A Relative-Time Semantics of Orc

Ian Wehrman, David Kitchin, William R. Cook, Jayadev Misra
The University of Texas at Austin

Task Orchestration

- Get data from remote services, perform computation, call other services with results
  - E.g. web scripting, workflow
- Orc is a language for orchestration
  - Services invoked by **calling sites**
  - Processes **publish** values to communicate

Orc is Simple

- Site calls: $M(v)$
- Combinators:
  1. Symmetric: $f \mid g$
  2. Sequential: $f >x> g$
  3. Asymmetric: $f <x< g$
- Definitions: $D(x) =_{\text{df}} g$

Site Calls

- Perform a computation, publish at most one result
- Remote services: $BBC(today)$
  - publishes $news$ (maybe)
- Local services: $add(5,3)$
  - publishes 8

Symmetric Combinator

- To evaluate $f \mid g$, evaluate f and g in parallel
  - $CNN(today) \mid BBC(today)$
- $f \mid g$ publishes v iff f or g publishes v

Sequential Combinator

- To evaluate f >x> g
  1. Begin by evaluating f
  2. For each v published by f, evaluate $[v/x]g$ in parallel
  3. f >x>g publishes w iff some $[v/x]g$ publishes w

Sequential Combinator

- f communicates with g by publishing
  - f >x> g binds x in g
- Many copies of g may evaluate in parallel
  - $(CNN \mid BBC)$ >x> $email(ian,x)$

Asymmetric Combinator

- To evaluate f <x< g
  1. Begin evaluating f and g in parallel
  2. f may block waiting for data from g
  3. if g publishes v, kill g and continue evaluating $[v/x]f$

Asymmetric Combinator

- g communicates with f by publishing
  - f <x< g binds x in f
- First value published by g is passed to f
  - $email(ian, x)$ <x< $(CNN \mid BBC)$

Definitions, etc.

- $Clk(x) =_{df} let(x) \mid (Rtimer(1) >> Clk(x+1))$
- $Clk(0)$ publishes $0, 1, 2, \ldots$ at unit intervals
  - $f >> g$ abbrev $f >x> g$ for x not free in g
  - $let(x)$ publishes x immediately
  - $Rtimer(t)$ publishes after waiting t units

More Examples

- $\mathit{fork\text{-}join} =_{\mathrm{df}} (\mathit{let}(x,y) <x< M) <y< N$
- $\mathit{sync} =_{\mathrm{df}} \mathit{fork\text{-}join} >x> (f \mid g)$
- $\mathit{delay} =_{\mathrm{df}} (\mathit{Rtimer}(1) >> \mathit{let}(x)) <x< M$
- $\mathit{priority} =_{\mathrm{df}} \mathit{let}(x) <x< (N \mid \mathit{delay})$

Semantics

- Previous work on asynchronous, synchronous-but-untimed semantics
- Today: **relative-time** semantics
- Describes delays from site calls, e.g. $Rtimer$.
  - $(Rtimer(2)$ >> $let(v))$ | $(Rtimer(3)$ >> $let(w))$

Operational Semantics

- Based on labeled transition systems
- Labels are time-event pairs
  - events indicate publication, site calls, etc.
  - time indicates when event occurs relative to start of evaluation.

Operational Semantics

$$f \xrightarrow{t,a} f'$$

- Expression f **may** engage in event a after t units of time, without engaging in other events, resulting in expression f**'**.

Sites

$$let(v) \xrightarrow{0,!v} \mathbf{0}$$

- Immediately publish value v
- Transition to an expression $\mathbf{0}$ that engages in no other events

Sites

$$Rtimer(t) \xrightarrow{t,!} \mathbf{0}$$

- Publish a signal after t time units

Combinators

$$\frac{f \xrightarrow{t,a} f'}{f \mid g \xrightarrow{t,a} f' \mid g}$$

- OK in asynchronous semantics, but **not** with time
  - $Rtimer(7) \mid Rtimer(2)$

Time Shifting

$$f^t$$

- Evaluate for t time units without an event
  - $Rtimer(5)^3 \approx Rtimer(2)$
- May not be possible:
  - $Rtimer(5)^7 \approx \bot$

Combinators

$$\frac{f \xrightarrow{t,a} f'}{f \mid g \xrightarrow{t,a} f' \mid g^t}$$

- Only if g^t is not $\bot$
- Rules for other combinators similarly extend the asynchronous semantics

Operational Semantics

$$\frac{[E(x) \;\underline{\Delta}\; f] \in \mathcal{D}}{E(p) \overset{0,\tau}{\to} [p/x].f} \quad (\textsc{Def})$$

$$\frac{f \overset{t,a}{\to} f' \qquad a \neq !m}{f >x> g \overset{t,a}{\to} f' >x> g} \quad (\textsc{Seq1N})$$

$$\frac{k \in \Sigma(M,m)}{M(m) \overset{0,\tau}{\to} ?k} \quad (\textsc{Call})$$

$$\frac{f \overset{t,!m}{\to} f'}{f >x> g \overset{t,\tau}{\to} (f' >x> g) \mid [m/x].g} \quad (\textsc{Seq1V})$$

$$\frac{(t,m) \in k}{?k \overset{t,!m}{\to} \mathbf{0}} \quad (\textsc{Return})$$

$$\frac{f \overset{t,a}{\to} f'}{f <x< g \overset{t,a}{\to} f' <x< g^t} \quad (\textsc{Asym1})$$

$$\frac{f \overset{t,a}{\to} f'}{f \mid g \overset{t,a}{\to} f' \mid g^t} \quad (\textsc{Sym1})$$

$$\frac{g \overset{t,!m}{\to} g'}{f <x< g \overset{t,\tau}{\to} [m/x].f^t} \quad (\textsc{Asym2V})$$

$$\frac{g \overset{t,a}{\to} g'}{f \mid g \overset{t,a}{\to} f^t \mid g'} \quad (\textsc{Sym2})$$

$$\frac{g \overset{t,a}{\to} g' \qquad a \neq !m}{f <x< g \overset{t,a}{\to} f^t <x< g'} \quad (\textsc{Asym2N})$$

Traces

- **Execution**: finite sequence of time-event pairs that f engages in
- **Trace**: execution without internal events
- $<f>$ = traces of f defined operationally

Denotational Semantics

- Trace sets form a **denotation**
- Denotation of f is $\mu(f)$
  - limit of a sequence of trace sets $\mu_0(f)$, $\mu_1(f)$,...
  - $\mu_i(f)$ defined recursively on f

Equivalence

- Operational and denotational semantics are equivalent.
- Theorem: $<f> = \mu(f)$.
- Allows for **compositional** reasoning.

Logical Semantics

in progress

- Specification language: FOL w/ predicate at.
- $at(f,t,b)$: f engages in event b at time t.
- Sound axioms + SMT solver = validity

Future Work

- Mutable state.

Thank You

- Questions?

Remote Sites

$$\frac{k \in \Sigma(M, m)}{M(m) \overset{0, M_k(m)}{\longrightarrow} ?k}$$

- Immediately transition to a **handle**
- Describes possible outcomes of a call

Handles

- Sets that may include:
  - time-value pairs – indicates that value v is published at time t
  - an element ω – indicates non-response

Handles

$$\frac{(t, v) \in k}{?k \xrightarrow{t, !v} \mathbf{0}}$$

- Expression may publish v at time t
- Note: no corresponding rule for ω