ is the projection $[\![(\Gamma, x : \bigwedge_i \tau_i) :: (\Delta, x : \kappa)]\!] \longrightarrow [\![\bigwedge_i \tau_i :: \kappa]\!]$ and for strategies $\sigma_i : [\![\Gamma :: \Delta]\!] \longrightarrow [\![\tau_i :: \kappa]\!]$ indexed by $i$, the strategy $\prod_i \sigma_i : [\![\Gamma :: \Delta]\!] \longrightarrow \bigwedge_i [\![\tau_i :: \kappa]\!]$ is the canonical map of the pullback.

**Lemma 13 (Componentwise Interpretation).** *Let $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$ be a derivation. Then $[\![M :: t]\!] = ([\![M]\!], [\![t]\!])$.*

*Proof.* By induction on $M$.

**Lemma 14 (Substitution).** *Suppose $[(\Gamma, x : \bigwedge_i \tau_i) :: (\Delta, x : \kappa)] \vdash [M :: t] : [\sigma :: \kappa']$ and $[\Gamma :: \Delta] \vdash [\prod_i N_i :: u] : [\bigwedge_i \tau_i :: \kappa]$. Then*

$$\langle \mathbf{id}_{[\![\Gamma :: \Delta]\!]}, [\![\prod_i N_i :: u]\!] \rangle ; [\![M :: t]\!] = [\![([\prod_i N_i / x]\ M) :: ([u/x]\ t)]\!].$$

*Proof.* By Lemma 13 and a well-know result for the standard interpretation [2].

**Lemma 15.** *Suppose $[\Gamma :: \Delta] \vdash [M :: t] : [\bigwedge_i \tau_i \to \sigma :: \kappa \to \kappa']$. Then*

$$[\![M :: t]\!] = [\![(\lambda x^{\bigwedge_i \tau_i} . M\ (\prod_i \mathsf{p}_i(x))) :: (\lambda x^{\kappa} . t\ x)]\!].$$

**Theorem 4 (Adequacy).** *Let $[\Gamma :: \Delta] \vdash [M_1 :: t_1] : [\tau :: \kappa]$ and $[\Gamma :: \Delta] \vdash [M_2 :: t_2] : [\tau :: \kappa]$ be two proofs such that $[M_1 :: t_1] =_{\beta\eta} [M_2 :: t_2]$. Then $[\![M_1 :: t_1]\!] = [\![M_2 :: t_2]\!]$.*

*Proof.* A consequence of the two lemmas above.

**Theorem 5 (Definability).** *Let $(\sigma, \bar{\sigma}) : [\![\Gamma :: \Delta]\!] \to [\![\tau :: \kappa]\!]$ be a winning strategy. There is a derivation $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$ such that $(\sigma, \bar{\sigma}) = [\![M :: t]\!]$.*

*Proof.* (Sketch) By the standard argument of definability [2], we have a proof term $M$ and a simply-typed lambda term $t$ such that $[\![M]\!] = \sigma : [\![\Gamma]\!] \longrightarrow [\![\tau]\!]$ and $[\![t]\!] = \bar{\sigma} : [\![\Delta]\!] \longrightarrow [\![\kappa]\!]$, where $[\![\cdot]\!]$ is the standard interpretation of typed lambda terms (here intersection $\wedge$ in $\Gamma$ and $\tau$ is interpreted as a product). If $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$ is a valid derivation, by Lemma 13, we have $[\![M :: t]\!] = (\sigma, \bar{\sigma})$ as required. Thus it suffices to show that $M :: t$, which can be shown by an easy induction. $\square$

We can use Church-style type-annotated terms in $\beta$-normal $\eta$-long form, called *canonical terms*, to represent winning strategies, which are terms-in-context of the form: $\Gamma \Vdash \mathsf{p}_i(x)\, M_1 \cdots M_n : q$ where $\Gamma = \cdots, x : \bigwedge_i \alpha_i, \cdots$ and $\alpha_i = \tau_1 \to \cdots \to \tau_n \to q$, and for each $k \in \{ 1, \ldots, n \}$,

$$M_k \;=\; \prod_{j \in J_k} \lambda y_{kj1}^{\tau_{kj1}} \ldots y_{kjr}^{\tau_{kjr}}.N_{kj} \;:\; \bigwedge_{j \in J_k} \beta_{kj} \;=\; \tau_k$$

such that for each $j \in J_k$, $\beta_{kj} = \tau_{kj1} \to \cdots \to \tau_{kjr} \to q_{kj}$ with $r = r_{kj}$ and $\Gamma, y_{kj1} : \tau_{kj1}, \cdots, y_{kjr} : \tau_{kjr} \Vdash N_{kj} : q_{kj}$ is a canonical term. (We assume that canonical terms are proof terms that represent derivations.)

By definition, canonical terms are not $\lambda$-abstractions. We call terms-in-context such as $M_k$ above canonical terms in (partially) *curried form*; they have the shape $\Gamma \Vdash \lambda\overline{x}.M : \tau_1 \to \cdots \to \tau_n \to q$. Note that in case $n = 0$, the curried form retains an outermost "dummy lambda" $\Gamma \Vdash \lambda.M : q$. With this syntactic convention, we obtain a tight correspondence between syntax and semantics.

**Lemma 16.** *Let $\tau :: \kappa$. There is a one-to-one correspondence between winning strategies over the two-level arena $[\![\tau :: \kappa]\!]$ and canonical terms in curried form of the shape $\emptyset \Vdash M : \tau$ (with $\eta$-long $\beta$-normal simply-typed term $t$ such that $M :: t$).*

*Proof.* First observe that a two-level arena is a forest; each move of the arena can be represented by the subtree rooted at the move. In other words, moves of $[\![\tau :: o]\!]$ correspond to (and can be named by) the prime subtypes of $\tau$. Consider the abstract syntax trees of these terms, so that the nodes at levels 0, 2, 4, etc. are labelled by lambdas (i.e. $\lambda\overline{x}$), and nodes at levels 1, 3, 5, etc. are labelled by variables. The idea is that a node labelled by a lambda (respectively variable) of prime type $\theta$ represents the O-move (respectively P-move) named by $\theta$. It suffices to observe that there is a one-one correspondence between the even-length paths in such a tree, and the even-length P-views in the corresponding winning strategy. (Note that an innocent strategy—*qua* set of legal positions—is determined by its subset of even-length P-views, which is just its view function.) We check that the term representation satisfies the axioms of winning strategy. P/O-alternation holds by construction of the canonical term; pointers to O-moves correspond to the standard lambda binding, and pointers to P-move correspond to the edges from a lambda node to its parent, which is a variable node. Colour-reflection, totality (leaves of a tree are by construction either a variable or $\top$) and contingent completeness all hold by definition of canonical term.

A strategy $(\sigma, \bar{\sigma})$ of $\mathcal{A} = (A, U, K)$ is *P-full* (respectively *O-full*) just if every P-move (respectively O-move) of $A$ occurs in $\sigma$. Suppose $(\sigma, \bar{\sigma})$ is a winning strategy of $[\![\tau :: \kappa]\!]$. Then: (i) If $(\sigma, \bar{\sigma})$ is P-full, then it is also O-full. (ii) There is a subtype $\tau' :: \kappa$ of $\tau$ such that $(\sigma, \bar{\sigma})$ is winning and P-full over $[\![\tau' :: \kappa]\!]$.

A derivation $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$ is *relevant* just if for each abstraction subterm $\lambda x^{\bigwedge_{i \in I} \tau_i}.M'$ of $M$ and $i \in I$, $M'$ has a free occurrence of $\mathsf{p}_i(x)$.

**Lemma 17.** $[\Gamma :: \Delta] \vdash [M :: t] : [\tau :: \kappa]$ *is relevant iff* $[\![M :: t]\!]$ *is P-full.*

*Proof.* The right-to-left direction is shown by a easy modification of the standard proof of definability (see [2, Proposition 7.4]). To prove the left-to-right direction, we first normalise $M :: t$ to the canonical form, say $M' :: t'$. It is easy to prove (syntactically) that normalisation preserves relevance of a derivation, so $M' :: t'$ is also relevant. Then by (easy) induction on canonical forms, we prove that $[\![M' :: t']\!]$ is full. By adequacy, $[\![M :: t]\!]$ is also full.

## 6 Applications to HORS Model-Checking

Fix a ranked alphabet $\Sigma$ and a HORS $G = \langle \Sigma, \mathcal{N}, S, \mathcal{R} \rangle$ we first give the game semantics $[\![G]\!]$ of $G$ (see [1] for a definition of HORS). Let $\mathcal{N} = \{ F_1 : \kappa_1, \ldots, F_n : \kappa_n \}$ with $F_1 = S$ (start symbol), and $\Sigma = \{ a_1 : r_1, \ldots, a_m : r_m \}$ where each $r_i = ar(a_i)$, the arity of $a_i$. Writing $[\![\Sigma]\!] := \prod_{i=1}^{m} [\![o^{r_i} \to o]\!]$ and $[\![\mathcal{N}]\!] := \prod_{i=1}^{n} [\![\kappa_i]\!]$, the *game semantics* of $G$, $[\![G]\!] : [\![\Sigma]\!] \longrightarrow [\![o]\!]$, is the composite

$$[\![\Sigma]\!] \xrightarrow{\Lambda(\mathbf{g})} ([\![\mathcal{N}]\!] \Rightarrow [\![\mathcal{N}]\!]) \xrightarrow{Y} [\![\mathcal{N}]\!] \xrightarrow{\{ S :: o \}} [\![o]\!]$$

in the cartesian closed category of $o$-coloured arenas and innocent strategies, where

- $\mathbf{g} = \langle g_1, \ldots, g_n \rangle : [\![\Sigma]\!] \times [\![\mathcal{N}]\!] \longrightarrow [\![\mathcal{N}]\!]$; each component $g_i = [\![\Sigma \cup \mathcal{N} \vdash \mathcal{R}(F_i) :: \kappa_i]\!]$ , and $\Lambda$(-) is currying
- $Y$ is the standard fixpoint strategy (see [2, §7.2]), and
- $\{ S :: o \} = \pi_1 : [\![\mathcal{N}]\!] \longrightarrow [\![o]\!]$ is the projection map.

*Remark 1.* Since the set of P-views of $[\![G]\!]$ coincide with the branch language[7] of the *value tree* of $G$ (i.e. the $\Sigma$-labelled tree generated by $G$; see [1]) and an innocent strategy is determined by its P-views, we identify the map $[\![G]\!]$ with the value tree of $G$.

Now fix a trivial automaton $\mathcal{B} = \langle Q, \Sigma, q_I, \delta \rangle$. We extend the game-semantic account to express the run tree of $\mathcal{B}$ over the value tree $[\![G]\!]$ in the category of $Q$-based two-level arenas and innocent strategies. First set

$$\begin{aligned} [\![\delta :: \Sigma]\!] &:= \textstyle\prod_{a \in \Sigma} \bigwedge_{(q,a,\overline{q}) \in \delta} [\![q_1 \to \ldots \to q_{ar(a)} \to q :: \underbrace{o \to \ldots \to o}_{ar(a)} \to o]\!] \\ &= ([\![\delta]\!], U, [\![\Sigma]\!]) \end{aligned}$$

where $[\![\delta]\!]$ is the $Q$-coloured arena $\prod_{a \in \Sigma} \prod_{(q,a,\overline{q}) \in \delta} [\![q_1 \to \ldots \to q_{ar(a)} \to q]\!]$ and $\overline{q} = q_1, q_2, \ldots, q_{ar(a)}$.

A *run tree* of $\mathcal{B}$ over $[\![G]\!]$ is just an innocent strategy $(\rho, [\![G]\!])$ of the arena $[\![\delta :: \Sigma]\!] \Rightarrow [\![q_I :: o]\!] = ([\![\delta]\!] \Rightarrow [\![q_I]\!], V, [\![\Sigma]\!] \Rightarrow [\![o]\!])$. Every P-view $\bar{p} \in [\![G]\!]$ has a unique "colouring" i.e. a P-view $p \in \rho$ such that $V(p) = \bar{p}$. This associates a colour (state) with each node of the value tree, which corresponds to a run tree in the concrete presentation.

[7] Let $m$ be the maximum arity of the symbols in $\Sigma$, and write $[m] = \{ 1, \cdots, m \}$. The *branch language* of $t : dom(t) \longrightarrow \Sigma$ consists of (i) $(f_1, d_1)(f_2, d_2) \cdots$ if there exists $d_1 \, d_2 \cdots \in [m]^\omega$ s.t. $t(d_1 \cdots d_i) = f_{i+1}$ for every $i \in \omega$; and (ii) $(f_1, d_1) \cdots (f_n, d_n) \, f_{n+1}$ if there exists $d_1 \cdots d_n \in [m]^*$ s.t. $t(d_1 \cdots d_i) = f_{i+1}$ for $0 \leq i \leq n$, and the arity of $f_{n+1}$ is 0.

### 6.1 Characterisation by Complete Type Environment

Using $G$ and $\mathcal{B}$ as before, Kobayashi [3] showed that $[\![G]\!]$ is accepted by $\mathcal{B}$ if, and only if, there is a *complete type environment* $\Gamma$, meaning that (i) $S : q_I \in \Gamma$, (ii) $\Gamma \vdash \mathcal{R}(F) : \theta$ for each $F : \theta \in \Gamma$. As a first application of two-level arena games, we give a semantic counterpart of the characterisation. Let $\Gamma = \{\, F_1 : \bigwedge_{j \in I_1} \tau_{1j} :: \kappa_1, \ldots, F_n : \bigwedge_{j \in I_n} \tau_{nj} :: \kappa_n \,\}$ be a type environment of $G$. Set $[\![\Gamma :: \mathcal{N}]\!] \;:=\; \prod_{i=1}^{n} \bigwedge_{j \in I_i} [\![\tau_{ij} :: \kappa_i]\!] = ([\![\Gamma]\!], U_1, [\![\mathcal{N}]\!])$ where $[\![\Gamma]\!] := \prod_{i=1}^{n} \prod_{j \in I_i} [\![\tau_{ij}]\!]$.

**Theorem 6.** *Using $\Sigma, G$ and $\mathcal{B}$ as before, $[\![G]\!]$ is accepted by $\mathcal{B}$ if, and only if, there exists $\Gamma$ such that*

*(i) $S : q_I \in \Gamma$, and*
*(ii) there exists a strategy $\sigma$ (say) of the $Q$-coloured arena $[\![\delta]\!] \times [\![\Gamma]\!] \Rightarrow [\![\Gamma]\!]$ such that $(\sigma, \mathbf{g})$ defines a winning strategy of the two-level arena*

$$([\![\delta :: \Sigma]\!] \times [\![\Gamma :: \mathcal{N}]\!]) \Rightarrow [\![\Gamma :: \mathcal{N}]\!] \;=\; ([\![\delta]\!] \times [\![\Gamma]\!] \Rightarrow [\![\Gamma]\!], V_1, [\![\Sigma]\!] \times [\![\mathcal{N}]\!] \Rightarrow [\![\mathcal{N}]\!])$$

*Proof.* Suppose we have $\Gamma$ and $\sigma$ that satisfy the conditions. The following composite map in the category of two-level arenas and innocent strategies

$$[\![\delta :: \Sigma]\!] \stackrel{\Lambda(\sigma,\mathbf{g})}{\longrightarrow} ([\![\Gamma :: \mathcal{N}]\!] \Rightarrow [\![\Gamma :: \mathcal{N}]\!]) \stackrel{(Y,Y)}{\longrightarrow} [\![\Gamma :: \mathcal{N}]\!] \stackrel{\{\, S : q_I \,\}}{\longrightarrow} [\![q_I :: o]\!]$$

gives the strategy $(\rho, [\![G]\!])$ over $([\![\delta]\!] \Rightarrow [\![q_I]\!], V, [\![\Sigma]\!] \Rightarrow [\![o]\!])$. To show $V(\rho) = [\![G]\!]$, it suffices to show that if $V(m_1) \cdots V(m_n) \cdot m \;\in\; [\![G]\!]$ and $m$ is an O-move of $[\![\Sigma]\!] \Rightarrow [\![o]\!]$, then $V(m') = m$ for some O-move $m'$ of $[\![\delta]\!] \Rightarrow [\![q_I]\!]$. But an O-move $m$ of $[\![\Sigma]\!] \Rightarrow [\![o]\!]$ is either the unique move of $[\![o]\!]$ or a move corresponding to an argument of a tree constructor in $\Sigma$ (i.e. a move corresponding to $o_i$ for some $a :: o_1 \to \cdots \to o_n \to o \in \Sigma$). By definition of $\delta$ and $q_I$, there exists a corresponding move $m'$ in $[\![\delta]\!] \Rightarrow [\![q_I]\!]$ for each O-move $m$ of $[\![\Sigma]\!] \Rightarrow [\![o]\!]$. The desired property then follows from the contingent completeness of $\rho$.

To prove the converse, suppose that $[\![G]\!]$ is accepted by $\mathcal{B}$. I.e. we have a run-tree given by a strategy $(\rho, [\![G]\!])$ over the two-level arena $([\![\delta]\!] \Rightarrow [\![q_I]\!], U_1, [\![\Sigma]\!] \Rightarrow [\![o]\!])$ such that $U_1(\rho) = [\![G]\!]$. By definition of $[\![G]\!]$, the following diagram (in the category of base arenas) commutes:

$$\begin{tikzcd}
{[\![\Sigma]\!]} \arrow[rr, "{[\![G]\!]}"] \arrow[dr, "\Lambda(\mathbf{g})"'] & & {[\![o]\!]} \\
& {[\![\mathcal{N}]\!] \Rightarrow [\![\mathcal{N}]\!]} \arrow[ur, "\mathbf{fix}"'] &
\end{tikzcd}$$

where $\mathbf{fix}$ is the composite

$$([\![\mathcal{N}]\!] \Rightarrow [\![\mathcal{N}]\!]) \stackrel{Y}{\longrightarrow} [\![\mathcal{N}]\!] \stackrel{\{\, S : o \,\}}{\longrightarrow} [\![o]\!].$$

By Subject Expansion (Theorem 3), there exist a two-level arena $\mathcal{A} = (A, U_2, \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket)$ and strategies $\sigma_1$ and $\sigma_2$ of $Q$-coloured arenas that make the diagram

$$\begin{tikzcd}
\llbracket \delta :: \Sigma \rrbracket \arrow[rr, "(\rho, \llbracket G \rrbracket)"] \arrow[dr, "(\sigma_1, \Lambda(\mathbf{g}))"'] & & \llbracket q_I :: o \rrbracket \\
& (A, U_2, \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket) \arrow[ur, "(\sigma_2, \mathbf{fix})"'] &
\end{tikzcd}$$

commutes.

By analysis of $\sigma_2$, there exist a $Q$-coloured arena $\Gamma$ (say) and hence a two-level arena $(\Gamma \Rightarrow \Gamma, U', \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket)$, a map $(\uparrow, \mathrm{id}_{\llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket})$ from $(A, U_2, \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket)$ to $(\Gamma \Rightarrow \Gamma, U', \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket)$, and a map $(\downarrow, \mathrm{id}_{\llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket})$ in the opposite direction, such that $\uparrow ; \downarrow = \mathrm{id}_{\mathcal{A}}$.

Thus we have the following commutative diagram:

$$\begin{tikzcd}
\llbracket \delta :: \Sigma \rrbracket \arrow[rr, "(\rho, \llbracket G \rrbracket)"] \arrow[dr, "(\sigma_1 ; \uparrow, \Lambda(\mathbf{g}))"'] & & \llbracket q_I :: o \rrbracket \ . \\
& (\Gamma \Rightarrow \Gamma, U', \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket) \arrow[ur, "(\downarrow ; \sigma_2, \mathbf{fix})"'] &
\end{tikzcd}$$

It follows from $U_1(\rho) = \llbracket G \rrbracket$ that $U(\sigma_1; \uparrow) = \Lambda(\mathbf{g})$ where $U$ is the relevant forgetful function. Hence, by uncurrying $(\sigma_1; \uparrow, \Lambda(\mathbf{g}))$, we obtain a total and hence winning strategy of $(\llbracket \delta :: \Sigma \rrbracket \times \llbracket \Gamma :: \mathcal{N} \rrbracket) \Rightarrow \llbracket \Gamma :: \mathcal{N} \rrbracket$ as desired. $\square$

### 6.2 Minimality of Traversals-induced Typing

Using the same notation as before, interaction sequences from $\mathbf{Int}(\Lambda(\mathbf{g}), \mathbf{fix}) \subseteq \mathrm{Int}(\llbracket \Sigma \rrbracket, \llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket, \llbracket o \rrbracket)$ form a tree, which is (in essence) the *traversal tree* in the sense of Ong [1].

*Prime types*, which are intersection types of the form $\theta = \bigwedge_{i \in I_1} \theta_{1i} \to \cdots \to \bigwedge_{i \in I_n} \theta_{ni} \to q$, are equivalent to *variable profiles* (or simply *profiles*) [1]. Precisely $\theta$ corresponds to profile $\widehat{\theta} := (\{ \widehat{\theta_{1i}} \mid i \in I_1 \}, \cdots, \{ \widehat{\theta_{ni}} \mid i \in I_n \}, q)$. We write profiles of ground kind as $q$, rather than $(q)$. Henceforth, we shall use prime types and profiles interchangeably.

Tsukada and Kobayashi [10] introduced (a kind-indexed family of) binary relations $\leq_\kappa$ between profiles of kind $\kappa$, and between sets of profiles of kind $\kappa$, by induction over the following rules.

(i) If for all $\theta \in A$ there exists $\theta' \in A'$ such that $\theta \leq_\kappa \theta'$ then $A \leq_\kappa A'$.
(ii) If $A_i \leq_{\kappa_i} A'_i$ for each $i$ then $(A_1, \ldots, A_n; q) \leq_{\kappa_1 \to \cdots \to \kappa_n \to o} (A'_1, \ldots, A'_n, q)$.

A *profile annotation* (or simply *annotation*) of the traversal tree $\mathbf{Int}(\Lambda(\mathbf{g}), \mathbf{fix})$ is a map of the nodes (which are move-occurrences of $M_{\llbracket \Sigma \rrbracket} + M_{\llbracket \mathcal{N} \rrbracket \Rightarrow \llbracket \mathcal{N} \rrbracket} + M_{\llbracket o \rrbracket}$) of the tree to profiles. We say that an annotation of the traversal tree is *consistent* just if whenever a move $m$, of kind $\kappa_1 \to \cdots \to \kappa_n \to o$ and simulates $q$, is annotated with profile $(A_1, \cdots, A_n, q')$, then (i) $q' = q$, (ii) for each $i$, $A_i$ is a set of profiles of kind $\kappa_i$,

(iii) if $m'$ is annotated with $\theta$ and $i$-points to $m$, then $\theta \in A_i$. Now consider *annotated moves*, which are moves paired with their annotations, written $(m, \theta)$. We say that a profile annotation is *innocent* just if whenever $u_1 \cdot (m_1, \theta_1)$ and $u_2 \cdot (m_2, \theta_2)$ are even-length paths in the annotated traversal tree such that $\ulcorner u_1 \urcorner = \ulcorner u_2 \urcorner$, then $m_1 = m_2$ and $\theta_1 = \theta_2$.

Every consistent (and innocent) annotation $\alpha$ of an (accepting) traversal tree gives rise to a typing environment, written $\Gamma_\alpha$, which is the set of bindings $F_i : \theta$ where $i \in \{ 1, \ldots, n \}$ and $\theta$ is the profile that annotates an occurrence of an initial move of $[\![\kappa_i]\!]$. Note that $\Gamma_\alpha$ is finite because there are only finitely many types of a given kind. We define a relation between annotations: $\alpha_1 \leq \alpha_2$ just if for each occurrence $m$ of a move of kind $\kappa$ in the traversal tree, $\alpha_1(m) \leq_\kappa \alpha_2(m)$.

**Theorem 7.** *(i) Let $\alpha$ be a consistent and innocent annotation of a traversal tree. Then $\Gamma_\alpha$ is a complete type environment.*

*(ii) There is $\leq$-minimal consistent and innocent annotation, written $\alpha_{\min}$. Then $\Gamma_{\alpha_{\min}} \leq \Gamma_\alpha$ meaning that for all $F : \theta \in \Gamma_{\alpha_{\min}}$ there exists $F : \theta' \in \Gamma_\alpha$ such that $\theta \leq \theta'$.*

*(iii) Every complete type environment $\Gamma$ determines a consistent and innocent annotation $\alpha_\Gamma$ of the traversal tree.*

### 6.3 Game-Semantic Proof of Completeness of GTRecS [9]

GTRecS [9] is a higher-order model checker proposed by Kobayashi. Although GTRecS is inspired by game-semantics, the formal development of the algorithm is purely type-theoretical and no concrete relationship to game semantics is known. Here we give a game-semantic proof of completeness of GTRecS based on two-level arena games.

The novelty of GTRecS lies in a function on type bindings, named **Expand**. For a set $\Gamma$ of nonterminal-type bindings, $\mathbf{Expand}(\Gamma)$ is defined as

$$\Gamma \cup \bigcup \{ \Gamma' \cup \{ F_i : \tau' \} \mid \Gamma \preceq_P \Gamma' \wedge \Gamma' \vdash \mathcal{R}(F_i) : \tau' \wedge \Gamma \preceq_O \{ F_i : \tau' \} \},$$

where $\Gamma' \vdash \mathcal{R}(F_i) : \tau'$ is relevant. Here for types $\tau_1$ and $\tau_2$, $\tau_1 \preceq_P \tau_2$ if the arena $[\![\tau_2]\!]$ is obtained by adding only proponent moves to $[\![\tau_1]\!]$. For example, $(\bigwedge \emptyset) \to q \preceq_P ((\bigwedge \emptyset) \to q') \to q$ but $(\bigwedge \emptyset) \to q \not\preceq_P (q'' \to q') \to q$, since $q'$ is at the proponent position and $q''$ at the opponent position. $\Gamma \preceq_P \Gamma'$ is defined as $\forall F : \tau' \in \Gamma'. \exists F : \tau \in \Gamma. \tau \preceq_P \tau'$. Similarly, $\tau \preceq_O \tau'$ and $\Gamma \preceq_O \Gamma'$ are defined.

Our goal is to analyse **Expand** game theoretically. The result is Lemma 21, which states that **Expand** overapproximates one step interaction of two strategies, $\sigma$ and **fix**. Completeness of GTRecS is a corollary of Lemma 21.

Fix a type environment $\Gamma$ and a winning strategy $\sigma : [\![\delta]\!] \longrightarrow (\Gamma^1 \Rightarrow \Gamma^2)$ (here we use superscripts to distinguish occurrences of $\Gamma$) that is induced from the derivation of $\vdash G : \Gamma$. The strategy $\mathbf{fix} : ([\![\Gamma^1]\!] \Rightarrow [\![\Gamma^2]\!]) \longrightarrow [\![q_I]\!]$ is defined as the composite of $([\![\Gamma]\!]^1 \Rightarrow [\![\Gamma]\!]^2) \xrightarrow{Y} [\![\Gamma]\!] \xrightarrow{\{ S : q_I \}} [\![q_I]\!]$. For $n \in \{1, 2, \ldots\}$, the *nth approximation of* **fix** is defined by $\lfloor \mathbf{fix} \rfloor_n = \{ s \in \mathbf{fix} \mid |s| \leq 2n + 1 \}$. Thus $\lfloor \mathbf{fix} \rfloor_n$ is a strategy that behaves like **fix** until the nth interaction, but stops after that. For the notational convenience, we define $\lfloor \mathbf{fix} \rfloor_\infty = \mathbf{fix}$.

Our goal is to show the concrete relationship between $\lfloor \mathbf{fix} \rfloor_n$ and $\mathbf{Expand}(\{S : q_I\})$ (Lemma 21). Completeness of GTRecS is an easy consequence of Lemma 21.

Let $n \in \{0, 1, 2, \ldots, \infty\}$. The nth approximation of **fix** induces approximation of arenas and strategies. An arena $\lfloor \Gamma^1 \Rightarrow \Gamma^2 \rfloor_n$ is defined as the restriction of $\Gamma^1 \Rightarrow \Gamma^2$ that consists of only moves appearing at $\mathbf{Int}(\sigma, \lfloor \mathbf{fix} \rfloor_n)$, and a strategy $\lfloor \sigma \rfloor_n : [\![\delta]\!] \longrightarrow \lfloor \Gamma^1 \Rightarrow \Gamma^2 \rfloor_n$ is the restriction of $\sigma$ to the arena.

**Lemma 18.** $\lfloor \Gamma^1 \rfloor_\infty = \lfloor \Gamma^2 \rfloor_\infty$ *and* $S : q_0 \in \lfloor \Gamma^2 \rfloor_\infty$.

*Proof.* Easy. □

*Remark 2.* It is not necessarily the case that $\lfloor \Gamma^1 \rfloor_\infty = \Gamma^1$ or $\lfloor \Gamma^2 \rfloor_\infty = \Gamma^2$.

**Lemma 19.** *For* $n \in \{0, 1, \ldots, \infty\}$, $\lfloor \sigma \rfloor_n$ *is a full and winning strategy.*

*Proof.* All properties other than totality come from the fact that $\lfloor \sigma \rfloor_n$ is a restriction of $\sigma$. To prove totality, a key observation is that every maximal sequence $s \in \lfloor \mathbf{fix} \rfloor_n$ ends with a O-move. Thus every maximal interaction sequence $s \in \mathbf{Int}(\sigma, \lfloor \mathbf{fix} \rfloor_n)$ ends with a O-move of $(\Gamma^1 \Rightarrow \Gamma^2) \to \langle q_0 \rangle$, since $\sigma$ is contingent complete and total. Therefore if $sm \in \lfloor \sigma \rfloor_n$ is maximal, then $m$ is a *P-move* of $(C \times \Gamma^1 \Rightarrow \Gamma^2)$. $\lfloor \sigma \rfloor_n$ is full by definition. □

If $\Gamma = \{F_i : \bigwedge_j \tau_{i,j} \mid F_i \in \mathcal{N}\}$, then the arena $[\![\delta]\!] \Rightarrow (\Gamma^1 \Rightarrow \Gamma^2)$ can be decomposed as $\prod_{i,j}([\![\delta]\!] \Rightarrow (\Gamma^1 \Rightarrow \tau_{i,j}))$. By the same way, the arena $[\![\delta]\!] \Rightarrow \lfloor \Gamma^1 \Rightarrow \Gamma^2 \rfloor_n$ is decomposed as $\prod_{i,j}([\![\delta]\!] \Rightarrow (\lfloor \Gamma^1 \rfloor_{n,i,j} \Rightarrow \lfloor \tau_{i,j} \rfloor_n))$.

Let $\lfloor \Gamma^1 \rfloor_n$ be the union of variable-type bindings corresponding to $\bigcup_{i,j} \lfloor \Gamma^1 \rfloor_{n,i,j}$ and $\lfloor \Gamma^2 \rfloor_n$ be the set of type bindings $\{F_i : \lfloor \tau_{i,j} \rfloor_n\}_{i,j}$.

**Lemma 20.** *For all* $n \in \{0, 1, \ldots, \infty\}$, *we have* $(\lfloor \Gamma^1 \rfloor_n \cup \lfloor \Gamma^2 \rfloor_n) \preceq_O \lfloor \Gamma^1 \rfloor_{n+1}$ *and* $(\lfloor \Gamma^1 \rfloor_n \cup \lfloor \Gamma^2 \rfloor_n) \preceq_P \lfloor \Gamma^2 \rfloor_{n+1}$ *(here* $\infty + 1 = \infty$*).*

*Proof.* Assume that $\lfloor \Gamma^1 \rfloor_n \cup \lfloor \Gamma^2 \rfloor_n \subsetneq \lfloor \Gamma^1 \rfloor_{n+1} \cup \lfloor \Gamma^2 \rfloor_{n+2}$. Let $m$ be an element of their difference. By definition of $\lfloor \Gamma^1 \rfloor_{n+1}$ and $\lfloor \Gamma^2 \rfloor_{n+1}$, there is a sequence $sm \in \mathbf{Int}(\Lambda(\sigma), \lfloor \mathbf{fix} \rfloor_{n+1}$ ending with $m$. Let $s'm_0$ be the maximal prefix of $sm$ such that $s'm_0 \in \mathbf{Int}(\Lambda(\sigma), \lfloor \mathbf{fix} \rfloor_n)$. Then $m_0$ is a O-move of $\Gamma^1$ or a P-move of $\Gamma^2$. (Otherwise $m_0$ must be a move of $C$, that implies $s'm_0$ is also maximal in $\mathbf{Int}(\Lambda(\sigma), \mathbf{fix})$, but this contradict to existence of its extension $sm \in \mathbf{Int}(\Lambda(\sigma), \lfloor \mathbf{fix} \rfloor_{n+1})$.) By the definition of $\lfloor \mathbf{fix} \rfloor_{n+1}$, $sm = s'm_0m'_0s''m$ for some $s''$, where $m'_0$ corresponds to $m_0$. This interaction sequence is maximal. Therefore $m$ is an O-move of $\Gamma^1$ or a P-move of $\Gamma^2$. Moreover $m$ is justified by a move of the view of $s'm_0m'_0s''$ , i.e., a move of $\lfloor \Gamma^1 \rfloor_n \cup \lfloor \Gamma^2 \rfloor_n$ or their counterpart. So the proposition holds. □

By Lemma 17 and Lemma 19, we have a relevant derivation of $\delta \cup \lfloor \Gamma^1 \rfloor_{n,i,j} \vdash \mathcal{R}(F_i) : \tau_{i,j}$. Combination of these derivations and Lemma 20 leads to the next lemma.

**Lemma 21.** $\lfloor \Gamma^1 \rfloor_n \cup \lfloor \Gamma^2 \rfloor_n \subseteq \mathbf{Expand}^n(\{S : q_0\})$.

*Proof.* By induction on $n$. The case $n = 0$ is trivial since $\lfloor \Gamma^1 \rfloor_0 = \emptyset$ and $\lfloor \Gamma^2 \rfloor_0 = \{S : q_0\}$.

Suppose $\lfloor \Gamma^1 \rfloor_n \cup \lfloor \Gamma^2 \rfloor_n \subseteq \mathbf{Expand}^n(\{S : q_0\})$. We use the following proposition (see [9, Appendix C].)

> If $\Gamma \preceq_O \Gamma'$ and $\theta \preceq_P \theta'$ and $\Gamma' \vdash \mathcal{R}(F) : \theta'$, then $\Gamma' \cup \{F : \theta'\} \subseteq \mathbf{Expand}(\Gamma \cup \{F : \theta\})$.

The lemma follows from the proposition and previous lemmas. □

**Conclusions and Further Directions** Two-level arena games are an accurate model of intersection types. Thanks to Subject Expansion, they are a useful semantic framework for reasoning about higher-order model checking.

For future work, we aim to (i) consider properties that are closed under disjunction and quantifications, and (ii) study a call-by-value version of intersection games. In orthogonal directions, it would be interesting to (iii) construct an intersection game model for untyped recursion schemes [10], and (iv) build a CCC of intersection games parameterised by an alternating parity tree automaton, thus extending our semantic framework to mu-calculus properties.

*Acknowledgement* This work is partially supported by Kakenhi $22 \cdot 3842$ and EPSRC EP/F036361/1. We thank Naoki Kobayashi for encouraging us to think about game-semantic proofs and for insightful discussions.

## References

1. Ong, C.H.L.: On model-checking trees generated by higher-order recursion schemes. In: LICS, IEEE Computer Society (2006) 81–90
2. Hyland, J.M.E., Ong, C.H.L.: On full abstraction for PCF: I, II, and III. Inf. Comput. **163**(2) (2000) 285–408
3. Kobayashi, N.: Types and higher-order recursion schemes for verification of higher-order programs. In Shao, Z., Pierce, B.C., eds.: POPL, ACM (2009) 416–428
4. Salvati, S.: On the membership problem for non-linear abstract categorial grammars. Journal of Logic, Language and Information **19**(2) (2010) 163–183
5. Kobayashi, N., Ong, C.H.L.: A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In: LICS, IEEE Computer Society (2009) 179–188
6. Hague, M., Murawski, A.S., Ong, C.H.L., Serre, O.: Collapsible pushdown automata and recursion schemes. In: LICS. (2008) 452–461
7. Salvati, S., Walukiewicz, I.: Krivine machines and higher-order schemes. In: ICALP (2). (2011) 162–173
8. Nielson, F.: Two-level semantics and abstract interpretation. Theor. Comput. Sci. **69**(2) (1989) 117–242
9. Kobayashi, N.: A practical linear time algorithm for trivial automata model checking of higher-order recursion schemes. In Hofmann, M., ed.: FOSSACS. Volume 6604 of Lecture Notes in Computer Science., Springer (2011) 260–274
10. Tsukada, T., Kobayashi, N.: Untyped recursion schemes and infinite intersection types. In Ong, C.H.L., ed.: FOSSACS. Volume 6014 of Lecture Notes in Computer Science., Springer (2010) 343–357