Experiments with Continuation Semantics for DNA Computing

Eneia Nicolae Todoran, Nikolaos Papaspyrou

TU Cluj-Napoca, Romania, TU Athens, Greece

9th International Conference on
Intelligent Computer Communication and Processing
(ICCP 2013)

Cluj-Napoca, Romania,

September 5-7, 2013

Aim and contribution

- We investigate the semantics of a process algebra L_{DNA}, incorporating some basic concepts of DNA computing
  - L_{DNA} was introduced [Cardelli-2011],[1] where a couple of so-called 'strand algebras' are presented
    - These formalisms can capture the massive concurrency available at molecular level in DNA systems
  - [Cardelli-2011] explains the relevance of L_{DNA} for DNA computing
- We offer a semantic investigation of L_{DNA} following the discipline of denotational semantics

[1] The syntax used in [Cardelli-2011] is slightly different

Aim and contribution

- We use the mathematical methodology of metric semantics [De Bakker and De Vink-1996]
  - The main mathematical tool Banach's fixed point Theorem
- We use continuations and powerdomains to represent nondeterministic behavior
  - An element of a powerdomain is a collection of sequences of observables representing DNA structures
- As far as we know this is the first paper that employs denotational semantics in the semantic investigation of DNA computing

Aim and contribution

- We present two denotational semantics, corresponding to two different notions of an observable item
  1. In the first denotational model $[\![\cdot]\!]_{\mathcal{G}}$ an observable is a L_{DNA} gate which captures an interaction
  2. In the second denotational model $[\![\cdot]\!]_{\mathcal{C}}$ an observable is a multiset of L_{DNA} elements representing a configuration of a system specified in L_{DNA}

Aim and contribution

- Behavior is described as a collection of sequences of DNA observables with no silent steps interspersed
- At present most researchers prefer operational semantics [Plotkin-2004]
  - In operational semantics behavior is expressed based on transitions between system configurations
  - Each transition can show the effect of an interaction
- We demonstrate that such operational effects can also be captured in denotational semantics by using continuation semantics for concurrency (CSC) [Todoran-2000]

Informal explanation

- L_{DNA} combines: signals, gates and populations
  - A signal $x, y, \ldots \in X$ is a symbol taken from an alphabet X
  - A gate is an operator $([x_1, \ldots, x_n], [y_1, \ldots, y_m])$ that joins the signals $x_1, \ldots, x_n$ and forks the signals $y_1, \ldots, y_m$
    - The order of signals in $[x_1, \ldots, x_n]$ and $[y_1, \ldots, y_m]$ is irrelevant, hence, $[x_1, \ldots, x_n]$ and $[y_1, \ldots, y_m]$ are multisets.
    - The signals $x_1, \ldots, x_n$ of a gate $([x_1, \ldots, x_n], [y_1, \ldots, y_m])$ represent a join pattern [Fournet and Gonthier-2002]
  - A population may be finite P^k ($k \in \mathbb{N}$) or unbounded P^*
    - The construct for unbounded (inexhaustible) populations is based on the replication primitive of π-calculus [Milner-1999].

Informal explanation

- Signals and gates combine in a multiset of elements - a 'chemical soup' - that proceed concurrently
  - '$\|$' is the operator for parallel composition in L_{DNA}
- An interaction betwen n signals $x_1, \ldots, x_n$ and a gate $([x_1, \ldots, x_n], [y_1, \ldots, y_m])$ can be described operationally

$$x_1 \parallel \cdots \parallel x_n \parallel ([x_1, \ldots, x_n], [y_1, \ldots, y_m]) \rightarrow y_1 \parallel \cdots \parallel y_m$$

  - Signals $x_1, \ldots, x_n$ and the gate are consumed
  - The signals $y_1, \ldots, y_m$ are released in the multiset
- Signals can interact with gates, but signals cannot interact with signals, nor gates with gates [Cardelli-2011]

Compositionality

- L_{DNA} is a process algebra, i.e. a formal language that can describe concurrent activities of multiple processes
  - In general, a process algebra only provides compositionality at the level of syntax
- In denotational semantics compositionality is provided at the level of semantics
  - *Language constructs denote values from a mathematical domain of interpretation*

$$[\![\cdot]\!] : \mathcal{L} \to \mathbf{D}$$

  - *Semantic definitions are compositional*

$$[\![\cdots x_1 \cdots x_2 \cdots]\!] = \cdots [\![x_1]\!] \cdots [\![x_2]\!] \cdots$$

$[\![\cdot]\!]_{\mathcal{G}}$ and $[\![\cdot]\!]_{\mathcal{C}}$ examples

- Let $P_1 = (x_1 \parallel ([x_1],[y_1])) \parallel (x_2 \parallel ([x_2],[y_2]))$, $P_1 \in L_{DNA}$
- $[\![P_1]\!]_{\mathcal{G}}(f_0)(null) =$
  $\{([x_1],[y_1])([x_2],[y_2]), ([x_2],[y_2])([x_1],[y_1])\}$
  - f_0 is the empty (synchronous) continuation
  - *null* is the empty synchronization context
- Let $P_2 = x \parallel (([x_1,x_2],[x_3]) \parallel ([x],[x_1,x_2])) \in L_{DNA}$
- $[\![P_2]\!]_{\mathcal{G}}(f_0)(null) = \{([x],[x_1,x_2])([x_1,x_2],[x_3])\}$

$[\![\cdot]\!]_{\mathcal{G}}$ and $[\![\cdot]\!]_{\mathcal{C}}$ examples

- $P_2 = x \parallel (([x_1, x_2], [x_3]) \parallel ([x], [x_1, x_2])) \in L_{DNA}$
- Operationally, P_2 behaves as follows [Cardelli-2011]
  $$P_2 \;\rightarrow\; x_1 \parallel x_2 \parallel ([x_1, x_2], [x_3]) \;\rightarrow\; x_3$$
- $[\![\cdot]\!]_{\mathcal{C}}$ can capture such (operational) effects denotationally:
  $$[\![P_2]\!]_{\mathcal{C}}(f_0)(null) = \{[x_1, x_2, ([x_1, x_2], [x_3])][x_3]\}$$
- The multiset $[x_1, x_2, ([x_1, x_2], [x_3])]$ is a semantic representation of the L_{DNA} term $x_1 \parallel x_2 \parallel ([x_1, x_2], [x_3])$

Formal syntax of L_{DNA}

$$P ::= 0 \mid x \mid g \mid P \parallel P \mid P^k \mid P^*$$

- $(x, y \in)X$ is a (countable) set of *signals*
- $(\overline{x}, \overline{y} \in)[X]$ is the set of all finite multisets of signals
- $(g \in)G = [X] \times [X]$ is the set of *gates*
  - A gate $g = (\overline{x}, \overline{y})(\in G)$ is a pair of multisets of signals

Synchronization contexts

The set $(w \in) W$ of synchronization contexts is defined by

$$W = \{\mu(w) \mid w \in \{null\} \cup (G \times [X])\}$$

where $\mu : \{null\} \cup (G \times [X]) \to Bool$ is given by

$$\mu(null) = true$$
$$\mu((\overline{x}, \overline{y}), \overline{x}') = (\overline{x}' \subseteq \overline{x})$$

$\mu(w) = true$ iff w could synchronize but not necessarily synchronizes (already)

Operations on synchronization contexts

- We define $\oplus : (W \times [X]) \to W$ by:

$$w \oplus \overline{x}'' = \begin{cases} ((\overline{x}, \overline{y}), \overline{x}' \uplus \overline{x}'') & \text{if } w = ((\overline{x}, \overline{y}), \overline{x}') \text{ and} \\ & \quad \overline{x}' \uplus \overline{x}'' \subseteq \overline{x} \\ w & \text{otherwise.} \end{cases}$$

  $\oplus$ adds a multiset of signals to a synchronization context
- We define $\sigma : W \to Bool$ by:

$$\sigma(null) = false$$
$$\sigma((\overline{x}, \overline{y}), \overline{x}') = (\overline{x}' = \overline{x})$$

  If $w \in W$ and $\sigma(w)$ we say that w synchronizes
- Remark $\sigma(w) \Rightarrow \mu(w)$ (if w synchronizes then w could synchronize)

Operations on synchronization contexts

- Let $(\cdot < \cdot), [\cdot < \cdot) : (W \times W) \to Bool$,

$$(w_1 < w_2) = \begin{cases} true & \text{if } w_1 = (g_1, \overline{x}_1) \text{ and } w_2 = null \\ true & \text{if } w_1 = (g_1, \overline{x}_1), w_2 = (g_2, \overline{x}_2), \\ & \quad g_1 = g_2, \text{ and } \overline{x}_2 \subset \overline{x}_1 \\ false & \text{otherwise.} \end{cases}$$

$$[w_1 < w_2) = (w_1 < w_2) \wedge \neg(\sigma(w_1))$$

  - Intuitively, $(w_1 < w_2)$ if $\mu(w_1)$ and w_1 is closer of synchronization than w_2
  - $[w_1 < w_2)$ if $(w_1 < w_2)$ and w_1 does not synchronize (yet)

- Remarks

(a) For any $w_1, w_2 \in W$, if $\sigma(w_2)$ then $\neg(w_1 < w_2)$.

(b) For any $w_1, w_2 \in W$, if $\sigma(w_2)$ then $\neg[w_1 < w_2)$.

Operations on synchronization contexts

- We define $c_w : W \to \mathbb{N} \cup \{\infty\}$ by:

  $$c_w(null) = \infty$$
  $$c_w((\overline{x}, \overline{y}), \overline{x}') = |\overline{x} \setminus \overline{x}'|$$

- We endow $\mathbb{N} \cup \{\infty\}$ with the total order $0 < 1 < 2 < \cdots < n < \cdots \infty$
  - $|\overline{x} \setminus \overline{x}'|$ is the cardinal number of the multiset $\overline{x} \setminus \overline{x}'$
- $c_w(w)$ that measures how far or close w is from synchronization

  Remarks

(a) $(w_1 < w_2) \Rightarrow c_w(w_1) < c_w(w_2)$.

(b) $\sigma(w) \Leftrightarrow c_w(w) = 0$.

Domain definitions for $[\![\cdot]\!]_{\mathcal{G}}$

$$(\phi \in)\mathbf{D} \cong \{d_0\} + \mathbf{Den}$$
$$(\varphi \in)\mathbf{Den} = \mathbf{F} \xrightarrow{1} W \to \mathbf{P}$$
$$(f \in)\mathbf{F} = \mathbf{K} \xrightarrow{1} W \to \mathbf{P} \quad \text{(synchronous continuations)}$$
$$(\kappa \in)\mathbf{K} = \tfrac{1}{2} \cdot \mathbf{D} \quad \text{(asynchronous continuations)}$$
$$(p \in)\mathbf{P} = \mathcal{P}_{nco}(\mathbf{Q})$$
$$(q \in)\mathbf{Q} \cong \{\epsilon\} + (G \times (\tfrac{1}{2} \cdot \mathbf{Q}))$$

- Remarks
  - In general, in CSC an asynchronous continuation is a more complex structure, e.g., a tree of computations
  - In the case of L_{DNA}, a continuation is a multiset packed into a single computation by means of parallel composition

Semantic operators

- $+ : (\mathbf{P} \times \mathbf{P}) \to \mathbf{P}$ is the operator for nondeterministic choice

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

- We define $(:) : (Bool \times \mathbf{P}) \to \mathbf{P}$ by:

  $true : p = p$
  $false : p = \{\epsilon\}$

- '$+$' is nonexpansive, associative, commutative and idempotent
- ':' is nonexpansive and

  $b : (p_1 + p_2) = (b : p_1) + (b : p_2),$
  $(b_1 \wedge b_2) : p = b_1 : (b_2 : p) = b_2 : (b_1 : p).$

Semantic operators - parallel composition

- Let $\| = fix(\Psi)$, $\Psi : \mathbf{Op} \to \mathbf{Op}$, $\mathbf{Op} = (\mathbf{D} \times \mathbf{D}) \xrightarrow{1} \mathbf{D}$

$$
\begin{aligned}
&\Psi(\psi)(d_0, d_0) = d_0\\
&\Psi(\psi)(d_0, \varphi) = \varphi\\
&\Psi(\psi)(\varphi, d_0) = \varphi\\
&\Psi(\psi)(\varphi_1, \varphi_2) =\\
&\quad \lambda f.\lambda w.(\varphi_1(\lambda \kappa_1.\lambda w_1.\\
&\qquad\qquad ((w_1 < w) : f(\psi(\kappa_1, \varphi_2))\, w_1)+\\
&\qquad\qquad ([w_1 < w) : \varphi_2(\lambda \kappa_2.f(\psi(\kappa_1, \kappa_2)))\, w_1))\, w+\\
&\qquad \varphi_2(\lambda \kappa_2.\lambda w_2.\\
&\qquad\qquad ((w_2 < w) : f(\psi(\kappa_2, \varphi_1))\, w_2)+\\
&\qquad\qquad ([w_2 < w) : \varphi_1(\lambda \kappa_1.f(\psi(\kappa_2, \kappa_1)))\, w_2))\, w)
\end{aligned}
$$

- Lemma $\Psi : Op \xrightarrow{\frac{1}{2}} Op$ (Ψ is a contraction, hence it has a unique fixed point, according to Banach's Theorem)

Semantic operators - left synchronization

- We define $\lfloor : (\mathbf{Den} \times \mathbf{Den}) \to \mathbf{Den}$ by:

$$
\begin{aligned}
&(\varphi_1 \lfloor \varphi_2) f w = \\
&\quad \varphi_1(\lambda \kappa_1 . \lambda w_1 . ((w_1 < w) : f(\kappa_1 \parallel \varphi_2)\, w_1) + \\
&\qquad\qquad\qquad ([\, w_1 < w) : \varphi_2(\lambda \kappa_2 . f(\kappa_1 \parallel \kappa_2))\, w_1))\, w
\end{aligned}
$$

- $\varphi_1 \parallel \varphi_2 = \lambda f . \lambda w . ((\varphi_1 \lfloor \varphi_2) f w + (\varphi_1 \lfloor \varphi_2) f w)$
  - $(\varphi_1 \lfloor \varphi_2)$ attempts to synchronize two computations φ_1, φ_2, in this order
  - No observable is produced before synchronization
  - $\parallel$ and $\lfloor$ are nonexpansive

Semantics of signals and gates

- $[\![\cdot]\!]^X_{\mathcal{G}} : X \to \mathbf{D}$

$$
\begin{array}{l}
[\![x]\!]^X_{\mathcal{G}} = \\
\quad \lambda f.\lambda w.\, \text{if } (w = \mathit{null}) \text{ then } \{\epsilon\} \\
\qquad\qquad\quad \text{else } \text{ let } w' = w \oplus [x] \\
\qquad\qquad\qquad\qquad \text{in } ((w' < w) : f(d_0)(w'))
\end{array}
$$

- $[\![\cdot]\!]^G_{\mathcal{G}} : G \to \mathbf{D}$

$$
\begin{array}{l}
[\![g]\!]^G_{\mathcal{G}} = \\
\quad \lambda f.\lambda w.\, \text{if } (w = \mathit{null}) \text{ then } f(d_0)(g, []) \text{ else } \{\epsilon\}
\end{array}
$$

Initial synchronous continuation

- Let $\Phi : \mathbf{F} \to \mathbf{F}$ be given by:

$$
\begin{array}{l}
\Phi(f)kw = \\
\quad \text{if } (\neg\, \sigma(w)) \text{ then } \{\epsilon\} \\
\quad \text{else } \text{ let } w = (g, \overline{x}') \\
\qquad\qquad g = (\overline{x}, [y_1, \ldots, y_m]) \\
\qquad\qquad \phi = \|^{m+1} (\kappa, [\![y_1]\!]^X_{\mathcal{G}}, \ldots, [\![y_m]\!]^X_{\mathcal{G}}) \\
\qquad \text{in } \text{ if } \phi = d_0 \text{ then } \{g\} \text{ else } g \cdot \phi(f)\mathit{null}
\end{array}
$$

- We define $f_0 = \mathit{fix}(\Phi)$

- Lemma Φ is a contraction, i.e. $\Phi : \mathbf{F} \overset{\frac{1}{2}}{\to} \mathbf{F}$

Semantic operator of unbounded populations

- Let Ω : **Den** $\rightarrow$ **Den** $\rightarrow$ **Den** be given by:

$$\Omega\varphi_1\varphi_2 f w = \varphi_1(\lambda\kappa_1.\lambda w_1.((w_1 < w) : f(\kappa_1 \parallel \varphi_2)\, w_1) + ([w_1 < w) : \Omega\varphi_1\varphi_2(\lambda\kappa_2.f(\kappa_1 \parallel \kappa_2))\, w_1))\, w$$

  - Ω is used in the equation for unbounded populations
  - Well-definedness of Ω follows by induction on $c_w(w)$

- Lemma Ω : **Den** $\stackrel{1}{\rightarrow}$**Den** $\stackrel{\frac{1}{2}}{\rightarrow}$**Den**
- Remark Let $\varphi \in$ **Den**. $\Omega(\varphi)$ is $\frac{1}{2}$ contractive. Let $\overline{\varphi} = fix(\Omega(\varphi))$. One can check that $\Omega(\varphi)(\overline{\varphi}) = \varphi \lfloor \overline{\varphi}$.

Denotational semantics $[\![\cdot]\!]_{\mathcal{G}}$

- We define $[\![\cdot]\!]_{\mathcal{G}} : L_{DNA} \to \mathbf{D}$ by:

$$\begin{array}{rcl}
[\![0]\!]_{\mathcal{G}} & = & d_0 \\
[\![x]\!]_{\mathcal{G}} & = & [\![x]\!]_{\mathcal{G}}^X \\
[\![g]\!]_{\mathcal{G}} & = & [\![g]\!]_{\mathcal{G}}^G \\
[\![P^k]\!]_{\mathcal{G}} & = & \|^k \left([\![P]\!]_{\mathcal{G}}, \ldots, [\![P]\!]_{\mathcal{G}}\right) \\
[\![P^*]\!]_{\mathcal{G}} & = & \begin{cases} d_0 & \text{if } [\![P]\!]_{\mathcal{G}} = d_0, \\ fix(\Omega([\![P]\!]_{\mathcal{G}})) & \text{otherwise} \end{cases} \\
[\![P_1 \parallel P_2]\!]_{\mathcal{G}} & = & [\![P_1]\!]_{\mathcal{G}} \parallel [\![P_2]\!]_{\mathcal{G}}
\end{array}$$

  Let $\mathcal{D}_{\mathcal{G}}[\![\cdot]\!] : L_{DNA} \to \mathbf{P}$ be given, for any $P \in L_{DNA}$, by:

$$\mathcal{D}_{\mathcal{G}}[\![P]\!] = [\![P]\!]_{\mathcal{G}}(f_0)(null)$$

- Remark $[\![P^*]\!]_{\mathcal{G}} = fix(\lambda\varphi.([\![P]\!]_{\mathcal{G}} \lfloor \varphi))$ (when $[\![P]\!]_{\mathcal{G}} \neq d_0$)

Semantics of unbounded populations

- The operator for unbounded populations should satisfy the property: $[\![P^*]\!]_{\mathcal{G}} = [\![P]\!]_{\mathcal{G}} \parallel [\![P^*]\!]_{\mathcal{G}}$ [Milner-1999]
  - $[\![P]\!]_{\mathcal{G}} \parallel [\![P^*]\!]_{\mathcal{G}}$ is a nondeteministic choice between $[\![P]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}}$ and $[\![P^*]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}}$. '$+$' is idempotent, hence it is enough to prove that $[\![P]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}} = [\![P^*]\!]_{\mathcal{G}} \lfloor [\![P]\!]_{\mathcal{G}}$
- $[\![P]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}} = ([\![P]\!]_{\mathcal{G}} \lfloor \cdots ([\![P]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}}) \cdots)$
- $[\![P^*]\!]_{\mathcal{G}} \lfloor [\![P]\!]_{\mathcal{G}} = ([\![P]\!]_{\mathcal{G}} \lfloor \cdots ([\![P]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}}) \cdots) \lfloor [\![P]\!]_{\mathcal{G}}$
  - Both $[\![P^*]\!]_{\mathcal{G}} \lfloor [\![P]\!]_{\mathcal{G}}$ and $[\![P]\!]_{\mathcal{G}} \lfloor [\![P^*]\!]_{\mathcal{G}}$ take as many copies of $[\![P]\!]_{\mathcal{G}}$ as necessary (but not more) to achieve a synchroniz.
  - The synchronization produces a $\frac{1}{2}$ contraction step
- After synchronization the continuations are executed in parallel with $[\![P^*]\!]_{\mathcal{G}} \parallel [\![P]\!]_{\mathcal{G}}$ and $[\![P^*]\!]_{\mathcal{G}}$, respectively.
- Hence, the relationship between $[\![P^*]\!]_{\mathcal{G}} \parallel [\![P]\!]_{\mathcal{G}}$ and $[\![P^*]\!]_{\mathcal{G}}$ is an invariant of the comput. [Ciobanu and Todoran-2013]

Experiments with $[\![\cdot]\!]_{\mathcal{G}}$

- Let $P_1, P_2, P_3 \in L_{DNA}$,

$$P_1 = (x_1 \parallel ([x_1],[y_1])) \parallel (x_2 \parallel ([x_2],[y_2]))$$
$$P_2 = x \parallel (([x_1,x_2],[x_3]) \parallel ([x],[x_1,x_2]))$$
$$P_3 = (y \parallel ([y,x_1],[x_2,y])^*) \parallel (x_1)^3$$

- One may check the following results:

$$\mathcal{D}_{\mathcal{G}}[\![P_1]\!] = \{([x_1],[y_1])([x_2],[y_2]), ([x_2],[y_2])([x_1],[y_1])\}$$
$$\mathcal{D}_{\mathcal{G}}[\![P_2]\!] = \{([x],[x_1,x_2])([x_1,x_2],[x_3])\}$$
$$\mathcal{D}_{\mathcal{G}}[\![P_3]\!] = \{ggg\}, \text{ where } g = ([y,x_1],[x_2,y])$$

- Let also $P_4 = x^* \parallel ([x],[y])^*$. The execution of P_4 never terminates. Our semantic interpreter produces:

$$\{([x],[y])([x],[y])([x],[y])\ldots\}$$

  - ...actually, only first n steps, for any n

Configurations

- We define the class $\alpha \in A$ of L_{DNA} elements inductively.
  - Any signal $x \in X$ or gate $g \in G$ is an L_{DNA} element, i.e. $X \subseteq A, G \subseteq A$.
  - If $\alpha_1, \ldots, \alpha_n \in A$ then $(*, [\alpha_1, \ldots, \alpha_n]) \in A$. We use the notation $[\alpha_1, \ldots, \alpha_n]^* = (*, [\alpha_1, \ldots, \alpha_n])$; here, $[\alpha_1, \ldots, \alpha_n]$ is a multiset of L_{DNA} elements.
- We define the class $\gamma \in \Gamma$ of L_{DNA} configurations by $\Gamma = [A]$; a configuration is a multiset of L_{DNA} elements.

Semantic domains

$$(\phi \in)\mathbf{D} \cong \{d_0\} + (\Gamma \times \mathbf{Den})$$

$$(\varphi \in)\mathbf{Den} = \mathbf{F} \stackrel{1}{\to} W \to \mathbf{P}$$

$$(f \in)\mathbf{F} = \mathbf{K} \stackrel{1}{\to} W \to \mathbf{P} \text{ (synchronous continuations)}$$

$$(\kappa \in)\mathbf{K} = \tfrac{1}{2} \cdot \mathbf{D} \text{ (asynchronous continuations)}$$

$$(p \in)\mathbf{P} = \mathcal{P}_{nco}(\mathbf{Q})$$

$$(q \in)\mathbf{Q} \cong \{\epsilon\} + (\Gamma \times (\tfrac{1}{2} \cdot \mathbf{Q}))$$

Parallel composition operator $\|$

$\|: (\mathbf{D} \times \mathbf{D}) \to \mathbf{D}$ acts as a multiset sum on configurations.
$d_0 \parallel d_0 = d_0$, $d_0 \parallel \phi = d_0 \parallel \phi = \phi$ and:

$$
\begin{array}{l}
(\gamma_1, \varphi_1) \parallel (\gamma_2, \varphi_2) = \\
\quad (\gamma_1 \uplus \gamma_2, \\
\quad \lambda f.\lambda w.(\varphi_1(\lambda\kappa_1.\lambda w_1. \\
\qquad\qquad ((w_1 < w) : f(\kappa_1 \parallel (\gamma_2, \varphi_2))\, w_1)+ \\
\qquad\qquad ([\, w_1 < w) : \varphi_2(\lambda\kappa_2.f(\kappa_1 \parallel \kappa_2))\, w_1))\, w+ \\
\qquad \varphi_2(\lambda\kappa_2.\lambda w_2. \\
\qquad\qquad ((w_2 < w) : f(\kappa_2 \parallel (\gamma_1, \varphi_1))\, w_2)+ \\
\qquad\qquad ([\, w_2 < w) : \varphi_1(\lambda\kappa_1.f(\kappa_2 \parallel \kappa_1))\, w_2))\, w)
\end{array}
$$

Semantics of signals and gates

$$
\begin{aligned}
&[\![x]\!]_{\mathcal{C}}^{X} = \\
&\quad ([x], \lambda f.\lambda w.\,\text{if } (w = \mathit{null}) \text{ then } \{\epsilon\} \\
&\qquad\qquad \text{else } \ \text{let } w' = w \oplus [x] \\
&\qquad\qquad\qquad \text{in } ((w' < w) : f(d_0)(w'))
\end{aligned}
$$

$$
\begin{aligned}
&[\![g]\!]_{\mathcal{C}}^{G} = \\
&\quad ([g], \lambda f.\lambda w.\,\text{if } (w = \mathit{null}) \text{ then } f(d_0)(g, []) \text{ else } \{\epsilon\}\ )
\end{aligned}
$$

Initial continuation

$$
\begin{array}{l}
f_0 k w = \text{ if } (\neg\, \sigma(w)) \text{ then } \{\epsilon\} \\
\qquad\qquad \text{else } \text{ let } w = ((\overline{x}, [y_1, \ldots, y_m]), \overline{x}') \\
\qquad\qquad\qquad\qquad \phi = \|^{m+1} (\kappa, [\![y_1]\!]_{\mathcal{C}}^{X}, \ldots, [\![y_m]\!]_{\mathcal{C}}^{X}) \\
\qquad\qquad\qquad \text{in } \text{ if } \phi = d_0 \text{ then } \{[]\} \text{ else} \\
\qquad\qquad\qquad\quad \text{let } \phi = (\gamma, \varphi) \text{ in } \gamma \cdot \varphi(f_0) \mathit{null}
\end{array}
$$

Unbounded populations

- We define the semantics of unbounded populations based on the operator $\Omega : \Gamma \to \mathbf{D} \to \mathbf{D} \to \mathbf{D}$,

$$\begin{aligned}&\Omega\gamma_2\varphi_1\varphi_2 f w = \\ &\quad \varphi_1(\lambda\kappa_1.\lambda w_1.((w_1 < w) : f(\kappa_1 \parallel (\gamma_2, \varphi_2))\, w_1) + \\ &\qquad\qquad ([w_1 < w) : \Omega\gamma_2\varphi_1\varphi_2(\lambda\kappa_2.f(\kappa_1 \parallel \kappa_2))\, w_1))\, w\end{aligned}$$

- For any $\gamma \in \Gamma, \varphi \in \mathbf{Den}$, $\Omega\gamma\varphi$ is $\frac{1}{2}$ contractive.

- If $\overline{\varphi} = fix(\Omega\gamma\varphi)$ then $\Omega\gamma\varphi\overline{\varphi} = \varphi \lfloor \overline{\varphi}$, where

$$\begin{aligned}&(\varphi_1 \lfloor \varphi_2) f w = \\ &\quad \varphi_1(\lambda\kappa_1.\lambda w_1.((w_1 < w) : f(\kappa_1 \parallel (\gamma_2, \varphi_2))\, w_1) + \\ &\qquad\qquad ([w_1 < w) : \varphi_2(\lambda\kappa_2.f(\kappa_1 \parallel \kappa_2))\, w_1))\, w\end{aligned}$$

Denotational semantics $[\![\cdot]\!]_{\mathcal{C}}$

We define $[\![\cdot]\!]_{\mathcal{C}} : L_{DNA} \to \mathbf{D}$ by:

$$
\begin{array}{rcl}
[\![0]\!]_{\mathcal{C}} & = & d_0 \\
[\![x]\!]_{\mathcal{C}} & = & [\![x]\!]_{\mathcal{C}}^{X} \\
[\![g]\!]_{\mathcal{C}} & = & [\![g]\!]_{\mathcal{C}}^{G} \\
[\![P^k]\!]_{\mathcal{C}} & = & \|^k \, ([\![P]\!]_{\mathcal{C}}, \ldots, [\![P]\!]_{\mathcal{C}}) \\
[\![P^*]\!]_{\mathcal{C}} & = & \begin{cases} d_0 & \text{if } [\![P]\!]_{\mathcal{C}} = d_0, \\ ([\gamma^*], fix(\Omega[\gamma^*]\varphi) & \text{if } [\![P]\!]_{\mathcal{C}} = (\gamma, \varphi) \end{cases} \\
[\![P_1 \parallel P_2]\!]_{\mathcal{C}} & = & [\![P_1]\!]_{\mathcal{C}} \parallel [\![P_2]\!]_{\mathcal{C}}
\end{array}
$$

Let $\mathcal{D}_{\mathcal{G}}[\![\cdot]\!] : L_{DNA} \to \mathbf{P}$ be given, for any $P \in L_{DNA}$, by:

$$\mathcal{D}_{\mathcal{C}}[\![P]\!] = [\![P]\!]_{\mathcal{C}}(f_0)(null)$$

Experiments with $[\![\cdot]\!]_{\mathcal{C}}$

Let $P_1, P_2, P_3 \in L_{DNA}$ (be as for $[\![\cdot]\!]_{\mathcal{G}}$)

$$P_1 = (x_1 \parallel ([x_1],[y_1])) \parallel (x_2 \parallel ([x_2],[y_2]))$$
$$P_2 = x \parallel (([x_1,x_2],[x_3]) \parallel ([x],[x_1,x_2]))$$
$$P_3 = (y \parallel ([y,x_1],[x_2,y])^*) \parallel (x_1)^3$$

One can check the following:

$$\begin{aligned}\mathcal{D}_{\mathcal{C}}[\![P_1]\!] &= \{[x_2,y_1,([x_2],[y_2])][y_1,y_2],\\ &\quad [x_1,y_2,([x_1],[y_1])][y_1,y_2]\}\\ \mathcal{D}_{\mathcal{C}}[\![P_2]\!] &= \{[x_1,x_2,([x_1,x_2],[x_3])][x_3]\}\\ \mathcal{D}_{\mathcal{C}}[\![P_3]\!] &= \{\gamma_1\gamma_2\gamma_3\}\end{aligned}$$

where

$$\gamma_1 = [x_1,x_1,x_2,y,[(([y,x_1],[x_2,y]))]^*]$$
$$\gamma_2 = [x_1,x_2,x_2,y,[(([y,x_1],[x_2,y]))]^*]$$
$$\gamma_3 = [x_2,x_2,x_2,y,[(([y,x_1],[x_2,y]))]^*]$$

Concluding remarks and future research

- We report on the first stage of an investigation of the denotational semantics of DNA computing
- In the future we will investigate the possibility to define a continuation semantics for the stochastic strand algebra given in section 4 of [Cardelli-2011]
- By using techniques from metric semantics we will study the formal relationship between the denotational semantics and the operational semantics of DNA computing

P.America, J.J.M.M. Rutten,
Solving reflexive domain equations in a category of complete metric spaces,
J. of Comput. System Sci., 39:343-375, 1989.

J.W. de Bakker, E.P. de Vink,
Control flow semantics.
MIT Press, 1996.

L. Cardelli,
Strand algebras for DNA computing,
Natural Computing 10(1): 407-428, 2011.

G. Ciobanu and E.N.Todoran,
Continuation semantics for asynchronous concurrency,
Fundamenta Informaticae, 2013 (in press).

C. Fournet, G. Gonthier,

The Join calculus: a language for distributed mobile programming,
LNCS 25:268–332, 2002.

R. Milner.
Communicating and mobile systems: the π caculus.
Cambridge Univ. Press, 1999.

G. Plotkin,
A structural approach to operational semantics,
J. Log. Algebr. Program. (60-61):17–139, 2004.

E.N.Todoran,
Metric semantics for synchronous and asynchronous communication: a continuation-based approach,
ENTCS 28:119–146, 2000.