

Why3

A Multi-Prover Platform for Program Verification

Jean-Christophe Filliâtre
CNRS

joint work with
Andrei Paskevich, Claude Marché, and François Bobot

ProVal team, Orsay, France

IFIP WG 1.9/2.14 “Verified Software”
June 2011



INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

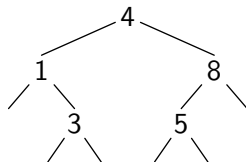
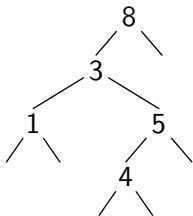
centre de recherche **BACLAY** - ÎLE-DE-FRANCE



UNIVERSITÉ
PARIS-SUD 11

Why3 Tutorial: Same Fringe

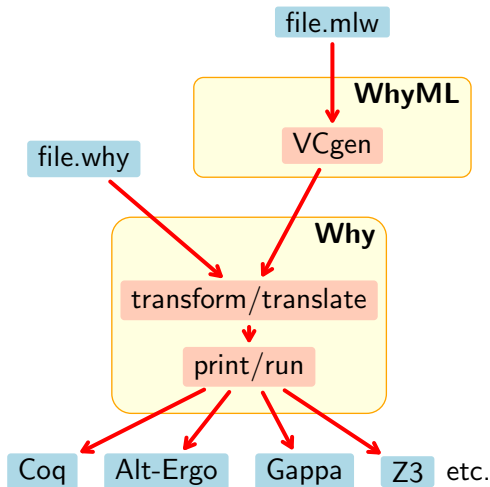
given two binary trees,
do they contain the same elements when traversed in order?



Why3 Tutorial: Sparse Arrays

from VACID-0 (Rustan Leino and Michał Moskal)

Why3 – Where Programs Meet Provers



A Little Bit of History

Why started 10 years ago

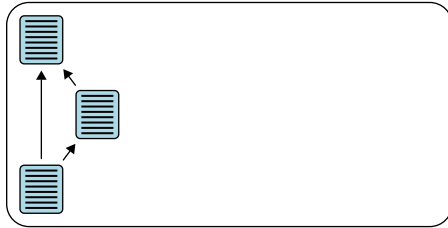
initially, as a Hoare-logic language, with Coq/PVS backends

evolutions

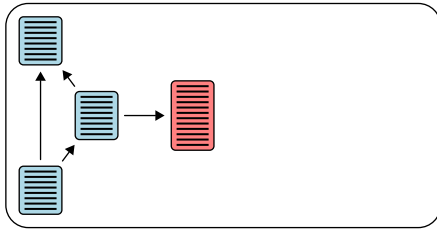
- ▶ logic: polymorphism, algebraic datatypes
- ▶ programming language: polymorphism, exceptions
- ▶ backend: more and more provers, more and more code

complete rewrite, started one year ago: **Why3**

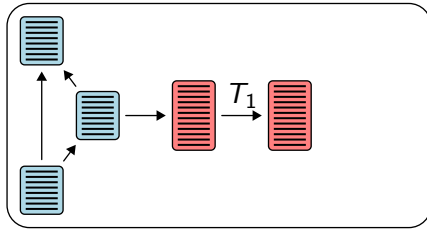
Why3: Architecture



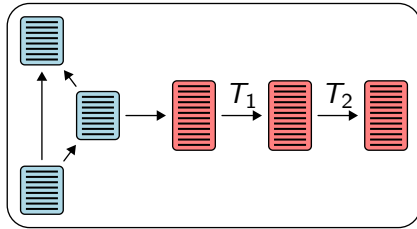
Why3: Architecture



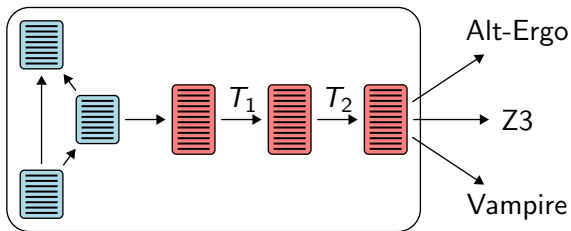
Why3: Architecture



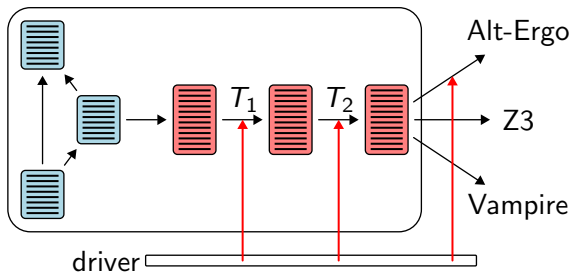
Why3: Architecture



Why3: Architecture



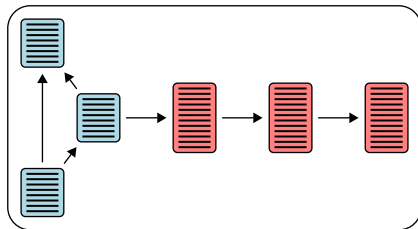
Why3: Architecture



Why3: Ocaml API and Plugins

Your code

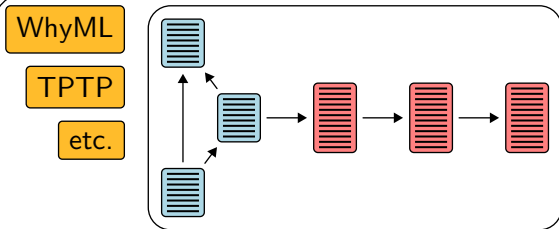
Why3 API



Why3: Ocaml API and Plugins

Your code

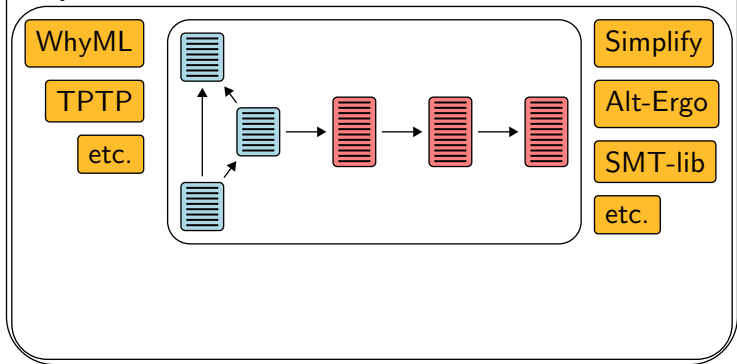
Why3 API



Why3: Ocaml API and Plugins

Your code

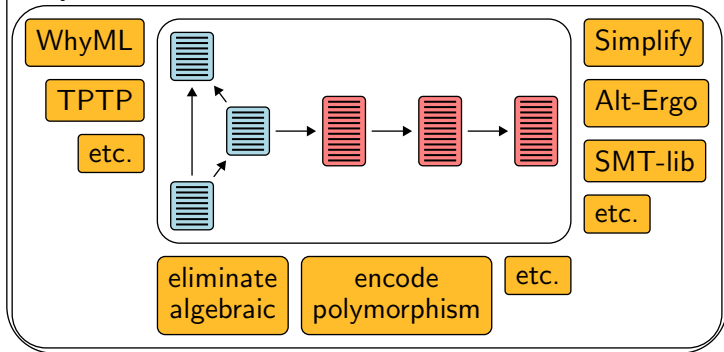
Why3 API



Why3: Ocaml API and Plugins

Your code

Why3 API



Why3: Logic

first-order polymorphic logic

- ▶ (mutually) (recursive) algebraic types
- ▶ (mutually) (recursive) functions and predicates
- ▶ (mutually) inductive predicates
- ▶ axioms / lemmas / goals

organized in theories

- ▶ a theory may **use** another theory (sharing)
- ▶ a theory may **clone** another theory (copy + substitution)

Demo: Einstein's Puzzle

simple logical puzzle formalized by Stéphane Lescuyer

highlights

- ▶ enumeration types
- ▶ cloning of theories

Why3: A Programming Language

WhyML

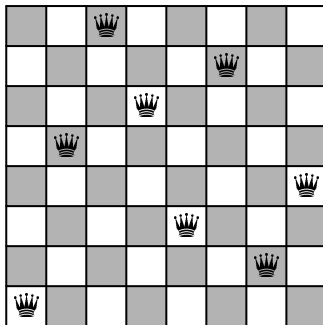
- ▶ first-order ML
- ▶ embeds logical terms
- ▶ annotations (pre/post/assert/loop invariants)
- ▶ WP calculus
- ▶ a notion of **modules**, analogous to theories

currently used

- ▶ as a VCG intermediate language
- ▶ to verify small but challenging programs (VSTTE'10, VACID-0, etc.)

Demo: N-Queens

classical combinatorial problem



highlights

- ▶ recursive functions
- ▶ exceptions
- ▶ why3ide

Perspectives

Why3

- ▶ higher-order logic
- ▶ Coq plugin to call external provers

WhyML

- ▶ ghost code
- ▶ Ocaml code extraction
- ▶ verified Ocaml libraries
- ▶ data invariants
- ▶ `clone` for modules
- ▶ higher-order programs