formalizing a turing-complete choreography calculus in coq

luís cruz-filipe

(joint work with fabrizio montesi & marco peressotti)

department of mathematics and computer science
university of southern denmark

types meeting
june 13th, 2019

motivation (i/ii)

choreographic programming

programming paradigm for concurrent systems, based on "alice-to-bob" communication

- high-level languages
- automatic compilation to process calculi
- deadlock-freedom by design

motivation (i/ii)

choreographic programming programming paradigm for concurrent systems, based on “alice-to-bob” communication

- high-level languages
- automatic compilation to process calculi
- deadlock-freedom by design

theoretical issues too many (published) proofs read “straightforward by structural induction”

- serious errors found recently in process calculi
- problems getting articles accepted

motivation (ii/ii)

goal formalize a research article (in coq)

- hopefully speed-up the refereeing process
- dispell doubts on correctness of proofs and methods

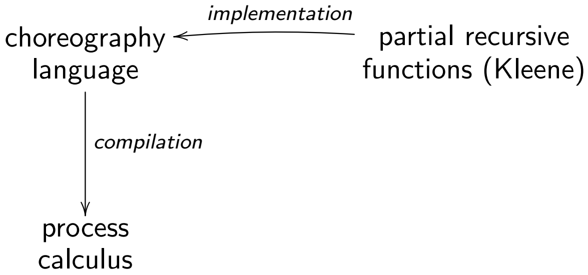
motivation (ii/ii)

goal formalize a research article (in coq)

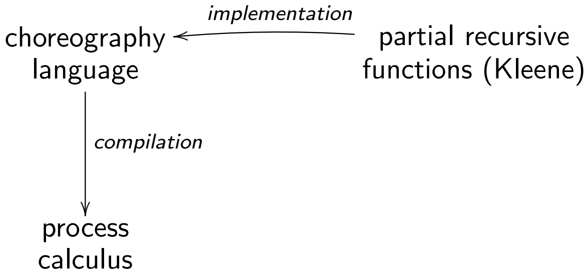
- hopefully speed-up the refereeing process
- dispell doubts on correctness of proofs and methods

main result turing-completeness of a core choreography calculus

general picture



general picture



challenges

- dependent types all over the place
- induction hypotheses are too weak

a concrete example

composition

given $g : \mathbb{N}^n \to \mathbb{N}$ and $f_1, \ldots, f_n : \mathbb{N}^k \to \mathbb{N}$, their composition is $h = C(g, \vec{f}) : \mathbb{N}^k \to \mathbb{N}$ with

$$h(x_1, \ldots, x_k) = g\left(f_n(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k)\right)$$

if all subterms are defined

a concrete example

composition

given $g : \mathbb{N}^n \to \mathbb{N}$ and $f_1, \ldots, f_n : \mathbb{N}^k \to \mathbb{N}$, their composition is $h = C(g, \vec{f}) : \mathbb{N}^k \to \mathbb{N}$ with

$$h(x_1, \ldots, x_k) = g\left(f_n(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k)\right)$$

if all subterms are defined

first attempt

type $\mathcal{PR}$ of partial recursive functions, with

$$\text{Composition} : \mathcal{PR} \to \text{list}(\mathcal{PR}) \to \mathcal{PR}$$

and a function arity : $\mathcal{PR} \to \mathbb{N}$

a concrete example

composition

given $g : \mathbb{N}^n \to \mathbb{N}$ and $f_1, \ldots, f_n : \mathbb{N}^k \to \mathbb{N}$, their composition is $h = C(g, \vec{f}) : \mathbb{N}^k \to \mathbb{N}$ with

$$h(x_1, \ldots, x_k) = g\,(f_n(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k))$$

if all subterms are defined

first attempt

type $\mathcal{PR}$ of partial recursive functions, with

$$\text{Composition} : \mathcal{PR} \to \text{list}(\mathcal{PR}) \to \mathcal{PR}$$

and a function arity : $\mathcal{PR} \to \mathbb{N}$
$\rightsquigarrow$ unclean...

a concrete example

composition

given $g : \mathbb{N}^n \to \mathbb{N}$ and $f_1, \ldots, f_n : \mathbb{N}^k \to \mathbb{N}$, their composition is $h = C(g, \vec{f}) : \mathbb{N}^k \to \mathbb{N}$ with

$$h(x_1, \ldots, x_k) = g\left(f_n(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k)\right)$$

if all subterms are defined

second attempt

dependent type $\Pi_{n:\mathbb{N}}.\mathcal{PR}(n)$ of partial recursive functions with arity n, and

$$\mathsf{Composition} : \Pi_{n,k}.\mathcal{PR}(n) \to \mathsf{Vec}_n(\mathcal{PR}(k)) \to \mathcal{PR}(k)$$

a concrete example

composition

given $g : \mathbb{N}^n \to \mathbb{N}$ and $f_1, \ldots, f_n : \mathbb{N}^k \to \mathbb{N}$, their composition is $h = C(g, \vec{f}) : \mathbb{N}^k \to \mathbb{N}$ with

$$h(x_1, \ldots, x_k) = g\left(f_n(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k)\right)$$

if all subterms are defined

second attempt

dependent type $\Pi_{n:\mathbb{N}}.\mathcal{PR}(n)$ of partial recursive functions with arity n, and

$$\text{Composition} : \Pi_{n,k}.\mathcal{PR}(n) \to \text{Vec}_n(\mathcal{PR}(k)) \to \mathcal{PR}(k)$$

- more faithful, but more complex
- problems with induction

a concrete example

composition given $g : \mathbb{N}^n \to \mathbb{N}$ and $f_1, \ldots, f_n : \mathbb{N}^k \to \mathbb{N}$, their composition is $h = C(g, \vec{f}) : \mathbb{N}^k \to \mathbb{N}$ with

$$h(x_1, \ldots, x_k) = g\left(f_n(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k)\right)$$

if all subterms are defined

second attempt dependent type $\Pi_{n:\mathbb{N}}.\mathcal{PR}(n)$ of partial recursive functions with arity n, and

$$\text{Composition} : \Pi_{n,k}.\mathcal{PR}(n) \to \text{Vec}_n(\mathcal{PR}(k)) \to \mathcal{PR}(k)$$

- more faithful, but more complex
- problems with induction

depth function induction on the depth of the proof that $f : \mathcal{PR}(n)$

$$\text{depth} : \Pi_n.\mathcal{PR}(n) \to \mathbb{N}$$

turing completeness of choreographies

mapping $\{\{\cdot\}\}$ from partial recursive functions to choreographies

- notion of function computed by a choreography
- soundness: $\{\{f\}\}$ computes f

status formalized definitions, soundness proved only for concrete examples

challenges

relations on choreographies

$\rightsquigarrow$ structural induction (again)

reduction $C, \sigma \to C', \sigma'$ (one-step execution) and *structural precongruence* $C \preceq C'$ (out-of-order execution)

challenges

relations on choreographies

problematic rules

$\rightsquigarrow$ structural induction (again)

reduction $C, \sigma \to C', \sigma'$ (one-step execution) and *structural precongruence* $C \leq C'$ (out-of-order execution)

$$\frac{C \leq C' \qquad C' \leq C''}{C \leq C''}$$

$$\frac{C_1 \leq C_1' \qquad C_1', \sigma_1 \to C_2', \sigma_2 \qquad C_2' \leq C_2}{C_1, \sigma_1 \to C_2, \sigma_2}$$

challenges

$\rightsquigarrow$ structural induction (again)

relations on choreographies

reduction $C, \sigma \to C', \sigma'$ (one-step execution) and *structural precongruence* $C \preceq C'$ (out-of-order execution)

problematic rules

$$\frac{C \preceq C' \qquad C' \preceq C''}{C \preceq C''}$$

$$\frac{C_1 \preceq C_1' \qquad C_1', \sigma_1 \to C_2', \sigma_2 \qquad C_2' \preceq C_2}{C_1, \sigma_1 \to C_2, \sigma_2}$$

our solution

induction on the number of steps in the derivation

challenges

$\rightsquigarrow$ structural induction (again)

relations on choreographies

reduction $C, \sigma \to C', \sigma'$ (one-step execution) and *structural precongruence* $C \preceq C'$ (out-of-order execution)

problematic rules

$$\frac{C \preceq_n C' \qquad C' \preceq_k C''}{C \preceq_{n+k} C''}$$

$$\frac{C_1 \preceq_k C_1' \qquad C_1', \sigma_1 \to_n C_2', \sigma_2 \qquad C_2' \preceq_m C_2}{C_1, \sigma_1 \to_{k+n+m} C_2, \sigma_2}$$

our solution

induction on the number of steps in the derivation

challenges

$\rightsquigarrow$ structural induction (again)

relations on choreographies

reduction $C, \sigma \to C', \sigma'$ (one-step execution) and *structural precongruence* $C \preceq C'$ (out-of-order execution)

problematic rules

$$\frac{C \preceq_n C' \qquad C' \preceq_k C''}{C \preceq_{n+k} C''}$$

$$\frac{C_1 \preceq_k C_1' \qquad C_1', \sigma_1 \to_n C_2', \sigma_2 \qquad C_2' \preceq_m C_2}{C_1, \sigma_1 \to_{k+n+m} C_2, \sigma_2}$$

our solution

induction on the number of steps in the derivation

$\rightsquigarrow$ soundness, but also canonical forms for reductions

conclusions

- work in progress
- main definitions in place
- similar problems in different places, uniform solutions
- better understanding of the theory
- better definitions?

thank you!