Transition systems over games

Paul Blain Levy
University of Birmingham
p.b.levy@cs.bham.ac.uk

Sam Staton
Radboud University Nijmegen
s.staton@cs.ru.nl

Abstract

We describe a framework for game semantics combining operational and denotational accounts. A game is a bipartite graph of "passive" and "active" positions, or a categorical variant with morphisms between positions.

The operational part of the framework is given by a labelled transition system in which each state sits in a particular position of the game. From a state in a passive position, transitions are labelled with a valid O-move from that position, and take us to a state in the updated position. Transitions from states in an active position are likewise labelled with a valid P-move, but silent transitions are allowed, which must take us to a state in the same position.

The denotational part is given by a "transfer" from one game to another, a kind of program that converts moves between the two games, giving an operation on strategies. The agreement between the two parts is given by a relation called a "stepped bisimulation".

The framework is illustrated by an example of substitution within a lambda-calculus.

Categories and Subject Descriptors F.3.2 [*Logic and meanings of programs*]: Semantics of programming languages

1. Introduction

This paper is concerned with two established lines of research in the semantics of higher-order calculi. One is game semantics using pointers [10], a form of denotational semantics that has been widely adapted successfully adapted to many language features, including general references [3, 27], control operators [19], exceptions [20] and polymorphism [22, 23]. The other is open (aka normal form) bisimulation [28], a convenient operational technique for establishing observational equivalences in various settings [13, 24, 25, 26, 29], based on a transition system constructed from the syntax of the calculus.

It is widely accepted that these two ideas have a lot in common, and that both operational and denotational perspectives are important [7, 14, 15, 16, 21]. The contribution of this paper is to elaborate the connections and to develop some principles that underlie the relationship.

Summary of the concepts

Our analysis is based on the following five key concepts:

1. *A game*, which is essentially a bipartite graph describing the moves that can be made as play passes between the different positions of the two players.
2. *A strategy for a game*, which specifies a particular way of playing a game, from a given starting position. Formally, it is a set of paths through the graph.
3. *A transition system over a game*, which is the operational part of the account. This can be thought of as an abstract machine that performs a strategy. Formally, it is a transition system labelled by legal moves of a game. (The set of legitimate actions changes over time, by contrast with conventional LTSs.)
4. *A way to transfer strategies from one game to another*, which is the denotational part of the account. We take as an example a composition operation, which takes two strategies in a particular game $\mathcal{G}$ and composes them. We understand this as a transfer from the tensor game $\mathcal{G} \otimes \mathcal{G}$ (whose strategies are roughly pairs of strategies) to the game $\mathcal{G}$.
5. *A stepped bisimulation across a transfer*, which relates the operational and denotational accounts. Our main theorem says that, if x and y are states of transition systems for $\mathcal{G}$ and $\mathcal{H}$ respectively, and they are stepped bisimilar across a transfer $\mathcal{O} : \mathcal{G} \to \mathcal{H}$, then the strategy performed by x is transfered by $\mathcal{O}$ to the strategy performed by y. For the example of composing strategies, we are able to relate the denotational operation of composition and the operational results of substitution.

Goals and further discoveries

When we embarked on this project, we had a number of goals. Primarily, we wanted to understand techniques for operating on strategies. We also wanted to set the fundamental notions within well-established mathematical frameworks, for example, transition systems as coalgebras, and renaming in terms of functor categories. As we report here, we have accomplished these things. But we have made some surprising discoveries too.

- We found a convenient diagrammatic notation for operations on strategies.
- Several well-known categorical concepts turn out to play an interesting role, in particular two-dimensional partial map (Sect. 5), bimodule (Sect. 8), and $*$-autonomous bicategory (Sect. 9).

Related Work

Apart from papers about denotational and/or operational game semantics, we briefly comment on some work that might seem broadly related to our work. First, it seems appropriate to emphasise the difference between open bisimilarity, in which a function is tested by calling it with a fresh identifier, and applicative bisimilarity [1], in which a function is tested by calling it with all possible arguments. For the latter there remains a big gap between oper-

CSL-LICS 2014, July 14–18, 2014, Vienna, Austria.

http://dx.doi.org/10.1145/2603088.2603150

ational and denotational accounts: Howe's ingenious congruence proof [9] appears unrelated to any denotational principles or models.

Second, we recall the 'bialgebraic semantics' of Turi and Plotkin [30], which relates operational and denotational semantics for simple first-order process calculi. But bialgebraic semantics seems different in spirit from our work. On the one hand bialgebraic semantics neatly explains when the structural operational semantics of a first-order process calculus is compositional by construction. On the other hand we are investigating compositionality properties for an abstract-machine-based operational semantics for higher-order calculi, where compositionality is not immediate.

In addition to these, there are many other operational analyses of game semantics, notably [5, 6] and the "traversal" technology of [4]. Games similar to those in Sect. 3 appear in [12, 17]. Bimodules (profunctors) are employed in game semantics in [8, 31].

2. Illustrative Example

2.1 Example calculus

The reader might expect at this point to see a rich example calculus, e.g. call-by-value typed λ-calculus with recursive types and general references. However, treating such a calculus would involve several complications (answer moves, ultimate pattern matching [25], renamings of references) that would distract somewhat from the points we are presenting in this paper. So instead we shall omit the general references, and consider just one recursive type

$$A = (A \times A) \to 0$$

which is just complicated enough to illustrate the points we are making in the paper. A value of this type is a function that is called with two arguments and (as in CPS) never returns. By abbreviating

$$\lambda(\mathtt{x},\mathtt{y}).\, M \stackrel{\text{def}}{=} \lambda \mathtt{z}.\, \mathtt{split\ z\ as}\ (\mathtt{x},\mathtt{y}).\, M$$

we obtain the following untyped (or uni-typed) calculus. We stress that it has no intrinsic importance; it is merely an illustrative fragment of the rich calculus described above.

Value	V	::=	$\mathtt{x} \mid \lambda(\mathtt{x},\mathtt{y}).M$
Nonreturning command	M	::=	$V(V,V)$

where λ binds $\mathtt{x}$ and $\mathtt{y}$; we work up to α-equivalence. For a finite set Γ of identifiers, we write $\Gamma \vdash^{\mathsf{v}} V$ to say that V is a value with free identifiers drawn from Γ, and $\Gamma \vdash^{\mathsf{nc}} M$ for a nonreturning command. We omit the typing rules, which are evident.

Operational semantics of an *open* command $\Gamma \vdash^{\mathsf{nc}} M$ is given by the *C-machine*, which β-reduces

$$\Gamma \vdash^{\mathsf{nc}} (\lambda(\mathtt{x},\mathtt{y}).M)(V,W) \rightsquigarrow M[V/\mathtt{x}, W/\mathtt{y}]$$

until reaching a command of the form $\mathtt{x}(V,W)$ for some $\mathtt{x} \in \Gamma$.

2.2 Example interaction

We consider interaction between two players called P (Proponent, Patricia, the program) and O (Opponent, Oliver, the environment). Each passes functions to the other, and the passed functions are represented as fresh function-names. To illustrate this, consider the program:

$$\mathtt{x},\mathtt{y} \vdash^{\mathsf{nc}} (\lambda(\mathtt{s},\mathtt{t}).\mathtt{s}(\mathtt{t},\mathtt{y}))(\mathtt{x}, \lambda(\mathtt{p},\mathtt{q}).(\lambda(\mathtt{u},\mathtt{v}).\mathtt{u}(\mathtt{v},\mathtt{q}))(\mathtt{p},\mathtt{x}))$$

For this program, here is an interaction between P and O. Initially, P has function-names $\mathtt{x}, \mathtt{y}$ (i.e. has the ability to call them) and O has none.

1. P firstly performs one β-reduction, giving
$$\mathtt{x}(\lambda(\mathtt{p},\mathtt{q}).(\lambda(\mathtt{u},\mathtt{v}).\mathtt{u}(\mathtt{v},\mathtt{q}))(\mathtt{p},\mathtt{x}),\mathtt{y})$$
so she calls $\mathtt{x}$ and passes to O two function-names b_0 and b_1 representing the functions $\lambda(\mathtt{p},\mathtt{q}).(\lambda(\mathtt{u},\mathtt{v}).\mathtt{u}(\mathtt{v},\mathtt{q}))(\mathtt{p},\mathtt{x})$ and $\mathtt{y}$ respectively. These names are *fresh*, i.e. not used previously.
2. Suppose O calls b_0, passing to P two fresh function-names $\mathtt{w}_0$ and $\mathtt{w}_1$.
3. Then P executes $(\lambda(\mathtt{p},\mathtt{q}).(\lambda(\mathtt{u},\mathtt{v}).\mathtt{u}(\mathtt{v},\mathtt{q}))(\mathtt{p},\mathtt{x}))(\mathtt{w}_0,\mathtt{w}_1) \rightsquigarrow (\lambda(\mathtt{u},\mathtt{v}).\mathtt{u}(\mathtt{v},\mathtt{w}_1))(\mathtt{w}_0,\mathtt{x}) \rightsquigarrow \mathtt{w}_0(\mathtt{x},\mathtt{w}_1)$ so she calls $\mathtt{w}_0$, passing two fresh function-names b_2 and b_3 representing the functions $\mathtt{x}$ and $\mathtt{w}_1$ respectively.
4. Suppose O calls b_1, passing to P two fresh function-names $\mathtt{w}_2$ and $\mathtt{w}_3$.
5. Then P executes $\mathtt{y}(\mathtt{w}_2,\mathtt{w}_3)$, i.e. immediately calls $\mathtt{y}$, passing to O two fresh function-names b_4 and b_5 representing the functions $\mathtt{w}_2$ and $\mathtt{w}_3$ respectively.

We see that each player moves by calling a function-name from their inventory, which grows over time. Consequently the set of legitimate moves keeps changing.

Remark We are not explicitly using justification pointers in the style of [10] but it is clear that, for example, when O calls b_1 in move 4, he could express this as "the second name I received in move 1". That is: as a justification pointer with some extra data. On the other hand, when P calls $\mathtt{y}$ in move 5, there is no justification pointer because she owned it at the start.

3. Games, Strategies, Transition Systems

We consider two kinds of games in this paper: a familiar "discrete" kind in this section, and a more sophisticated "categorical" kind in Sect. 4.

3.1 Games

DEFINITION 1. *A* (discrete) game $\mathcal{G}$ *is a bipartite directed multigraph. Explicitly:*

- *a set* $\mathcal{G}^{\mathrm{pass}}$ *of* passive positions
- *a set* $\mathcal{G}^{\mathrm{act}}$ *of* active positions
- *for each passive position* P*,*
  - *a set* $\mathrm{Omove}\, P$ *of* Opponent-moves *from* P
  - *for each* $m \in \mathrm{Omove}\, P$ *an active* result position $P.m$
- *for each active position* Q*,*
  - *a set* $\mathrm{Pmove}\, Q$ *of* Proponent-moves *from* Q
  - *for each* $n \in \mathrm{Pmove}\, Q$ *a passive* result position $Q.n$.

Note that no position is designated "initial". We write $P \overset{m}{\bullet\!\!\longrightarrow} Q$ to mean that $m \in \mathrm{Omove}\, P$ and $Q = P.m$, and we write $Q \overset{n}{\bullet\!\!\longrightarrow} P$ to mean that $n \in \mathrm{Pmove}\, Q$ and $P = Q.n$.

Our main example of a game uses the notion of a *gen-set*, intuitively a set of names with a facility for generating fresh ones. Formally it is a set A equipped with a set of *permitted* finite subsets of A, and for each permitted set R an element $\nu R \in A \setminus R$ such that $R^+ \stackrel{\text{def}}{=} R \cup \{\nu R\}$ is permitted. Here are some examples.

1. $\mathbb{N}$ is a gen-set with $\$n \stackrel{\text{def}}{=} \{m \in \mathbb{N} \mid m < n\}$ permitted for all $n \in \mathbb{N}$, and $\nu(\$n) = n$. This is similar to O's function-names in our example, viz. $b_0, b_1, b_2, b_3, b_4, b_5$.
2. If B is a set then $B + \mathbb{N}$ is a gen-set with $U + \$n$ permitted for all finite $U \subseteq B$ and $n \in \mathbb{N}$, and $\nu(U + \$n) = \mathsf{inr}\, n$. This is similar to P's function-names in our example, viz. $\mathtt{x}, \mathtt{y}, \mathtt{w}_0, \mathtt{w}_1, \mathtt{w}_2, \mathtt{w}_3$. We would take B to be some set containing $\mathtt{x}$ and $\mathtt{y}$.
3. [Example with thanks to L. Moss; not used in the sequel.] The universe of sets forms a gen-set, or strictly speaking a gen-*class*,

with every finite set R permitted and $\nu R \stackrel{\text{def}}{=} \{x \in R \mid x \notin x\}$, which is simply R if the Axiom of Foundation is assumed.

For a permitted set R and $n \in \mathbb{N}$, we write $R^{+n} \stackrel{\text{def}}{=} R^{\overbrace{+\cdots+}^{n}}$ and $\nu_n R \stackrel{\text{def}}{=} \nu R^{+n}$, giving a sequence $(\nu_n R)_{n\in\mathbb{N}}$ of distinct names not in R. For permitted sets R, S and function $f : R \to S$ and $n \in \mathbb{N}$, we write $f^{+n} : R^{+n} \to S^{+n}$ for the extended function $f\,[\nu_i R \mapsto \nu_i S]_{i<n}$.

Let us now fix two gen-sets of *P's function-names* and of *O's function-names* respectively. We form a game $\lambda\mathbf{Game}$ in which

- a passive position is a pair $\Gamma \parallel \Delta$, where Γ and Δ are permitted sets of P's names and O's names respectively
- an active position is the same
- an O-move from $\Gamma \parallel \Delta$ is $b \in \Delta$, with result position $\Gamma^{+2} \parallel \Delta$
- a P-move from $\Gamma \parallel \Delta$ is $\mathtt{x} \in \Gamma$, with result position $\Gamma \parallel \Delta^{+2}$.

This is in keeping with the interaction in Section 2: O moves by selecting a function-name from his inventory, putting two new function-names into P's, and vice versa.

3.2 Strategies

Let $\mathcal{G}$ be a game. A *play* in $\mathcal{G}$ is any sequence of consecutive moves. Strategies may be described as sets of plays, as follows.

DEFINITION 2. *1. A* strategy *for $\mathcal{G}$ starting from a passive position P is a set σ of passive-ending plays*

$$P \xrightarrow{m_0} \cdot \xrightarrow{n_0} \cdot \qquad \cdots \qquad \cdot \xrightarrow{m_{r-1}} \cdot \xrightarrow{n_{r-1}} \cdot$$

such that the empty play is in σ, and $smn \in \sigma$ implies $s \in \sigma$, and $tn, tn' \in \sigma$ implies $n = n'$. We write $\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} P$ for the set of all such strategies.

2. A strategy *for $\mathcal{G}$ starting from an active position Q is a set σ of passive-ending plays*

$$Q \xrightarrow{n_0} \cdot \xrightarrow{m_0} \cdot \qquad \cdots \qquad \cdot \xrightarrow{n_r} \cdot$$

such that $smn \in \sigma$ implies $s \in \sigma$, and $tn, tn' \in \sigma$ implies $n = n'$. We write $\mathrm{Strat}^{\mathrm{act}}_{\mathcal{G}} Q$ for the set of all such strategies.

3.3 Transition systems

In the following, it is essential not to confuse positions with states. For a computer program playing chess, the position describes the current arrangement of the chessboard, which determines what moves are legitimate, whereas a state describes the values of memory cells etc. used to determine how to play. Think of a position as the "type" of a state.

DEFINITION 3. *A* small-step system *over a game $\mathcal{G}$ consists of the following data.*

- *For each passive position P a set $\mathbb{S}^{\mathrm{pass}} P$ of* passive states in position P.
- *For each active position Q a set $\mathbb{S}^{\mathrm{act}} Q$ of* active states in position Q.
- *For each passive position P a function*

$$\zeta^{\mathrm{pass}}_P : \mathbb{S}^{\mathrm{pass}} P \quad \to \quad \prod_{m \in \mathrm{Omove}\, P} \mathbb{S}^{\mathrm{act}}\, P.m$$

- *For each active position Q a function*

$$\zeta^{\mathrm{act}}_Q : \mathbb{S}^{\mathrm{act}} Q \quad \to \quad (\sum_{n \in \mathrm{Pmove}\, Q} \mathbb{S}^{\mathrm{pass}}\, Q.n) + \mathbb{S}^{\mathrm{act}}\, Q$$

For $x \in \mathbb{S}^{\mathrm{pass}} P$ and $m \in \mathrm{Omove}\, P$ we write $x@m$ for $\zeta^{\mathrm{pass}}_P(x)(m)$. For $y \in \mathbb{S}^{\mathrm{act}} P$ we write $y \overset{n}{\rightsquigarrow} x$ or $y \rightsquigarrow z$ (*silent transition*) according as $\zeta^{\mathrm{act}}_P(y)$ is $\mathsf{inl}\ (n, x)$ or $\mathsf{inr}\ z$.

For our main example (§2), we form a transition system $\lambda\mathbf{Syst}$ over $\lambda\mathbf{Game}$ as follows.

- A passive state in (passive) position $\Gamma \parallel \Delta$ is a family of values $(V_a)_{a\in\Delta}$, where $\Gamma \vdash^{\mathsf{v}} V_a$ for each $a \in \Delta$.
- An active state in (active) position $\Gamma \parallel \Delta$ consists of a family of values $(V_a)_{a\in\Delta}$ and a nonreturning command M, where $\Gamma \vdash^{\mathsf{v}} V_a$ for each $a \in \Delta$ and $\Gamma \vdash^{\mathsf{nc}} M$. We write $(V_a)_{a\in\Delta} \blacktriangleright M$ for an active state.
- For a passive state $(V_a)_{a\in\Delta}$ in position $\Gamma \parallel \Delta$, we define $(V_a)_{a\in\Delta}@a \stackrel{\text{def}}{=} (V_a)_{a\in\Delta} \blacktriangleright V_a(\nu_0\Gamma, \nu_1\Gamma)$.
- For an active state $(V_a)_{a\in\Delta} \blacktriangleright M$ in position $\Gamma \parallel \Delta$, we set

$$(V_a)_{a\in\Delta} \blacktriangleright M \begin{cases} \rightsquigarrow & (V_a)_{a\in\Delta} \blacktriangleright M' \quad \text{if } M \rightsquigarrow M' \\ \overset{\mathtt{x}}{\rightsquigarrow} & (V_a)_{a\in\Delta}\,[\nu_i\Delta \mapsto V_i]_{i=0,1} \\ & \qquad \text{if } M = \mathtt{x}(V_0, V_1) \end{cases}$$

We often want to ignore the silent transitions, in which case we use the following.

DEFINITION 4. *A* big-step system *over a game $\mathcal{G}$ consists of the following data.*

- *For each passive position P a set $\mathbb{S}^{\mathrm{pass}} P$ of* passive states in position P.
- *For each active position Q a set $\mathbb{S}^{\mathrm{act}} Q$ of* active states in position Q.
- *For each passive position P a function*

$$\zeta^{\mathrm{pass}}_P : \mathbb{S}^{\mathrm{pass}} P \quad \to \quad \prod_{m \in \mathrm{Omove}\, P} \mathbb{S}^{\mathrm{act}}\, P.m$$

- *For each active position Q a function*

$$\zeta^{\mathrm{act}}_Q : \mathbb{S}^{\mathrm{act}} Q \quad \to \quad (\sum_{n \in \mathrm{Pmove}\, Q} \mathbb{S}^{\mathrm{pass}}\, Q.n) + 1$$

For $x \in \mathbb{S}^{\mathrm{pass}} P$ and $m \in \mathrm{Omove}\, P$ we write $x@m$ for $\zeta^{\mathrm{pass}}_P(x)(m)$. For $y \in \mathbb{S}^{\mathrm{act}} P$ we write $y \overset{n}{\Longrightarrow} x$ and $y \Uparrow$ according as $\zeta^{\mathrm{act}}_P(y)$ is $\mathsf{inl}\ (n, x)$ or $\mathsf{inr}\ ()$. A small-step system always gives rise to a big-step one: we set $y \overset{n}{\Longrightarrow} x$ when $y \rightsquigarrow^* \overset{n}{\rightsquigarrow} x$, and $y \Uparrow$ when $y \rightsquigarrow^\omega$.

We may go a step further and dispense with the active states:

DEFINITION 5. *A* passive system *over a game $\mathcal{G}$ consists of the following data.*

- *For each passive position P a set $\mathbb{S}^{\mathrm{pass}} P$ of* passive states in position P.
- *For each passive position P a function* $\zeta^{\mathrm{pass}}_P : \mathbb{S}^{\mathrm{pass}} P \quad \to$

$$\prod_{m \in \mathrm{Omove}\, P} ((\sum_{n \in \mathrm{Pmove}\, P.m} \mathbb{S}^{\mathrm{pass}}\, P.m.n) + 1)$$

For $x \in \mathbb{S}^{\mathrm{pass}} P$ and $m \in \mathrm{Omove}\, P$ we write $x@m \overset{n}{\rightsquigarrow} w$ and $x@m \Uparrow$ according as $\zeta^{\mathrm{act}}_P(x)(m)$ is $\mathsf{inl}\ (n, w)$ or $\mathsf{inr}\ ()$. (Here $@\rightsquigarrow$ and $@\Uparrow$ are quaternary and binary predicates respectively, and $x@m$ has no meaning in a passive system.) Clearly a big-step system gives rise to a passive one by taking just the passive states.

We could consider bisimulations for small-step or big-step systems, but it turns out that the most useful notion is for passive systems.

DEFINITION 6. *Let $\mathbb{S}$ be a passive system over a game $\mathcal{G}$. A* passive bisimulation *on $\mathbb{S}$ associates to each passive position P a binary relation $\mathcal{R}_P$ on $\mathbb{S}^{\mathrm{pass}} P$, such that if $x\,(\mathcal{R}_P)\,x'$ and $m \in \mathrm{Omove}\, P$, either*

- $x@m \stackrel{n}{\Longrightarrow} w$ *and* $x'@m \stackrel{n}{\Longrightarrow} w'$ *for some* $n \in \mathrm{Pmove}\, P.m$ *and* $w\ (\mathcal{R}_{P.m.n})\ w'$
- *or* $x@m \Uparrow$ *and* $x'@m \Uparrow$.

3.4 From transition systems to strategies

Each state has an associated strategy that describes the plays it may perform.

PROPOSITION 1. *1. Let* $\mathbb{S}$ *be a passive system over a game* $\mathcal{G}$*, and* x *a passive state in position* P*. Write* $[\![x]\!]_P^{\mathrm{pass}}$ *for the set of* traces *of* x*, i.e. passive-ending plays* $m_0 n_0 \ldots m_{k-1} n_{k-1}$ *from* P *that arise from a sequence of states*

$$x = x_0 \quad x_0@m_0 \stackrel{n_0}{\rightsquigarrow} x_1 \quad \cdots \quad x_{k-1}@m_{k-1} \stackrel{n_{k-1}}{\rightsquigarrow} x_k$$

Then $[\![x]\!]_P^{\mathrm{pass}} \in \mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}}\, P$.

2. Let $\mathbb{S}$ *be a big-step system over a game* $\mathcal{G}$*, and* y *an active state in position* Q*. Write* $[\![y]\!]_Q^{\mathrm{act}}$ *for the set of* traces *of* y *i.e. passive-ending plays* $n_0 m_0 n_1 \ldots m_{k-1} n_k$ *from* Q *that arise from a sequence of states*

$$y \stackrel{n_0}{\rightsquigarrow} x_0 \quad x_0@m_0 \stackrel{n_1}{\rightsquigarrow} x_1 \quad \cdots \quad x_{k-1}@m_{k-1} \stackrel{n_k}{\rightsquigarrow} x_k$$

Then $[\![y]\!]_Q^{\mathrm{act}} \in \mathrm{Strat}_{\mathcal{G}}^{\mathrm{act}}\, Q$.

As usual for deterministic systems, trace equivalence and bisimilarity coincide:

PROPOSITION 2. *1. Let* $\mathbb{S}$ *be a passive system over a game* $\mathcal{G}$*, and* x, x' *passive states in position* P*. Then* $[\![x]\!]_P^{\mathrm{pass}} = [\![x']\!]_P^{\mathrm{pass}}$ *iff there is a passive bisimulation* $\mathcal{R}$ *on* $\mathbb{S}$ *such that* $x\ (\mathcal{R}_P)\ x'$.

2. Let $\mathbb{S}$ *be a big-step system over a game* $\mathcal{G}$*, and* y, y' *active states in position* Q*. Then* $[\![y]\!]_Q^{\mathrm{act}} = [\![y']\!]_Q^{\mathrm{pass}}$ *iff there is a passive bisimulation* $\mathcal{R}$ *on* $\mathbb{S}$ *such that either*

- $y \stackrel{n}{\Longrightarrow} w$ *and* $y' \stackrel{n}{\Longrightarrow} w'$ *for some* $n \in \mathrm{Pmove}\, P.m$ *and* $w\ (\mathcal{R}_{Q.n})\ w'$
- *or* $y \Uparrow$ *and* $y' \Uparrow$.

4. Position Morphisms

We return to our example (§2). Given functions $p : \Gamma \to \Gamma'$ and $q : \Delta' \to \Delta$, any passive state $(V_a)_{a \in \Delta}$ in position $\Gamma \parallel \Delta$ can be transformed into a passive state $(p^* V_{q(a)})_{a \in \Delta'}$ in position $\Gamma' \parallel \Delta'$, where p^* indicates renaming. We would expect that the operational meaning of the latter state can be obtained from that of the former by the following operation transforming strategies σ on $\Gamma \parallel \Delta$ to strategies on $\Gamma' \parallel \Delta'$.

> *An O-move from the latter position is converted into an O-move from the former by applying* q*. If we feed this to* σ *and it responds with a P-move from* $\Gamma^{+2} \parallel \Delta$*, we play a P-move from* $\Gamma'^{+2} \parallel \Delta'$ *by applying* p^{+2}*. If we receive another O-move, we continue in the same way.*

The correctness of this construction is an instance of a general fact, Proposition 5 below. The key idea is that the pair $p \parallel q$ in the preceding discussion may be called a *morphism* $\Gamma \parallel \Delta \to \Gamma' \parallel \Delta'$, and it is then evident that the passive positions form a category (and likewise the active positions).

4.1 Categorical games

DEFINITION 7. *A* categorical game $\mathcal{G}$ *is a game together with the following additional data.*

- *For each pair of passive positions* P, P'*, a set* $\mathcal{G}^{\mathrm{pass}}(P, P')$ *of* passive position morphisms $P \to P'$.
- *For two passive position morphisms* $P \xrightarrow{f} P' \xrightarrow{f'} P''$ *a composite* $P \xrightarrow{f;f'} P''$*, and an identity* $P \xrightarrow{\mathrm{id}_P} P$ *for each passive position* P*, satisfying the usual left and right identity and associativity laws.*
- *Likewise for active positions.*
- *For each passive position morphism* $f : P \to P'$ *and* $m' \in \mathrm{Omove}\, P'$*, a move* $(\mathrm{Omove}\, f)(m') \in \mathrm{Omove}\, P$ *and active position morphism* $f.m' : P.(\mathrm{Omove}\, f)(m') \to P'.m'$*, satisfying equations for identity and composition described below.*
- *For each active position morphism* $g : Q \to Q'$ *and each* $n \in \mathrm{Pmove}\, Q$*, a move* $(\mathrm{Pmove}\, g)(n) \in \mathrm{Pmove}\, Q'$ *and passive position morphism* $g.n : Q.n \to Q'.(\mathrm{Pmove}\, g)(n)$*, satisfying equations for identity and composition described below.*

We introduce a helpful diagrammatic notation. We shall draw an *O-move square*

$$\begin{array}{ccc} P & \overset{m}{\circ\!\!\cdots\!\!\to} & Q \\ {\scriptstyle f}\downarrow & & \vdots{\scriptstyle g} \\ P' & \underset{m'}{\circ\!\!\to} & Q' \end{array} \tag{1}$$

to say that $m = (\mathrm{Omove}\, f)(m')$ and $g = f.m'$, and likewise draw a *P-move square*

$$\begin{array}{ccc} Q & \overset{n}{\bullet\!\!\to} & P \\ {\scriptstyle g}\downarrow & & \vdots{\scriptstyle f} \\ Q' & \underset{n'}{\bullet\!\!\cdots\!\!\to} & P' \end{array} \tag{2}$$

to say that $n' = (\mathrm{Pmove}\, g)(n)$ and $f = g.n$. Note the conventions used: downwards arrows are position morphisms, rightwards arrows are moves, and dashed arrows are derived.

The equations mentioned in Def.7 stipulate that identities and composites of O-move squares

$$\begin{array}{ccc} P & \overset{m}{\circ\!\!\to} & Q \\ {\scriptstyle \mathrm{id}}\downarrow & & \downarrow{\scriptstyle \mathrm{id}} \\ P & \underset{m}{\circ\!\!\to} & Q \end{array} \qquad \begin{array}{ccc} P & \overset{m}{\circ\!\!\cdots\!\!\to} & Q \\ {\scriptstyle f}\downarrow & & \vdots{\scriptstyle g} \\ P' & \underset{m'}{\circ\!\!\cdots\!\!\to} & Q' \\ {\scriptstyle f'}\downarrow & & \vdots{\scriptstyle g'} \\ P'' & \underset{m''}{\circ\!\!\to} & Q'' \end{array}$$

are O-move squares, and likewise for P-move squares.

We may now describe how our main example $\lambda\mathbf{Game}$ forms a categorical game. In both the passive and active position categories a morphism

$$p \parallel q : \Gamma \parallel \Delta \to \Gamma' \parallel \Delta' \tag{3}$$

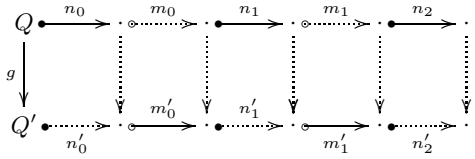
consists of functions $p : \Gamma \to \Gamma'$ and $q : \Delta' \to \Delta$, with identity and composite morphisms defined in the evident way. The squares are

$$\begin{array}{ccc} \Gamma \parallel \Delta & \overset{l(b)}{\circ\!\!\cdots\!\!\to} & \Gamma^{+2} \parallel \Delta \\ {\scriptstyle p \parallel q}\downarrow & & \vdots{\scriptstyle p^{+2} \parallel q} \\ \Gamma' \parallel \Delta' & \underset{b}{\circ\!\!\to} & \Gamma'^{+2} \parallel \Delta' \end{array} \qquad \text{O-move square}$$

$$\begin{array}{ccc} \Gamma \parallel \Delta & \overset{\mathtt{x}}{\bullet\!\!\to} & \Gamma \parallel \Delta^{+2} \\ {\scriptstyle p \parallel q}\downarrow & & \vdots{\scriptstyle p \parallel q^{+2}} \\ \Gamma' \parallel \Delta' & \underset{k(\mathtt{x})}{\bullet\!\!\cdots\!\!\to} & \Gamma' \parallel \Delta'^{+2} \end{array} \qquad \text{P-move square}$$

4.2 Position morphisms acting on strategies

The operation on strategies described at the start of the section may now be given generally. Let $\mathcal{G}$ be a categorical game. A $\mathcal{G}$-*interaction sequence* is just a sequence of O-move and P-move squares, e.g.



We may describe this interaction sequence from g by the sequence of moves $n_0 m'_0 n_1 m'_1 n_2$ depicted as solid, since they determine the other moves and morphisms. The play depicted along an interaction sequence's upper edge is its *internal play*; the one depicted along its lower edge is its *external play*.

PROPOSITION 3. *For any passive position morphism $f : P \to P'$ and $\sigma \in \mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} P$, let $(\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} f)(\sigma)$ be the set of external plays of passive-ending $\mathcal{G}$-interaction sequences from f whose internal play is in σ. Then $(\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} f)(\sigma) \in \mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} P'$. Likewise for an active position morphism.*

PROPOSITION 4. *(Functoriality of $\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}}$ and $\mathrm{Strat}^{\mathrm{act}}_{\mathcal{G}}$)*

1. *For a strategy $\sigma \in \mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} P$ we have*
$$(\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} \mathrm{id}_P)(\sigma) = \sigma$$
and likewise for active positions.
2. *For passive position morphisms $P \xrightarrow{f} P' \xrightarrow{g} P''$, and strategy $\sigma \in \mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} P$, we have*
$$(\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} (f;g))(\sigma) = (\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} g)(\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} f)(\sigma)$$
and likewise for active position morphisms.

4.3 Transition systems over a categorical game

As in Sect.3.3, we define small-step, big-step and passive systems over a categorical game.

DEFINITION 8. *Let $\mathcal{G}$ be a categorical game. A* small-step system *over $\mathcal{G}$ is a small-step system over the discrete game $\mathcal{G}$ (as in Definition 3), with the following additional data.*

- *For each passive position morphism $f : P \to P'$ a function $\mathbb{S}^{\mathrm{pass}} f : \mathbb{S}^{\mathrm{pass}} P \to \mathbb{S}^{\mathrm{pass}} P'$.*
- *For each active position morphism $g : Q \to Q'$ a function $\mathbb{S}^{\mathrm{act}} g : \mathbb{S}^{\mathrm{act}} Q \to \mathbb{S}^{\mathrm{act}} Q'$.*

The following conditions must be satisfied.

- *For $x \in \mathbb{S}^{\mathrm{pass}} P$ we have $(\mathbb{S}^{\mathrm{pass}} \mathrm{id}_P)(x) = x$, and likewise for active positions.*
- *For passive position morphisms $P \xrightarrow{f} P' \xrightarrow{g} P''$ and $x \in \mathbb{S}^{\mathrm{pass}} P$ we have $(\mathbb{S}^{\mathrm{pass}} f; g)(x) = (\mathbb{S}^{\mathrm{pass}} g)(\mathbb{S}^{\mathrm{pass}} f)(x)$, and likewise for active position morphisms.*
- *For every O-move square (1) and $x \in \mathbb{S}^{\mathrm{pass}} P$, we have $(\mathbb{S}^{\mathrm{act}} g)(x @ m) = ((\mathbb{S}^{\mathrm{pass}} f)(x)) @ m'$.*
- *For every active position morphism $g : Q \to Q'$ and $y \in \mathbb{S}^{\mathrm{act}} Q$, if $y \rightsquigarrow z$ then $(\mathbb{S}^{\mathrm{act}} g)(y) \rightsquigarrow (\mathbb{S}^{\mathrm{act}} g)(z)$.*
- *For every P-move square (2) and $y \in \mathbb{S}^{\mathrm{act}} Q$, if $y \overset{n}{\rightsquigarrow} x$ then $(\mathbb{S}^{\mathrm{act}} g)(y) \overset{n'}{\rightsquigarrow} (\mathbb{S}^{\mathrm{pass}} f)(x)$.*

We likewise define a big-step system and passive system over $\mathcal{G}$. Once again a small-step system gives rise to a big-step system, and a big-step system to a passive system.

Our example $\lambda\mathbf{Syst}$ forms a small-step system over the categorical game $\lambda\mathbf{Game}$: a passive morphism (3) transforms $(V_a)_{a\in\Delta}$ to $(p^* V_{q(b)})_{b\in\Delta'}$, and an active morphism (3) transforms $(V_a)_{a\in\Delta} \blacktriangleright M$ to $(p^* V_{q(b)})_{b\in\Delta'} \blacktriangleright p^* M$.

4.4 Compositionality theorem for position morphisms

We now substantiate the claim at the start of the section: it is an instance of the following "compositionality theorem".

PROPOSITION 5. *(Naturality of $[\![-]\!]^{\mathrm{pass}}$ and $[\![-]\!]^{\mathrm{act}}$)*

1. *Let $\mathbb{S}$ be a passive system over a categorical game $\mathcal{G}$. For any passive position morphism $f : P \to P'$ and state $x \in \mathbb{S}^{\mathrm{pass}} P$ we have*
$$[\![(\mathbb{S}^{\mathrm{pass}} f)(x)]\!]^{\mathrm{pass}}_{P'} \quad = \quad (\mathrm{Strat}^{\mathrm{pass}}_{\mathcal{G}} f)\,[\![x]\!]^{\mathrm{pass}}_{P}$$
2. *Let $\mathbb{S}$ be a big-step system over a categorical game $\mathcal{G}$. For any active position morphism $g : Q \to Q'$ and state $y \in \mathbb{S}^{\mathrm{act}} Q$ we have*
$$[\![(\mathbb{S}^{\mathrm{act}} g)(y)]\!]^{\mathrm{act}}_{Q'} \quad = \quad (\mathrm{Strat}^{\mathrm{act}}_{\mathcal{G}} g)\,[\![y]\!]^{\mathrm{act}}_{Q}$$

5. Categorical Games via Two-Dimensional Partial Maps

In this section, which is not used in the sequel, we present a concise formulation of categorical game in terms of two-dimensional partial maps [18].

DEFINITION 9. *(Element category) Let $\mathcal{C}$ be a category. For a functor $G : \mathcal{C} \to \mathbf{Set}$, we write $\mathrm{El}(\mathcal{C}, G)$ for the category of pairs of $A \in \mathcal{C}$ and $x \in GA$. Its morphisms are $\mathcal{C}$-morphisms preserving the element. For a functor $G : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$, we define $\mathrm{opEl}(\mathcal{C}, G) = (\mathrm{El}(\mathcal{C}^{\mathrm{op}}, G))^{\mathrm{op}}$. Each has a forgetful functor to $\mathcal{C}$.*

DEFINITION 10. *(Families construction) Let $\mathcal{C}$ be a category. We write $\mathrm{Fam}(\mathcal{C})$ for the category that has as objects families of objects from $\mathcal{C}$; the homset from $(A_i)_{i\in I}$ to $(B_j)_{j\in J}$ is $\prod_{i\in I} \sum_{j\in J} \mathcal{C}(A_i, B_j)$. We write $\mathrm{opFam}(\mathcal{C})$ for $(\mathrm{Fam}(\mathcal{C}^{\mathrm{op}}))^{\mathrm{op}}$. These are respectively the free category with coproducts and the free category with products on $\mathcal{C}$.*

DEFINITION 11. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories.*

1. *A* two-dimensional partial map *$\mathcal{C} \nrightarrow \mathcal{D}$ is a functor $F : \mathcal{C} \to \mathrm{Fam}(\mathcal{D})$. Equivalently: it consists of a functor $F_0 : \mathcal{C} \to \mathbf{Set}$ and a functor $F_1 : \mathrm{El}(\mathcal{C}, F_0) \to \mathcal{D}$. Any functor $H : \mathcal{D} \to \mathbf{Set}$ gives a functor $\sum_F H : \mathcal{C} \to \mathbf{Set}$ defined to be the composite of H with the coproduct-preserving extension of H to $\mathrm{Fam}(\mathcal{D})$. Equivalently: the left Kan extension of HF^1 along the forgetful functor $\mathrm{El}(\mathcal{C}, F_0) \to \mathcal{C}$.*
2. *A* 2-dimensional op-partial map *$\mathcal{C} \nrightarrow \mathcal{D}$ is a functor $F : \mathcal{C} \to \mathrm{opFam}(\mathcal{D})$. Equivalently: it consists of a functor $F_0 : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ and a functor $F_1 : \mathrm{opEl}(\mathcal{C}, F_0) \to \mathcal{D}$. Any functor $H : \mathcal{D} \to \mathbf{Set}$ gives a functor $\prod_F H : \mathcal{C} \to \mathbf{Set}$ defined to be the composite of H with the product-preserving extension of H to $\mathrm{opFam}(\mathcal{D})$. Equivalently: the right Kan extension of HF^1 along the forgetful functor $\mathrm{opEl}(\mathcal{C}, F_0) \to \mathcal{C}$.*

Now we reformulate our key definitions.

PROPOSITION 6. *A categorical game $\mathcal{G}$ consists of*

- *(small) categories $\mathcal{G}^{\mathrm{pass}}$ and $\mathcal{G}^{\mathrm{act}}$*
- *a two-dimensional op-partial map $\mathrm{Omove} : \mathcal{G}^{\mathrm{pass}} \nrightarrow \mathcal{G}^{\mathrm{act}}$*
- *a two-dimensional partial map $\mathrm{Pmove} : \mathcal{G}^{\mathrm{act}} \nrightarrow \mathcal{G}^{\mathrm{pass}}$.*

PROPOSITION 7. *Let $\mathcal{G}$ be a categorical game.*

1. *A small-step system over $\mathcal{G}$ is a coalgebra for the endofunctor on $\mathbf{Set}^{\mathcal{G}^{\mathrm{pass}}} \times \mathbf{Set}^{\mathcal{G}^{\mathrm{act}}}$ sending*

$$(\mathbb{S}^{\mathrm{pass}}, \mathbb{S}^{\mathrm{act}}) \mapsto (\prod_{\mathrm{Omove}} \mathbb{S}^{\mathrm{act}}, (\sum_{\mathrm{Pmove}} \mathbb{S}^{\mathrm{pass}}) + \mathbb{S}^{\mathrm{act}})$$

2. *A big-step system over $\mathcal{G}$ is a coalgebra for the endofunctor on $\mathbf{Set}^{\mathcal{G}^{\mathrm{pass}}} \times \mathbf{Set}^{\mathcal{G}^{\mathrm{act}}}$ sending*

$$(\mathbb{S}^{\mathrm{pass}}, \mathbb{S}^{\mathrm{act}}) \mapsto (\prod_{\mathrm{Omove}} \mathbb{S}^{\mathrm{act}}, (\sum_{\mathrm{Pmove}} \mathbb{S}^{\mathrm{pass}}) + 1)$$

3. *A passive system over $\mathcal{G}$ is a coalgebra for the endofunctor on $\mathbf{Set}^{\mathcal{G}^{\mathrm{pass}}}$ sending*

$$\mathbb{S}^{\mathrm{pass}} \mapsto \prod_{\mathrm{Omove}} ((\sum_{\mathrm{Pmove}} \mathbb{S}^{\mathrm{pass}}) + 1)$$

More surprisingly, a game is a coalgebra too:

PROPOSITION 8. *A categorical game is a coalgebra for the endofunctor on $\mathbf{Cat}^2$ sending*

$$(\mathcal{G}^{\mathrm{pass}}, \mathcal{G}^{\mathrm{act}}) \mapsto (\mathrm{opFam}(\mathcal{G}^{\mathrm{act}}), \mathrm{Fam}(\mathcal{G}^{\mathrm{pass}}))$$

6. Tensoring

6.1 Tensor games

We often want to run two games $\mathcal{G}$ and $\mathcal{G}'$ in parallel, and we describe this by a "tensor" game $\mathcal{G} \otimes \mathcal{G}'$, following e.g. [2, 11]. A passive position in the tensor consists of a passive position from each game. O can move in either component; then that component becomes active and P can only respond within it, whereupon both components are again passive.

DEFINITION 12. *For games $\mathcal{G}$ and $\mathcal{G}'$, let $\mathcal{G} \otimes \mathcal{G}'$ be the following game:*

- *A passive position is (P, P') where P and P' are passive in $\mathcal{G}$ and $\mathcal{G}'$ respectively.*
- *An active position is either $\mathsf{inl}(Q, P')$ with Q active in $\mathcal{G}$ and P' passive in $\mathcal{G}'$, or $\mathsf{inr}(P, Q')$ with P passive in $\mathcal{G}$ and Q' active in $\mathcal{G}'$.*
- *An O-move from (P, P') is either $\mathsf{inl}\, m$ with $m \in \mathrm{Omove}\, P$, which has target $\mathsf{inl}(P.m, P')$, or $\mathsf{inr}\, m$ with $m \in \mathrm{Omove}\, P'$, which has target $\mathsf{inr}(P, P'.m)$.*
- *A P-move from $\mathsf{inl}(Q, P')$ is $n \in \mathrm{Pmove}\, Q$, and its target is $(Q.n, P')$. Likewise from an inr active position.*

DEFINITION 13. *For categorical games $\mathcal{G}$ and $\mathcal{G}'$, define the categorical game $\mathcal{G} \otimes \mathcal{G}'$ as in Definition 12, with the following additional data.*

- *A passive position morphism from (P, P') to (P'', P''') is a pair (f, f') with $f : P \to P''$ and $f' : P' \to P'''$. Composite and identity morphisms are defined componentwise.*
- *Likewise for inl active positions, and likewise for inr active positions. There are no morphisms from inl to inr active positions or vice versa.*

6.2 Tensor strategies

Any play in $\mathcal{G} \otimes \mathcal{G}'$ has a *left play* and a *right play*. This is illustrated in Fig. 1.

PROPOSITION 9. *Let $\mathcal{G}$ and $\mathcal{G}'$ be games. For passive positions P, P' in $\mathcal{G}, \mathcal{G}'$ respectively, and $\sigma \in \mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}} P$ and $\sigma' \in \mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}} P'$, let $\sigma \otimes \sigma'$ be the set of passive-ending plays in $\mathcal{G} \otimes \mathcal{G}'$ from (P, P') whose left play is in σ and whose right play is in σ'. Then $\sigma \otimes \sigma' \in \mathrm{Strat}_{\mathcal{G} \otimes \mathcal{G}'}^{\mathrm{pass}} (P, P')$. Likewise for inl and for inr active positions.*

PROPOSITION 10. *(Naturality of $\otimes$)*
Let $\mathcal{G}$ and $\mathcal{G}'$ be categorical games. For passive position morphisms $f : P \to P''$ in $\mathcal{G}$ and $f' : P' \to P'''$ in $\mathcal{G}'$, and $\sigma \in \mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}} P$ and $\sigma' \in \mathrm{Strat}_{\mathcal{G}'}^{\mathrm{pass}} P'$, we have

$$(\mathrm{Strat}_{\mathcal{G} \otimes \mathcal{G}'}^{\mathrm{pass}} (f, f'))\, (\sigma \otimes \sigma') = (\mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}} f)\, (\sigma) \otimes (\mathrm{Strat}_{\mathcal{G}'}^{\mathrm{pass}} f')\, (\sigma')$$

Likewise for inl and for inr active positions.

6.3 Tensor systems

DEFINITION 14. *Let $\mathbb{S}$ and $\mathbb{S}'$ be small-step systems over games $\mathcal{G}$ and $\mathcal{G}'$ respectively. The small-step system $\mathbb{S} \otimes \mathbb{S}'$ over $\mathcal{G} \otimes \mathcal{G}'$ is as follows.*

- *A passive state in position (P, P') is a pair (x, x') of states $x \in \mathbb{S}^{\mathrm{pass}}\, P$ and $x' \in \mathbb{S}'^{\mathrm{pass}}\, P'$.*
- *An active state in position $\mathsf{inl}(Q, P')$ is a pair (y, x') of states $y \in \mathbb{S}^{\mathrm{act}}\, Q$ and $x' \in \mathbb{S}^{\mathrm{pass}}\, P'$, and likewise for inr.*
- *$(x, x')@(\mathsf{inl}\ m) \stackrel{\mathrm{def}}{=} (x@m, x')$, and likewise for $\mathsf{inr}\ m$.*
- *$\mathsf{inl}\,(y, x') \begin{cases} \rightsquigarrow & \mathsf{inl}\,(z, x') \quad \text{if } y \rightsquigarrow z \\ \overset{n}{\rightsquigarrow} & (x, x') \quad \text{if } y \overset{n}{\rightsquigarrow} x \end{cases}$*
- *likewise for $\mathsf{inr}(x, y')$.*

We likewise define the tensor of big-step or passive systems. The following shows that this does not cause ambiguity.

PROPOSITION 11. *1. For small-step systems $\mathbb{S}$ and $\mathbb{S}'$ over games $\mathcal{G}$ and $\mathcal{G}'$ respectively, the tensor of the big-step forms of $\mathcal{G}$ and $\mathcal{G}'$ is the big-step form of $\mathcal{G} \otimes \mathcal{G}'$.*
2. For big-step systems $\mathbb{S}$ and $\mathbb{S}'$ over games $\mathcal{G}$ and $\mathcal{G}'$ respectively, the tensor of the passive forms of $\mathcal{G}$ and $\mathcal{G}'$ is the passive form of $\mathcal{G} \otimes \mathcal{G}'$.

The tensor of systems over categorical games may be defined in the evident way, though we shall not use this.

6.4 Compositionality theorem for tensors

We see how to obtain the operational meaning of a tensor state from those of its components.

PROPOSITION 12. *Let $\mathcal{G}$ and $\mathcal{G}'$ be games.*

1. *Let $\mathbb{S}$ and $\mathbb{S}'$ be passive systems over $\mathcal{G}$ and $\mathcal{G}'$ respectively. For $x \in \mathbb{S}^{\mathrm{pass}}\, P$ and $x' \in \mathbb{S}'^{\mathrm{pass}}\, P'$ we have*

$$[\![(x, x')]\!]^{\mathrm{pass}}_{(P,P')} \quad = \quad [\![x]\!]^{\mathrm{pass}}_{P} \otimes [\![x']\!]^{\mathrm{pass}}_{P'}$$

2. *Let $\mathbb{S}$ and $\mathbb{S}'$ be big-step systems over $\mathcal{G}$ and $\mathcal{G}'$ respectively. For $y \in \mathbb{S}^{\mathrm{act}}\, Q$ and $x' \in \mathbb{S}'^{\mathrm{pass}}\, P'$, we have*

$$[\![(y, x')]\!]^{\mathrm{act}}_{\mathsf{inl}\,(Q,P')} \quad = \quad [\![y]\!]^{\mathrm{act}}_{Q} \otimes [\![x']\!]^{\mathrm{pass}}_{P'}$$

Likewise for inr.

By Proposition 11 this result also holds for small-step systems.

7. Operating on strategies: the example of substitution

We return to the transition system $\lambda\mathbf{Syst}$ for our example calculus. For terms $\mathtt{x}, \mathtt{y} \vdash^{\mathsf{nc}} M$ and $\mathtt{z} \vdash^{\mathsf{v}} V$, suppose we have been given two black boxes: the left one containing active state $\blacktriangleright\ M$ in position $\mathtt{x}, \mathtt{y} \parallel$ and the right one containing passive state $(a \mapsto V)$ in position $\mathtt{z} \parallel a$. We can play O-moves into them and wait to see what P-moves come out, but cannot see the syntax. We wish to simulate the behaviour of active state $\blacktriangleright\ M[V/\mathtt{y}]$, formed by substitution, in position $\mathtt{x}, \mathtt{z} \parallel$, just using the behaviour from the left and right black boxes. How shall we proceed? We play on three interfaces: with the left and right boxes and with the external world.

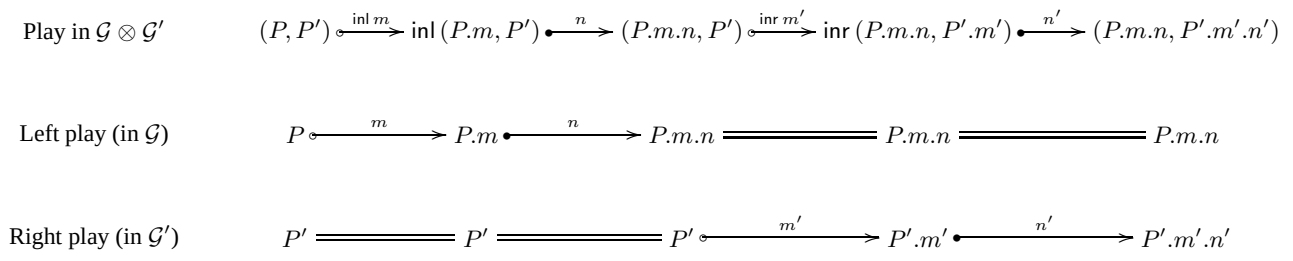


Figure 1. Left and right play

We first wait for the left box $\blacktriangleright M$. If it plays $\mathtt{x}$, then $M \rightsquigarrow^* \mathtt{x}(W_0, W_1)$ and O receives fresh names b_0 and b_1 representing W_0 and W_1 respectively. Thus $M[V/\mathtt{y}] \rightsquigarrow^* \mathtt{x}(W_0[V/\mathtt{y}], W_1[V/\mathtt{y}])$ and we make the latter command our current "external state"—the state we are trying to simulate. So we play $\mathtt{x}$ externally, resulting in position $\mathtt{x}, \mathtt{z} \parallel b_0, b_1$, and our current external state becomes $(b_0 \mapsto W_0[V/\mathtt{y}], b_1 \mapsto W_1[V/\mathtt{y}])$. We record the fact that b_0 and b_1 at the external interface were respectively obtained from b_0 and b_1 at the left interface, so that whenever external Opponent plays b_0, we play b_0 into the left box.

On the other hand, suppose the left box plays $\mathtt{y}$. Then $M \rightsquigarrow^* \mathtt{y}(W_0, W_1)$ and O receives fresh names b_0 and b_1 representing W_0 and W_1 respectively. Thus $M[V/\mathtt{y}] \rightsquigarrow^* V(W_0[V/\mathtt{y}], W_1[V/\mathtt{y}])$ and we make the latter command our current external state. We do not play $\mathtt{y}$ externally (indeed the external position is $\mathtt{x}, \mathtt{z} \parallel$). Instead, we play a into the right box, resulting in state $a \mapsto V \blacktriangleright V(\mathtt{w}_0, \mathtt{w}_1)$ in position $\mathtt{z}, \mathtt{w}_0, \mathtt{w}_1 \parallel a$. We record that $\mathtt{w}_0$ and $\mathtt{w}_1$ at the right interface were respectively obtained from b_0 and b_1 at the the left interface, so that whenever the right box plays $\mathtt{w}_0$, we play b_0 into the left box.

If now the right box plays $\mathtt{z}$, then $V(\mathtt{w}_0, \mathtt{w}_1) \rightsquigarrow^* \mathtt{z}(W_2, W_3)$ so $V(W_0[V/\mathtt{y}], W_1[V/\mathtt{y}]) \rightsquigarrow^*$

$$\mathtt{z}(W_2 \left[\begin{array}{c} W_0[V/\mathtt{y}]/\mathtt{w}_0 \\ W_1[V/\mathtt{y}]/\mathtt{w}_1 \end{array}\right], W_3 \left[\begin{array}{c} W_0[V/\mathtt{y}]/\mathtt{w}_0 \\ W_1[V/\mathtt{y}]/\mathtt{w}_1 \end{array}\right])$$

and we make the latter command our current external state. Therefore we play $\mathtt{z}$ at the external interface. Several points may be learnt from this discussion:

1. When we await the external Opponent, all interfaces are passive. When we await a box, it and the external interface are active; the other box is passive.
2. We need to maintain a "linker" function saying where each O-name in the external interface, and each P-name in each internal interface, came from.
3. At any time there is a current external state that we are trying to simulate. It is constructed from chains of substitutions that grow as play continues.
4. When we move between internal (left and right) interfaces, the external state does not change. We must rule out an infinite sequence of consecutive such events, to ensure the external state makes progress during our simulation.

8. Transfers Between Games

8.1 Transfers

We introduce a notion of *transfer* from one game to another, which is a recipe for converting a strategy for the first game into a strategy for the second. As we shall see, the procedure outlined in Section 7 forms a transfer $\lambda\mathbf{Game} \otimes \lambda\mathbf{Game} \to \lambda\mathbf{Game}$.

DEFINITION 15. *Let $\mathcal{G}$ and $\mathcal{H}$ be (discrete) games. A* transfer $\mathcal{O} : \mathcal{G} \to \mathcal{H}$ *consists of the following data.*

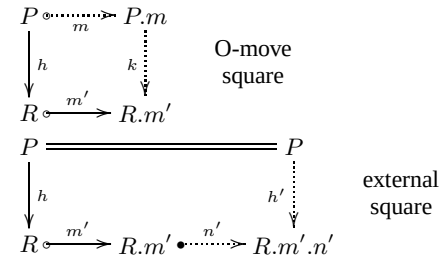
- *For each pair of passive positions P in $\mathcal{G}$ and R in $\mathcal{H}$, a set $\mathcal{O}^{\mathrm{pass}}(P, R)$ of* passive linkers $P \to R$.
- *For each pair of active positions Q in $\mathcal{G}$ and S in $\mathcal{H}$, a set $\mathcal{O}^{\mathrm{act}}(Q, S)$ of* active linkers $Q \to S$.
- *For each passive linker $h : P \to R$ a function $\gamma_h^{\mathrm{pass}} \in$*

$$\prod_{m' \in \mathrm{Omove}\, R} ((\sum_{m \in \mathrm{Omove}\, P} \mathcal{O}^{\mathrm{act}}(P.m, R.m')) + (\sum_{n' \in \mathrm{Pmove}\,(R.m')} \mathcal{O}^{\mathrm{pass}}(P, R.m'.n')))$$

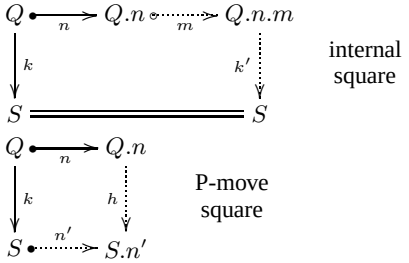
- *For each active linker $k : Q \to S$ a function $\gamma_k^{\mathrm{act}} \in$*

$$\prod_{n \in \mathrm{Pmove}\, Q} ((\sum_{m \in \mathrm{Omove}\,(Q.n)} \mathcal{O}^{\mathrm{act}}(Q.n.m, S)) + (\sum_{n' \in \mathrm{Pmove}\, S} \mathcal{O}^{\mathrm{pass}}(Q.n, S.n'))$$

We depict the cases $\gamma_h^{\mathrm{pass}}(m)$ of $\mathsf{inl}\,(m, k)$ and $\mathsf{inr}\,(n', h')$ as respectively:

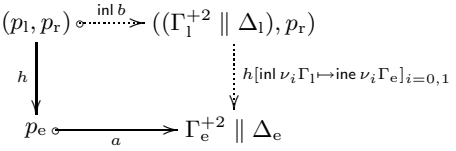


We depict the cases $\gamma_k^{\mathrm{act}}(n')$ of $\mathsf{inl}\,(m, k')$ and $\mathsf{inr}\,(n', h)$ as respectively:

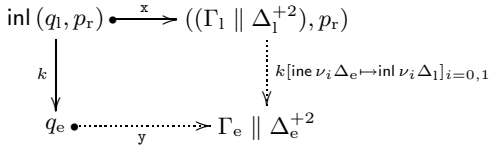


As promised at the start of the section, we give a transfer $\lambda\mathbf{Comp}$ from $\lambda\mathbf{Game} \otimes \lambda\mathbf{Game}$ to $\lambda\mathbf{Game}$. A passive linker $(p_\mathrm{l}, p_\mathrm{r}) \to p_\mathrm{e}$, where $p_\mathrm{l} = \Gamma_\mathrm{l} \parallel \Delta_\mathrm{l}$ etc., is a function $\Gamma_\mathrm{l} + \Gamma_\mathrm{r} + \Delta_\mathrm{e} \to \Delta_\mathrm{l} + \Delta_\mathrm{r} + \Gamma_\mathrm{e}$ mapping each name to one at a different interface. (We use $\mathsf{inl}, \mathsf{inr}, \mathsf{ine}$ as ternary sum constructors.) Likewise for active linkers $\mathsf{inl}\,(q_\mathrm{l}, p_\mathrm{r}) \to q_\mathrm{e}$ or $\mathsf{inr}\,(p_\mathrm{l}, q_\mathrm{r}) \to q_\mathrm{e}$. The squares are

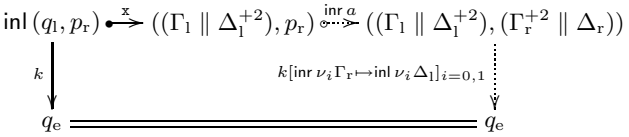
- if $h(\mathsf{ine}\, a) = \mathsf{inl}\, b$ then O-move square



  and likewise if $h(\mathsf{ine}\, a) = \mathsf{inr}\, b$
- if $k(\mathsf{inl}\, \mathsf{x}) = \mathsf{ine}\, \mathsf{y}$ then P-move square



  and if $k(\mathsf{inl}\, \mathsf{x}) = \mathsf{inr}\, a$ then internal square



- likewise for inr.

To define a transfer between categorical games, we need the well-known notion of bimodule (also called "profunctor" and "distributor").

DEFINITION 16. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A $(\mathcal{C}, \mathcal{D})$-*bimodule *$\mathcal{M}$ consists of the following data:*

- *for each $X \in \mathcal{C}$ and $Y \in \mathcal{D}$ a set $\mathcal{M}(X, Y)$ of $\mathcal{M}$-*morphisms *$X \to Y$*
- *for any $\mathcal{C}$- and $\mathcal{M}$-morphisms $X' \xrightarrow{f} X \xrightarrow{g} Y$ an $\mathcal{M}$-morphism $X' \xrightarrow{f;g} Y$*
- *for any $\mathcal{M}$- and $\mathcal{D}$-morphisms $X \xrightarrow{g} Y \xrightarrow{h} Y'$ an $\mathcal{M}$-morphism $X \xrightarrow{g;h} Y'$*

The following five equations must be satisfied:

- *For a $\mathcal{M}$-morphism $g : X \to Y$ we have $\mathrm{id}_X; g = g = g; \mathrm{id}_Y$.*
- *For two $\mathcal{C}$-morphisms and an $\mathcal{M}$-morphism:*

$$X'' \xrightarrow{f'} X' \xrightarrow{f} X \xrightarrow{g} Y$$

  we have $(f'; f); g = f'; (f; g)$
- *For a $\mathcal{C}$-morphism and $\mathcal{M}$-morphism and $\mathcal{D}$-morphism:*

$$X' \xrightarrow{f} X \xrightarrow{g} Y \xrightarrow{h} Y'$$

  we have $(f; g); h = f; (g; h)$
- *For a $\mathcal{M}$-morphism and two $\mathcal{D}$-morphisms*

$$X \xrightarrow{g} Y \xrightarrow{h} Y' \xrightarrow{h'} Y''$$

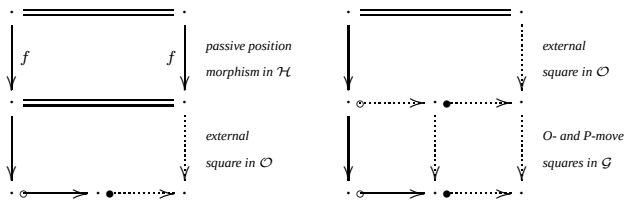
  we have $(g; h); h' = g; (h; h')$.

Concisely: a $(\mathcal{C}, \mathcal{D})$-bimodule is a functor $\mathcal{C}^{\mathrm{op}} \times \mathcal{D} \to \mathbf{Set}$.

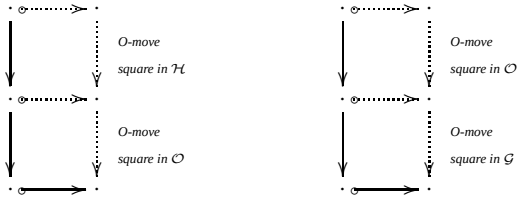
DEFINITION 17. *Let $\mathcal{G}$ and $\mathcal{H}$ be categorical games. A* transfer *$\mathcal{O} : \mathcal{G} \to \mathcal{H}$ is defined as in Def. 15, with the following additional data.*

- *Composition of a passive linker $P \to R$ with a passive position morphism $P' \to P$ in $\mathcal{G}$, and with a passive position morphism $R \to R'$ in $\mathcal{H}$, satisfying the five equations in Def. 16.*
- *Likewise for active linkers.*

We require all composites



to be external squares, all composites



to be O-move squares, and likewise for P-move and internal squares.

Our example $\lambda\mathbf{Comp}$ is a categorical transfer: linker composition is given by function composition in the evident way.

8.2 Transfer operating on strategies

To define how a transfer $\mathcal{O}$ operates on strategies, we consider "$\mathcal{O}$-interaction sequences" comprised of O-move, external, P-move and internal squares. An example is shown in Fig. 2. We may describe this interaction sequence from h by the sequence of moves $m_0', n_0, n_1, m_1', n_2, m_2'$ depicted as solid, since they determine the other moves and morphisms. The play from P appearing along the upper edge is called the *internal play*; the play from R appearing along the lower edge is called the *external play*.

PROPOSITION 13. *Let $\mathcal{O} : \mathcal{G} \to \mathcal{H}$ be a transfer. For any passive linker $h : P \to R$ and strategy $\sigma \in \mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}}\, P$ let $(\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, h)\sigma$ be the set of all external plays of passive-ending $\mathcal{O}$-interaction sequences from h whose internal play is in σ. Then $(\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, h)\sigma \in \mathrm{Strat}_{\mathcal{H}}^{\mathrm{pass}}\, R$. Likewise for an active linker.*

PROPOSITION 14. *Let $\mathcal{O} : \mathcal{G} \to \mathcal{H}$ be a transfer of categorical games.*

1. *For passive position morphism in $\mathcal{G}$ and linker*

$$P' \xrightarrow{f} P \xrightarrow{h} R$$

   and strategy in $\mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}}\, R'$ we have

$$(\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, (f; h))\, (\sigma) = (\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, h)\, (\mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}}\, f)\, (\sigma)$$

   Likewise for active positions.
2. *For passive linker and position morphism in $\mathcal{H}$*

$$P \xrightarrow{h} R \xrightarrow{f} R'$$

   and strategy in $\mathrm{Strat}_{\mathcal{G}}^{\mathrm{pass}}\, R$ we have

$$(\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, (h; f))\, (\sigma) = (\mathrm{Strat}_{\mathcal{H}}^{\mathrm{pass}}\, f)\, (\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, h)\, (\sigma)$$

   Likewise for active positions.

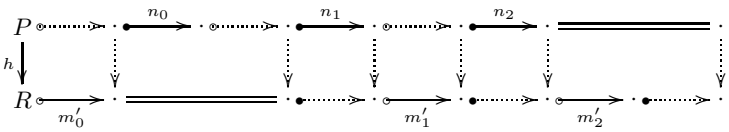


Figure 2. An interaction sequence

8.3 Stepped bisimulation across a transfer

We now establish the relationship between transition systems and transfers, i.e. between operational and denotational game semantics. We shall consider only discrete games.

To ensure that we do not perform infinitely many operations at the internal interface without making progress, we "grade" the states to give a finite quota of internal operations.

DEFINITION 18. *Given a transfer $\mathcal{O} : \mathcal{G} \to \mathcal{H}$ and small-step systems $\mathbb{S}$ over $\mathcal{G}$ and $\mathbb{T}$ over $\mathcal{H}$, a* stepped bisimulation *from $\mathbb{S}$ to $\mathbb{T}$ across $\mathcal{O}$ consists of the following data.*

- *For each active linker $k : Q \to S$, an increasing sequence $(U^i_k)_{i\in\mathbb{N}}$ of subsets of $\mathbb{S}^{\mathrm{act}}\, Q$.*
- *For each passive linker $h : P \to R$, a relation $\mathcal{R}^{\mathrm{pass}}_h$ from $\mathbb{S}^{\mathrm{pass}}\, P$ to $\mathbb{T}^{\mathrm{pass}}\, R$.*
- *For each active linker $k : Q \to S$, a relation $\mathcal{R}^{\mathrm{act}}_k$ from $\mathbb{S}^{\mathrm{act}}\, Q$ to $\mathbb{T}^{\mathrm{act}}\, S$.*

These are required to satisfy the following conditions.

- *For any active linker $k : Q \to S$ and $i \in \mathbb{N}$ and $y \in U^i_k$, if $y \overset{n}{\rightsquigarrow} x$ and n completes to an internal square*

$$\begin{array}{ccc} Q & \xrightarrow[n]{} & Q.n \xrightarrow[m]{} Q.n.m \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle k'} \\ S & = & S \end{array} \qquad (4)$$

  then $x@m \in U^j_{k'}$ for some $j < i$.
- *For any passive linker $h : P \to R$ and $x\ (\mathcal{R}^{\mathrm{pass}}_h)\ x'$ and $m' \in \mathrm{Omove}\, R$,*
  - *if m completes to an O-move square*

$$\begin{array}{ccc} P & \xrightarrow[m]{} & P.m \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle k} \\ R & \xrightarrow{m'} & R.m' \end{array} \qquad \text{then } x@m\ (\mathcal{R}^{\mathrm{act}}_k)\ x'@m'$$

  - *if m completes to an external square*

$$\begin{array}{ccccc} P & = & P & & \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle h'} & & \\ R & \xrightarrow{m'} & R.m' & \xrightarrow{n'} & R.m'.n' \end{array}$$

    then $x'@m' \overset{n}{\rightsquigarrow} w'$ with $x\ (\mathcal{R}^{\mathrm{pass}}_{h'})\ w'$.
- *For any active linker $k : Q \to S$ and $y\ (\mathcal{R}^{\mathrm{act}}_k)\ y'$,*
  - *$y \in U^i_k$ for some $i \in \mathbb{N}$*
  - *if $y \rightsquigarrow z$ then $y' \rightsquigarrow z'$ with $z\ (\mathcal{R}^{\mathrm{act}}_k)\ z'$*
  - *if $y \overset{n}{\rightsquigarrow} x$ and n completes to a P-move square*

$$\begin{array}{ccc} Q & \xrightarrow[n]{} & Q.n \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle h} \\ S & \xrightarrow{n'} & S.n' \end{array}$$

    then $y' \overset{n'}{\rightsquigarrow} x'$ with $x\ (\mathcal{R}^{\mathrm{pass}}_h)\ x'$
  - *if $y \overset{n}{\rightsquigarrow} x$ and n completes to an internal square (4) then $x@m\ (\mathcal{R}^{\mathrm{act}}_{k'})\ y'$.*

Recall our example transfer $\lambda\mathbf{Comp} : \lambda\mathbf{Game}\otimes\lambda\mathbf{Game} \to \lambda\mathbf{Game}$. Our aim is to give a stepped bisimulation $(U, \mathcal{R})$ from $\lambda\mathbf{Syst} \otimes \lambda\mathbf{Syst}$ to $\lambda\mathbf{Syst}$ across this transfer.

Given a passive linker $h : (p_{\mathrm{l}}, p_{\mathrm{r}}) \to p_{\mathrm{e}}$, an *$h$-grading* ϕ consists of a natural number $|\phi|$ (the number of "generations") and maps $\phi_{\mathrm{l}} : \Delta_{\mathrm{l}} \to \$|\phi|$ and $\phi_{\mathrm{r}} : \Delta_{\mathrm{r}} \to \$|\phi|$ (saying when each name was generated). For an h-grading ϕ and $i \in \mathbb{N}$ we write $\mathsf{Older}^{\phi}_{\mathrm{l}}\, i \subseteq \Gamma_{\mathrm{l}}$ and $\mathsf{Older}^{\phi}_{\mathrm{r}}\, i \subseteq \Gamma_{\mathrm{r}}$ for the names that are older than generation i. Explicitly: $\mathsf{Older}^{\phi}_{\mathrm{l}}\, i$ consists of those $\mathtt{x} \in \Gamma_{\mathrm{l}}$ such that $k(\mathsf{inl}\,\mathtt{x})$ is either $\mathsf{inr}\, a$ with $\phi_{\mathrm{r}}(a) < i$ or $\mathsf{ine}\,\mathtt{y}$. (External P-names are deemed to have always existed.)

Now a passive state in position $(p_{\mathrm{l}}, p_{\mathrm{r}})$ takes the form

$$((V^{\mathrm{l}}_a), (V^{\mathrm{r}}_a)_{a\in\Delta_{\mathrm{r}}}) \qquad (5)$$

The state (5) is *graded* by an h-grading ϕ when for all $a \in \Delta_{\mathrm{l}}$ we have $\mathsf{Older}^{\phi}_{\mathrm{l}}\, \phi_{\mathrm{l}}(a) \vdash^{\mathsf{v}} V^{\mathrm{l}}_a$ and likewise for $a \in \Delta_{\mathrm{r}}$. If (5) is h-gradeable (i.e. is graded by some h-grading), then we shall define its *h-substitution*, a passive state in position p_{e} as follows. First associate to each $u \in \Delta_{\mathrm{l}} + \Delta_{\mathrm{r}} + \Gamma_{\mathrm{e}}$ a value $\Gamma_{\mathrm{e}} \vdash^{\mathsf{v}} W_u$ by the equations

$$\begin{array}{rcl} W_{\mathsf{inl}\, a} & = & V^{\mathrm{l}}_a[W_{h(\mathsf{inl}\,\mathtt{x})}/\mathtt{x}]_{\mathtt{x}\in\Gamma_{\mathrm{l}}} \\ W_{\mathsf{inr}\, a} & = & V^{\mathrm{r}}_a[W_{h(\mathsf{inr}\,\mathtt{x})}/\mathtt{x}]_{\mathtt{x}\in\Gamma_{\mathrm{r}}} \\ W_{\mathsf{ine}\,\mathtt{x}} & = & \mathtt{x} \end{array}$$

which have a unique solution by induction over a grading. Then define the h-substitution of (5) to be $(W_{h(\mathsf{ine}\, a)})_{a\in\Delta_{\mathrm{e}}}$.

We set $\mathcal{R}^{\mathrm{pass}}_h$ to be the partial function relating each h-gradeable state (5) to its h-substitution.

Given an active linker $k : \mathsf{inl}(q_{\mathrm{l}}, p_{\mathrm{r}}) \to q_{\mathrm{e}}$ we likewise define a *k-grading*. An active state in position $\mathsf{inl}(q_{\mathrm{l}}, p_{\mathrm{r}})$ takes the form

$$((V^{\mathrm{l}}_a)_{a\in\Delta_{\mathrm{l}}} \blacktriangleright M, (V^{\mathrm{r}}_a)_{a\in\Delta_{\mathrm{r}}}) \qquad (6)$$

The state (6) is *graded* by a k-grading ϕ when for all $a \in \Delta_{\mathrm{l}}$ we have $\mathsf{Older}^{\phi}_{\mathrm{l}}\, \phi_{\mathrm{l}}(a) \vdash^{\mathsf{v}} V^{\mathrm{l}}_a$ and likewise for $a \in \Delta_{\mathrm{r}}$. For $i \in \mathbb{N}$, the state (6) is graded *at level i* by ϕ when the following additional condition holds:

$$M = V(W, W') \quad \text{implies} \quad \mathsf{Older}^{\phi}_{\mathrm{l}}\, i \vdash^{\mathsf{v}} V.$$

If (6) is k-gradeable, we define its *k-substitution*, an active state in position q_{e}, to be

$$(W_{k(\mathsf{ine}\, a)})_{a\in\Delta_{\mathrm{e}}} \blacktriangleright M[W_{k(\mathsf{inl}\,\mathtt{x})}/\mathtt{x}]_{\mathtt{x}\in\Gamma_{\mathrm{l}}}$$

where W_u is defined as in the passive case.

We set $\mathcal{R}^{\mathrm{act}}_k$ to be the partial function relating each k-gradeable state (6) to its k-substitution. We set U^i_k to be the set of active states (6) that are k-gradeable at level i. Likewise for the inr case.

Then $(U, \mathcal{R})$ is a stepped bisimulation across $\lambda\mathbf{Comp}$.

8.4 Compositionality theorem for transfers

PROPOSITION 15. *Given a transfer $\mathcal{O} : \mathcal{G} \to \mathcal{H}$ and small-step systems $\mathbb{S}$ over $\mathcal{G}$ and $\mathbb{T}$ over $\mathcal{H}$, let $(U, \mathcal{R})$ be a stepped bisimulation from $\mathbb{S}$ to $\mathbb{T}$ across $\mathcal{O}$.*

1. *For a passive linker $h : P \to R$, if $x\,(\mathcal{R}_h^{\mathrm{pass}})\,x'$ then*
$$[\![x']\!]_R^{\mathrm{pass}} \quad = \quad (\mathrm{Strat}_{\mathcal{O}}^{\mathrm{pass}}\, h)\,[\![x]\!]_P^{\mathrm{pass}}$$
2. *For an active linker $k : Q \to S$, if $y\,(\mathcal{R}_k^{\mathrm{act}})\,y'$ then*
$$[\![y']\!]_S^{\mathrm{pass}} \quad = \quad (\mathrm{Strat}_{\mathcal{O}}^{\mathrm{act}}\, k)\,[\![y]\!]_Q^{\mathrm{act}}$$

We illustrate this with our example. Let the active linker

$$k : \mathsf{inl}\,((\mathtt{x}, \mathtt{y} \parallel), (\mathtt{z} \parallel a)) \to (\mathtt{x}, \mathtt{z} \parallel)$$

send $\mathsf{inl}\,\mathtt{x} \mapsto \mathsf{ine}\,\mathtt{x}$ and $\mathsf{inl}\,\mathtt{y} \mapsto \mathsf{inr}\,a$ and $\mathsf{inr}\,\mathtt{z} \mapsto \mathsf{ine}\,\mathtt{z}$. The state $\mathsf{inl}\,(\blacktriangleright M, (a \mapsto V))$ is graded by the k-grading $(1, (\,), (a \mapsto 0))$ and its k-substitution is $\blacktriangleright M[V/\mathtt{y}]$. Since $(U, \mathcal{R})$ is a stepped bisimulation across $\lambda\mathbf{Comp}$, we obtain

$$[\![\blacktriangleright M[V/\mathtt{y}]]\!]^{\mathrm{act}}_{\mathtt{x},\mathtt{z}\parallel}$$

$$= \quad (\mathrm{Strat}^{\mathrm{act}}_{\lambda\mathbf{Comp}}\, k)\,[\![\mathsf{inl}\,(\blacktriangleright M, (a \mapsto V))]\!]^{\mathrm{act}}_{\mathsf{inl}\,((\mathtt{x},\mathtt{y}\parallel),(\mathtt{z}\parallel a))} \qquad \text{(by Prop. 15)}$$

$$= \quad (\mathrm{Strat}^{\mathrm{act}}_{\lambda\mathbf{Comp}}\, k)\,([\![\blacktriangleright M]\!]^{\mathrm{act}}_{\mathtt{x},\mathtt{y}\parallel} \otimes [\![a \mapsto V]\!]^{\mathrm{pass}}_{\mathtt{z}\parallel a}) \qquad \text{(by Prop. 12)}$$

This validates the procedure described in Sect. 7.

9. Dual Games and Transfers

There is an evident involution on games:

DEFINITION 19. *For a categorical game*

$$\mathcal{G} = (\mathcal{G}^{\mathrm{pass}}, \mathcal{G}^{\mathrm{act}}, \mathrm{Omove}, \mathrm{Pmove})$$

its dual *is given by*

$$\mathcal{G}^{\perp} \stackrel{\text{def}}{=} ((\mathcal{G}^{\mathrm{act}})^{\mathrm{op}}, (\mathcal{G}^{\mathrm{pass}})^{\mathrm{op}}, \mathrm{Pmove}^{\mathrm{op}}, \mathrm{Omove}^{\mathrm{op}})$$

This provides a concise formulation of transfer:

PROPOSITION 16. *Let $\mathcal{G}$ and $\mathcal{H}$ be categorical games. A transfer $\mathcal{G} \to \mathcal{H}$ is a total passive system over $\mathcal{G} \multimap \mathcal{H} \stackrel{\text{def}}{=} (\mathcal{G} \otimes \mathcal{H}^{\perp})^{\perp}$*

Remark Transfers cannot be composed, because of the possibility of "infinite chattering" [2], but *partial* transfers can be, giving rise to a $*$-autonomous bicategory.

10. Conclusions and Future Work

We have described a basic framework with a "checklist" for formulating a game model of a language:

- give a small-step system over a categorical game
- for each syntactic operation, give a transfer, and a stepped bisimulation across it to demonstrate its correctness.

It remains to examine the many game models in the literature that use justification pointers to see how well they fit this framework. We shall need to study transfers that create multiple threads of the strategy they act on, as in [11]. We also should treat *nondeterministic* systems, where there is a proliferation of notions of equivalence, hence of strategy. A final intriguing question is whether our sequential framework can be adapted for concurrent systems.

References

[1] Abramsky, S.: The lazy λ-calculus. In: Research topics in Functional Programming, pp. 65–117. Addison Wesley (1990)

[2] Abramsky, S.: Semantics of interaction: an introduction to game semantics. In: 1996 CLiCS Summer School. Cambridge U. Press (1997)

[3] Abramsky, S., Honda, K., McCusker, G.: A fully abstract game semantics for general references. In: 13th LICS (1998)

[4] Blum, W., Ong, C.H.L.: The safe lambda calculus. Logical Methods in Computer Science 5(1) (2009)

[5] Curien, P.L.: Abstract Böhm trees. Math. Structures in CS 8(6) (1998)

[6] Danos, V., Herbelin, H., Regnier, L.: Game semantics and abstract machines. In: 11th LICS (1996)

[7] Ghica, D.R., Tzevelekos, N.: A system-level game semantics. 28th Mathematical Foundations in Programming Semantics, ENTCS 286

[8] Harmer, R., Hyland, M., Melliès, P-A.: Categorical combinatorics for innocent strategies. LICS 2007: 379-388

[9] Howe, D.J.: Proving congruence of bisimulation in functional programming languages. Information and Computation 124(2) (1996)

[10] Hyland, J.M.E., Ong, C.H.L.: On full abstraction for PCF: I, II, and III. Information and Computation 163(2) (2000)

[11] Hyland, J.M.E.: Game semantics. In: Semantics and Logics of Computation. Cambridge University Press (1997)

[12] Hyvernat, P.: A linear category of polynomial diagrams. Mathematical Structures in Computer Science 24 (2013)

[13] Jagadeesan, R., Pitcher, C.S., Riely, J.: Open bisimulation for aspects. In: Aspect-Oriented System Development. ACM Press (2007)

[14] Jagadeesan, R., Pitcher, C., Rathke, J., Riely, J.: Local memory via layout randomization. In: 24th Computer Security Foundations Symposium (2011),

[15] Jeffrey, A., Rathke, J.: Java Jr.: Fully abstract trace semantics for a core Java language. European Symp. on Programming, LNCS 3444 (2005)

[16] Jeffrey, A., Rathke, J.: A fully abstract may testing semantics for concurrent objects. Theoretical Computer Science 338 (2005)

[17] Kozen, D.: Realization of coinductive types. In: 27th Conference on the Math. Foundations of Programming Semantics ENTCS 276 (2011)

[18] Lack, S., Street, R.: The formal theory of monads II. Journal of Pure and Applied Algebra 175 (2002)

[19] Laird, J.: Full abstraction for functional languages with control. In: 12th LICS (1997)

[20] Laird, J.: A fully abstract game semantics of local exceptions. In: 16th LICS (2001)

[21] Laird, J.: A fully abstract trace semantics for general references. In: 34th ICALP, LNCS 4596 (2007)

[22] Laird, J.: Game semantics for a polymorphic programming language. In: LICS (2010)

[23] Laird, J.: Game semantics for call-by-value polymorphism. In: 37th ICALP, LNCS 6199 (2010)

[24] Lassen, S.B.: Eager normal form bisimulation. In: 20th LICS (2005)

[25] Lassen, S.B., Levy, P.B.: Typed normal form bisimulation. In: 23rd CSL, LNCS, 4646 (2007)

[26] Lassen, S.B., Levy, P.B.: Typed normal form bisimulation for parametric polymorphism. In: LICS (2008)

[27] Murawski, A.S., Tzevelekos, N.: Game semantics for good general references. In: LICS (2011)

[28] Sangiorgi, D.: The lazy lambda calculus in a concurrency scenario. Information and Computation 111(1), 120–153 (1994)

[29] Støvring, K., Lassen, S.: A complete, co-inductive syntactic theory of sequential control and state. In: Semantics and Algebraic Specification, P. Mosses 60th birthday festschrift, LNCS 5700 (2009)

[30] Turi, Plotkin: Towards a mathematical operational semantics. In: LICS: IEEE Symposium on Logic in Computer Science (1997)

[31] Winskel, G: Strategies as Profunctors. In: FoSSaCS 2013.