Amb breaks well-pointedness, ground amb doesn’t

Paul Blain Levy

School of Computer Science, University of Birmingham, UK
Nondeterministic Operators

- We can extend a functional language with:
  - binary erratic choice \( M \) or \( N \)
  - countable erratic choice \( \text{choose } n \in \mathbb{N}. M_n \)
  - ambiguous choice \( M \text{ amb } N \)

- To evaluate \( M \text{ amb } N \), we run \( M \) and \( N \) on an arbitrary fair scheduler, and return whatever we get first.

- Thus \( M \text{ amb } N \) can diverge iff \( M \) and \( N \) can both diverge.
Small Language

A call-by-name language, with two ground types, and (unary) sum types.

Types

\[ A ::= \text{bool} \mid 1 \mid LA \]

Terms

\[ M ::= x \mid \text{rec } x.\ M \mid M \text{ or } M \mid M \text{ amb } M \mid \text{true} \mid \text{false} \mid \text{if } M \text{ then } M \text{ else } M \mid \text{top} \mid M;M \mid \text{up } M \mid \text{pm } M \text{ as up } x.\ M \]
Operational Semantics

Terminal Terms

\[ T ::= \text{true} \mid \text{false} \mid \text{top} \mid \text{up } M \]

Remember: in a call-by-name sum type, we don’t evaluate under the constructor.

\( M \downarrow T \) is defined inductively.

\( M \uparrow \) is defined coinductively.
(Crude) Meaning Of A Type

For each type $B$, we define the set $[B]$ by induction on $B$:

\[
[\text{bool}] = \mathcal{P}(\{\text{true, false, } \perp\}) \\
[1] = \mathcal{P}(\{\top, \perp\}) \\
[LA] = \mathcal{P}([A]_\perp)
\]

Could restrict to nonempty sets—doesn’t matter.

For each closed term $M : B$, we define the operational meaning $[M] \in [B]$, by induction on $B$.

E.g. $[M] \overset{\text{def}}{=} \{\text{up } [N] \mid M \Downarrow \text{up } N\} \cup \{\perp \mid M \Uparrow\}$ for $B = LA$. 

Amb breaks well-pointedness, ground amb doesn’t – p. 5/21
Big Question

- Programs $\vdash M, M' : \texttt{bool}$ are behaviourally equivalent when $[M] = [M']$.

- We would like a denotational semantics such that for programs $\vdash M, M' : \texttt{bool}$, we have

  $$[M] = [M']$$

  iff $M$ and $M'$ are behaviourally equivalent.

- Is this possible?
A semantics is divergence-least when
- terms denote element of a poset
- all constructs are monotone
- \( \text{diverge} \overset{\text{def}}{=} \text{rec } x. \ x \) denotes least element \( \perp \).

This is the case if \( \text{rec} \) denotes least prefixed point.

Example: domain semantics
What goes wrong with divergence-least

(folklore, also Lassen, Levy, Panangaden, APPSEM 2005)

- On the one hand
  \[ \text{true or diverge} \leq \text{true or true} = \text{true} \]

- On the other hand, monotonicity of \( \text{amb} \) gives
  \[
  \text{true} = \text{if (false amb diverge) then diverge else true}
  \leq \text{if (false amb true) then diverge else true}
  = \text{true or diverge}
  \]

- So \( \text{true or diverge} = \text{true} \)

- Each powerdomain theory either gives this equation, or makes \( \text{amb} \) non-monotone.

Amb breaks well-pointedness, ground amb doesn't – p. 8/21
Well-pointed semantics is one where a term in context $\Gamma$ denotes a function from a set of environments.

E.g. a term $x : L1 \vdash M : L1$ should denote a function from $[L1]$ to itself.

And $[\text{rec } x. M]$ should be a fixpoint of this function.

We need some way of computing this fixpoint.
Operational Question 1 (Lassen 1998)

- Two closed terms $\vdash M, M : A$ are convex bisimilar when $[M] = [M']$.  
  This is robust (preserved by every context).

- Two terms $\Gamma \vdash M, M : A$ are convex applicatively bisimilar when $M[N/x]$ and $M'[N/x]$ are convex bisimilar for every $\Gamma$-environment $N/x$.  
  Is this a congruence?

- Without amb, the answer is yes.
Operational Question 2 (Lassen 1999)

- Two terms $\Gamma \vdash M, M' : A$ are contextually equivalent when $CM$ and $CM'$ are behaviourally equivalent for every ground context $C[\cdot]$.

- Two $\Gamma \vdash M, M' : A$ terms are Closed Instantiation equivalent when $M[N/x]$ and $M'[N/x]$ are contextually equivalent for every $\Gamma$-environment $\overrightarrow{N/x}$.

- The context lemma says that CI equivalence implies contextual equivalence. This is true without amb.

- Is it true in the presence of amb?
Both of these questions has a variant where we use inclusion of behaviour sets rather than equality.

This makes \( \text{amb} \) monotone.
Divergence Ordering

Alternatively, we can use equality for convergence, but inclusion for divergence.

\[
\begin{align*}
\{d\} & \quad \{t, d\} & \quad \{f, d\} & \quad \{t, f, d\} \\
\{\} & \quad \{t\} & \quad \{f\} & \quad \{t, f\}
\end{align*}
\]

This makes \texttt{amb} monotone.
Breaking well-pointedness

- We will exhibit two terms $x : L_1 \vdash M, M'' : L_1$ and a context $C[\cdot] : 1$ such that
  - $C[M] \nleq$ and $C[M''] \uhr$
  - but $M$ and $M''$ represent the same selfmap $f$ on $[L_1]$.
  - This refutes all 6 operational conjectures, and shows the impossibility of a well-pointed denotational semantics.

- The terms $\text{rec } x. M$ and $\text{rec } x. M''$ represent different fixpoints of $f$. 
The Terms

\[ M \overset{\text{def}}{=} (\text{up top}) \text{ amb } (\text{pm } x \text{ as } \text{up } z. \text{up } (\text{top or } z)) \]

\[ M' \overset{\text{def}}{=} \text{up } (\text{top or } \text{pm } (x \text{ amb } \text{up } \text{top}) \text{ as } \text{up } y.y) \]

\[ M'' \overset{\text{def}}{=} M \text{ or } M' \]

Consider \( M[N/x] \) and \( M''[N/x] \).

- Neither may diverge.
- Both may return \( \text{up } P \), where \( P \downarrow \text{top} \) and \( P \uparrow \).
- Neither may return \( \text{up } P \), where \( P \not\downarrow \text{top} \).
- If \( N \downarrow \text{up } Q \), where \( Q \uparrow \), then both may return \( \text{up } P \), where \( P \downarrow \text{top} \) and \( P \uparrow \).
- Otherwise, neither may.
The distinguishing context

\[
M \quad \overset{\text{def}}{=} \quad (\text{up top}) \ \text{amb} \ (\text{pm } x \ \text{as up } z. \ \text{up} \ (\text{top or } z)) \\
M' \quad \overset{\text{def}}{=} \quad \text{up} \ (\text{top or pm} \ (\text{up top amb } x) \ \text{as up } y. \ y) \\
M'' \quad \overset{\text{def}}{=} \quad M \ \text{or} \ M' \\
C[\cdot] \quad \overset{\text{def}}{=} \quad \text{pm} \ (\text{up top amb} \ (\text{rec } x.[\cdot])) \ \text{as up } y. \ y
\]

\(C[M'']\) may diverge: just keep choosing to go right, using

\[\text{rec } x. M'' \downarrow \text{up} \ (\text{top or } C[M''])\]

\(C[M]\) cannot diverge because if \(\text{rec } x. M \downarrow \text{up} \ N\) then \(N = (\text{top or } )^n \text{ top}\), which cannot diverge.
That Big Question

We would like a denotational semantics such that for programs $\vdash M, M' : \text{bool}$, we have

$$[[M]] = [[M']] \iff [M] = [M']$$

Is this possible?

Still open.
General Amb vs Ground Amb

- Our example uses \texttt{amb} at type \( L_1 \), not just at ground type.

- All 6 conjectures are \texttt{true} if we restrict to ground \texttt{amb}.

- The proofs are mild adaptations of the proofs without \texttt{amb}.

- These results can be extended to a full type system with recursive types.

- It can be call-by-name, call-by-value or call-by-push-value.

Cf. O’Hearn’s monad for ground storage, Laird’s semantics of ground control.
Uses

- A **use** is a special kind of ground context.
  - A **use** for `bool` is `if [·] then N else N'`
  - A **use** for `1` is `[·]; N`
  - A **use** for `LA` is `pm [·] as up x. N`.

- Closed terms $M, M' : A$ are **Uses** equivalent when they are behaviourally equivalent under every use.

- The **Uses** theorem says that **Uses** equivalence implies contextual equivalence.

- Again 2 variants using inclusion.

- Context lemma + **Uses** theorem = CIU theorem
We define two terms $\vdash M, M' : L1$ and a context $C[\cdot] : 1$ such that

- $M$ and $M'$ are Uses equivalent
- $C[M] \not\uparrow$ but $C[M'] \uparrow$

$$M \overset{\text{def}}{=} \text{diverge or up top}$$

$$M' \overset{\text{def}}{=} M \text{ or up } (\text{top or diverge})$$

$$C[\cdot] \overset{\text{def}}{=} \text{pm } ([\cdot] \text{ amb up top}) \text{ as } x. \ x$$

With ground amb, the CIU theorem holds.
Conclusion (denotational slant)

- $\text{amb}$ cannot have a well-pointed denotational semantics.
- ground $\text{amb}$ might have.