Mechanized foundations of finite group theory

The work of the *Mathematical Components* team at Microsoft Research and INRIA.

MICROSOFT RESEARCH
INRIA
JOINT CENTRE

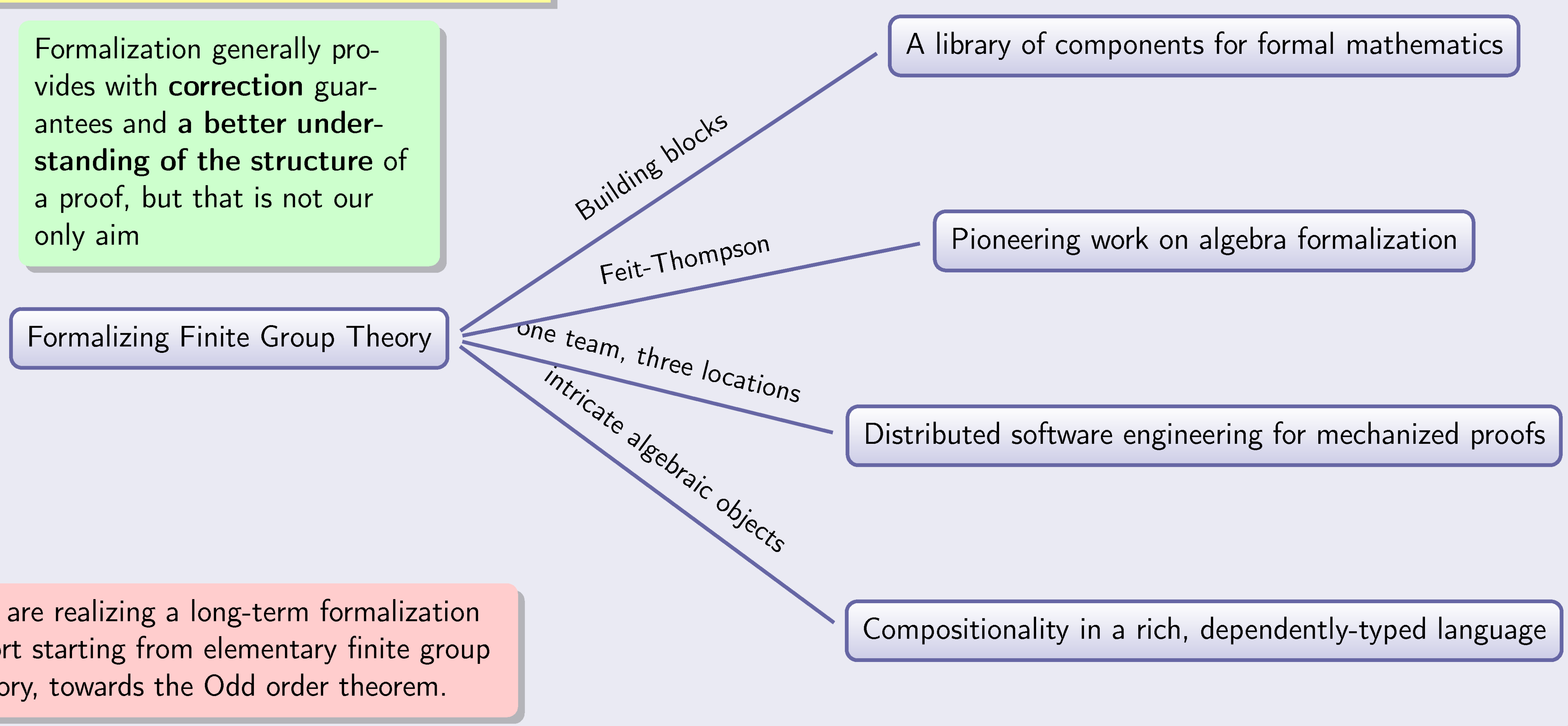
François Garillot
francois.garillot@inria.fr

INRIA - Microsoft Research Joint Lab
91890 Orsay Cedex
France

Microsoft Research Summer School MSR Cambridge July 2008

Motivation

Why formalizing finite group theory ?



Formalization generally provides with **correction** guarantees and **a better understanding of the structure** of a proof, but that is not our only aim

Formalizing Finite Group Theory
Building blocks
A library of components for formal mathematics
Feit-Thompson
Pioneering work on algebra formalization
one team, three locations
Distributed software engineering for mechanized proofs
intricate algebraic objects
Compositionality in a rich, dependently-typed language

We are realizing a long-term formalization effort starting from elementary finite group theory, towards the Odd order theorem.

The Feit-Thompson Theorem

Group theory ≡ the study of reversible composition laws.

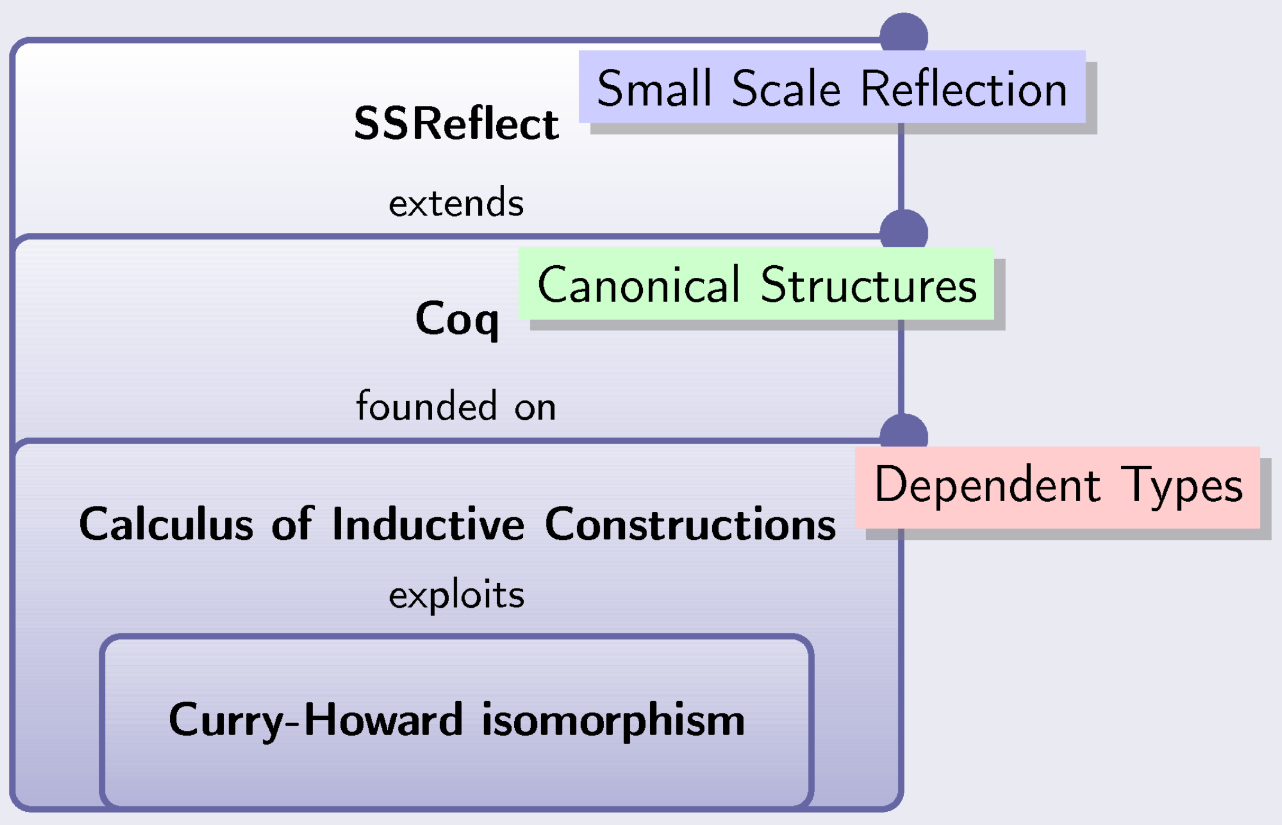
A finite group can be decomposed in *simple groups*.

Those simple groups can all be exactly described in a classification, achived in 1983. It is **widely recognized as one of the greatest achievements of twentieth century mathematics**. The proof of this *enormous theorem* is disseminated in hundreds of journal articles. The foundational breakthrough of this classification came in 1962, when Feit & Thompson showed that

Every finite group of odd order is solvable.

The proof is **255 pages long**, and **one of the reasons J.G. Thompson has been awarded the Abel prize in 2008**.

Tools and Theories

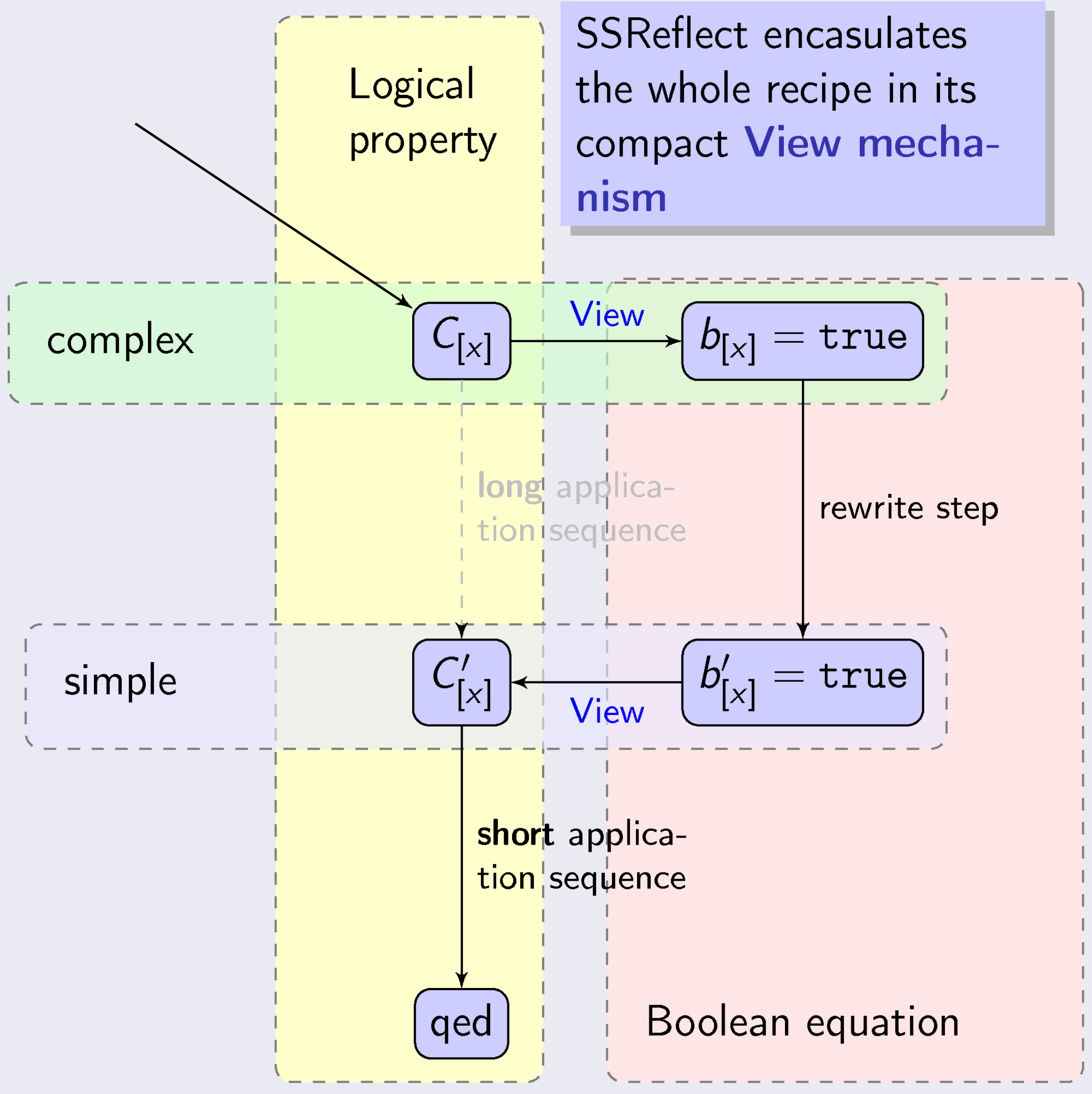


Small Scale Reflection

Recipe.
To prove a goal that uses a predicate P.

- Implement in Coq a partial decision procedure for P that **reflects** the truth of P with an algorithm returning booleans.
- Prove that P holds if and only if the procedure returns `true`.
- Prove individual instances by applying that soundness theorem with reflexivity proofs.

SSReflect encasulates the whole recipe in its compact **View mechanism**



Logical property
complex
$C_{[x]}$
View
$b_{[x]} = \texttt{true}$
long application sequence
rewrite step
simple
$C'_{[x]}$
View
$b'_{[x]} = \texttt{true}$
short application sequence
qed
Boolean equation

Canonical Structures

Type classes are essentially implicitly passed dictionaries.

In Coq, they correspond to Canonical Structures and become implicitly passed proofs., providing us with "*the shorthand that makes mathematics usable*" (Bourbaki).

```
 Structure monoid : Type := Monoid {
   sort :> Type;
   add : sort -> sort -> sort;
   unit: sort;
 }

Variable (mT: monoid A).

 Fixpoint sum (s:list (mT)) :=
   match s with
   | h::t -> add h (sum t)
   | _ -> unit
   end;;
```

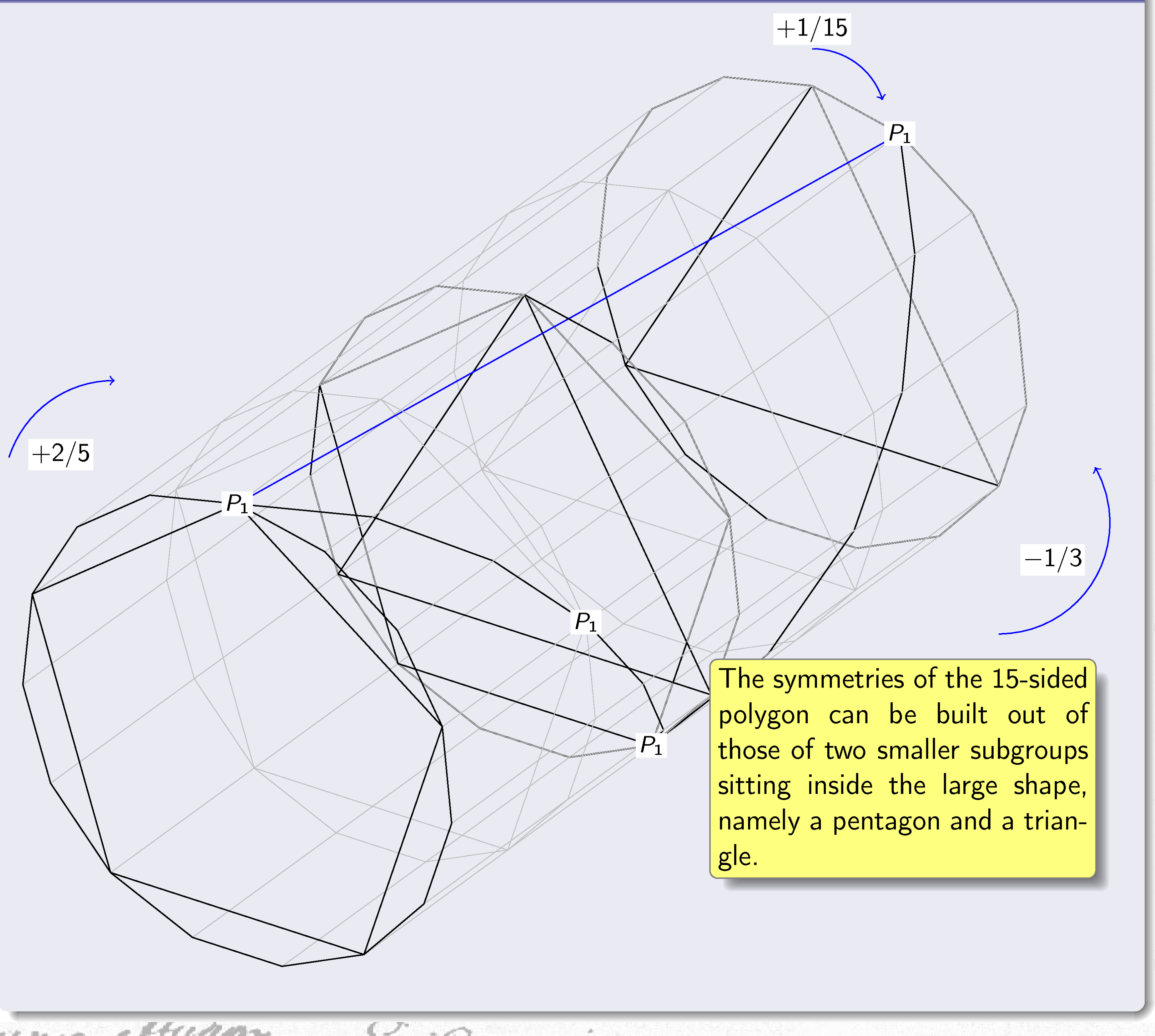
These structures allow-us to thin-slice group properties, offering many levels of abstraction for our properties. They form **narrowly-focused definitions**, on which we can **interface** our developments.

SSReflect

New release 1.1 !

- A renewed tactic shell for Coq that accelerates proof development using Small Scale Reflection
- Property bookkeeping is eliminated or shortened;
- A set of reusable **components** that structures proofs on finite domains;
- try it:
  `http://www.msr-inria.inria.fr/`

Group decomposition example : the pentadecagon



The symmetries of the 15-sided polygon can be built out of those of two smaller subgroups sitting inside the large shape, namely a pentagon and a triangle.

Results obtained, future steps

Most group formalizations stop at the Lagrange theorem. We have formalised:

- the Sylow theor
- the Cayley-Ham
- the Frobenius-C
- the simplicity of the alternating group
- the Schur-Zassenhaus theorem
- the Jordan-Hölder theorem
- . . . and counting

We already have one of the most advanced formalizations of finite algebra !

We now expect:

- to continue using programming language constructs to express theory (notations, phantom types, etc . . .)
- to find a way to quickly relate properties on an object to those of one of its isomorphic images, doing **property transfer** on a well-behaved mapping

and pursue our development of group theory!

INRIA
Microsoft Research