formalising choreographic programming

luís cruz-filipe
(joint work with fabrizio montesi & marco peressotti)

department of mathematics and computer science
university of southern denmark

vest workshop
july 12th, 2021

the goal

long-term

a certified framework for choreographic programming

the goal

long-term

a certified framework for choreographic programming

in this talk

the first steps

- a core choreographic language
- a proof of turing completeness
- a core process calculus
- a proof of the epp theorem

the goal

long-term

a certified framework for choreographic programming

in this talk

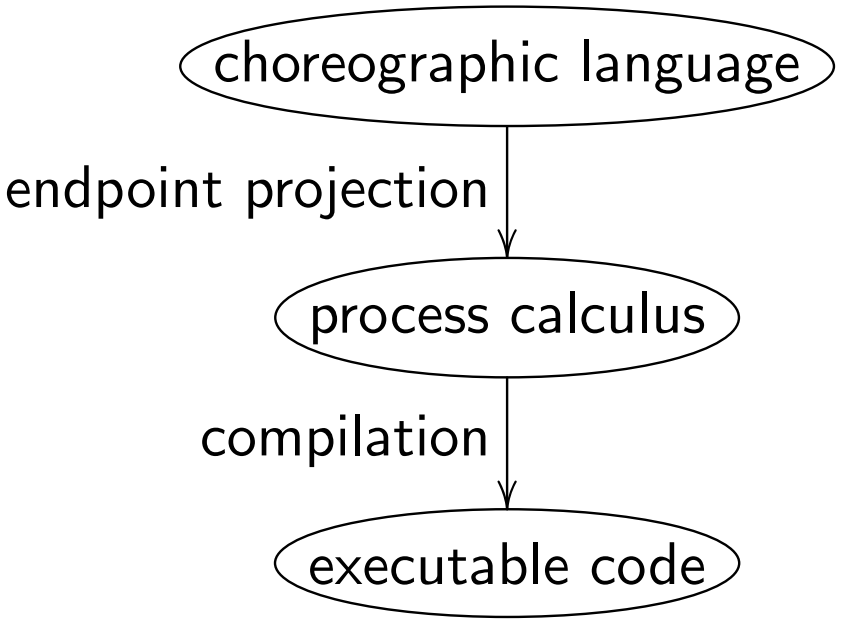
the first steps

- a core choreographic language
- a proof of turing completeness
- a core process calculus
- a proof of the epp theorem

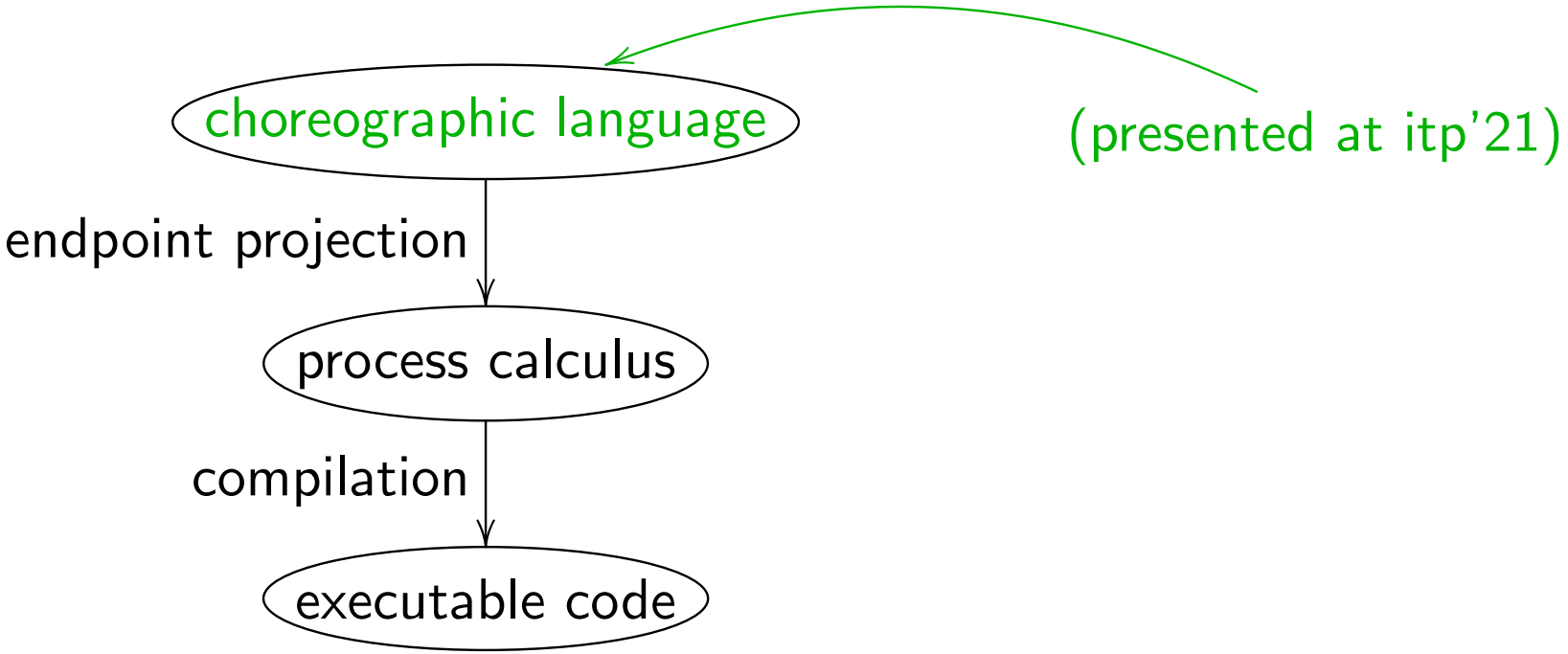
very brief summary

initially presented at types'19, publications at itp'21 & ictac'21

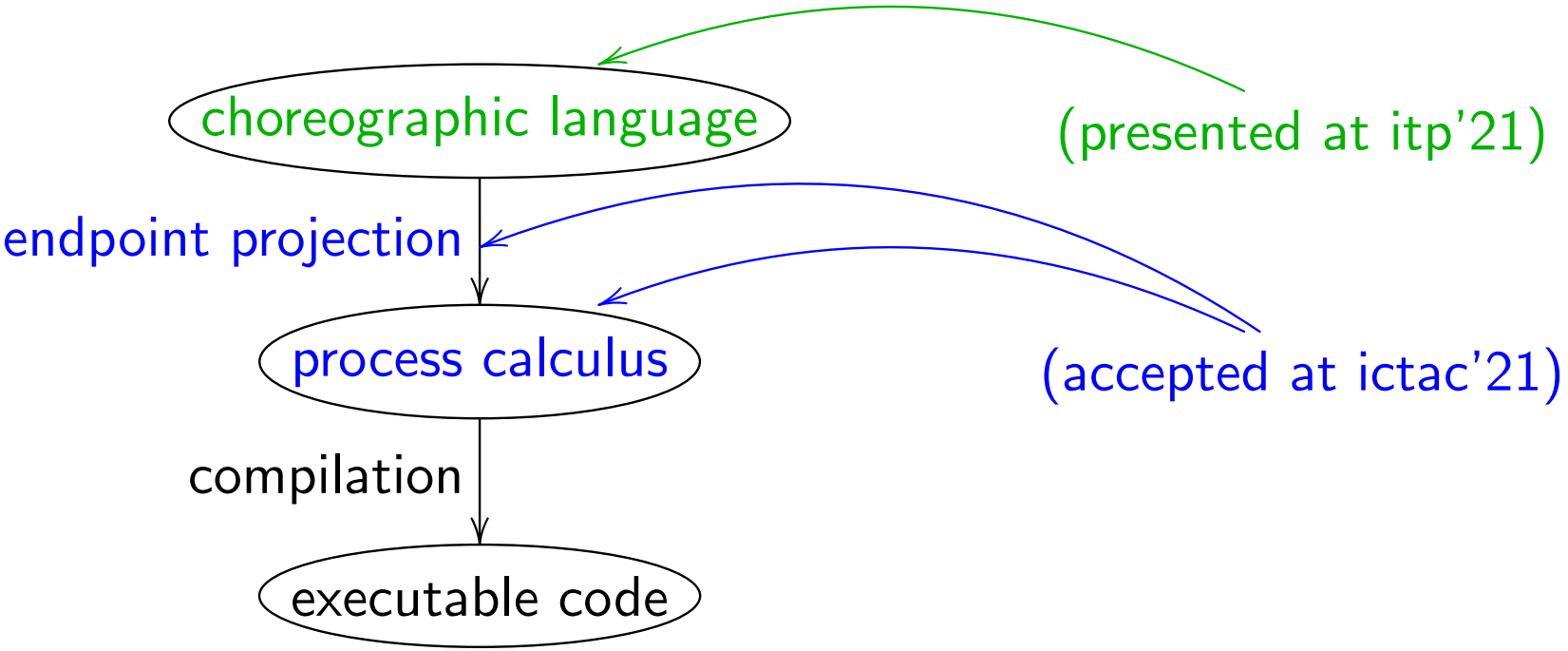
a bird's-eye view



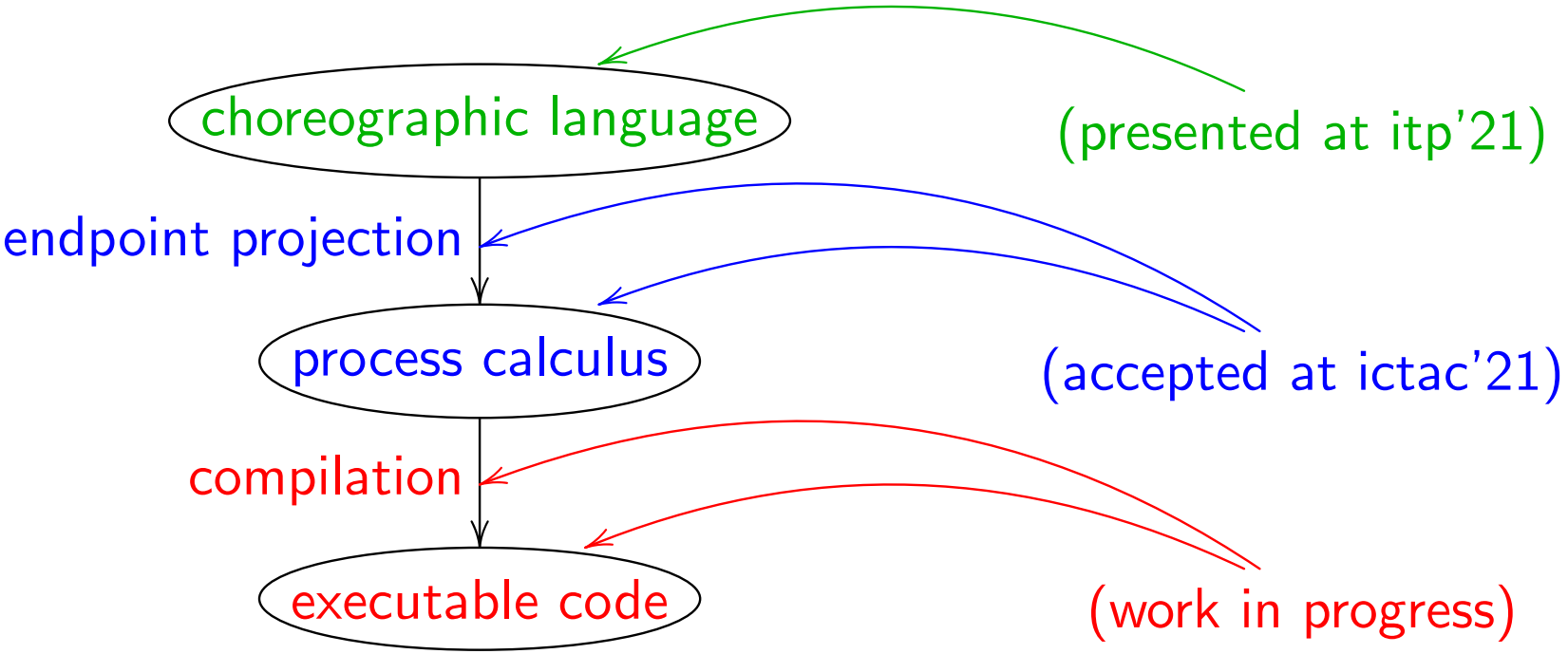
a bird's-eye view



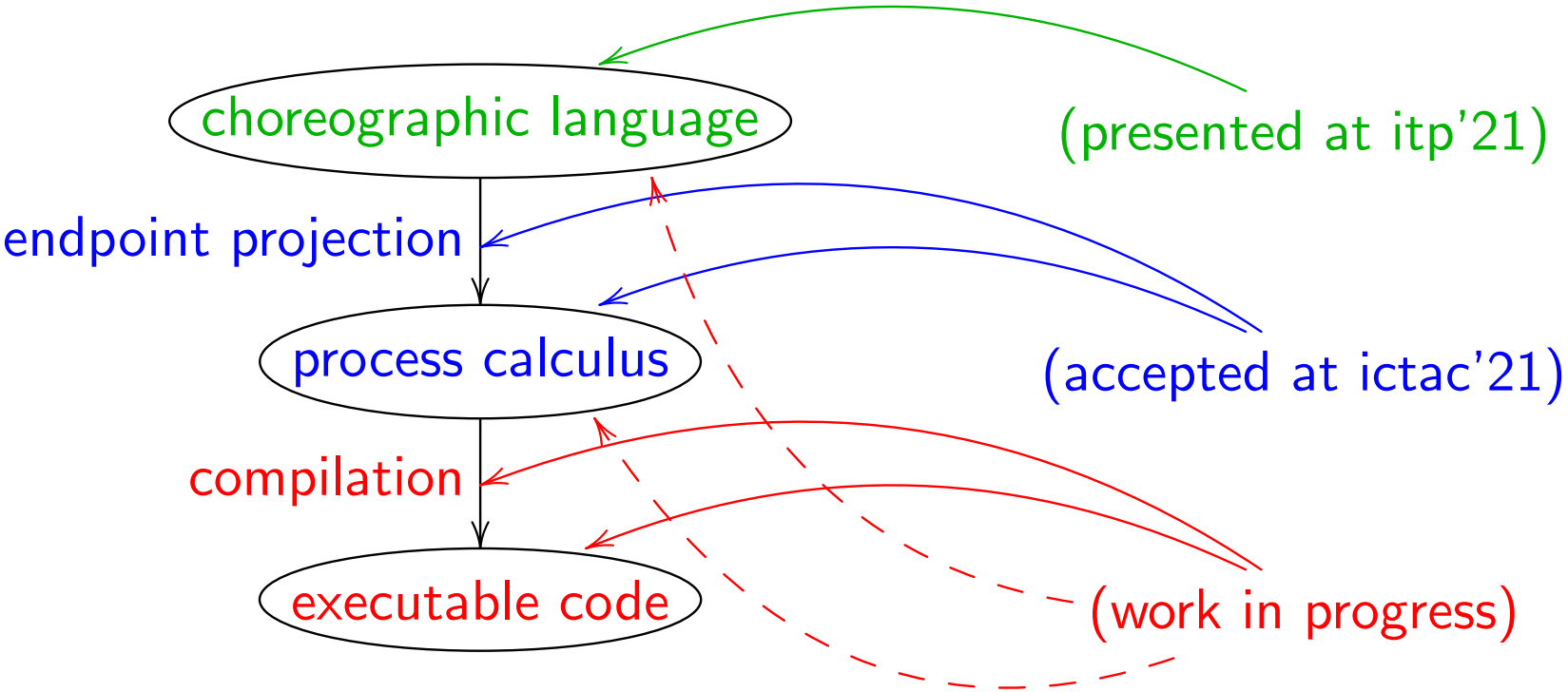
a bird's-eye view



a bird's-eye view



a bird's-eye view



our language

a minimal language

- value communication
- label selections (for projection)
- conditionals
- trailing procedure calls (for recursion)

our language

a minimal language

- value communication
- label selections (for projection)
- conditionals
- trailing procedure calls (for recursion)

agnostic language

- parametric on expressions and values
- only two labels

the first step

choreographic language

- syntax and semantics
- progress and deadlock-freedom
- properties of the semantics: determinism, confluence
- turing-completeness from the communication structure

(itp'21)

a bit on the process

first attempt: a miserable failure

- bad model of *out-of-order execution*
- pen-and-paper definition by means of a structural precongruence (ugh)
- properties are very "intuitive" and never* actually proved
- the number of auxiliary results exploded, with no end in sight

*to the best of the speaker's knowledge

a weird coincidence?

- oddly enough, this is also where students get stuck

is this good or bad?

second attempt: a success story with side-effects

- model out-of-order execution using an lts
- "intuitive" properties no longer needed (or can be proved)
- auxiliary lemmas disappeared
- final proof of confluence around 25% of the size of the previous (incomplete) development

is this good or bad?

second attempt: a success story with side-effects

- model out-of-order execution using an lts
- "intuitive" properties no longer needed (or can be proved)
- auxiliary lemmas disappeared
- final proof of confluence around 25% of the size of the previous (incomplete) development

and the cherry on top of the cake

our students also liked the new definitions :-)

the second step

the epp theorem

- definition of a suitable process calculus
- formalisation of endpoint projection
- challenges: partial functions (branching terms, merging, projection)
- different solutions (dedicated terms, auxiliary types, indirect definitions)
- case explosion (partially) handled by automation

(ictac'21)

what's next?

implementation

using coq's extraction mechanism, we can obtain a certified compiler from choreographies to processes

- next step: build an (uncertified?) compiler to a real programming language
- extend the choreographic language (and the process calculus) with other interesting constructs

conclusions

formalising choreographic programming:

- is feasible
- is useful
- can speed up things

conclusions

formalising choreographic programming:

- is feasible
  - we did it (at least partially)
- is useful
  - our theory benefitted from it
- can speed up things
  - convincing coq is faster than convincing reviewers...

thank you!