Dinaturality Meets Genericity: A Game Semantics of Bounded Polymorphism

James Laird ✉
Department of Computer Science, University of Bath, UK

Abstract

We study subtyping and parametric polymorphism, with the aim of providing direct and tractable semantic representations of type systems with these expressive features. The *liveness order* uses the Player-Opponent duality of game semantics to give a simple representation of subtyping: we generalize it to include graphs extracted directly from second-order intuitionistic types, and use the resulting complete lattice to interpret bounded polymorphic types in the style of System $\mathsf{F}_{<:}$, but with a more tractable subtyping relation.

To extend this to a semantics of terms, we use the type-derived graphs as arenas, on which strategies correspond to dinatural transformations with respect to the canonical coercions ("on the nose" copycats) induced by the liveness ordering. This relationship between the interpretation of generic and subtype polymorphism thus provides the basis of the semantics of our type system.

2012 ACM Subject Classification Theory of computation → Program semantics

Keywords and phrases Subtyping, Bounded Polymorphism, Game Semantics, Dinaturality

Digital Object Identifier 10.4230/LIPIcs.FSCD.2023.33

1 Introduction and Related Work

Subtype and parametric polymorphism both provide powerful principles for data abstraction. Combining them via bounded quantification increases this expressive power: they may be used to write programs which are generic, but range over a constrained set of types (a program of type $\forall(X <: S).T$ may be instantiated only with a subtype of S). They have been used to develop formal theories of key aspects of object oriented languages such as inheritance [3, 17]. This combination is not without its challenges: for example, discovering type systems in which the fundamental problems, such as typechecking of terms, are efficiently decidable [19].

Our aim is to describe and relate simple, concrete notions of subtype and generic, parametric polymorphism and show that they can work together to give an interpretation of bounded polymorphism which is both tractable and intensional, yielding a formal semantic account of the constraints on behaviour which can be expressed in such a setting. This allows for models which combine bounded polymorphism with computational effects (in particular, state) and, potentially, for semantics-based subtyping theories which capture aspects of program behaviour.

Earlier models of subtyping, and bounded polymorphism in particular, are based on realizability-style interpretations (partial equivalence relations) [2]. These have made an important contribution to the semantic understanding and development of typing systems such as System $\mathsf{F}_{<:}$ [4, 7], but do not give a direct and effective characterization of the subtyping relation. Indeed, the subtyping relation of System $\mathsf{F}_{<:}$ (which they validate) is itself problematic from an algorithmic point of view – in particular, it is undecidable [19]. There is a body of work dedicated to giving typing systems for bounded quantification which are more tractable, but still expressive [12, 23, 10]. As advocated in [5], semantics should be a guide to this search, and this is one of our motivations. We show that our interpretation of subtyping may be used to interpret a typing system which subsumes several proposed

restrictions of System $\mathsf{F}_{<:}$ by distinguishing the introduction and elimination forms of bounded quantification: in [16] this type system, and its subtyping and type-checking algorithms is studied in more detail.

The nature of the subtype order makes it difficult to capture in a simple and finitary way using semantic structures such as domains: a subtype may be a restriction of its supertype (e.g. signed and unsigned integers) or an extension of it (e.g. record types). This dependence on interaction with consuming contexts can be captured via the Player/Opponent duality of game semantics in the *liveness ordering* [6]. We generalize this ordering to graph structures which may be derived from AJM-style games, directly from types themselves via the subtype ordering, and Hyland-Ong style arenas (which are are a widely-used basis for sequential and concurrent semantics of functional and object-oriented programming languages), and and use its lattice structure to construct an interpretation of bounded quantification.

The game semantics of parametric polymorphism underlying our interpretation is a novel presentation of the well-bracketed second order games model [13, 14], which is based on instantiating pairs of question-answer moves with copycat behaviour, structure which is made more immediate by using second-order types as moves. The glue which binds this to subtyping (and allows for a sound model of bounded quantification) is the *dinaturality* of strategy instantiation with respect to copycat strategies – canonical subtyping coercions which (as we show) are characteristic of the liveness order. Dinaturality has been proposed as a core semantic principle for modelling polymorphism [1] but as a general property is neither preserved by composition, nor possessed by all terms of System F [8]. However, copycat dinaturality is the key to soundly modelling bounded abstraction and instantiation.

1.1 Contribution of this work

This work establishes the relevance and tractability of the calculus for bounded polymorphism described in [16] by giving a concrete denotational semantics for it, in a setting (HO game semantics) which is readily extendable with relevant computational effects such as stateful objects. Giving such a semantics requires the integration of two kinds of polymorphism – subtyping and parametric polymorphism. The more general contribution is to show that this can be achieved in such a setting, using dinaturality and copycat strategies to relate rather diverse elements of game semantics – the liveness order and the well-bracketing condition. Finally, by generalising the liveness order, and showing that it may be described directly at the syntactic level, on types, it is hoped to draw it to wider attention as a way of understanding and studying subtyping.

2 Subtyping Graphs

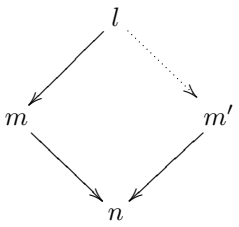
The game semantic interpretation of subtyping as a *liveness ordering* was introduced by Chroboczek [6] for games presented positionally, as sequences of moves. Here, we generalize it to graph structures of disjoint sets of nodes, which include Hyland-Ong (HO) style arenas [11], which provide a general setting for interpreting types in sequential and concurrent game semantics. In the next section we will describe the derivation of these graphs directly from the subtyping order on second-order types.

▶ **Definition 1.** *A graph-arena is a rooted, directed graph with a partition of its non-root nodes: given as a tuple* $(O, P, \triangleright)$ *consisting of :*

- *Disjoint sets of nodes* O *and* P*, not containing the distinguished root node* $\top$*.*
- *An edge relation* $\triangleright \subseteq (O \cup P \cup \{\top\}) \times (O \cup P)$

with the following properties:

Well-Foundedness. *There is no infinite chain* $\ldots \triangleright m_{i+1} \triangleright m_i \triangleright \ldots \triangleright m_0$
Quasi-Arborescence. *If* $l \triangleright m$, $m \triangleright n$ *and* $m' \triangleright n$ *then* $l \triangleright m'$.



Quasi-Partition. *If* $m \triangleright n$ *and* $m' \triangleright n$ *then* $m, m' \in O$ *or* $m, m' \in P$

Fixing a graph $U = (O, P, \triangleright)$, for any set of nodes $A \subseteq O \cup P$, let $E(A) = \{u \in O \cup P \mid \exists u \in A \cup \{\top\}.u \triangleright v\}$ be the set of vertices accessible from A. Writing $\triangleright_A$ for the restriction of $\triangleright$ to $A \cup \{\top\}$:

▶ **Definition 2.** *A sub-arena of* U *is a non-empty subset* $A \subseteq O \cup P$ *such that* $\top \triangleright_A^* a$ *for all* $a \in A$ *(i.e. a root-connected subgraph of* U*).*

▶ **Proposition 3.** $A \subseteq O \cup P$ *is a sub-arena if and only if* $A \subseteq E(A)$.

Proof. Evidently, if A is root-connected, every node in A is connected to one in $A \cup \{\top\}$. The converse follows by Noetherian induction: if a is initial then it is evidently hereditarily enabled in A. Otherwise there exists $a' \in A$ such that $a' \triangleright a$. By hypothesis, this is connected to $\top$ through A and hence so is a. ◀

Partitioning $E(A)$ into the sets $P(A) = E(A) \cap P$ and $O(A) = E(A) \cap O$, we now define the liveness order on sub-arenas.

▶ **Definition 4.** *Let* $(\mathcal{S}(U), \preccurlyeq)$ *be the partially ordered set of sub-arenas of* U*, where* $\preccurlyeq$ *is the liveness order:*

$A \preccurlyeq B$ *if and only if* $O(A) \cap B \subseteq A$ *and* $P(B) \cap A \subseteq B$.

In other words, $A \preccurlyeq B$ if all of the P-nodes in A which are accesible from B are already in B, and all O-nodes in B which are accessible from A are in A.

For example AJM-style games are given by sets of plays (alternating sequences over a set of moves partitioned between Opponent and Proponent). Their arena graphs are given by taking O and P to be the sets of plays ending in Proponent and Opponent moves, respectively, with an edge from s to t if the latter has the form sa. In this case $A \preccurlyeq B$ if for any Opponent move a, $s \in A$ and $sa \in B$ implies $sa \in A$, and for any Proponent move b, $s \in B$ and $sb \in A$ implies $sb \in B$. This is the liveness ordering defined in [6].

The following lemma is used to show that this is a well-defined partial order,

▶ **Lemma 5.** *If* $A \preccurlyeq B$ *and* $B \preccurlyeq C$ *then* $E(A) \cap E(C) \subseteq E(B)$.

Proof. By Noetherian induction. Suppose $b \in E(A) \cap E(C)$. If $\top \triangleleft b$ then $b \in E(B)$ as required. Otherwise, there exist $a \in A$ and $c \in C$ such that $a \triangleright b$ and $c \triangleright b$, and $a, c \in P$ or $a, c \in O$ by quasi-partition. Supposing the former, by connectedness of A there exists $a' \in A \cup \{\top\}$ such that $a' \triangleright a$. By quasi-arboresecence, $a' \triangleright c$, and so $c \in E(A) \cap E(C)$. By induction hypothesis, $c \in E(B)$ and so $c \in P(B) \cap C$. Thus $c \in B$ (since $B \preccurlyeq C$) and so $b \in E(B)$ as required. The case where $a, c \in O$ is symmetric. ◀

▶ **Proposition 6.** *$(\mathcal{S}(U), \preccurlyeq)$ is a partial order.*

Proof. Reflexivity is evident.

For transitivity, suppose $A \preccurlyeq B$ and $B \preccurlyeq C$. Then $O(A) \cap C \subseteq O(A) \cap E(C) \cap C$ by connectedness of C
$\subseteq O(A) \cap O(B). \cap C$ (by Lemma 5)
$\subseteq O(A) \cap B$ (since $B \preccurlyeq C$)
$\subseteq A$ (since $A \preccurlyeq B$). By symmetry, $P(C) \cap A \subseteq C$.

For antisymmetry, suppose $A \preccurlyeq B$ and $B \preccurlyeq A$. Then $A = A \cap E(A)$ by connectedness
$\subseteq A \cap E(B)$ by Lemma 5
$= A \cap O(B) \cup P(B) = (A \cap O(B)) \cup (A \cap P(B))$
$\subseteq B \cup B = B$ as $A \preccurlyeq B$ and $B \preccurlyeq A$.

By symmetry, $B \subseteq A$ and hence $A = B$ as required. ◀

In fact $(\mathcal{S}(U), \preccurlyeq)$ is a complete lattice.

▶ **Definition 7.** *The maximal sub-arena of a set $X \subseteq O \cup P$ is $X^\circ = \bigcup\{A \in \mathcal{S}(U) \mid A \subseteq X\}$.*

X° is root-connected – $X^\circ = \bigcup\{A \in \mathcal{S}(U) \mid A \subseteq X\} \subseteq \bigcup\{E(A) \in \mathcal{S}(U) \mid A \subseteq X\} = E(X^\circ)$ – and thus a well-defined sub-arena.

▶ **Lemma 8.** *For any $X \subseteq O \cup P$, $E(X^\circ) \cap X = X^\circ$*

Proof. $X^\circ \subseteq E(X^\circ)$ and $X^\circ \subseteq X$ by definition. Conversely, suppose $x \in X \cap E(X^\circ)$. Then $X^\circ \cup \{x\} \subseteq X$ and $X^\circ \cup \{x\} \subseteq E(X^\circ)$ and so $X^\circ \cup \{x\} \subseteq X^\circ$ – i.e. $x \in X^\circ$ and so $X \cap E(X^\circ) \subseteq X^\circ$ as required. ◀

The $\preccurlyeq$-infimum of a set of sub-arenas $\Delta \subseteq \mathcal{S}(U)$ is the maximal sub-arena of the maximal subset of $\bigcup \Delta$ in which any P-node which has an enabling node in two sub-arenas $A, B \in \Delta$ is in their intersection.

▶ **Definition 9.** *Given $\Delta \subseteq \mathcal{S}(U)$, let*

$$\underline{\bigwedge \Delta} = \{m \in \bigcup \Delta \mid m \in P(A) \cap P(B) \implies m \in A \cap B\}$$

and define

$$\bigwedge \Delta = (\underline{\bigwedge \Delta})^\circ.$$

▶ **Proposition 10.** *$\bigwedge \Delta$ is the greatest lower bound of Δ in $\mathcal{S}(U)$.*

Proof. We suppose $A \in \Delta$ and show that $\bigwedge \Delta \preccurlyeq A$. First if $m \in O(\bigwedge \Delta) \cap A$ then $m \in \underline{\bigwedge \Delta}$, as the condition $m \notin P(B) \cap P(C)$ for all $B, C \in \Delta$. So $m \in E(\bigwedge \Delta) \cap \underline{\bigwedge \Delta}$
$= \bigwedge \Delta$ by Lemma 8.
Now suppose $m \in P(A) \cap \bigwedge \Delta$, so that $m \in B \subseteq E(B)$ for some $B \in \Delta$.
Then $m \in P(A) \cap P(B)$, implying that $m \in A \cap B \subseteq A$ as required, since $m \in \underline{\bigwedge \Delta}$.

Now we suppose $C \preccurlyeq A$ for all $A \in \Delta$ and show that $C \preccurlyeq \bigwedge \Delta$. First, if $m \in O(C) \cap \bigwedge \Delta$ then $m \in O(C) \cap A$ for some $A \in \Delta$ with $C \preccurlyeq A$ and so $m \in C$ as required. Now suppose $m \in P(\bigwedge \Delta) \cap C$. Then $m \in P(A) \cap C \subseteq A$ for some $A \in \Delta$, so $A \in \bigcup \Delta$. If $m \in P(A) \cap P(B)$ for $A, B \in \Delta$ (so $C \preccurlyeq A, B$) then $m \in P(A) \cap P(B) \cap (C) \subseteq A \cap B$ – i.e. $m \in P(A) \cap P(B))$ implies $m \in A \cap B$ and thus $m \in \underline{\bigwedge \Delta}$. So $m \in E(\bigwedge \Delta) \cap \underline{\bigwedge \Delta} = \bigwedge \Delta$ by Lemma 8. ◀

3 Arena Graphs from Types

As an example showing the relationship between the liveness ordering and subtyping, (and a step towards a semantics of bounded quantification) we derive arena-graphs directly from the subtyping relation of System $\mathrm{F}_\top$, which is System F (the second-order λ-calculus [9, 21]) extended with with products and a supertype $\top$ [4] – i.e. its set of types is given by the grammar:

$$S ::= \top \mid X \mid S \to S \mid S \times S \mid \forall X.S$$

The $\top$ type allows for a for non-trivial subtyping relation, given by the following derivation rules:

$$\frac{}{T <: \top}\ \mathsf{Top} \qquad \frac{}{T <: T}\ \mathsf{Refl} \qquad \frac{T <: T' \qquad T' <: T''}{T <: T''}\ \mathsf{Trans}$$

$$\frac{S' <: S \qquad T <: T'}{S \to T <: S' \to T'} \to \qquad \frac{S <: S' \qquad T <: T'}{S \times T <: S' \times T'} \times \qquad \frac{T <: T'}{\forall X.T <: \forall X.T'} \forall$$

For example, the standard System F representation of the Booleans – the type $\forall X.X \to X \to X$ has the subtypes $\forall X.\top \to X \to X$, $\forall X.X \to \top \to X$ and $\forall X.\top \to \top \to X$.

3.1 Type Arenas for System $\mathbf{F}_\top$

We now derive an arena-graph in which to interpret System $\mathrm{F}_\top$ subtyping. The sets O and P of O-nodes and P-nodes (respectively) consist of the types given by the grammars:

$$o ::= X \mid \top \to o \mid p \to \top \mid \top \times o \mid o \times \top \mid \forall X.o$$
$$p ::= \top \to p \mid o \to \top \mid \top \times p \mid p \times \top \mid \forall X.p$$

where X ranges over an unbounded set of *type variables*. Note that each such *node-type* m contains exactly one occurrence of a type-variable X, which may be bound or free, so we may write it as $m[X]$. It is an O-node if X occurs "positively" and a P-node if it occurs "negatively".

The edge relation for our second-order type arena-graph is derived from the subtyping order. Let $\lessdot$ be the covering relation for its restriction to $O \cup P \cup \{\top\}$ – i.e. $m \lessdot m'$ if $m <: m'$, $m \neq m'$ and if $m <: m'' <: m'$ then $m'' = m$ or $m'' = m'$.

▶ **Definition 11.** *Let $\triangleright$ be the least relation on $(O \cup P \cup \{\top\}) \times (O \cup P)$ such that:*

- *If o is $\lessdot$-minimal (i.e. $o' \lessdot o$ implies $o = o'$) then $\top \triangleright o$.*
- *If $p \lessdot p'$, $p' \triangleright o$ and $o \lessdot p'$ then $o \triangleright p$.*
- *If $o \lessdot o'$, $o \triangleright p$ and $o' \lessdot p$ then $p \triangleright o$.*

In other words, a node m is initial (enabled by $\top$) if its type variable occurs on the right of every arrow ($\to$) in m. Otherwise it has the form $C[m' \to \top]$, for some unique initial node m', and is enabled by any node of the form $C[\top \to n]$, where n is an initial node. Quasi-arboresecence follows from this characterization: while $\triangleright$ is not a tree, (e.g. $\top \to (X \times \top) \triangleright X \to \top$ and $\top \to (\top \times X) \triangleright X \to \top$, both $\top \to (X \times \top)$ and $\top \to (\top \times X)$ are initial. Hence $(O, P, \triangleright)$ is an arena – it is a bipartite graph by construction, and it is well-founded, as any chain $\ldots o_{n+1} \triangleright p_n \triangleright o_n \triangleright \ldots \triangleright p_1 \triangleright o_1$ contains an infinite descending chain $\ldots <: o_{n+1} <: o_n <: \ldots <: o_1$: a straightforward induction establishes that no such chain exists.

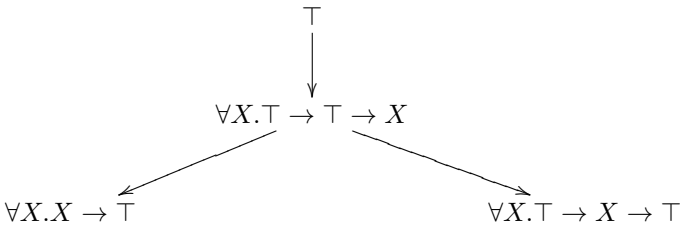
We may now define a sub-arena (its *type-arena*) for each type.

▶ **Definition 12.** *Let $\sqsubseteq_\top$ be the least congruence on types such that $S \sqsubseteq_\top \top$ for all types S. Then for any type T, let $[\![T]\!] = \{m \in O \cup P \| \ m \sqsubseteq_\top T\}^\circ$*

In other words, $[\![T]\!]$ is the (root-connected) set of nodes which may be obtained from T by replacing subterms of T with $\top$. For example, the Boolean type $\forall X.X \to X \to X$ denotes the set containing the O-node $\forall X.\top \to X$ and the P-nodes $\forall X.\top \to X \to \top$ and $\forall X.X \to \top$. The edge-relation restricts to these nodes as follows:

- $\top \triangleright \forall X.\top \to \top \to X$, since $\forall X.\top \to \top \to X$ is a $<:$-minimal type.
- $\forall X.\top \to \top \to X \triangleright \forall X.\top \to X \to \top, \forall X.X \to \top$ since $\top \triangleright \forall X.\top \to \top \to X$, and $\forall X.\top \to \top \to X <: \forall X.\top \to X \to \top, \forall X.X \to \top$, $\forall X.\top \to X \to \top, \forall X.X \to \top <: \top$.

yielding the expected graph structure:



Type-arenas may also be defined compositionally:

- $[\![\top]\!] = \varnothing$, $[\![X]\!] = \{X\}$,
- $[\![S \times T]\!] = [\![S]\!] \times [\![T]\!] \triangleq \{m \times \top \mid m \in S\} \cup \{\top \times m \mid m \in T\}$,
- $[\![S \to T]\!] = [\![S]\!] \to [\![T]\!] \triangleq (\{m \to \top \mid m \in S\} \cup \{\top \to m \mid m \in T\})^\circ$,
- $[\![\forall X.S]\!] = \forall X.M \triangleq \{\forall X.m \mid m \in S\}$.

Using this decomposition, and the soundness of each of the subtyping derivation rules: e.g. if $A' \preccurlyeq A$ and $B \preccurlyeq B'$ then $A \to B \preccurlyeq A' \to B'$, it follows that subtyping is sound with respect to the liveness order:

▶ **Proposition 13.** *If $S <: T$ then $[\![S]\!] \preccurlyeq [\![T]\!]$.*

4 Bounded Quantification

We now describe an interpretation of bounded (universal) quantification types. The standard typing system for the second-order λ-calculus with such types is System $\mathsf{F}_{<:}$ [7, 4, 18], in which terms of type $\forall(X <: S).T$ may only be instantiated with a subtype of S. (Thanks to the presence of a $<:$-greatest type $\top$, System F may be viewed as a proper subsystem of System $\mathsf{F}_{<:}$ by reading unbounded quantification $\forall X.S$ as the bounded quantification $\forall(X <: \top).S$.) However, System $\mathsf{F}_{<:}$ lacks reasonable algorithmic characteristics – in particular, its subtyping relation (and thus also typechecking) is undecidable [19]. The culprit is the rule for subtyping bounded quantification:

$$\frac{\mathcal{E} \vdash T_0 <: S_0 \qquad \mathcal{E}, X <: T_0 \vdash S_1 <: T_1}{\mathcal{E} \vdash \forall(X <: S_0).S_1 <: \forall(X <: T_0).T_1} \forall - \mathsf{Orig}$$

where $\mathcal{E}$ is a subtyping context – a sequence of subtyping assumptions $X_1 <: S_1, \ldots, X_n <: S_n$ such that $X_1, \ldots, X_{i-1} \vdash S_i$ for each $1 <: i <: n$.

This problem is also reflected in the difficulty of giving a direct game semantics of the System $\mathsf{F}_{<:}$ subtyping relation. The game semantics for bounded quantification given in [15] constructs a subtyping relation from the derivation system: in particular, the above rule

does not respect the liveness in that model. Various modifications to $\mathsf{F}_{<:}$ have been proposed with the aim of giving a more tractable system [12, 23, 10, 5], including three restrictions of this rule.

The first restricts the subtype order to quantified types which have identical bounds:

$$\frac{\mathcal{E}, X <: S \vdash T <: T'}{\mathcal{E} \vdash \forall(X <: S).T <: \forall(X <: S).T'} \forall - \mathsf{Fun}$$

This yields a calculus, Kernel $\mathsf{F}_{<:}$, which is well-behaved (subtyping and type-checking are efficiently decidable) at the cost of expressiveness.

The second rule (proposed by Castagna and Pierce as the basis of System $\mathsf{F}^{\top}_{<:}$ [5]) does not use assumptions about the bounds on variables when inferring the subtype relation between the bodies.

$$\frac{\mathcal{E} \vdash T_0 <: S_0 \qquad \mathcal{E}, X <: \top \vdash S_1 <: T_1}{\forall(X <: S_0).S_1 <: \forall(X <: T_0).T_1} \forall - \mathsf{Top}$$

It has an expressive subtyping relation with nice properties, including decidability. However, it lacks the minimal typing property, and thus an evident typechecking algorithm.

The third rule [5] uses the greater of the two bounds when inferring the subtype relation between the bodies.

$$\frac{\mathcal{E} \vdash T_0 <: S_0 \qquad \mathcal{E}, X <: S_0 \vdash S_1 <: T_1}{\mathcal{E} \vdash \forall(X <: S_0).S_1 <: \forall(X <: T_0).T_1} \forall - \mathsf{Loc}$$

This is expressive (subsuming both rules $\forall - \mathsf{Fun}$ and $\forall - \mathsf{Top}$) but lacks a sound and complete (let alone, decidable) subtyping algorithm.

In [16], it is shown that these three rules may coexist in a single system ($\mathsf{F}^{\mathsf{F}\top}_{<:}$) for subtyping bounded quantification by distinguishing the introduction and elimination forms of bounded quantification, and making the former a subtype of the latter. This avoids the pitfalls of $\forall - \mathsf{Top}$ and $\forall - \mathsf{Loc}$ but allows for a more expressive typing relation than Kernel $\mathsf{F}_{<:}$.

4.1 System $\mathsf{F}^{\mathsf{F}\top}_{<:}$

The type system (System $\mathsf{F}^{\mathsf{F}\top}_{<:}$) decorates bounded quantifiers with the superscripts $\{\mathsf{F}, \top\}$ (F for the introduction form, which obeys the rule of Kernel $\mathsf{F}_{<:}$ and $\top$ for the elimination form, which obeys that of System $\mathsf{F}^{\top}_{<:}$). Raw types are given by the grammar:

$$T ::= \top \mid X \mid T \to T \mid \forall^{\mathsf{F}}(X <: T).T \mid \forall^{\top}(X <: T).T$$

Table 1 gives rules defining subtyping judgments, $\mathcal{E} \vdash S <: T$, where $\mathcal{E}$ is a context of type variable bounds. These replace the single original typing rule for bounded quantification of System $\mathsf{F}_{<:}$ with the rules $\forall - \mathsf{Top}$, $\forall - \mathsf{Fun}$ and $\forall - \mathsf{Loc}$.

4.2 Semantics of Bounded Quantification

To interpret bounded quantification, we first observe that type-variable substitution can be extended to arenas. Node-types (which properly come with a context of free variables $\Theta \vdash m$) are closed under substitution.

▶ **Lemma 14.** *Given nodes $\Theta, X, \Theta'' \vdash m$ and $\Theta, \Theta' \vdash n$, the substitution of n for any free occurrence of X in m yields a node $\Theta, \Theta', \Theta'' \vdash m[n/X]$.*

Table 1 Subtyping Rules for System $\mathsf{F}^{\mathsf{F}\top}_{<:}$.

$$\frac{\Theta, X <: T, \Theta' \vdash \top}{\Theta, X <: T, \Theta' \vdash X <: T}\ \mathsf{Var} \qquad \frac{\Theta \vdash T}{\Theta \vdash T <: \top}\ \mathsf{Top} \qquad \frac{\Theta \vdash T}{\Theta \vdash T <: T}\ \mathsf{Refl}$$

$$\frac{\Theta \vdash T <: T' \qquad \Theta \vdash T' <: T''}{\Theta \vdash T <: T''}\ \mathsf{Trans}$$

$$\frac{\Theta \vdash S' <: S \quad \Theta \vdash T <: T'}{\Theta \vdash S \to T <: S' \to T'} \to \qquad \frac{\Theta, X <: S \vdash T <: T'}{\Theta \vdash \forall^{\mathsf{F}}(X <: S).T <: \forall^{\mathsf{F}}(X <: S).T'}\ \forall - \mathsf{Fun}$$

$$\frac{\Theta \vdash T_0 <: S_0 \qquad \Theta, X <: S_0 \vdash S_1 <: T_1}{\Theta \vdash \forall^{\mathsf{F}}(X <: S_0).S_1 <: \forall^{\top}(X <: T_0).T_1}\ \forall - \mathsf{Loc}$$

$$\frac{\Theta \vdash T_0 <: S_0 \qquad \Theta, X <: \top \vdash S_1 <: T_1}{\Theta \vdash \forall^{\top}(X <: S_0).S_1 <: \forall^{\top}(X <: T_0).T_1}\ \forall - \top$$

Noting that $m[n/X] \in O$ if $m, n \in O$ or $m, n \in P$, and $m[n/X] \in P$ otherwise, we define substitution into sub-arenas separately for negative and positive occurrences. Writing $\Theta \vdash A$ if $\Theta \vdash m$ for every $m \in A$:

▶ **Definition 15.** *The substitution of sub-arenas $\Theta, \Theta' \vdash B, C$ into $\Theta, X, \Theta'' \vdash A$ is defined:*

$$\Theta, \Theta', \Theta'' \vdash A(B,C)_X = \{m[n/X] \mid m \in A \cap P, n \in B\} \cup \{m[n/X] \mid m \in A \cap O, n \in C\}^{\circ}.$$

This substitution operation is antitone (with respect to the liveness ordering) in the first argument and monotone in the second:

▶ **Lemma 16.** *If $B' \preccurlyeq B$ and $C \preccurlyeq C'$ then $A(B,C)_X \preccurlyeq A(B',C')_X$.*

Proof. Suppose $m \in E(A(B,C)_X) \cap O(A(B',C')_X)$. Then $m = m'[n/X]$ for some $m' \in A$ and $n \in B' \cup C'$. There are three possibilities.

- $m' \in O(A)$ and $n \in O(C')$. Then $n \in E(C) \cap O(C')$ and so $n \in C$ and $m = m'[n/X] \in A(B,C)_X$ as required.
- $m' \in P(A)$ and $n \in P(B')$. Then $n \in E(B) \cap P(B')$ and so $n \in B$ and $m = m'[n/X] \in A(B,C)_X$ as required.
- X is not free in m' – then $m = m' \in A(B,C)_X$ as required.

Similarly, $E(A(B',C')_X) \cap P(E(A,B)) \subseteq A(B',C')_X$. ◀

A subtyping constraint of the form $X <: S$ corresponds to the ability to subsume any term of type X into the type S – i.e. there should be a canonical coercion of $[\![X]\!]$ into S. Thus a S-bounded type-variable may be represented as the arena $\Theta, X \vdash \{X\} \wedge [\![S]\!]$ (cf [18]) – i.e. $X <: S \vdash T$ denotes $[\![T]\!](\{X\} \wedge [\![S]\!], X \wedge [\![S]\!])$, so that $[\![X <: S \vdash X]\!] = \{X\} \wedge [\![S]\!] \preccurlyeq [\![X <: S \vdash S]\!]$.

However, these coercions are are only actually used for type-variables which occur negatively. This allows for two possible interpretations of bounded quantification – as $\forall X.[\![T]\!](X \wedge S, X \wedge S)_X$ or as $\forall X.[\![T]\!](X \wedge S, X)_X$. Noting that:

- $\forall X.[\![T]\!](X \wedge S, X \wedge S)_X \preccurlyeq \forall X.[\![T]\!](X \wedge S, X)_X$ by Lemma 16.
- Subtypings inferred using the bound hold for the first interpretation but not the second.
- The second interpretation is antitone in the variable bound but not the first.

we extend the interpretation of System $\mathrm{F}_{\top}$ types to System $\mathsf{F}^{\mathsf{F}\top}_{<:}$.

$$[\![\mathcal{E} \vdash \forall^{\mathsf{K}}(X <: S).T]\!] = \forall X.[\![\mathcal{E}, X <: S \vdash T]\!]$$
$$[\![\mathcal{E} \vdash \forall^{\top}(X <: S).T]\!] = \forall X.[\![\mathcal{E}, X <: \top \vdash T]\!](\{X\} \wedge [\![\mathcal{E} \vdash S]\!], \{X\})$$

Using Lemma 16, we show that this is sound with respect to the three subtyping rules for bounded quantification:

$\forall -$ **Fun:** If $A(\{X\} \wedge S, \{X\} \wedge S)_X \preccurlyeq B(\{X\} \wedge S, \{X\} \wedge S)_X$ then $\forall X.A(\{X\} \wedge S, \{X\} \wedge S)_X \preccurlyeq \forall X.B(\{X\} \wedge S, \{X\} \wedge S)_X$.

$\forall -$ **Loc:** If $D \preccurlyeq C$ and $A(\{X\} \wedge C, \{X\} \wedge C)_X \preccurlyeq B(\{X\} \wedge C, \{X\} \wedge C)_X$ then $\forall X.A(\{X\} \wedge C, \{X\} \wedge C)_X \preccurlyeq \forall X.B(\{X\} \wedge C, \{X\} \wedge C)_X \preccurlyeq \forall X.B(\{X\} \wedge C, \{X\})_X \preccurlyeq \forall X.B(\{X\} \wedge D, \{X\})_X$.

$\forall -$ **Top:** If $D \preccurlyeq C$ and $A(\{X\}, \{X\}) \preccurlyeq B(\{X\}, \{X\})$ then $\forall X.A(\{X\} \wedge C, \{X\}) \preccurlyeq \forall X.B(\{X\} \wedge C, \{X\})_X \preccurlyeq \forall X.B(\{X\} \wedge D, \{X\})_X$.

Thus (by induction on the length of derivation of $\Theta \vdash S <: T$):

▶ **Proposition 17** (Soundness). *If $\mathcal{E} \vdash S <: T$ then $[\![\mathcal{E} \vdash \mathcal{S}]\!] \preccurlyeq [\![\mathcal{E} \vdash T]\!]$.*

5 Copycat Strategies

Having described an interpretation of bounded polymorphism at the level of subtyping, it is now necessary to show that this carries through to the term level. Interpreting terms as *strategies* on our type-arenas yields a semantics for subtype and parametric polymorphism which is based on *copycat strategies*. As canonical coercions, these give an alternative characterization of the liveness ordering – a "hereditarily total" copycat strategy exists between two arenas if and only if they are in the ordering. We now identify nodes explicitly with moves.

▶ **Definition 18.** *A* legal sequence *on an arena graph is a finite sequence of its moves which starts with an Opponent move, alternates between Opponent and Proponent moves and is equipped with a* justification pointer *[11] from each non-initial move to some preceding, enabling move.*

- Given arenas A, B, let $L(A, B)$ denote the set of legal sequences from A to B – that is, legal sequences on the arena $\overline{A} \oplus B$, where $\overline{(O, P, \triangleright)} \triangleq (P, O, \triangleright)$ swaps Proponent and Opponent moves, and $(O, P, \triangleright) \oplus (O', P', \triangleright') \triangleq (O \uplus O', P \uplus P', \triangleright \oplus \triangleright')$ is the smash sum of rooted graphs.
- Let $C(A, B)$ denote the set of *copycat sequences* from A to B – that is, legal sequences $t \in L(A, B)$ such that for every even-prefix $s \sqsubseteq^E t$, $s{\upharpoonright}A = s{\upharpoonright}B$.

It is easy to see that $C(A, B)$ is a deterministic strategy from A to B – that is, a non-empty set of even-length sequences in $L(A, B)$ which is even-prefix-closed and even-branching (if $s, t \in C(A, B)$ then $s \cap t$ has even length). Moreover:

▶ **Lemma 19.** *If $A \preccurlyeq B$ then $C(A, B)$ is a* hereditarily total *strategy: for any sequence $sm \in L(A, B)$ with $s \in C(A, B)$ there exists n such that $smn \in \sigma$.*

Proof. Suppose $s \in C(A, B)$ and $s(b.r) \in L(A, B)$, where b is an Opponent move in B. Either b is initial, or else it has a justifier in $s{\upharpoonright}B$, which therefore also occurs in $s{\upharpoonright}A$. So $b \in E(A) \cap O(B) \subseteq A$ (since $A \preccurlyeq B$) and $s(b.r)(b.l) \in C(A, B)$ as required. The case where $s(a.l) \in L(A, B)$ for some Proponent move in A is similar. ◀

The proof of the converse uses the following lemma.

▶ **Lemma 20.** *Suppose $C(A,B)$ is heredarily total.*
(i) *If $m \in E(A) \cap O(B)$ then there is a justified sequence of the form $s(m.r)(m.l)$ in $C(A,B)$.*
(ii) *If $m \in E(B) \cap P(A)$ then there is a justified sequence of the form $s(m.l)(m.r)$ in $C(A,B)$.*

Proof. By Noetherian induction. If m is initial then by totality, if $m \in O(B)$ then $(m.r)(m.l) \in C(A,B)$ and if $m \in P(B)$ then $(m.l)(m.r) \in C(A,B)$. For the induction case suppose, for example, that $m \in E(A) \cap O(B)$ is enabled by $a \in A$ and $b \in B$, which must both be Proponent or both Opponent moves by quasi-bipartiteness. In the former case, there exists $b' \in B \cup \{\top\}$ such that $b' \triangleright B$, and by quasi-arborescence $b' \triangleright a$ – i.e. $a \in E(B) \cap P(A)$, and so by inductive hypothesis there exists a justified sequence of the form $s(a.l)(a.r) \in C(A,B)$. Then $s(a.l)(a.r).(m.r)$ (with a pointer from $m.r$ to $a.r$) is a well-formed justfied sequence, and so by totality $s(a.l)(a.r)(m.r)(m.l) \in C(A,B)$ as required. The other cases are similar. ◀

Evidently, (i) and (ii) imply that $A \preccurlyeq B$ and so:

▶ **Proposition 21.** *$C(A,B)$ is hereditarily total if and only if $A \preccurlyeq B$.*

6 QA-arenas

To describe the interpretation of parametric polymorphism requires further structure on arenas – a *question-answer labelling* and *question-answer relation* supporting the interpretation of type abstraction and instantiation – introduced in [13, 14], (to which we refer for further details and proofs).

▶ **Definition 22.** *A QA-arena over a set $\mathcal{L}$ of labels (not containing "Q" and "A") is an arena $(O, P, \triangleright)$ with the following additional structure:*

- *a labelling function $\lambda : O \cup P \to \mathcal{L} \cup \{Q, A\}$ partitioning moves into sets of* questions,answers *and $\mathcal{L}$-labelled* holes.
- *a* scoped question answer/relation *– a ternary relation on moves – $\triangleleft \subseteq (O \cup P) \times \lambda)^{-1}(Q) \times (\lambda)^{-1}(A)$. The relation $m \triangleleft (q,a)$ holds if a can answer q within the scope of m: Proponent questions are scoped by Proponent moves and answered by Opponent moves, and Opponent moves are scoped by Opponent moves and answered by Proponent moves.*

The type-arena defined in Section 3 is refined to a QA-arena for each type-variable context Θ by defining a QA-labelling and QA-relation over Θ for moves $\Theta \vdash m$. This is defined inductively via judgments of the form $\Theta \vdash^{QA} m : L$ (where $\Theta \vdash m$ and $L \in \{Q, A\} \cup \Theta$) and $\Theta \vdash^{QA} m \triangleleft (q,a)$ (where $\Theta \vdash m, q, a$), for which derivation rules are given in Table 2.

Informally, a move $m[X]$ is labelled with X if X occurs free in M. Otherwise (if X is bound):

- m is a question if X occurs in m with the same polarity as its binder.
- Its answers are moves $m'[X]$ in which X occurs with the opposite polarity to its binder.
- They are scoped by moves of the form $C[\forall X.m'']$ where m' is an initial move.

For example, in the sub-arena $\forall X.X \to X \to X$, the initial (Opponent) move $\forall X.\top \to \top \to X$ is a question, with answers $\forall X.\top \to X \to \top$ and $\forall X.X \to \top$. The scoping move for both question-answer pairs is the initial move $\forall X.\top \to \top \to X$. In other words, this corresponds to the usual QA-labelling for the arena of Booleans.

Table 2 Typing Judgments for QA-Labelling and QA-relation.

$$\frac{}{\Theta \vdash^{QA} X : X} \qquad \frac{\Theta \vdash^{QA} m : X}{\Theta \vdash^{QA} \forall X.m : Q} m \in = O \qquad \frac{\Theta \vdash^{QA} m : X}{\Theta \vdash^{QA} \forall X.M : A} m \in P \qquad \frac{\Theta \vdash^{QA} m : L}{\Theta \vdash^{QA} \forall X.m : L} L \neq X$$

$$\frac{\Theta \vdash^{QA} m : L}{\Theta \vdash^{QA} M \to \top : L} \qquad \frac{\Theta \vdash^{QA} m : L}{\Theta \vdash^{QA} \top \to m : L} \qquad \frac{\Theta \vdash^{QA} m : L}{\Theta \vdash^{QA} m \times \top : L} \qquad \frac{\Theta \vdash^{QA} m : L}{\Theta \vdash^{QA} \top \times m : L}$$

$$\frac{\Theta \vdash \top \triangleright m \quad \Theta \vdash^{QA} m' : X \quad \Theta \vdash^{QA} m'' : X}{\Theta \vdash^{QA} \forall X.m \triangleleft (\forall X.m', \forall X.m'')} (m', m'') \in (O, P)$$

$$\frac{\Theta \vdash^{QA} m \triangleleft (m', m'')}{\Theta \vdash^{QA} m \to \top \triangleleft (m' \to \top, m'' \to \top)} \qquad \frac{\Theta \vdash^{QA} m \triangleleft (m', m'')}{\Theta \vdash^{QA} \top \to m \triangleleft (\top \to m', \top \to m'')}$$

$$\frac{\Theta \vdash^{QA} m \triangleleft (m', m'')}{\Theta \vdash^{QA} m \times \top \triangleleft (m' \times \top, m'' \times \top)} \qquad \frac{\Theta \vdash^{QA} m \triangleleft (m', m'')}{\Theta \vdash^{QA} \top \times m \triangleleft (\top \times m', \top \times m'')}$$

6.1 Well-bracketed Strategies

A QA-arena over the empty set of labels is *closed*: matching questions and answers as opening and closing parentheses induces a relation between the moves of any legal sequence over a closed QA-arena : it is *well-bracketed* if this relation is contained within the QA-relation. More precisely, let the *pending question* of a sequence of moves t over such a closed arena be the prefix of t given (where defined) by:

$$\mathsf{pending}(sm) = \begin{cases} sm & \text{if } m \text{ is a question} \\ \mathsf{pending}(s') & \text{if } m \text{ is an answer and } \mathsf{pending}(s) = s'm' \end{cases}.$$

▶ **Definition 23.** *A well-bracketed sequence on a closed arena A is a legal sequence t on A which satisfies the* bracketing condition*:* Whenever $sa \sqsubseteq t$, where a is an answer, then $\mathsf{pending}(s) = s'q$, where $m \triangleleft (q, a)$ for some move m in s' which hereditarily justifies both q and a.

The *closure* of a QA-arena A over $\mathcal{L}$ is obtained by converting $\mathcal{L}$-labelled Opponent and Proponent moves to questions and answers, respectively, and extending the QA-relation by making initial moves into scoping moves for pairs of (Opponent,Proponent) moves with the same label.

▶ **Definition 24.** *Given $\mathcal{L} \vdash A$, the closed arena $\forall A$ has the same underlying arena. The QA-labelling is:*

$$\lambda_{\forall A}(m) = \begin{cases} Q & \text{if } \lambda_A(m) \in \mathcal{L} \text{ and } m \in O \\ A & \text{if } \lambda_A(m) \in \mathcal{L} \text{ and } m \in P \\ \lambda_A(m) & \text{otherwise} \end{cases}$$

and the QA-relation is:

$$m \triangleleft_{\forall A} (q, a) \text{ if } m \triangleleft_A (q, a) \text{ or } \lambda_A(q) = \lambda_A(a) = l \in \mathcal{L} \text{ and } \top \triangleright m$$

A strategy σ from A to B is *well-bracketed* if every $s \in \sigma$ is a well-bracketed sequence on $\forall(\overline{A} \uplus B)$. It is *thread-independent* if:

For any well-bracketed r, s, t, where r is an interleaving of s and t:
$r \in \sigma$ if and only if $s, t \in \sigma$.

Let $CW(A, B)$ be the set of well-bracketed copycat sequences from A to B: this is well-bracketed by definition, and since any legal interleaving of legal sequences s and t is a copycat sequence if and only if r and s are copycats, it is a thread-independent strategy.

▶ **Definition 25.** *For a QA-arena U, the category $\mathcal{G}(U)$ consists of (objects) the sub-arenas of U with (morphisms) thread-independent strategies between them. Strategies are composed by parallel composition plus hiding*

$$\{s \in w(A, C) \mid \exists t \in (A + B + C)^*.t\restriction A, B \in \sigma \wedge t\restriction B, C \in \tau\}$$

The identity on A is the well-bracketed copycat $CW(A, A)$.

Well-bracketed copycats have a more general identity property.

▶ **Proposition 26.** *If $A' \preccurlyeq A$ and $B \preccurlyeq B'$ then for any $\sigma : A \to B$, $CW(A, A'); \sigma; CW(B, B') = \sigma \cap L(A', B')$.*

In particular, if $A \preccurlyeq B$ and $B \preccurlyeq C$ then $CW(A, B); CW(B, C) = CW(A, C)$, justifying:

▶ **Definition 27.** *Let $CW : \mathcal{S}(U) \to \mathcal{G}(U)$ be the identity-on-objects functor from the lattice $\mathcal{S}(U)$ (considered as a thin category) into $\mathcal{G}(U)$, sending $A \preccurlyeq B$ to the copycat $CW(A, B)$.*

7 Copycat Dinaturality

Returning to our family of second-order type arenas, the instantiation of a sub-arena into a strategy is given (as in [14]) by playing copycat between the arenas plugged into the holes. To define this on type arenas we make use of a restriction operation – a partial inverse to subsitution on moves:

▶ **Definition 28.** *Given moves $\Theta, \Theta', \Theta'' \vdash m$ and $\Theta, X, \Theta'' \vdash n[X]$ the restriction of m to $n[X]$ is defined:*

$$m\restriction n[X] = \begin{cases} l & \text{if } m = n[l/X] \\ \text{undefined} & \text{otherwise} \end{cases}.$$

By definition, $m[n/X]\restriction m = n$ and if $m\restriction n[X]$ is defined, then $m[(m\restriction n[X])/X] = m$.

This operation lifts to justified sequences by applying it pointwise to the moves on which it is defined, and omitting moves on which it is not defined.

▶ **Definition 29.** *Given a justified sequence t, and move n, let $t\restriction n$ be the justified sequence such that $\varepsilon\restriction n = \varepsilon$ and*

$$(tm)\restriction n = \begin{cases} (t\restriction n)(m\restriction n) & \text{if } m\restriction n \downarrow \\ t\restriction n & \text{otherwise} \end{cases}$$

where $m\restriction n$ points to $m'\restriction n'$ in $t\restriction n$, if both are defined and m points to m' in t. (Otherwise $m\restriction n$ is initial and requires no pointer.)

We may now describe the instantiation of an arena into a strategy on type-arenas. Suppose t is a well-bracketed sequence from $A(C, C)_X$ to $B(C, C)_X$, such that $t \restriction \forall(\overline{A} \uplus B)$ is a well-bracketed sequence. For any even prefix $sp[X] \sqsubseteq t\restriction\forall\overline{A} \uplus B$, $p[X]$ is a Proponent answer

in $\forall\overline{A} \uplus B$, so by well-bracketing $\mathsf{pending}(s) = s'o[X]$, where $o[X]$ is an Opponent question in $\forall\overline{A} \uplus B$. Say that t is a copycat-instantiated sequence for X if it is a copycat between every such pair of moves – i.e. for any even-length sequence t' with $s \sqsubseteq t' \sqsubseteq t$, $t\restriction p[X] = t\restriction o[X]$.

▶ **Definition 30.** *Given a strategy* $\Theta, X, \Theta' \vdash \sigma : A \to B$*,let* $\Theta, \Theta', \Theta'' \vdash \sigma[C]_X : A(C,C)_X \to B(C,C)_X$ *be the set of sequences* t *on* $\forall A(C,C)_X \to B(C,C)_X$ *which are copycat-instantiated for* X*, such that there exists* $s \in \sigma$ *with* $s = t\restriction\forall(\overline{A} \uplus B)$*.*

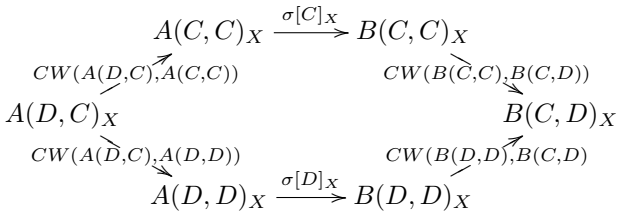
This yields a family of strategies $\sigma[\_]_X =$

$$\{\Theta, \Theta', \Theta'' \vdash \sigma[C]_X : A(C,C)_X \to B(C,C)_X \mid \Theta, \Theta' \vdash C\}.$$

By Proposition 16, $A(\_,\_)_X$ and $B(\_,\_)_X$ act as mixed variance functors from $\mathcal{S}(\Theta, \Theta')$ to $\mathcal{S}(\Theta, \Theta', \Theta'')$, which may be composed with the copycat functor $CW : \mathcal{S}(\Theta, \Theta', \Theta'') \to \mathcal{G}(\Theta, \Theta', \Theta'')$.

▶ **Proposition 31.** $\sigma[\_]_X$ *is a* dinatural transformation *from* $CW \cdot A(\_,\_)_X$ *to* $CW \cdot B(\_,\_)_X$*.*

Proof. In other words, for any arenas $\Theta, \Theta' \vdash C \preccurlyeq D$, the dinaturality hexagon:



commutes. By Proposition 26, this is equivalent to requiring that $\sigma[C]_X \cap W(C,D) = \sigma[D]_X \cap W(C,D)$, which follows from the definition of copycat instantiation. ◀

Note the importance of the restriction to copycat (or, at least, strict) strategies – the dinaturality hexagon above need not commute for all morphisms from C to D (even if σ is the denotation of a term of System F : see [8] for examples).

Instantiation extends to tuples of arenas – $B(A_1, A_1')_{X_1} \dots (A_n, A_n')_{X_n}$ substitutes the moves of A_1 for negative occurrences of X_1 in B, the moves of A_1' for positive occurrences of X_1 and so on, giving a family of n-ary mixed variance functors on $\mathcal{S}(\Theta)$ to $\mathcal{S}(\Theta)$ which is closed under composition. Using these we may define a hyperdoctrine model of System F [20].

- Let $\mathcal{I}$ be the category in which objects are type-variable contexts and morphisms from Θ to $\Theta' = X_1, \dots, X_n$ are substitutions – n-tuples of arenas $\langle \Theta \vdash A_1, \dots, \Theta \vdash A_n\rangle$ composed by instantiation: $\langle B_1, \dots, B_m\rangle \cdot \langle A_1, \dots, A_n\rangle =$ $\langle B_1(A_1, A_1)_{X_1} \dots (A_n, A_n)_{X_n}, \dots, B_m(A_1, A_1)_{X_1} \dots (A_n, A_n)_{X_n}\rangle$.
- For $\langle A_1, \dots, A_n\rangle : \Theta \to \Theta'$, the *instantiation functor* $\mathcal{G}\langle A_1, \dots, A_n\rangle : \mathcal{G}(\Theta') \to \mathcal{G}(\Theta)$ sends $\Theta' \vdash B$ to $\Theta \vdash B(A_1, A_1)_{X_1} \dots (A_n, A_n)_{X_n}$, and $\Theta' \vdash \sigma : B \to C$ to $\Theta' \vdash \sigma[A_1]_{X_1} \dots [A_n]_{X_n}$.
- Each category $\mathcal{G}(\Theta)$ is cartesian closed ($A \times B$ is the cartesian product of A and B, and $A \to B$ is their internal hom) and instantiation preserves this structure. Thus we have a functor $\mathcal{G}$ from $\mathcal{I}^{op}$ to the category of cartesian closed categories sending Θ to $\mathcal{G}(\Theta)$ and $\langle A_1, \dots, A_n\rangle$ to $\mathcal{G}\langle A_1, \dots, A_n\rangle$.
- For context Θ, Θ, X is the product of Θ with X in $\mathcal{I}$, and the image of the corresponding projections, $\mathcal{G}(\pi) : \mathcal{G}(\Theta) \to \mathcal{G}(\Theta, X)$ has an indexed left adjoint $\forall X$. Since instantiation commutes with type-variable abstraction, this satisfies the Beck-Chevalley condition.

Table 3 Typing Judgments for System $\mathsf{F}^{\mathsf{F}\top}_{<:}$.

$$\frac{\mathcal{E} \vdash \Gamma \qquad \mathcal{E} \vdash \top}{\Theta \vdash \mathsf{top} : \top}\ \top \qquad \frac{\mathcal{E} \vdash \Gamma, x : T, \Gamma'}{\mathcal{E}; \Gamma, x : T, \Gamma' \vdash x : T}\ \mathsf{var}$$

$$\frac{\mathcal{E}; \Gamma \vdash t : T \qquad \mathcal{E} \vdash T <: T'}{\mathcal{E}; \Gamma \vdash t : T'}\ \mathsf{sub}$$

$$\frac{\mathcal{E}; \Gamma, x : S \vdash t : T}{\mathcal{E}; \Gamma \vdash \lambda x : S.t : S \to T} \to -\mathsf{i} \qquad \frac{\mathcal{E}; \Gamma \vdash t : S \to T \qquad \mathcal{E}; \Gamma \vdash s : S}{\mathcal{E}; \Gamma \vdash ts : T} \to -\mathsf{e}$$

$$\frac{\mathcal{E}, X <: S; \Gamma \vdash t : T \qquad \mathcal{E} \vdash \Gamma}{\mathcal{E}; \Gamma \vdash \Lambda(X <: S).t : \forall^{\mathsf{F}}(X <: S).T} \forall - \mathsf{i}$$

$$\frac{\mathcal{E}; \Gamma \vdash t : \forall^{\top}(X <: S).T \qquad \mathcal{E} \vdash S' <: S}{\mathcal{E}; \Gamma \vdash t\{S'\} : T[S'/X]} \forall - \mathsf{e}$$

8 System $\mathsf{F}^{\mathsf{F}\top}_{<:}$ and its Semantics

The raw terms of System $\mathsf{F}^{\mathsf{F}\top}_{<:}$ are those of System $\mathsf{F}_{<:}$ – given by the grammar:

$$t ::= \mathsf{top} \mid x \mid \lambda(x : T).t \mid \Lambda(X <: T).t \mid t\,t \mid t\{T\}$$

Typing judgments (derived according to the rules in Table 3) take the form $\mathcal{E}; \Gamma \vdash t : T$, where Γ is a context of term-variables $x_1 : T_1, \ldots x_n : T_n$ such that $\mathcal{E} \vdash \Gamma$ – i.e. $\mathcal{E} \vdash T_1, \ldots, \mathcal{E} \vdash T_n$. Type-variable abstraction ($\forall$-introduction) is typed using $\forall^{\mathsf{F}}$ and instantiation ($\forall$-elimination) is typed using $\forall^{\top}$ – subsumption allows conversion from the former to the latter.

Each term-in-context $\mathcal{E}; x_1 : S_1, \ldots, x_n : S_n \vdash t : T$ denotes a morphism from $[\![\mathcal{E} \vdash S_1]\!] \times \ldots \times [\![\mathcal{E} \vdash S_n]\!]$ to $[\![\mathcal{E} \vdash T]\!]$ in the category $\mathcal{G}(|\mathcal{E}|)$, where $|X_1 <: R_1, \ldots, X_m <: R_m| = X_1, \ldots, X_m$. To interpret instantiation for bounded variables we require an operation taking a strategy $\sigma : A \to \forall X.B(\{X\} \wedge C, \{X\})_X$ and a bounded argument $D \preccurlyeq C$ to a strategy on $A \to B(D, D)_X$. The obvious way to do this is by instantiation of D for X in σ, followed by subsumption. However, the substitution operation on arenas does not respect the lattice structure of the liveness ordering.

▶ **Lemma 32.** *$(C \wedge A)(B, B)_X \neq C(B, B)_X \wedge C(B, B)_X$ in general.*

Proof. The meet $\{X\} \wedge A$ is the union $\{X\} \cup A$ (since X contains only a single initial move): the instantiation $(\{X\} \wedge A)(B, B)_X$ is thus equal to the union $A \cup B$, which is not the same as $A \wedge B$ in general. ◀

However, when substituting in bounded types we can use the following lemma

▶ **Lemma 33.** *If $C \preccurlyeq B$ then $C \preccurlyeq (B \cup C)$.*

Proof. $O(C) \cap (C \cup B) = (O(C) \cap C) \cup O(C) \cap B \subseteq C \cup C = C$. $P(B \cup C) \cap C \subseteq C \subseteq B \cup C$. ◀

So if $C \preccurlyeq D$, then $A(X \wedge B, X)_X(C, C)_X = A(C \cup B, C)_X \preccurlyeq A(C, C)_X$, and we may interpret bounded instantiation by substitution followed by subsumption – if $\mathcal{E} \vdash S <: S'$ then $[\![\mathcal{E}; \Gamma \vdash t\{S'\} : T[S'/X]]\!] =$

$$[\![\mathcal{E}; \Gamma \vdash t : \forall^{\top}(X <: S)]\!]; CW([\![E, X <: S \vdash T]\!]([\![E \vdash S']\!], [\![E \vdash S']\!])_X, [\![E \vdash T[S'/X]]\!]).$$

Since use of the subsumption rule may yield multiple derivations of the same typing judgment it is not immediately obvious that this defines a unique denotation for each term-in-context – we need to show that any derivation for a given term in context yields the same denotation (coherence).

▶ **Proposition 34.** *Every derivable typing judgment* $\Theta; \Gamma \vdash t : T$ *denotes a unique morphism* $[\![\Theta; \Gamma \vdash M : T]\!]$.

Proof. In [16] we define a derivation system for *minimal types*, such that if t is typable in context $\mathcal{E}; \Gamma$ there is a unique derivation of a minimal type $\mathcal{E}; \Gamma \vdash t : T$ such that if $\mathcal{E}; \Gamma \vdash t : T'$ then $\mathcal{E} \vdash T <: T'$. To establish coherence, we show that any denotation for $\mathcal{E}; \Gamma \vdash t : T'$ satisfies $[\![\mathcal{E}; \Gamma \vdash t : T']\!] = [\![\mathcal{E}; \Gamma \vdash t : T]\!]; cw([\![\mathcal{E} \vdash T]\!], [\![\mathcal{E} \vdash T]\!])$. ◀

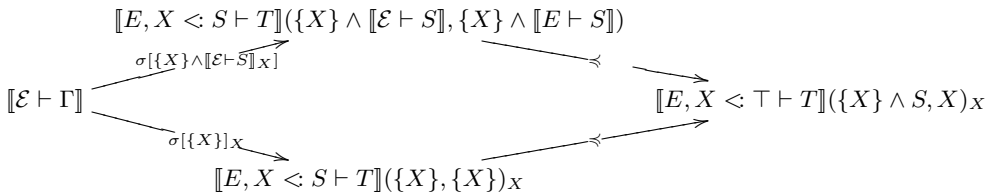
Copycat dinaturality is the key to showing that the semantics is sound with respect to second-order β and η-equivalence:

▶ **Lemma 35.** *The model soundly interprets the rules:*

$$\frac{\mathcal{E}, X <: S; \Gamma \vdash t : T \qquad \mathcal{E}; \Gamma \vdash R <: S}{\mathcal{E}; \Gamma \vdash (\Lambda(X <: S).t)\{R\} = t[R/X] : T[R/X]}\ \beta_2$$

$$Y \notin \mathsf{dom}(\mathcal{E})\ \frac{\mathcal{E}; \Gamma \vdash t : \forall^{\top} X <: S.T}{\mathcal{E}; \Gamma \vdash \Lambda(Y <: S).(t\{Y\}) = t : \forall^{\top}(X <: S).T}\ \eta_2$$

Proof. We give the case of second-order η-equivalence as an example. Suppose σ is the uncurrying of $[\![\mathcal{E}; \Gamma \vdash t : \forall^{\top}(X <: S).T]\!]$. The diagram



commutes by copycat dinaturality, and so $[\![\mathcal{E}; \Gamma \vdash t : \forall^{\top} X <: S.T]\!] = [\![\mathcal{E}; \Gamma \vdash \Lambda X.t\{X\} : \forall^{\top}(X <: S).T]\!]$ as required. ◀

▶ **Proposition 36.** *If* $\mathcal{E}; \Gamma \vdash t =_{\beta\eta} t' : T$ *then* $[\![\mathcal{E}; \Gamma \vdash t]\!] = [\![\mathcal{E}; \Gamma \vdash t' : T]\!]$.

9 Conclusions and Further Directions

Our interpretation of the subtyping relation on Hyland-Ong arenas may be applied to a wide range of games models which employ this basic structure. We have also shown that it can be integrated, via dinaturality, with an interpretation of generic polymorphism based on bracketing structure, providing a way to use this principle which might be further explored in reasoning about program equivalence. This gives us the ingredients to develop existing games semantics for stateful objects with more expressive type theories such as Dependent Object Types [17, 22] (with an appropriate treatment for subtyping bounded quantification – see also [10]) as well as bounded abstract data types.

References

1 E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P.J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70(1):35–64, 1990. doi:10.1016/0304-3975(90)90055-m.

2 K. Bruce and G. Longo. A modest model of records, inheritance and bounded quantification. *Information and Computation*, 87(1/2):196–240, 1990. doi:10.1109/lics.1988.5099.

3 L. Cardelli and P. Wegner. On understanding types, data abstraction and polymorphism. *Computing Surveys*, 17(4):471–522, 1985.

4 Luca Cardelli, John C. Mitchell, Simone Martini, and Andre Scedrov. An extension of System F with subtyping. *Information and Computation*, 109(1–2):4–56, 1994.

5 G. Castagna and B. C. Pierce. Decidable bounded quantification. In *Proceedings of POPL '94*, pages 1–29, 1994. doi:10.1145/174675.177844.

6 J. Chroboczek. Game semantics and subtyping. In *Proceedings of the fifteenth annual symposium on Logic in Computer Science*, pages 192–203. IEEE press, 2000. doi:10.1109/lics.2000.855769.

7 P.-L. Curien and G. Ghelli. Coherence of subsumption, minimum typing and type-checking in $\mathsf{F}_{<:}$. *Mathematical Structures in Computer Science*, 2(1):55–91, 1992. doi:10.1007/3-540-52590-4_45.

8 J. de Lataillade. Dinatural terms in System F. In *Proceedings of the 24th annual symposium on Logic in Computer Science, LICS '09*. IEEE Press, 2009. doi:10.1109/lics.2009.30.

9 J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.

10 J. Z. S. Hu and O. Lhoták. Undecidability of $\mathsf{D}_{<:}$ and its decidable fragments. *Proceedings of the ACM on Programming Languages (POPL)*, 4(9):1–30, 2020. doi:10.1145/3371077.

11 J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation*, 163:285–408, 2000.

12 D. Katiyar and S. Sankar. Completely bounded quantification is decidable. In *Proceedings of the ACM SIGPLAN Workshop on ML and its Applications*, pages 68–77, 1992.

13 J. Laird. Game semantics for a polymorphic programming language. In *Proceedings of LICS '10*. IEEE Press, 2010.

14 J. Laird. Game semantics for a polymorphic programming language. *Journal of the ACM*, 60(4), 2013.

15 J. Laird. Game semantics for bounded polymorphism. In *Proceedings of FoSSaCS '16*, number 9634 in LNCS. Springer, 2016. doi:10.1007/978-3-662-49630-5_4.

16 J. Laird. Revisiting decidable bounded quantification, via dinaturality. In *Proceedings of Mathematical Foundations of Program Semantics '22*, Electronic Notes in Theoretical Computer Science, 2022. To appear.

17 N. Amin, S. Grütter, M. Odersky, T. Rompf, and S. Stucki. The essence of dependent object types. In *A List of Successes That Can Change the World - Essays Dedicated to Philip Wadler on the Occasion of His 60th Birthday*, number 9600 in LNCS, pages 249–272. Springer, 2016. doi:10.1007/978-3-319-30936-1_14.

18 B. C. Pierce. *Programming with Intersection Types and Bounded Polymorphism*. PhD thesis, Carnegie Mellon University, 1991. URL: https://dl.acm.org/doi/10.5555/145640.

19 Benjamin C. Pierce. Bounded quantification is undecidable. In *POPL*, pages 305–315, 1992. doi:10.1006/inco.1994.1055.

20 A. M. Pitts. Relational properties of domains. *Information and Computation*, 127:66–90, 1996.

21 J. C. Reynolds. Syntactic Control of Interference. In *Conf. Record 5^{th} ACM Symposium on Principles of Programming Languages*, pages 39–46, 1978.

22 Tiark Rompf and Nada Amin. From F to DOT: type soundness proofs with definitional interpreters. *CoRR*, abs/1510.05216, 2015. arXiv:1510.05216.

23 Sergei Vorobyov. Structural decidable extensions of bounded quantification. In *Proceedings of POPL '95*, pages 164–175, 1995. doi:10.1145/199448.199479.