A fully abstract games model of exceptions

Jim Laird

COGS, University of Sussex
1 Background: Exceptions

- Dynamically bound exceptions are *different* from statically bound exceptions — for instance, they cannot be macro-expressed using first-class continuations.
- Dynamically bound exceptions are a kind of “hybrid” of state and control effects — they can be *implemented* using continuations and references.
- *Local Declaration* of exceptions raises many of the same identity-related issues as local declaration of references.
2 Background: Game Semantics

By relaxing constraints on the original Hyland-Ong model of PCF, we can form fully abstract games models of sequential functional languages with features including:

- First-class continuations (call/cc or equivalent),
- Idealized-Algol-style ground-type references,
- ML-style general references.
A question: Where do exceptions fit into this “hierarchy” of games models, and what does this tell us about their expressive power?

General References

Idealized Algol

PCF → call/cc
3 IAx — Idealized Algol with Exceptions

Types:

\[ T ::= \text{comm} | \text{0} | \text{nat} | \text{var} | \text{exn} | T \Rightarrow T \]

Terms:

\[ M ::= x | \text{skip} | 0 | \text{succ} \; M | \text{pred} \; M | \text{IF} \; 0 \; M | | \lambda x. \; M | M \; M | \text{Y} \; M \]
\[ M; M | \text{new} \; M | M ::= M | !M | \text{new}\_\text{exn} \; M | \text{raise} \; M | \text{handle} \; M \; M \]

Typing judgements for exceptions:

\[
\begin{align*}
\Gamma \vdash M : \text{exn} &\Rightarrow T & \Gamma \vdash M \; \text{exn} &\Rightarrow \text{exn} & \Gamma \vdash N : \text{0} &\Rightarrow \text{0} \\
\Gamma \vdash \text{new}\_\text{exn} \; M : T & & \Gamma \vdash \text{raise} \; M \; \text{0} & & \Gamma \vdash \text{handle} \; M \; N : \text{comm}
\end{align*}
\]
Evaluation contexts:
\[ E[\cdot] ::= \cdot | E[\cdot] M | \text{succ} E[\cdot] | \text{pred} E[\cdot] | \text{IF0} E[\cdot] | Y E[\cdot] | \]
raise \(E[\cdot]) | \text{handle} E[\cdot] M | \text{handle h} E[\cdot] \]

Reduction rules for exceptions:
\[ E[\text{new\_exn} M], \mathcal{E} \rightarrow E[M h], \mathcal{E} \cup \{h\} \quad (h \notin \mathcal{E}) \]
\[ E[\text{handle h} E[h][\text{raise h}]] \rightarrow E[\text{skip}] \]

Observational equivalence: Say that \(M \preceq N\) if for all contexts \(C[\cdot] : \text{comm}\), whenever \(C[M] \rightarrow \text{skip}\) then \(C[N] \rightarrow \text{skip}\).
4 Games and control

In Hyland-Ong Games, locality of control flow is enforced by the \textit{bracketing condition}.

- A strategy is \textit{well-bracketed} if each \textit{answer} responds to the pending question, where:
  \begin{align*}
  \text{pending}(sa) &= a, \text{ if } a \text{ is a question}, \\
  \text{pending}(satb) &= \text{pending}(s), \text{ if } b \text{ is an answer to } a.
  \end{align*}

- Interactions between well-bracketed strategies form a nested sequence of parentheses:
  \[(\[\ldots]\[\ldots\])\ldots]\]

- Abandoning the bracketing condition completely yields a model of first-class continuations, equivalent to a cps model.
To model exceptions, we add *contingency pointers* to sequences, subject to the following constraints:

- Every Player move is contingent on some previous move,
- Player moves are contingent on Opponent moves and vice-versa,
- Answers are contingent on questions.

A strategy $\sigma$ is *weakly bracketed* if every Player move in $\sigma$ is contingent on an open question, where:

- $\text{open}(\varepsilon) = \{\}$,
- $\text{open}(sa) = \{a\}$, if $a$ is not contingent,
- if $b$ is contingent on $a$, then:
  - $\text{open}(satb) = \text{open}(s)$ if $\lambda^{QA}(b) = A$,
  - $\text{open}(satb) = \text{open}(sa) \cup \{b\}$, otherwise.
Figure 1: A violation of weak bracketing
5 Control Games

We form a category $\mathcal{CG}$ of control games which has arenas as objects and weakly-bracketed strategies on $A \Rightarrow B$ as morphisms from $A$ to $B$.

- A strategy $\sigma$ in $\mathcal{CG}$ is well-bracketed if every Player move in $\sigma$ is contingent on the pending question.
- The well-bracketed strategies form a subcategory of $\mathcal{CG}$ which is isomorphic to the category $\mathcal{G}$ of games without contingency pointers and well-bracketed strategies.
- Thus we have a semantics of Idealized Algol in $\mathcal{CG}$ with the property that every well-bracketed and finite strategy is definable as a term.
\[(0 \Rightarrow 0) \Rightarrow ([0] \Rightarrow [0])\]

Figure 2: A play of \(id_{[0]\Rightarrow[0]}\)
6 Semantics of Exceptions

Like references, exceptions are modelled as a product of “methods” (raising and handling).

- \([\text{exn}] = ([0] \Rightarrow [\text{comm}]) \times [0]\)
- \([\Gamma \vdash \text{handle} M N] = ([\Gamma \vdash M]; \pi_l, [\Gamma \vdash N]); \text{App}\)
- \([\Gamma \vdash \text{raise} M : B] = ([\Gamma \vdash M]; \pi_r\)
- \([\Gamma \vdash \text{new} \_\text{exn} M] = ([\Gamma \vdash M] \times \text{xcell}); \text{App}\).

Where \(\text{xcell} : ([0] \Rightarrow [\text{comm}]) \times [0]\) is a strategy which violates well-bracketing and uses contingency pointers in an essential way.
Figure 3: A typical play of xcell
7 Full abstraction

We prove that every finite, weakly-bracketed strategy over an $\text{exn}$-free type-object is definable as a term of $\text{IAx}$ using a factorization of weakly-bracketed strategies into well-bracketed strategies.

**Proposition 7.1** If $\sigma : A$ is a finite and weakly-bracketed strategy then there is some finite and well-bracketed strategy $\hat{\sigma} : [\text{exn}] \rightarrow A$ such that $x_{\text{cell}}; \hat{\sigma} = \sigma$.

Definability is used to prove full abstraction (unlike the original model of PCF, no “collapse under intrinsic preorder” is required.

**Theorem 7.2 (Full abstraction)** Let $M, N : T$ be $\text{IAx}$ terms of $\text{exn}$-free type. Then $M \preceq N$ if and only if $[M]_{C^g} \subseteq [N]_{C^g}$.
8 Contingency and expressiveness

**Proposition 8.1** There is a weakly bracketed strategy $\text{not}_\text{id} : [(0 \Rightarrow 0) \Rightarrow (0 \Rightarrow 0)]$ such that $|\text{not}_\text{id}| = |\text{id}[0 \Rightarrow 0]|$ but $\text{id}[0 \Rightarrow 0] \neq \text{not}_\text{id}$.

$$
([0] \Rightarrow [0]) \Rightarrow ([0] \Rightarrow [0])
$$

Figure 4: A typical play of $\text{not}_\text{id}$

$\text{not}_\text{id} = [\lambda f. \lambda x. \text{new}_\text{exn} \lambda h. (\text{handle} h (f \text{raise} h)); x]$
We can use the distinction between not_id and id to show that exceptions cannot be modelled in $G$, and hence cannot be expressed using continuations and references.

**Proposition 8.2** There is a strategy

$$\text{trunc} : \left[ (0 \Rightarrow 0) \Rightarrow (0 \Rightarrow 0) \Rightarrow (0 \Rightarrow 0) \Rightarrow 0 \Rightarrow 0 \right]$$

such that for all well-bracketed strategies $\sigma : \left[ (0 \Rightarrow 0) \Rightarrow 0 \Rightarrow 0 \right]$, $\sigma; \text{trunc} \subseteq \text{id}$, but not_id; trunc = not_id.

(I.e. $\text{trunc} = [\lambda g : T.\text{new} \lambda z.\lambda f.\lambda x.\lambda z := 0; ((g \ M_1) \ M_2)]$, where $M_1 = \lambda y.\text{IF0} !z \text{ then } (z := 1; (f \ y)) \text{ else } \Omega$ and $M_2 = \text{IF0} !z \text{ then } \Omega \text{ else } x$.)

**Corollary 8.3** There is no adequate model of IAx in $G$ which conservatively extends the semantics of IA.
9 Conclusions

- We can model call/cc and exception handling (and general references) in $\mathcal{G}$ by dropping the bracketing condition. However, this is not equivalent to a cps model.

- We can extract from this model examples of terms which are observationally equivalent without exceptions, and observationally distinguishable with exceptions.

- Enriching games with additional pointers which track the follow of control appears to be more generally useful, for example to model multiple threads of control.