Nameless Formalization of HOcore in Coq

Initial Bachelor Seminar Talk

Lukas Convent

Advisor: Tobias Tebbi

Supervisor: Prof. Dr. Gert Smolka

SAARLAND UNIVERSITY
COMPUTER SCIENCE
Programming Systems

March 18, 2016

Nameless Formalization of HOcore in Coq

- **HOcore is a process calculus**
  - Modelling concurrent systems
  - *others*: CCS, π-Calculus
- . . . **with binders**
  - Processes can receive and deliver values
  - *e.g.* $\overline{a}\langle R\rangle \parallel a(x).P \xrightarrow{\tau} \emptyset \parallel P\{{}^{R}/_{x}\}$
- We aim at **nameless formalization** (De Bruijn indices) of HOcore and some proofs about it in Coq

What is special about HOcore?

CCS

- **No value passing**, only synchronization
- $start!.P \parallel start?.Q \xrightarrow{\tau} P \parallel Q$

π-Calculus

- **Channels** passed as values, Turing-complete
- $\overline{chgCh}\langle n\rangle.P \parallel chgCh(x).\overline{x}\langle msg\rangle.Q \xrightarrow{\tau} P \parallel \overline{n}\langle msg\rangle.Q$

HOcore (Higher-Order)

- **Processes** passed as values, Turing complete
- $\overline{exe2}\langle P\rangle \parallel exe2(x).(x \parallel x) \xrightarrow{\tau} \emptyset \parallel (P \parallel P)$

Previous work

- Lanese, Pérez, Sangiorgi, Schmitt: **On the Expressiveness and Decidability of Higher-Order Process Calculi**. *Proceedings of LICS'08*
- Maksimovi, Schmitt: **HOCore in Coq**. *Interactive Theorem Proving, Vol. 9236, 2015*

HOcore Processes

Example

$$\overline{exe2}\langle P\rangle \parallel exe2(x).(x \parallel x) \xrightarrow{\tau} \emptyset \parallel (P \parallel P)$$

$$\overline{exe2}\langle P\rangle \parallel exe2.(0 \parallel 0) \xrightarrow{\tau} \emptyset \parallel (P \parallel P)$$

		De Bruijn	
$P, Q ::=$	$\overline{a}\langle P\rangle$	$\overline{a}\langle P\rangle$	Output process
	$\mid a(x).P$	$a.P$	Input prefixed process
	$\mid x$	$x \in \mathbb{N}$	Process variable
	$\mid P \parallel Q$	$P \parallel Q$	Parallel composition
	$\mid \emptyset$	$\emptyset$	Empty process

- After transmission: Terminate ($\emptyset$-process)
- All channels are global

HOcore transitions (1)

De Bruijn

$$\frac{}{\overline{a}\langle P\rangle \xrightarrow{\overline{a}\langle P\rangle} \emptyset} \quad \textsc{Out} \quad \frac{}{\overline{a}\langle P\rangle \xrightarrow{\overline{a}\langle P\rangle} \emptyset}$$

$$\frac{}{a(x).P \xrightarrow{a} \lambda x.P} \quad \textsc{In} \quad \frac{}{a.P \xrightarrow{a} P}$$

$$\frac{P \xrightarrow{\overline{a}\langle R\rangle} P' \quad Q \xrightarrow{a} \lambda x.Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'\{^{R}/_{x}\}} \quad \textsc{SynL} \quad \frac{P \xrightarrow{\overline{a}\langle R\rangle} P' \quad Q \xrightarrow{a} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'[R :: id]}$$

Example

$$\frac{\dfrac{}{\overline{exe2}\langle P\rangle \xrightarrow{\overline{exe2}\langle P\rangle} \emptyset}\ \textsc{Out} \qquad \dfrac{}{exe2.(0 \parallel 0) \xrightarrow{exe2} 0 \parallel 0}\ \textsc{In}}{\overline{exe2}\langle P\rangle \parallel exe2.(0 \parallel 0) \xrightarrow{\tau} \emptyset \parallel (P \parallel P)}\ \textsc{SynL}$$

HOcore transitions (2)

De Bruijn

ParTauL
$$\frac{P \xrightarrow{\tau} P}{P \parallel Q \xrightarrow{\tau} P' \parallel Q}$$

ParOutL
$$\frac{P \xrightarrow{\overline{a}\langle R\rangle} P'}{P \parallel Q \xrightarrow{\overline{a}\langle R\rangle} P' \parallel Q}$$

$$\frac{P \xrightarrow{a} \lambda x.P' \qquad x \notin fv(Q)}{P \parallel Q \xrightarrow{a} \lambda x.(P' \parallel Q)}$$

ParInL
$$\frac{P \xrightarrow{a} P'}{P \parallel Q \xrightarrow{a} P' \parallel Q[\uparrow]}$$

Bisimulation & Bisimilarity

Bisimulation in CCS

$$\textit{Bisimulation}\ \mathcal{R} :\Leftrightarrow \quad \begin{tikzcd} P \arrow[r, dash, "\mathcal{R}"] \arrow[d, "\alpha"'] & Q \arrow[d, dotted, "\alpha"] \\ P' \arrow[r, dash, dotted, "\mathcal{R}"] & Q' \end{tikzcd} \quad \wedge \quad \begin{tikzcd} P \arrow[r, dash, "\mathcal{R}"] \arrow[d, dotted, "\alpha"'] & Q \arrow[d, "\alpha"] \\ P' \arrow[r, dash, dotted, "\mathcal{R}"] & Q' \end{tikzcd}$$

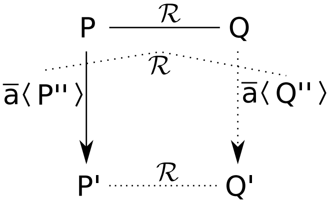
Bisimilarity in CCS

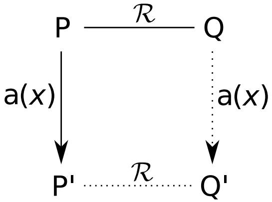
$$P \sim Q :\Leftrightarrow \quad \exists\ \textit{Bisimulation}\ \mathcal{R}.\ \ (P, Q) \in \mathcal{R}$$

- In CCS, bisimulation demands identical actions
  $\Rightarrow$ But for HOcore we want: $\overline{a}\langle P \parallel Q\rangle.\emptyset \ \ \sim \ \ \overline{a}\langle Q \parallel P\rangle.\emptyset$
- There are several options for a definition of bisimilarity
- A straightforward one is **IO-Bisimilarity**

IO Bisimilarity

$\mathcal{R}$ is an **IO Bisimulation** if the following properties hold:





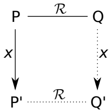
Does this suffice? No, it is not yet a **congruence**:

$$\textit{We want:} \qquad 0 \not\sim 3$$

$$\textit{because:} \quad \overline{a}\langle P\rangle \parallel a.0 \not\sim \overline{a}\langle P\rangle \parallel a.3$$

Use **Variable Bisimilarity**:

$$\textsc{Rem}\ \frac{}{x \xrightarrow{x} \emptyset} \qquad \textsc{ParRemL}\ \frac{P \xrightarrow{x} P'}{P \parallel Q \xrightarrow{x} P' \parallel Q}$$



Inductive vs. Coinductive Definitions (1)

$$\frac{}{nil \in S} \qquad \frac{x \in \mathbb{N} \quad xs \in S}{x :: xs \in S}$$

Rule functional

$$\mathcal{F}(S) = \{nil\} \cup \{x :: xs \mid {}^{x \in \mathbb{N} \wedge}_{xs \in S}\}$$

Finite Lists $L_{fin} =$ **Least fixed-point** of $\mathcal{F}$

Least set which is closed under the rules:

$$L_{fin} = \{nil\} \cup \{x :: xs \mid {}^{x \in \mathbb{N} \wedge}_{xs \in L_{fin}}\}$$
$$= \mathcal{F}(L_{fin})$$

Finite and Infinite Lists $L_\omega =$ **Greatest fixed-point** of $\mathcal{F}$

Largest set which is closed under the rules:

$$L_\omega = \{nil\} \cup \{x :: xs \mid {}^{x \in \mathbb{N} \wedge}_{xs \in L_\omega}\}$$
$$= \mathcal{F}(L_\omega)$$

Which fixed-point?

{finite lists over $\mathbb{N}$} ... {finite and infinite lists over $\mathbb{N}$}

Inductive vs. Coinductive Definitions (2)

$$\frac{}{nil \in S} \qquad \frac{x \in \mathbb{N} \quad xs \in S}{x :: xs \in S}$$

Rule functional

$$\mathcal{F}(S) = \{nil\} \cup \{x :: xs \mid \begin{smallmatrix} x \in \mathbb{N} \wedge \\ xs \in S \end{smallmatrix}\}$$

Finite Lists $L_{fin} =$
Least fixed-point of $\mathcal{F}$

Inductive Definition by $\mathcal{F}$

Finite and Infinite Lists $L_\omega =$
Greatest fixed-point of $\mathcal{F}$

Coinductive Definition by $\mathcal{F}$

Fixed-Point Theorem of Knaster-Tarski

If $\mathcal{F}$ is **monotone**,

Least fixed-point of $\mathcal{F}$

$= \bigcap \{T \mid \mathcal{F}(T) \subseteq T\}$

$= \bigcap \{$Pre-fixed-points$\}$

Greatest fixed-point of $\mathcal{F}$

$= \bigcup \{T \mid T \subseteq \mathcal{F}(T)\}$

$= \bigcup \{$Post-fixed-points$\}$

Formalizing Bisimilarity

Rule Functional of Bisimilarity $\sim$

$$\mathcal{F}(\,\mathrm{B}\,) = \{(P,Q) \mid \begin{array}{c}\forall P'.P \longrightarrow P' \Rightarrow \\ \exists Q'.Q \longrightarrow Q' \wedge P' \ \mathrm{B}\ Q'\end{array} \bigwedge \begin{array}{c}\forall Q'.Q \longrightarrow Q' \Rightarrow \\ \exists P'.P \longrightarrow P' \wedge P' \ \mathrm{B}\ Q'\end{array}\}$$

$\sim$

$$\begin{aligned}\sim &= \text{Greatest fixed-point of } \mathcal{F} \\ &= \cup\,\{\mathcal{B} \mid \mathcal{B} \subseteq \mathcal{F}(\mathcal{B})\} = \cup\,\{\,\text{Bisimulations } \mathcal{B}\,\}\end{aligned}$$

Proof technique for $\sim$

If $\quad \mathcal{B} \subseteq \mathcal{F}(\mathcal{B}) \quad$ and $\quad (P,Q) \in \mathcal{B}$

then $\quad (P,Q) \in \sim$

Bisimulation-up-to Relations

$\mathcal{R}$ is a **bisimulation-up-to-U** if:

$$\begin{tikzcd}
P \arrow[rrr, dash, "\mathcal{R}"] \arrow[d, "\alpha"'] & & & Q \arrow[d, dotted, "\alpha"] \\
P' \arrow[r, dotted, dash, "U"] & P'' \arrow[r, dotted, dash, "\mathcal{R}"] & Q'' \arrow[r, dotted, dash, "U"] & Q'
\end{tikzcd}$$

Sound up-to relation U for $\mathcal{F}$

U is a sound up-to relation for $\mathcal{F}$ if :

For any $\mathcal{R} \subseteq \mathcal{F}(U \circ \mathcal{R} \circ U) = \mathcal{F}(\mathcal{R}^U)$

we have $\mathcal{R} \subseteq \sim$

Bisimulation-up-to-Bisimilarity

U is a sound up-to relation for $\mathcal{F}$ if:

For any $\mathcal{R} \subseteq \mathcal{F}(U \circ \mathcal{R} \circ U)$ we have $\mathcal{R} \subseteq \sim$

$\sim$ is a sound up-to relation for $\sim$

Let $\mathcal{R}$ be a **bisimulation-up-to-$\sim$** rel.: $\mathcal{R} \subseteq \mathcal{F}(\sim \circ \mathcal{R} \circ \sim)$

$$
\begin{aligned}
\mathcal{R} \subseteq \quad \mathcal{R}^{\sim} &= \sim \circ \mathcal{R} \circ \sim && \sim \text{ is reflexive} \\
&\subseteq \sim \circ \mathcal{F}(\mathcal{R}^{\sim}) \circ \sim && \text{Assumption} \\
&= \sim \circ \mathcal{F}(\sim \circ \mathcal{R} \circ \sim) \circ \sim && \text{Def. up-to} \\
&= \mathcal{F}(\sim) \circ \mathcal{F}(\sim \circ \mathcal{R} \circ \sim) \circ \mathcal{F}(\sim) && \sim \text{ is FP of } \mathcal{F} \\
&\subseteq \mathcal{F}(\sim \circ \sim \circ \mathcal{R} \circ \sim \circ \sim) && \\
& && \mathcal{F}(A) \circ \mathcal{F}(B) \subseteq \mathcal{F}(A \circ B) \\
&= \mathcal{F}(\sim \circ \mathcal{R} \circ \sim) && \sim \text{ is transitive} \\
&= \mathcal{F}(\mathcal{R}^{\sim}) \subseteq \sim &&
\end{aligned}
$$

Conclusion & Outlook

- *De Bruijn* indices allow an easy method to get around α renaming
- Regarding bisimilarity as a greatest fixed-point of a functional makes general proofs about bisimilarity possible
- Next steps:
  - Relating (different?) bisimilarities to each other
  - How can proofs be done in a compositional way in Coq? (Paco Library)
  - Other properties of bisimilarities (decidability)

Thank you!

References

- Lanese, Pérez, Sangiorgi, Schmitt: **On the Expressiveness and Decidability of Higher-Order Process Calculi**. *Proceedings of LICS'08*
- Pérez: *http://www.cs.unibo.it/∼perez/talks/coplas.pdf*
- Maksimovi, Schmitt: **HOCore in Coq**. *Interactive Theorem Proving, Vol. 9236, 2015*
- Sangiorgi: Presentation on **Bisimulation and Coinduction**: *http://www.fing.edu.uy/inco/eventos/SEFM2011/cursos/Davide.pdf*