Introduction to Programming and Proving in Cedille

Chris Jenkins, Colin McDonald, Aaron Stump
Computer Science
The University of Iowa
Iowa City, Iowa

Plan for the tutorial

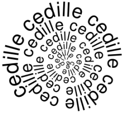
ç?

⊢ *CeDilLE*

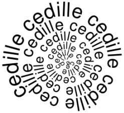
cedille

$\rightsquigarrow \texttt{cedille}_{\texttt{core}}$

c d ll



Plan for the tutorial

Ç?	Motivation and background for Cedille
⊢ *CeDilLE*	Syntax and semantics
`cedille`	Tooling: emacs frontend ↔ backend
⇝ $\texttt{cedille}_{\texttt{core}}$	Elaboration to Cedille Core
`c d ll`	Spine-local type inference
	Future directions

Ç?

Motivation and background for Cedille

A little history

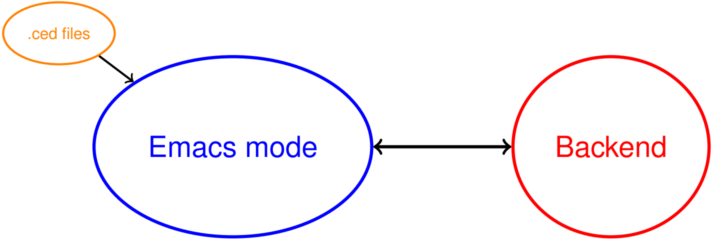
A little history

System F (Girard, Reynolds, early 1970s)

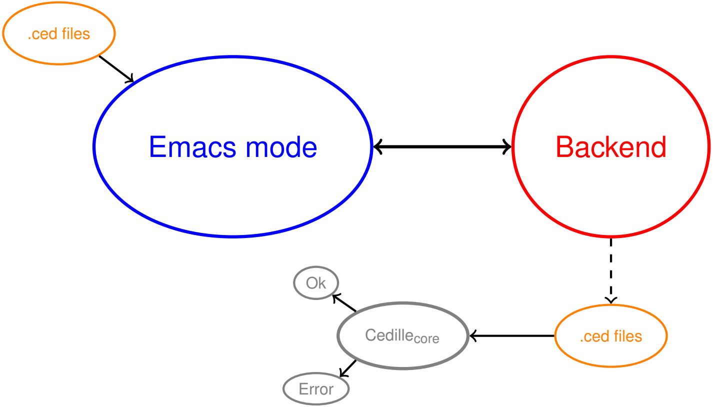
System F (Girard, Reynolds, early 1970s)

- Raw power (impredicativity!)
- A little crude (no Curry-Howard)

Architecture of Cedille

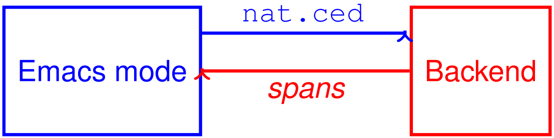


Architecture of Cedille



Emacs mode (*to be revised*)

- Cedille has an emacs mode for editing Cedille files
- Based on a generic structured-editing mode by Carl Olson



- A span is [*label*, *start-pos*, *end-pos*, *attributes*]
- Spans communicated in JSON
- Cedille sends all type information, in span attributes
- Monadic style for writing the backend (type checker)