



Eisbach

A proof method language for Isabelle

Daniel Matichuk | Research Engineer/PhD Student

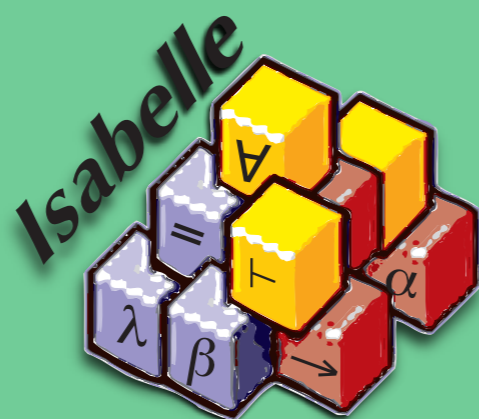
October 2015

www.csiro.au



Isabelle Concepts

Isar, Proof Methods, and ML



Isabelle Stack



jEdit (Scala)



Isar



Isabelle/ML



Poly/ML



theorem *Knaster-Tarski'*:

assumes *mono[intro!]*: $\bigwedge x y. x \leq y \implies f x \leq f y$

shows $f (\bigcap \{x. f x \leq x\}) = \bigcap (\{x. f x \leq x\})$ (**is** $f ?a = ?a$)

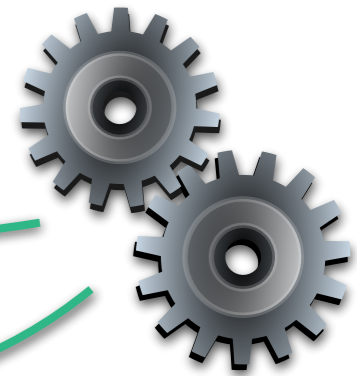
proof –

have *: $f ?a \leq ?a$ **by** (*clarsimp, rule order.trans, fastforce*)

also have $?a \leq f ?a$ **by** (*fastforce intro!: **)

finally show $f ?a = ?a$.

qed



Proof Methods



Method Expression

have *: $f ?a \leq ?a$ **by** (*clarsimp*, *rule order.trans*, *fastforce*)

Goal

Method

Combinator

also have $?a \leq f ?a$ **by** (*fastforce intro!*: *)

Proof Methods

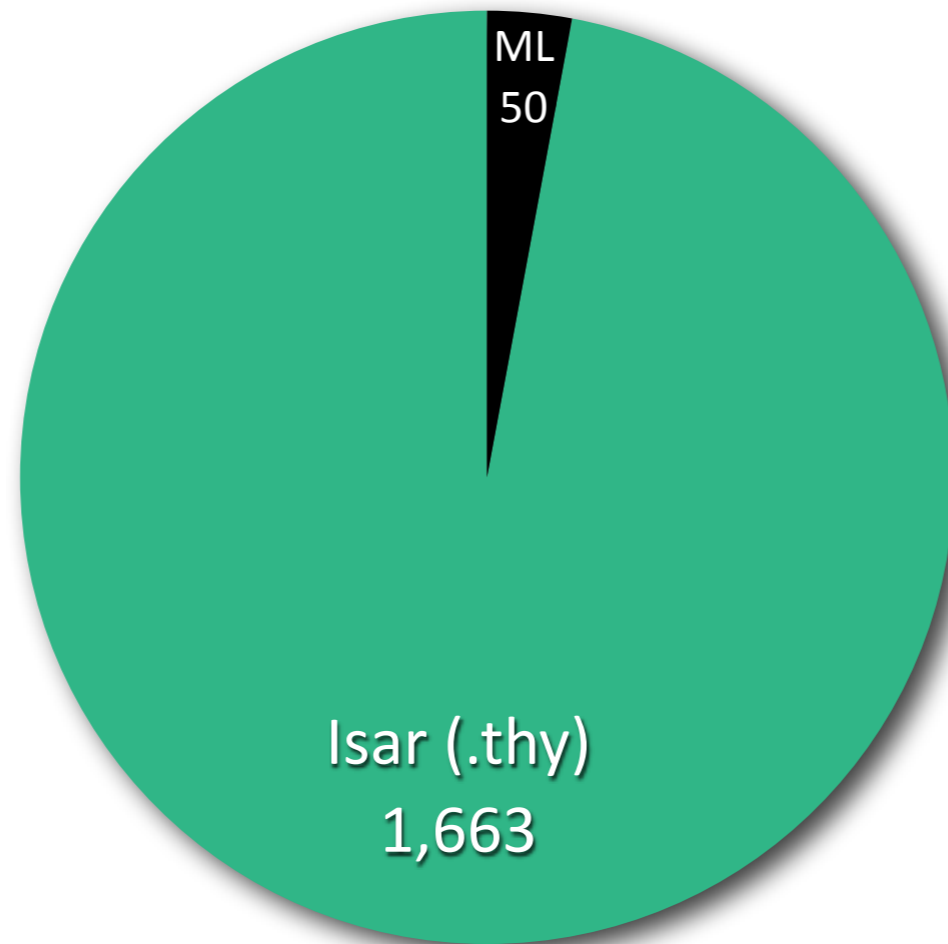


- Syntactic layer for *tactics*
 - LCF-style reasoning to guarantees soundness
- Perform arbitrary (potentially unsafe) transformations
 - e.g. **rule** FalseE
- Extensible through *declaration attributes*
 - e.g. **declare** my_fact[*intro!*]
- Combinators used to make *method expressions*
 - e.g. `((subst foo | (rule baz; simp?))+)[1]`
- Implemented in Isabelle/ML
 - requires knowledge of Isabelle's implementation
 - often break with API changes

Isar vs. ML



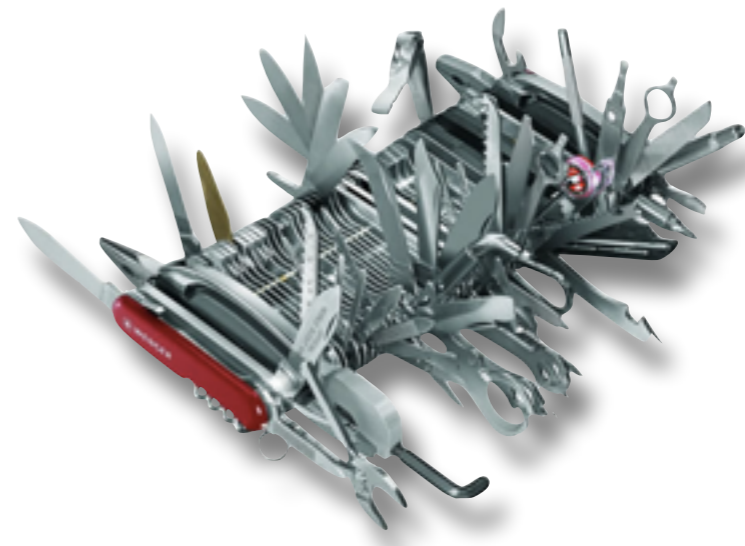
Files in Isabelle's AFP



seL4

Our experience

- Full functional correctness proof
 - Open source proof and code
 - <http://seL4.systems> for more info
- Isabelle proof methods developed
 - wp/wpc
 - vcg for monadic hoare logic
 - sep-cancel, sep_solve ...
 - automating separation logic
- Proof Engineers want more!



Eisbach

Easy Custom Proof Methods





Eisbach

Isar

Demo

Tactic languages are not new



- Ltac
 - Untyped High-level tactic language for Coq
 - Goal matching, iteration, recursion

- VeriML
 - Dependently typed tactic language
 - Provides strong static guarantees

- Mtac
 - Typed tactic language for Coq
 - Leverages built-in Coq notion of computation
 - Strong static guarantees

What distinguishes Eisbach?



- Extensive backtracking support
 - In Isabelle's combinators and match method
- Named theorems
 - Efficient and convenient databases of facts
- Powerful pattern matching
 - Using Isabelle's unifier
- Extensible
 - Language extensions can be built as proof methods (in ML)
 - e.g. match is simply another proof method



Thank You!

SSRG

Daniel Matichuk

Research Engineer/PhD Student

e daniel.matichuk@nicta.com.au

w <http://ssrg.nicta.com.au/people/?cn=Daniel+Matichuk>

www.csiro.au

