Abstractness of Continuation Semantics for Asynchronous Concurrency

Gabriel Ciobanu, Eneia Nicolae Todoran

"A.I. Cuza" University, Iasi, Technical University, Cluj-Napoca, Romania

Federated Logic Conference 2018
Workshop Domains

Oxford, UK

July 7-8, 2018

Motivation and Aim

- A continuation is a semantic representation of the rest a computation [Stratchey and Wadsworth 1974]
- Traditional continuations can express: non-local exits, coroutines, even multitasking and ADA-like rendez-vous
  - However, the traditional continuations do not work well enough in the presence of concurrency [Hieb, Dybvig and Anderson 1994]

Motivation and Aim

- In [Todoran 2000, Ciobanu and Todoran 2014] we introduced a continuation semantics for concurrency (CSC)
- CSC can express concurrent composition as well as various communication and synchronization mechanisms
  - Intuitively, CSC is a denotational scheduler
- In the CSC approach continuations are application-specific structures of computations
  - Rather than the functions to some answer used in the classic technique of continuations

Motivation and Aim

- In this talk we survey some applications of CSC and we investigate the abstractness of CSC
- We present an optimality criterion specific of continuation semantics that we name weak abstractness which
  - Relaxes the completeness condition
  - Preserves the correctness condition of the classic full abstractness criterion [Milner 1977]

Continuation Semantics for Traditional Concurrent Programming Concepts

CSP-like synchronous communication and asynchronous communication [Todoran 2000]	Warren's Andorra Model [Todoran and Papaspyrou 2000]
CSP Extended with Multiple Channels Communication [Ciobanu and Todoran 2015]	

CSP Extended with Multiple Channels Communication [Ciobanu and Todoran 2015]

$$j ::= c?v \mid j\&j$$
$$a ::= v := e \mid c!e \mid j$$
$$s ::= a \mid y \mid s;s \mid s+s \mid s \parallel s$$

$$c_1!e_1 \parallel \cdots \parallel c_n!e_n \parallel (c_1?v_1\& \cdots c_n?v_n)$$
$$\equiv (v_1, \ldots, v_n := e_1, \ldots, e_n)$$

Warren's Andorra Model [Todoran and Papaspyrou 2000]

$$p ::= (y = x;)^* x$$
$$x ::= g \mid \ll o \gg \mid \langle l \rangle \mid \#\langle l \rangle$$
$$\mid y \mid x \parallel x$$
$$g ::= a \mid \text{fail}$$
$$l ::= \epsilon \mid g?x\ (+g?x)^*$$
$$o ::= \epsilon \mid g : x\ (+g : x)^*$$

$$\ll o \gg \parallel a \parallel \langle a_1?x_1 + a_2?x_2 \rangle \parallel \langle l_1 \rangle \parallel \cdots \parallel \langle l_n \rangle$$

Continuation Passing Semantics for Nature Inspired Formalisms

Membrane Computing
[Ciobanu and Todoran 2017]

$\rho = (D; x), \quad x = o_1 \parallel o_4$

$$
\begin{aligned}
D = \; & \text{membrane } M_0 \, \{ \\
& \quad [o_1, o_4] \Rightarrow o_2 \parallel o_4; \\
& \quad [o_2] \Rightarrow o_3 \parallel \text{new}(M_1, l_1, o_1 \parallel o_5); \\
& \quad [o_2] \Rightarrow o_4; \\
& \quad [o_3] \Rightarrow \text{in}(l_1, o_5) \parallel o_5; \\
& \quad [o_4, o_4, o_5] \Rightarrow o_5; \; \}; \\
& \} \\
& \text{membrane } M_1 \, \{ \\
& \quad [o_1] \Rightarrow o_4 \parallel \text{out}(o_4); \\
& \quad [o_5, o_5] \Rightarrow \delta \; \}. \\
& \}
\end{aligned}
$$

Spiking Neural P-Systems
[Ciobanu and Todoran 2018]

$\rho' = (D', x'),$

$x' = \text{send}(\langle a^{2k-1} \rangle, \{N_1\}) \parallel \text{send}(a, \{N_3\})$

$$
\begin{aligned}
D' = \; & \text{neuron } N_0 \, \{ \, r_\epsilon \mid \{N_1, N_2, N_3\} \, \} \\
& \text{neuron } N_1 \, \{ \, a^+/[a] \rightarrow a; 2 \mid \{N_2\} \, \} \\
& \text{neuron } N_2 \, \{ \, [a^k] \rightarrow a; 1 \mid \{N_3\} \, \} \\
& \text{neuron } N_3 \, \{ \, [a] \rightarrow a; 0 \mid \{N_0\} \, \}
\end{aligned}
$$

On the Abstractness of Continuation Semantics

- The full abstractness condition is in general difficult to establish [Milner 1977]
  - Even more difficult in continuation semantics
- We are not aware of any full abstractness result for a concurrent language designed with continuations
- Continuation-passing semantics for sequential languages are not fully abstract [Cartwright, Curien & Felleisen 1994]

On the Abstractness of Continuation Semantics

- Weak abstractness may be useful when full abstractness is difficult (or impossible) to achieve
- We offer a denotational semantics $[\![\cdot]\!]$ for an asynchronous formalism; we use a domain of continuations

$$\mathbf{D} = \mathbf{Cont} \to \mathbf{P} \qquad\qquad \mathbf{Cont} = \cdots \mathbf{D} \cdots$$

- The semantics is designed by using metric domains [De Bakker and Zucker 1982, America and Rutten 1989]
  - Like the classic domains [Scott 1976, Scott 1982], metric spaces can also be used to express denotational semantics
  - We prove that $[\![\cdot]\!]$ is weakly abstract w.r.t. an $\mathcal{O}[\![\cdot]\!]$

Classic Full Abstractness [Milner 1977]

- A denotational semantics $\mathcal{D} : L \rightarrow \mathbf{D}$ is said to be fully abstract with respect to a (corresponding) operational semantics $\mathcal{O} : L \rightarrow \mathbf{O}$ if
  - $\mathcal{D}$ it is correct with respect to $\mathcal{O}$
    $$\forall x_1, x_2 \in L[\mathcal{D}(x_1) = \mathcal{D}(x_2) \;\Rightarrow\; \forall S[\mathcal{O}(S(x_1)) = \mathcal{O}(S(x_2))]]$$
  - $\mathcal{D}$ and complete with respect to $\mathcal{O}$
    $$\forall x_1, x_2 \in L[\mathcal{D}(x_1) \neq \mathcal{D}(x_2) \;\Rightarrow\; \exists S[\mathcal{O}(S(x_1)) \neq \mathcal{O}(S(x_2))]]$$

(S is an L syntactic context)

Abstractness of Continuation Semantics

- In continuation semantics, $\mathcal{D} : L \to \mathbf{D}$, $\mathbf{D} = \mathbf{Cont} \to \mathbf{F}$, the completeness condition (of full abstractness) is:

$$\forall x_1, x_2 \in L[\,(\exists\gamma \in \mathbf{Cont}[\mathcal{D}(x_1)\gamma \neq \mathcal{D}(x_2)\gamma]) \Rightarrow (\exists S[\mathcal{O}(S(x_1)) \neq \mathcal{O}(S(x_2))])\,]$$

- In continuation semantics for concurrency (CSC) [Todoran 2000, Ciobanu and Todoran 2014] we cannot prove this condition for the whole domain of continuations
  - A weaker condition can be established if we restrict the investigation to the class of denotable continuations
    - Which is closed under arbitrary computations
  - This condition we call weak completeness

The Class of Metric Domains for CSC

- A metric denotational domain of CSC is given as the unique solution of an equation [America and Rutten 1989]

$$\mathbf{D} \cong \mathbf{\Gamma} \stackrel{1}{\to} \mathbf{F}$$

- The class $(\mathbb{\Gamma} \in)\mathcal{DCONT}$ of domains for CSC is:

$$\mathbf{\Gamma} ::= \tfrac{1}{2} \cdot \mathbf{D} \mid \mathbf{M} \to \mathbf{\Gamma} \mid \mathbf{M} \times \mathbf{\Gamma} \mid \mathbf{M} + \mathbf{\Gamma} \mid \mathbf{\Gamma} \times \mathbf{\Gamma} \mid \mathbf{\Gamma} + \mathbf{\Gamma}$$

($\mathbf{M}$ is an arbitrary set $(m \in)M$ endowed with the discrete metric)

Resumptions and Denotable Continuations

- In the CSC approach a continuation is a structured configuration of (partially evaluated) denotations
- Following [De Bakker and De Vink 1996] we use the term resumption as an operational counterpart of the term continuation
- Let $(x \in)X$ be a fixed set. We define a class $(R \in)\mathcal{RES}(X)$ of sets of resumptions for X by:

$$R ::= X \mid M \to R \mid M \times R \mid M + R \mid R \times R \mid R + R$$

(here $(m \in)M$ is an arbitrary set)

Resumptions and Denotable Continuations

- For any language $(x \in)L$ and CSC domain $\mathbf{\Gamma}$ we can construct a corresponding set of resumptions by a homomorphism $res^L(\cdot) : \mathcal{DCONT} \to \mathcal{RES}(L)$

$$
\begin{aligned}
&res^L(\tfrac{1}{2} \cdot \mathbf{D}) = L \\
&res^L(\mathbf{M} \to \mathbf{\Gamma}) = M \to res^L(\mathbf{\Gamma}), \\
&res^L(\mathbf{M} \times \mathbf{\Gamma}) = M \times res^L(\mathbf{\Gamma}) \\
&res^L(\mathbf{M} + \mathbf{\Gamma}) = M + res^L(\mathbf{\Gamma}) \\
&res^L(\mathbf{\Gamma}_1 \times \mathbf{\Gamma}_2) = res^L(\mathbf{\Gamma}_1) \times res^L(\mathbf{\Gamma}_2) \\
&res^L(\mathbf{\Gamma}_1 + \mathbf{\Gamma}_2) = res^L(\mathbf{\Gamma}_1) + res^L(\mathbf{\Gamma}_2)
\end{aligned}
$$

- $res^L(\cdot)$ maps a complete metric space to a plain set
  - $\mathbf{M}$ is a metric space, given by an arbitrary set $(m \in)M$ endowed with the discrete metric

The Class of Denotable Continuations

- Let $(x \in) L$ be a language. Let $\mathcal{D} : L \to \mathbf{D}$, $\mathbf{D} \cong \mathbf{\Gamma} \xrightarrow{1} \mathbf{F}$, be a denotational semantics of L designed with CSC
- $(\gamma \in) \mathbf{\Gamma}^{Cls} = \{ [\![c]\!]^L_{\mathbf{\Gamma}} \mid c \in res^L(\mathbf{\Gamma}) \}$ is the class of denotable continuations for $\mathcal{D}$, where for any $\mathbf{\Gamma} \in \mathcal{DCONT}$, $[\![\cdot]\!]^L_{\mathbf{\Gamma}} : res^L(\mathbf{\Gamma}) \to \mathbf{\Gamma}$ is given by

$$
\begin{array}{rcl}
[\![x]\!]^L_{\frac{1}{2} \cdot \mathbf{D}} & = & \mathcal{D}(x) \\
[\![\lambda m \,.\, c]\!]^L_{\mathbf{M} \to \mathbf{\Gamma}} & = & \lambda m \,.\, [\![c]\!]^L_{\mathbf{\Gamma}} \\
[\![(m, c)]\!]^L_{\mathbf{M} \times \mathbf{\Gamma}} & = & (m, [\![c]\!]^L_{\mathbf{\Gamma}}) \\
[\![(i, c)]\!]^L_{\mathbf{M} + \mathbf{\Gamma}_2} & = & \begin{cases} (1, c) & \text{if} \quad i = 1 \\ (2, [\![c]\!]^L_{\mathbf{\Gamma}_2}) & \text{if} \quad i = 2 \end{cases} \\
[\![(c_1, c_2)]\!]^L_{\mathbf{\Gamma}_1 \times \mathbf{\Gamma}_2} & = & ([\![c_1]\!]^L_{\mathbf{\Gamma}_1}, [\![c_2]\!]^L_{\mathbf{\Gamma}_2}) \\
[\![(i, c)]\!]^L_{\mathbf{\Gamma}_1 + \mathbf{\Gamma}_2} & = & \begin{cases} (1, [\![c]\!]^L_{\mathbf{\Gamma}_1}) & \text{if} \quad i = 1 \\ (2, [\![c]\!]^L_{\mathbf{\Gamma}_2}) & \text{if} \quad i = 2 \end{cases}
\end{array}
$$

The (Metric) Domain of Denotable Continuations

- Let $(\gamma \in)\Gamma^{Dom} = co(\Gamma^{Cls}|\mathbf{\Gamma})$ be the metric completion of Γ^{Cls} with respect to $\mathbf{\Gamma}$ constructed as in the following Remark.
  - We call Γ^{Dom} the domain of denotable continuations for $\mathcal{D}$.

Remark

Let (M, d) be a complete metric space and let X be a subset of M, $X \subseteq M$. We use the notation $co(X|M)$ to represent the set

$$co(X|M) \stackrel{not.}{=} \{x \mid x \in M, x = \lim_i x_i, \forall i \in \mathbb{N} : x_i \in X, \\ (x_i)_i \text{ is a Cauchy sequence in } X\}$$

where the limits are taken with respect to d (as (M, d) is complete $\lim_i x_i \in M$). If we endow $co(X|M)$ with $d_{co(X|M)} = d\restriction co(X|M)$ and X with $d_X = d\restriction X$, then $(co(X|M), d_{co(X|M)})$ is a metric completion of (X, d_X). Recall that each metric space has a completion which is unique up to isometry

The (Metric) Domain of Denotable Continuations

- We have
  - $\Gamma^{Cls} \lhd \Gamma^{Dom}$, and (by construction) $\Gamma^{Dom} \lhd \Gamma$, but
  - in general, $\Gamma^{Dom} \neq \Gamma$

$$((M, d) \lhd (M', d'), \text{ whenever } M \subseteq M' \text{ and } d' \restriction M = d)$$

- In general we do not know whether Γ^{Cls} itself is a complete metric space (where it is, $\Gamma^{Cls} = \Gamma^{Dom}$)

- For a simple asynchronous language, in [Ciobanu and Todoran 2012] it is constructed a continuation $\gamma_\epsilon \in \Gamma$ such that $d(\gamma, \gamma_\epsilon) \geq \frac{1}{2}$, for any $\gamma \in \Gamma^{Dom}$

Weak Abstractness for CSC

Definition

Let $(x \in)L$ be a language and let $\mathcal{D} : L \to \mathbf{D}$, $\mathbf{D} \cong \Gamma \xrightarrow{1} \mathbf{F}$, be a denotational semantics of L designed with CSC. Let also $\mathcal{O} : L \to \mathbf{O}$ be an operational semantics of L and S a typical element of the class of syntactic contexts for L.

(a) *$\mathcal{D}$ is correct with respect to $\mathcal{O}$ iff*

$$\forall x_1, x_2 \in L[\mathcal{D}(x_1) = \mathcal{D}(x_2) \Rightarrow \forall S[\mathcal{O}(S(x_1)) = \mathcal{O}(S(x_2))]]$$

(b) *Let $(\gamma \in)\Gamma^{Dom}$ be the domain of denotable continuations for $\mathcal{D}$. We say that $\mathcal{D}$ is weakly complete with respect to $\mathcal{O}$ iff*

$$\forall x_1, x_2 \in L[(\exists \gamma \in \Gamma^{Dom}[\mathcal{D}(x_1)\gamma \neq \mathcal{D}(x_2)\gamma]) \Rightarrow (\exists S[\mathcal{O}(S(x_1)) \neq \mathcal{O}(S(x_2))])]$$

(c) *We say that $\mathcal{D}$ is weakly abstract with respect to $\mathcal{O}$ iff $\mathcal{D}$ is correct and weakly complete with respect to $\mathcal{O}$.*

Weak Abstractness for CSC

- It suffices to verify the weak completeness property for the class of denotable continuations Γ^{Cls}
  - If weak completeness can be established for Γ^{Cls} then it holds for the whole domain of denotable continuations Γ^{Dom}

Lemma

Let $(x \in)L$ be a language and let $\mathcal{D} : L \to \mathbf{D}$, $\mathbf{D} \cong \Gamma \stackrel{1}{\to} \mathbf{F}$, be a denotational semantics of L designed with CSC. Let also $\mathcal{O} : L \to \mathbf{O}$ be an operational semantics of L and S a typical element of the class of syntactic contexts for L. $\mathcal{D}$ is weakly complete w.r.t. $\mathcal{O}$ iff

$$\forall x_1, x_2 \in L[(\exists \gamma \in \Gamma^{Cls}[\mathcal{D}(x_1)\gamma \neq \mathcal{D}(x_2)\gamma]) \Rightarrow (\exists S[\mathcal{O}(S(x_1)) \neq \mathcal{O}(S(x_2))])]$$

where Γ^{Cls} is the class of denotable continuations for $\mathcal{D}$.

The Language $\mathcal{L}$ - a Paradigm for Asynchronous Communication [Boer, Kok, Palamidessi, Rutten 1993]

Definition

(a) *(Statements)* $x(\in X) ::= a \mid y \mid x + x \mid x; x \mid x \parallel\!\!\lfloor x \mid x \parallel x$

(b) *(Guarded statements)* $g(\in G) ::= a \mid g + g \mid g; x \mid g \parallel\!\!\lfloor x \mid g \parallel g$

(c) *(Declarations, Programs)* $(D \in) Decl = Y \rightarrow G$, $(\rho \in)\mathcal{L} = Decl \times X$

- $(a \in) Act$ is a set of atomic actions ($\delta \in Act$)
- $(y \in) Y$ is a set of recursion variables
- $I : Act \rightarrow \Sigma \rightarrow (\{\uparrow\} \cup \Sigma)$ is an interpretation function
  - If $I(a)(\sigma) = \uparrow$ then a is suspended in σ $(I(\delta)(\sigma) = \uparrow, \forall \sigma \in \Sigma)$

- Instances of the paradigm: asynchronous CCS, asynchronous CSP [Jifeng, Josephs, Hoare 1990], concurrent constraint programming [Saraswat 1993], await statement [Owicki, Gries 1976]

Structure of Continuations (and Resumptions) for $\mathcal{L}$

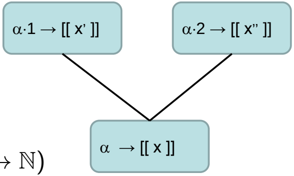
- $(\alpha \in) A = \{1, 2\}^*$ - identifiers (ϵ is the empty sequence)
- $\alpha \leq \alpha'$ iff $\alpha' = \alpha \cdot i_1 \cdots i_n$ $(n \geq 0)$ - partial order

Notation for partially ordered bags (multisets) of computations

$$\{\!|\mathbf{X}|\!\} \quad = \quad \mathcal{P}_{fin}(A) \times (A \to \mathbf{X})$$

- A continuation is a cactus stack (finite tree, active elements at leaves)
- $\nu : (A \times \{\!|\mathbf{X}|\!\}) \to Bool$ $\quad (\nu(\alpha, (\pi, \theta))$ iff α is a leaf in $(\{\alpha\} \cup \pi, \leq_{\{\alpha\} \cup \pi}))$

$$[\![(x' \parallel x''); x]\!] \quad \Rightarrow$$



- Cannot define multisets using $(\mathbf{X} \to \mathbb{N})$

Operational Semantics $\mathcal{O}[\![\cdot]\!]$

Semantic universe, configurations, (consistent) resumptions

$$(p \in)\mathbf{P} = \mathcal{P}_{nco}(\Sigma^* \cup \Sigma^* \cdot \{\delta\} \cup \Sigma^\omega)$$

$$Conf = (X \times CRes' \times \Sigma) \cup (KRes \times \Sigma)$$

$$CRes = A \times KRes, \quad (k \in)KRes = \{\!|X|\!\}, \; k_0 = (\emptyset, \lambda\alpha \,.\, \delta) \in KRes$$

$$CRes' = \{(\alpha, k) \mid \alpha \in A, k \in KRes, \nu(\alpha, k)\}$$

Operational semantics $\mathcal{O}[\![\cdot]\!] : X \to \Sigma \to \mathbf{P}$ $\quad (\mathcal{O} : Conf \to \mathbf{P})$

$$\mathcal{O}[\![x]\!](\sigma) = \mathcal{O}(x, (\alpha_0, k_0), \sigma)$$

$$\mathcal{O}(t) = \begin{cases} \{\Lambda\} & \text{if } t \text{ terminates} \\ \{\delta\} & \text{if } t \text{ blocks} \\ \bigcup\{\sigma \cdot \mathcal{O}(k, \sigma) \mid t \to (k, \sigma)\} & \text{otherwise} \end{cases}$$

Transition System Specification for $\mathcal{L}$

$$(a, (\alpha, k), \sigma) \rightarrow (k, \sigma') \qquad \text{if } I(a)(\sigma) = \sigma'$$

$$(y, (\alpha, k), \sigma) \nearrow (D(y), (\alpha, k), \sigma)$$

$$(x_1 + x_2, (\alpha, k), \sigma) \nearrow (x_1, (\alpha, k), \sigma)$$

$$(x_1 + x_2, (\alpha, k), \sigma) \nearrow (x_2, (\alpha, k), \sigma)$$

$$(x_1 ; x_2, (\alpha, k), \sigma) \nearrow (x_1, (\alpha \cdot 1, [k \mid \alpha \mapsto x_2]), \sigma)$$

$$(x_1 \lfloor\!\lfloor x_2, (\alpha, k), \sigma) \nearrow (x_1, (\alpha \cdot 1, [k \mid \alpha \cdot 2 \mapsto x_2]), \sigma)$$

$$(x_1 \parallel x_2, (\alpha, k), \sigma) \nearrow (x_1, (\alpha \cdot 1, [k \mid \alpha \cdot 2 \mapsto x_2]), \sigma)$$

$$(x_1 \parallel x_2, (\alpha, k), \sigma) \nearrow (x_2, (\alpha \cdot 1, [k \mid \alpha \cdot 2 \mapsto x_1]), \sigma)$$

$$(k, \sigma) \nearrow (k(\alpha), (\alpha, k \setminus \{\alpha\}), \sigma) \qquad \forall \alpha \in max(id(k))$$

CSC Domains and Evaluation Mechanism

CSC domains

$$\begin{array}{rcl} (\varphi \in)\mathbf{D} & \cong & \mathbf{Cont} \stackrel{1}{\to} \Sigma \to \mathbf{P} \\ (\gamma \in)\mathbf{Cont} & = & A \times \mathbf{Kont} \\ (\kappa \in)\mathbf{Kont} & = & \{\!|\, \frac{1}{2} \cdot \mathbf{D} \,|\!\} \end{array}$$

CSC evaluation mechanism

- **Cont** - open continuations (evaluation contexts)
- **Kont** - closed continuations (manipulated by the scheduler)
- ”Evaluate-normalize-schedule” loop

Auxiliary Operators

Nondeterministic choice ($+ : (\mathbf{P} \times \mathbf{P}) \to \mathbf{P}$)

$$p_1 + p_2 = \{q \mid q \in p_1 \cup p_2, q \neq \delta\} \cup \{\delta \mid \delta \in p_1 \cap p_2\}$$

Denotational scheduler

- $kc : \mathbf{Kont} \to \Sigma \to \mathbf{P}$

$$\begin{aligned} kc(\kappa)(\sigma) \;=\; & \text{if } (id(\kappa) = \emptyset) \text{ then } \{\Lambda\} \\ & \text{else } +_{\alpha \in max(id(\kappa))} \kappa(\alpha)(\alpha, \kappa \setminus \{\alpha\})(\sigma) \end{aligned}$$

$[\![x]\!](\alpha, \kappa)(\sigma) =_c p$ is an abbreviation for:

$$[\![x]\!](\alpha, \kappa)(\sigma) = \begin{cases} p & \text{if } \nu(\alpha, \kappa) \\ \{\delta\} & \text{otherwise} \end{cases}$$

Denotational Semantics $[\![\cdot]\!] : X \to \mathbf{D}$ $(\mathcal{D}[\![\cdot]\!] : X \to \Sigma \to \mathbf{P})$

$$
\begin{array}{rcl}
[\![a]\!](\alpha,\kappa)(\sigma) & =_c & \text{if } (I(a)(\sigma) = \uparrow) \text{ then } \{\delta\} \\
 & & \text{else } I(a)(\sigma)\cdot kc(\kappa)(I(a)(\sigma)) \\
[\![y]\!](\alpha,\kappa)(\sigma) & =_c & [\![D(y)]\!](\alpha,\kappa)(\sigma) \\
[\![x_1 + x_2]\!](\alpha,\kappa)(\sigma) & =_c & [\![x_1]\!](\alpha,\kappa)(\sigma) + [\![x_2]\!](\alpha,\kappa)(\sigma) \\
[\![x_1 ; x_2]\!](\alpha,\kappa)(\sigma) & =_c & [\![x_1]\!](\alpha\cdot 1, [\kappa \mid \alpha \mapsto [\![x_2]\!]])(\sigma) \\
[\![x_1 \mathbin{\|\!\!\lfloor} x_2]\!](\alpha,\kappa)(\sigma) & =_c & [\![x_1]\!](\alpha\cdot 1, [\kappa \mid \alpha\cdot 2 \mapsto [\![x_2]\!]])(\sigma) \\
[\![x_1 \parallel x_2]\!](\alpha,\kappa)(\sigma) & =_c & [\![x_1]\!](\alpha\cdot 1, [\kappa \mid \alpha\cdot 2 \mapsto [\![x_2]\!]])(\sigma) + \\
 & & [\![x_2]\!](\alpha\cdot 1, [\kappa \mid \alpha\cdot 2 \mapsto [\![x_1]\!]])(\sigma)
\end{array}
$$

$$
\mathcal{D}[\![x]\!] \quad = \quad [\![x]\!](\alpha_0,\kappa_0), \quad \alpha_0 = \epsilon, \kappa_0 = (\emptyset, \lambda\alpha.[\![\delta]\!])
$$

Isomorphic Resumptions

Definition

Two open resumptions $(\alpha_1, k_1), (\alpha_2, k_2) \in CRes$ *are isomorphic, written* $(\alpha_1, k_1) \cong (\alpha_2, k_2)$*, iff either (1) or (2) is satisfied:*

(1) $\neg\nu(\alpha_1, k_1)$ *and* $\neg\nu(\alpha_2, k_2)$ *(*(α_1, k_1) *and* (α_2, k_2) *are both inconsistent)*

(2) $\nu(\alpha_1, k_1)$ *and* $\nu(\alpha_2, k_2)$ *(*(α_1, k_1) *and* (α_2, k_2) *are both consistent) and there exists a bijection* $\mu : (\{\alpha_1\} \cup id(k_1)) \to (\{\alpha_2\} \cup id(k_2))$ *such that:*

  (i) $\mu(\alpha_1) = \alpha_2$

  (ii) $\mu(\alpha') \leq \mu(\alpha'') \Leftrightarrow \alpha' \leq \alpha'', \quad \forall \alpha', \alpha'' \in (\{\alpha_1\} \cup id(k_1))$

  (iii) $k_2(\mu(\alpha')) = k_1(\alpha'), \quad \forall \alpha' \in id(k_1)$

Class of Denotable Continuations

Definition

We define $[\![\cdot]\!] : KRes \to \mathbf{Kont}$, $[\![k]\!] = (id(k), \lambda\alpha.[\![k(\alpha)]\!])$. *The class of (open) denotable continuations for* $\mathcal{L}$ *is*

$$Cont^{Cls} = \{(\alpha, [\![k]\!]) \mid (\alpha, k) \in CRes\} = A \times Kont^{Cls}$$

where $(\kappa \in)Kont^{Cls} = \{[\![k]\!] \mid k \in KRes\}$.

Syntactic Contexts for $\mathcal{L}$

Definition

The class of syntactic contexts for $\mathcal{L}$ is given by:

$$S ::= (\cdot) \mid a \mid y \mid S; S \mid S + S \mid S \| S \mid S \lfloor\!\lfloor S$$

$S(x)$ is the result of substituting x for all occurrences of $(\cdot)$ in S.

Let $x, \overline{x} \in X$. When $[\![S(x)]\!](\alpha, [\![k]\!]) = [\![S(\overline{x})]\!](\overline{\alpha}, [\![\overline{k}]\!])$ $^{(*)}$ for for all contexts S and for all isomorphic resumptions $(\alpha, k) \cong (\overline{\alpha}, \overline{k})$ $(\in CRes)$, we write:

$$x \simeq \overline{x}$$

$^{(*)}$ implies $\mathcal{D}[\![S(x)]\!] = \mathcal{D}[\![S(\overline{x})]\!]$, for all contexts S

Concurrency Laws in Continuation Semantics [Ciobanu and Todoran 2014]

$$
\begin{array}{rcl}
x_1 + x_2 & \simeq & x_2 + x_1 \\
(x_1 + x_2) + x_3 & \simeq & x_1 + (x_2 + x_3) \\
x + x & \simeq & x \\
(x_1 + x_2); x3 & \simeq & x_1 ; x_3 + x_2 ; x_3 \\
x_1 ; (x_2 + x_3) & \simeq & x_1 ; x_2 + x_1 ; x_3 \\
x_1 ; (x_2 ; x_3) & \simeq & (x_1 ; x_2); x_3 \\
x + \delta & \simeq & x \\
\delta x & \simeq & \delta \\
x_1 \parallel x_2 & \simeq & x_1 \mathbin{\lfloor\!\lfloor} x_2 + x_2 \mathbin{\lfloor\!\lfloor} x_1 \\
a \mathbin{\lfloor\!\lfloor} x & \simeq & a; x \\
a; x_1 \mathbin{\lfloor\!\lfloor} x_2 & \simeq & a; (x_1 \parallel x_2) \\
(x_1 + x_2) \mathbin{\lfloor\!\lfloor} x_3 & \simeq & x_1 \mathbin{\lfloor\!\lfloor} x_3 + x_2 \mathbin{\lfloor\!\lfloor} x_3
\end{array}
$$

Concurrency Laws over the Domain of Denotable Continuations

- We do not know whether the class $Cont^{Cls}$ of denotable continuations is a complete metric space

- However, in CSC computations are (nonexpansive and thus) continuous functions, because

$$\mathbf{D} \cong \mathbf{Cont} \xrightarrow{1} \Sigma \to \mathbf{P}$$

- It is reasonable to study semantic properties in the metric completion of the class $Cont^{Cls}$, i.e., in the domain of denotable continuations

$$Cont^{Dom} = co(Cont^{Cls}|\mathbf{Cont})$$

Concurrency Laws over the Domain of Denotable Continuations

Definition

We say that two open denotable continuations $(\alpha, \kappa), (\overline{\alpha}, \overline{\kappa}) \in Cont^{Dom}$ are isomorphic, and we write $(\alpha, \kappa) \cong (\overline{\alpha}, \overline{\kappa})$, iff there exist sequences $(\alpha_i, k_i)_i, (\overline{\alpha}_i, \overline{k}_i)_i$, (with $(\alpha_i, k_i), (\overline{\alpha}_i, \overline{k}_i) \in CRes, \forall i \in \mathbb{N}$) such that:

- $(\alpha, \kappa) = \lim_i(\alpha_i, [\![k_i]\!]), (\overline{\alpha}, \overline{\kappa}) = \lim_i(\overline{\alpha}_i, [\![\overline{k}_i]\!])$, *and*
- $(\alpha_i, k_i) \cong (\overline{\alpha}_i, \overline{k}_i), \forall i \in \mathbb{N}$.

Concurrency laws hold for the whole domain $Cont^{Dom}$ of denotable continuations

Let $x, \overline{x} \in X$. When $[\![S(x)]\!](\alpha, \kappa) = [\![S(\overline{x})]\!](\overline{\alpha}, \overline{\kappa})$ for all $\mathcal{L}$ syntactic contexts S and for all isomorphic denotable continuations $(\alpha, \kappa) \cong (\overline{\alpha}, \overline{\kappa}) \in Cont^{Dom}$, we write:

$$x \sim \overline{x}$$

Proposition

$x \simeq \overline{x} \Rightarrow x \sim \overline{x}$, for all $x, \overline{x} \in X$.

Remark

*Not all continuations are denotable: $Cont^{Cls} \lhd Cont^{Dom}$ and $Cont^{Dom} \lhd$ **Cont**, but $Cont^{Dom} \neq$ **Cont** [Ciobanu and Todoran 2012]. As a consequence, $[\![\cdot]\!]$ is not fully abstract.*

Weak Abstractness for $\mathcal{L}$

- Recall that $\mathbf{D} \cong \mathbf{Cont} \xrightarrow{1} (\Sigma \to \mathbf{P})$, $\mathbf{Cont} = A \times \{\!|\, \frac{1}{2} \cdot \mathbf{D} \,|\!\}$.
- If we expand the notation $\{\!| \cdot |\!\}$ we get ($\mathbf{Cont} \in \mathcal{DCONT}$):

  $$\mathbf{Cont} = A \times (\mathcal{P}_{fin}(A) \times (A \to \tfrac{1}{2} \cdot \mathbf{D}))$$

  The corresponding set of resumptions is ($CRes \in \mathcal{RES}(X)$)

  $$CRes = res^X(\mathbf{Cont}) = A \times (\mathcal{P}_{fin}(A) \times (A \to X))$$

- The class of denotable continuations is

  $$\begin{aligned} Cont^{Cls} &= \{ [\![(\alpha, k)]\!]^X_{\mathbf{Cont}} \mid (\alpha, k) \in res^X(\mathbf{Cont}) \} \\ &= \{ (\alpha, [\![k]\!]) \mid (\alpha, k) \in CRes \}. \end{aligned}$$

Lemma

For any $x \in X$, $(\alpha, k) \in CRes$ there is an $\mathcal{L}$ syntactic context S such that: $[\![x]\!](\alpha, [\![k]\!]) = \mathcal{D}[\![S(x)]\!] = [\![S(x)]\!](\alpha_0, \kappa_0)$. Furthermore, S does not depend on x, it only depends on (α, k).

Weak Abstractness for $\mathcal{L}$

Theorem

The denotational semantics $[\![\cdot]\!]$ of $\mathcal{L}$ is weakly abstract with respect to the operational semantics $\mathcal{O}[\![\cdot]\!]$.

Proof.

- One can check that $\mathcal{D}[\![x]\!] = \mathcal{O}[\![x]\!], \forall x \in X$, which implies correctness of $[\![\cdot]\!]$ with respect to $\mathcal{O}[\![\cdot]\!]$ [Todoran 2000, Ciobanu and Todoran 2012]
- For weak completeness, suppose $x_1, x_2 \in X$, $(\alpha, k) \in CRes$ are such that $[\![x_1]\!](\alpha, [\![k]\!]) \neq [\![x_2]\!](\alpha, [\![k]\!])$. By previous Lemma $\exists S$ such that
$$\mathcal{D}[\![S(x_1)]\!] = [\![x_1]\!](\alpha, [\![k]\!]) \neq [\![x_2]\!](\alpha, [\![k]\!]) = \mathcal{D}[\![S(x_2)]\!]$$
- Hence, $\mathcal{O}[\![S(x_1)]\!] = \mathcal{D}[\![S(x_1)]\!] \neq \mathcal{D}[\![S(x_2)]\!] = \mathcal{O}[\![S(x_2)]\!]$,
- We conclude that $[\![\cdot]\!]$ is (weakly complete and thus) weakly abstract with respect to $\mathcal{O}[\![\cdot]\!]$.

$\square$

Concluding Remarks and Future Research

- We introduce an optimality criterion specific of continuation semantics that we call weak abstractness
  - Which relaxes the completeness condition of the classic full abstractness criterion [Milner 1977].
- To illustrate the approach we presented a weakly abstract continuation semantics for an asynchronous language
  - We believe similar weak abstractness results can be obtained for various advanced concurrent control concepts
    - Andorra Model [Todoran and Papaspyrou 2000]
    - Multiparty interactions [Ciobanu and Todoran 2015]
    - Nature inspired formalisms [Ciobanu and Todoran 2017]

References

P. America, J.J.M.M. Rutten,
"Solving reflexive domain equations in a category of complete metric spaces,"
J. of Comp. Syst. Sci, vol. 39(3), pp. 343–375, 1989.

J.W. De Bakker, J. Zucker,
"Processes and the Denotational Semantics of Concurrency,"
Information and Control, vol. 54, pp. 70–120, 1982.

J.W. De Bakker, E.P. De Vink,
Control Flow Semantics,
MIT Press, 1996.

References

J.C.M. Baeten, W.P. Weijland,
Process Algebra,
Cambridge Univ. Press, 1990.

F.S. De Boer, J.N. Kok, C. Palamidessi, J.J.M.M. Rutten,
"A Paradigm for Asynchronous Communication and its Application to Concurrent Constraint Programming,"
In K.R. Apt, J.W. De Bakker, J.J.M.M. Rutten, eds., *Logic Programming Languages: Constraints, Functions and Objects*, pp. 82-114, MIT Press, 1993.

S.D. Brookes,
"Full Abstraction for a Shared-Variable Parallel Language,"'
Information and Computation, vol. 127(2), pp. 145–163, 1996.

R. Cartwright, P.L. Curien, M. Felleisen,
Fully Abstract Semantics for Observably Sequential Languages,
Information and Computation, vol. 111(2), 297401, 1994.

References

G. Ciobanu and E.N. Todoran,
"Abstract Continuation Semantics for Asynchronous Concurrency,"
Technical Report FML-12-02, Romanian Academy, 2012. Available at
http://iit.iit.tuiasi.ro/TR/reports/fml1202.pdf

G. Ciobanu, E.N. Todoran,
"Continuation Semantics for Asynchronous Concurrency,"
Fundamenta Informaticae, vol. 131(3-4), pp. 373–388, 2014.

G. Ciobanu, E.N. Todoran,
"Continuation Semantics for Concurrency with Multiple Channels Communication,"
Proceedings of 17th International Conference on Formal Engineering Methods (ICFEM 2015), Lecture Notes in Computer Science, vol. 9407, pp. 400–416, Springer, 2015.

G. Ciobanu, E.N. Todoran,
"Denotational Semantics of Membrane Systems by using Complete Metric Spaces,"
Theoretical Computer Science, Elsevier, 2017 (in press).

G. Ciobanu, E.N. Todoran,
A Semantic Investigation of Spiking Neural P-Systems,
19th Int. Conf. on Membrane Computing (CMC 19), Dresden, 2018 (accepted).

References

O. Danvy,
"On Evaluation Contexts, Continuations and the Rest of the Computation,"
Proceedings of 4th ACM SIGPLAN Continuations Workshop, pp. 13–23, 2004.

R. Hieb, R. K. Dybvig, C.W. Anderson,
Subcontinuations,
Lisp and Symbolic Computation, vol.7(1), 83110, 1994.

K. Honda, M. Tokoro,
"An Object Calculus for Asynchronous Communication,"
Lecture Notes in Computer Science, vol. 512, pp. 133–147, Springer, 1991.

References

H. Jifeng, M. Josephs, C.A.R. Hoare.
A theory of synchrony and asynchrony.
In *Proc. of IFIP Working Conference on Programming Concepts and Methods*, pp. 459-478, 1990.

R. Milner,
"Fully Abstract Models of Typed λ-Calculi,"
Theoretical Computer Science, vol. 4, pp. 1–22, 1977.

S. Owicki, D. Gries.
An Axiomatic Proof Technique for Parallel Programs.
Acta Informatica, 6:319–340, 1976.

References

C. Palamidessi,
”Comparing the Expressive Power of the Synchronous and the Asynchronous π-Calculus,”
Math. Structures in Comp. Sci., vol. 13(5), pp. 685–719, 2003.

J.J.M.M. Rutten,
”Semantic Correctness for a Parallel Object Oriented Language,”
SIAM Journal of Computing, vol. 19(2), pp. 341–383, 1990.

V. Saraswat.
Concurrent constraint programming.
MIT Press, 1993.

D.S. Scott.
Data Types as Latices.
SIAM J. Comput., vol. 5, pp. 522-587, 1976.

D.S. Scott.
Domains for denotational semantics.
Proc. 9th ICALP, pp. 577-613, LNCS 140, 1982.

References

C. Strachey and C.P. Wadsworth,
Continuations: A Mathematical Semantics for Handling Full Jumps,
Higher-Order and Symbolic Computation, vol. 13(1/2), pp. 135152, 2000. Reprint of the technical monograph PRG-11, Oxford University.

E.N. Todoran,
"Metric Semantics for Synchronous and Asynchronous Communication: a Continuation-based Approach,"
Electronic Notes in Theoretical Computer Science, vol. 28, pp. 101–127, Elsevier, 2000.

E.N. Todoran and N. Papaspyrou,
"Continuations for Parallel Logic Programming,"
Proceedings of the 2nd International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP 2000), pp. 257–267, ACM Press, 2000.