Games on Position Categories (Work in progress)

Paul Blain Levy[1] Sam Staton[2]

[1]University of Birmingham

[2]University of Cambridge

January 15, 2013

Operational game semantics

An approach to game semantics initiated in 2007 by

- Jagadeesan, Pitcher, Riely
- Laird
- Lassen, Levy

Intended to capture the intuition that game semantics describes the behaviour of a term.

We build transition systems out of terms.

The meaning of a term is its set of traces.

Outline

1. Discrete Games
2. Example
3. Categorical Games
4. Operating on Strategies

The counter game

Two players, Proponent and Opponent, play alternately.
On the table is either

- a light colour (pink, cyan or yellow)—Proponent's turn
- a dark counter (purple or brown)—Opponent's turn.

When the counter is pink, Proponent has 3 legal moves

$$6, 9, 23$$

and the counter turns purple, brown or purple respectively.
When the counter is purple, Opponent has 4 legal moves

$$1, 6, 8, 12$$

and the counter turns yellow, pink, pink or pink respectively.
And so forth.

Games

A game consists of the following data:

- A set ActPos of active positions.
- A set PassPos of passive positions.
- For each active position p, a countable set $\mathrm{Pmove}(p)$ of P-moves (outputs).
- Each move $a \in \mathrm{Pmove}(p)$ has an outcome $p.a$, a passive position.
- For each passive position p, a countable set $\mathrm{Omove}(p)$ of O-moves (inputs).
- Each move $a \in \mathrm{Omove}(p)$ has an outcome $p.a$, an active position.

Games

A game consists of the following data:

- A set ActPos of active positions.
- A set PassPos of passive positions.
- For each active position p, a countable set $\mathrm{Pmove}(p)$ of P-moves (outputs).
- Each move $a \in \mathrm{Pmove}(p)$ has an outcome $p.a$, a passive position.
- For each passive position p, a countable set $\mathrm{Omove}(p)$ of O-moves (inputs).
- Each move $a \in \mathrm{Omove}(p)$ has an outcome $p.a$, an active position.

It's just a bipartite graph.

A game consists of the following data:

- A set ActPos of active positions.
- A set PassPos of passive positions.
- For each active position p, a countable set $\mathrm{Pmove}(p)$ of P-moves (outputs).
- Each move $a \in \mathrm{Pmove}(p)$ has an outcome $p.a$, a passive position.
- For each passive position p, a countable set $\mathrm{Omove}(p)$ of O-moves (inputs).
- Each move $a \in \mathrm{Omove}(p)$ has an outcome $p.a$, an active position.

It's just a bipartite graph.

By contrast with e.g. Lamarche games, there is no initial position.

Transition system over a game $\mathfrak{G}$

A transition system over $\mathfrak{G}$ consists of the following data.

- A set of nodes, each of which has a position in G.
- We write $X(p)$ for the set of nodes in passive position p.
- We write $Y(p)$ for the set of nodes in active position p.
- For every passive position p

$$X(p) \longrightarrow \prod_{a \in \mathrm{Omove}(p)} Y(p.a) \qquad \textit{deterministic}$$

- For every active position p

$$Y(p) \longrightarrow \mathrm{Maybe} \sum_{a \in \mathrm{Pmove}(p)} X(p.a)$$

Transition system over a game $\mathfrak{G}$

A transition system over $\mathfrak{G}$ consists of the following data.

- A set of nodes, each of which has a position in G.
- We write $X(p)$ for the set of nodes in passive position p.
- We write $Y(p)$ for the set of nodes in active position p.
- For every passive position p

$$X(p) \longrightarrow \prod_{a \in \mathrm{Omove}(p)} Y(p.a) \qquad \textit{deterministic}$$

- For every active position p

$$Y(p) \longrightarrow \mathrm{Maybe} \sum_{a \in \mathrm{Pmove}(p)} X(p.a)$$

Coalgebra for endofunctor $\mathrm{Maybe}^{\mathfrak{G}}$ on $\mathbf{Set}^{\mathrm{ActPos}} \times \mathbf{Set}^{\mathrm{PassPos}}$.
(Cf. Kozen's notion.)

One Active Position, One Passive Position

Game consists of a set I of inputs and a set O of outputs.
A transition system is two sets X and Y and

$$\begin{array}{rcl} X & \longrightarrow & I \to Y \\ Y & \longrightarrow & \text{Maybe}\,(O \times X) \end{array}$$

Variants

- A passive transition system is a set X and function

$$X \longrightarrow I \to \text{Maybe}\,(O \times X)$$

- An active transition system is a set Y is a

$$Y \longrightarrow \text{Maybe}\,(O \times (I \to Y))$$

Strategies from a position p

We write $\mathrm{strat}\ ^{\mathfrak{G}}p$ for the set of strategies (for Proponent) starting from position p.

Abstract definition

$\mathrm{strat}\ ^{\mathfrak{G}}$ is the final coalgebra for our functor Maybe$^{\mathfrak{G}}$.

Concrete definition

- A play from p is a path in the game diagram.
- A strategy from p is a set σ of passive paths (i.e. ending in a passive position) satisfying prefix-closure and determinacy and containing ε (if p is passive).

Example: Calculus of No Return

The target calculus of the CPS transform.

Summary of Syntax

Types $A ::= \sum_{i \in I} A_i \mid 1 \mid A \times A \mid \neg A \mid \mathtt{rec}\ \mathtt{X}.\ A$

Terms Values $\Gamma \vdash^{\mathsf{v}} V : A$ and non-returning computations $\Gamma \vdash^{\mathsf{nc}} M$

$\neg A$ is the type of functions that take an argument of type A

and never return.

Innocent game for Calculus of No Return

Active positions and nodes

An active position is Γ^{P}, a finite set of names "belonging to P" each with function type.
A node in this position is a command $\Gamma^{\mathsf{P}} \vdash^{\mathsf{nc}} M$.

Passive positions and nodes

A passive position is Γ^{P}, a finite set of names "belonging to "P", and Γ^{O}, a finite set of names "belonging to O". Each has function type.
A node in this position provides a value $\Gamma \vdash^{\mathsf{v}} V : \neg A$ for each $(\bar{\mathtt{u}} : \neg A) \in \Gamma^{\mathsf{O}}$.

O-moves

Suppose we're in passive position

$$\mathtt{x} : \neg A, \mathtt{u} : \neg((\neg B \times \neg B') + \neg C)$$

in node $\mathtt{u} \mapsto V$, and Opponent plays $\overline{\mathtt{x}}$ inl $\langle -, - \rangle$.

The outcome is active position

$$\mathtt{x} : \neg A, \mathtt{y} : \neg B, \mathtt{z} : \neg C$$

and the node is V inl $\langle \mathtt{y}, \mathtt{z} \rangle$.

O-moves

Suppose we're in passive position

$$\mathtt{x} : \neg A, \mathtt{u} : \neg((\neg B \times \neg B') + \neg C)$$

in node $\mathtt{u} \mapsto V$, and Opponent plays $\overline{\mathtt{x}}\ \mathsf{inl}\ \langle -, - \rangle$.

The outcome is active position

$$\mathtt{x} : \neg A, \mathtt{y} : \neg B, \mathtt{z} : \neg C$$

and the node is $V\ \mathsf{inl}\ \langle \mathtt{y}, \mathtt{z} \rangle$.

The names $\mathtt{y}$ and $\mathtt{z}$ must be fresh, i.e. distinct and $\notin \{\mathtt{x}\}$.

These are provided deterministically by a gensym function. So they are not just fresh but canonically fresh.

Renaming Lemma

For a renaming $\Gamma \xrightarrow{\theta} \Gamma'$, we can obtain the meaning of $\theta^* c$ from that of c.

Easy to prove.

Question Can we get this for free?

Substitution Lemma

We can obtain the meaning of $M[V/\mathtt{x}]$ from that of M and V.

As we do this we have a set of names that includes all those used by M, by V and $M[V/\mathtt{x}]$.

As we work out the meaning of $M[V/\mathtt{x}]$, we have to maintain three sets of names, and dictionaries relating them, and apply gensym to each set separately.

Question Can't we maintain a single global set of names, and apply gensym to that instead?

Bisimulation up to renaming

A relation $\mathcal{R}$ on nodes in the same position is a bisimulation up to renaming when

- if M, M' are $\mathcal{R}$-related active nodes, and M moves in a way that passes 3 functions, then M' makes the same move, and there is some choice of 3 fresh names making the resulting passive nodes $\mathcal{R}$-related
- if $\overrightarrow{V}, \overrightarrow{V'}$ are $\mathcal{R}$-related passive nodes, then for any O-move that passes 5 functions, there is some choice of 5 fresh names making the resulting active nodes $\mathcal{R}$-related.

This is a sound principle, i.e. any such relation is contained in bisimilarity.

Question Can't we obtain this for free?

Problems

- Want renaming lemma for free.
- Want to prove substitution lemma using a single global set of names.
- Want soundness of bisimulation up to renaming for free.

Problems

- Want renaming lemma for free.
- Want to prove substitution lemma using a single global set of names.
- Want soundness of bisimulation up to renaming for free.

We want

- canonical fresh names in our semantics
- the freedom to use convenient fresh names in our reasoning (cf. Gabbay-Pitts freshness quantifier).

The Category $\mathsf{Fam_c}(\mathcal{C})$

Any category $\mathcal{C}$ gives rise to another category $\mathsf{Fam_c}(\mathcal{C})$ with countable coproducts.

An object of $\mathsf{Fam_c}(\mathcal{C})$ is a countable family of $\mathcal{C}$-objects $(A_i)_{i \in I}$, i.e.

- a countable set I
- for each $i \in I$, a $\mathcal{C}$-object A_i.

The homset from $(A_i)_{i \in I}$ to $(B_j)_{j \in J}$ is

$$\prod_{i \in I} \sum_{j \in J} \mathcal{C}(A_i, B_j)$$

Functors from $\mathsf{Fam_c}(\mathcal{C})$

The fully faithful functor $\mathcal{C} \longrightarrow \mathsf{Fam_c}(\mathcal{C})$ sends A to $(A)_{i\in 1}$.

It is universal: every functor $F : \mathcal{C} \longrightarrow \mathcal{D}$, where $\mathcal{D}$ has countable coproducts, has an essentially unique extension

$$\begin{tikzcd} \mathcal{C} \arrow[r] \arrow[dr, "F"'] & \mathsf{Fam_c}(\mathcal{C}) \arrow[d, dotted, "F^{\Sigma}"] \arrow[dl, phantom, "\cong"] \\ & \mathcal{D} \end{tikzcd}$$

by setting $F^{\Sigma} : (A_i)_{i\in I} \mapsto \sum_{i\in I} FA_i$

The Category $\mathsf{opFam}_{\mathsf{c}}(\mathcal{C})$

The category $\mathsf{opFam}_{\mathsf{c}}(\mathcal{C}) \stackrel{\mathrm{def}}{=} (\mathsf{Fam}_{\mathsf{c}}(\mathcal{C}^{\mathsf{op}}))^{\mathsf{op}}$ has countable products.

An object is a family of $\mathcal{C}$-objects.

The homset from $(A_i)_{i \in I}$ to $(B_j)_{j \in J}$ is

$$\prod_{j \in J} \sum_{i \in I} \mathcal{C}(A_i, B_j)$$

Functors from $\mathsf{opFam_c}(\mathcal{C})$

The fully faithful functor $\mathcal{C} \longrightarrow \mathsf{opFam_c}(\mathcal{C})$ sends A to $(A)_{i \in 1}$.

It is universal: every functor $F : \mathcal{C} \longrightarrow \mathcal{D}$, where $\mathcal{D}$ has countable products, has an essentially unique extension

$$\begin{tikzcd} \mathcal{C} \arrow[r] \arrow[dr, "F"'] & \mathsf{opFam_c}(\mathcal{C}) \arrow[d, dotted, "F^{\prod}"] \arrow[dl, phantom, "\cong"] \\ & \mathcal{D} \end{tikzcd}$$

by setting $F^{\prod} : (A_i)_{i \in I} \mapsto \prod_{i \in I} FA_i$

Categorical Game

A categorical game consists of

- A category ActPos of active positions
- A category PassPos of passive positions
- Functors

$$\begin{aligned} \mathrm{Omove} : \mathrm{PassPos} &\longrightarrow \mathrm{opFam_c(ActPos)} \\ \mathrm{Pmove} : \mathrm{ActPos} &\longrightarrow \mathrm{Fam_c(PassPos)} \end{aligned}$$

If ActPos and PassPos are discrete, this is the same as before.

Example: innocent game for Calculus of No Return

A morphism of active positions $\Gamma^{\mathsf{P}} \longrightarrow \Delta^{\mathsf{P}}$ is a renaming.

A morphism of passive positions $\Gamma^{\mathsf{P}}, \Gamma^{\mathsf{O}} \longrightarrow \Delta^{\mathsf{P}}, \Delta^{\mathsf{O}}$ is

- a renaming $\Gamma^{\mathsf{P}} \longrightarrow \Delta^{\mathsf{P}}$
- and a renaming $\Delta^{\mathsf{O}} \longrightarrow \Gamma^{\mathsf{O}}$.

Transition System on a Categorical Game $\mathfrak{G}$

A transition system should be carried by a pair of presheaves

$$\begin{array}{rcl} X & \in & \mathbf{Set}^{\mathsf{PassPos}} \\ Y & \in & \mathbf{Set}^{\mathsf{ActPos}} \end{array}$$

Transition System on a Categorical Game $\mathfrak{G}$

A transition system should be carried by a pair of presheaves

$$\begin{array}{ccc} X & \in & \mathbf{Set}^{\mathrm{PassPos}} \\ Y & \in & \mathbf{Set}^{\mathrm{ActPos}} \end{array}$$

Endofunctor $H_{\mathfrak{G}}$ on $\mathbf{Set}^{\mathrm{PassPos}} \times \mathbf{Set}^{\mathrm{ActPos}}$

$H_{\mathfrak{G}}(X, Y)$ has passive part

$$\mathrm{PassPos} \xrightarrow{\mathrm{Omove}} \mathrm{opFam_c(ActPos)} \xrightarrow{Y^{prod}} \mathbf{Set}$$

and active part

$$\mathrm{ActPos} \xrightarrow{\mathrm{Pmove}} \mathrm{Fam_c(PassPos)} \xrightarrow{X^{\Sigma}} \mathbf{Set} \xrightarrow{H} \mathbf{Set}$$

An H-transition system on $\mathfrak{G}$ is an $H_{\mathfrak{G}}$-coalgebra.

We're adding structure, but not changing the semantics

Renaming Lemma

For a renaming $p \xrightarrow{\theta} p'$ and node $c \in X(p)$, the meaning of $X(\theta)c$ is obtained from that of x.

Bisimulation up to isomorphism

A relation $\mathcal{R}$ on nodes in the same position is a bisimulation up to isomorphism when

- if M, M' are $\mathcal{R}$-related active nodes, and M moves in a way that passes 3 functions, then M' makes the same move, and there is some isomorphism we can apply that makes the resulting passive nodes $\mathcal{R}$-related
- if $\overrightarrow{V}, \overrightarrow{V'}$ are $\mathcal{R}$-related passive nodes, then for any O-move that passes 5 functions, there is some isomorphism we can apply that makes the resulting active nodes $\mathcal{R}$-related.

This is a sound principle, i.e. any such relation is contained in bisimilarity.

Conclusions

Doing operational game semantics with discrete games is possible but necessitates renaming bureaucracy.

So let's work with categorical games instead.

The actual semantics is unaffected.

Converting strategies

We want to convert strategies on game $\mathfrak{G}$ into strategies on game $\mathfrak{H}$.
More precisely, we have a span

$$\begin{tikzcd} & R \arrow[dl, "\pi"'] \arrow[dr, "\pi'"] & \\ \mathrm{PassPos}(\mathfrak{G}) & & \mathrm{PassPos}(\mathfrak{H}) \end{tikzcd} \qquad \begin{tikzcd} & S \arrow[dl, "\pi"'] \arrow[dr, "\pi'"] & \\ \mathrm{ActPos}(\mathfrak{G}) & & \mathrm{ActPos}(\mathfrak{H}) \end{tikzcd}$$

For $r \in R$, we want to convert a strategy for πr into a strategy for $\pi' r$, where $r \in R$. Likewise for S.
This can be achieved using a transfer along the span.

Transfer along a span

Given $r \in R$ and an O-move on the right

$$m \in \mathrm{Omove}(\pi' r)$$

we want to "transfer" it to either an O-move on the left or a P-move on the right. That is, we give either

- $n \in \mathrm{Omove}(\pi r)$ together with $s \in S$ over $\pi r.n$ and $\pi' r.m$
- or $n \in \mathrm{Pmove}(\pi' r.n)$ together with $s \in S$ over πr and $\pi' r.m.n$

Similar requirement for $s \in S$ and P-move on the left.

The tensor game $\mathfrak{G} \otimes \mathfrak{H}$

- A passive position of $\mathfrak{G} \otimes \mathfrak{H}$ is a pair of passive positions.
- An active position is a pair of positions, where just one is active.
- In passive position (p, p'), Opponent chooses to play on the left or on the right.
- But in active position (p, p'), Proponent has to play in whichever side is currently active.

This is symmetric monoidal, wrt an obvious notion of isomorphism of games.

What We Want

- A virtual bicategory of games, in which a morphism is a transfer along a span.
- The definition of transfer should be relaxed to replace position equality by a morphism.
- Symmetric monoidal structure given by $\otimes$.
- Two "bangs" (not exactly comonads): a non-backtracking one (cf. Hyland) and a backtracking one.
- Each operation on games has an accompanying operation on strategies, related to transition systems by a suitable lemma.