

# Integrating Readable Proofs and Modules

Clemens Ballarin



Technische Universität München

# Overview

- Isar: Isabelle's language for readable proofs
- Locales: the module system
- Their combination
- NB: Isabelle's Meta-Logic is an intuitionistic fragment of HOL

# A Proof

**Theorem** complete\_lattice(| carrier = { $H.H \leq G$ }, le =  $\leq$  |)

*Proof.*

...

Every non-empty set  $\mathcal{A}$  of subgroups of  $G$  has the greatest lower bound  $\bigcap \mathcal{A}$ .

...

Observe that  $\bigcap \mathcal{A} \leq G$ .

...

In order to show that  $\bigcap \mathcal{A}$  is a lower bound of  $\mathcal{A}$ , let  $H \in \mathcal{A}$ . Show  $\bigcap \mathcal{A} \leq H$ .

$\bigcap \mathcal{A} \subseteq H$  follows from  $H \in \mathcal{A}$ .

$\bigcap \mathcal{A}$  is closed wrt. operations of  $G$ , because it is a subgroup of  $G$ .

...

# A Proof

Theorem complete\_lattice() New objects of discourse

*Proof.*

...

Every non-empty set  $\mathcal{A}$  of subgroups of  $G$  has the greatest lower bound  $\bigcap \mathcal{A}$ .

...

Observe that  $\bigcap \mathcal{A} \leq G$ .

...

In order to show that  $\bigcap \mathcal{A}$  is a lower bound of  $\mathcal{A}$ , let  $H \in \mathcal{A}$ . Show  $\bigcap \mathcal{A} \leq H$ .

$\bigcap \mathcal{A} \subseteq H$  follows from  $H \in \mathcal{A}$ .

$\bigcap \mathcal{A}$  is closed wrt. operations of  $G$ , because it is a subgroup of  $G$ .

...

# A Proof

**Theorem** complete\_lattice()

**Local assumptions**

*Proof.*

...

Every non-empty set  $\mathcal{A}$  of subgroups of  $G$  has the greatest lower bound  $\bigcap \mathcal{A}$ .

...

Observe that  $\bigcap \mathcal{A} \leq G$ .

...

In order to show that  $\bigcap \mathcal{A}$  is a lower bound of  $\mathcal{A}$ , let  $H \in \mathcal{A}$ .

Show  $\bigcap \mathcal{A} \leq H$ .

$\bigcap \mathcal{A} \subseteq H$  follows from  $H \in \mathcal{A}$ .

$\bigcap \mathcal{A}$  is closed wrt. operations of  $G$ , because it is a subgroup of  $G$ .

...

# A Proof

**Theorem** complete\_lattice()

**Intermediate goal**

*Proof.*

...

Every non-empty set  $\mathcal{A}$  of subgroups of  $G$  has the greatest lower bound  $\bigcap \mathcal{A}$ .

...

Observe that  $\bigcap \mathcal{A} \leq G$ .

...

In order to show that  $\bigcap \mathcal{A}$  is a lower bound of  $\mathcal{A}$ , let  $H \in \mathcal{A}$ . Show  $\bigcap \mathcal{A} \leq H$ .

$\bigcap \mathcal{A} \subseteq H$  follows from  $H \in \mathcal{A}$ .

$\bigcap \mathcal{A}$  is closed wrt. operations of  $G$ , because it is a subgroup of  $G$ .

...

# A Proof

Theorem complete\_lattice()

Subgoal solving  
part of outer goal



*Proof.*

...

Every non-empty set  $\mathcal{A}$  of subgroups of  $G$  has the greatest lower bound  $\bigcap \mathcal{A}$ .

...

Observe that  $\bigcap \mathcal{A} \leq G$ .

...

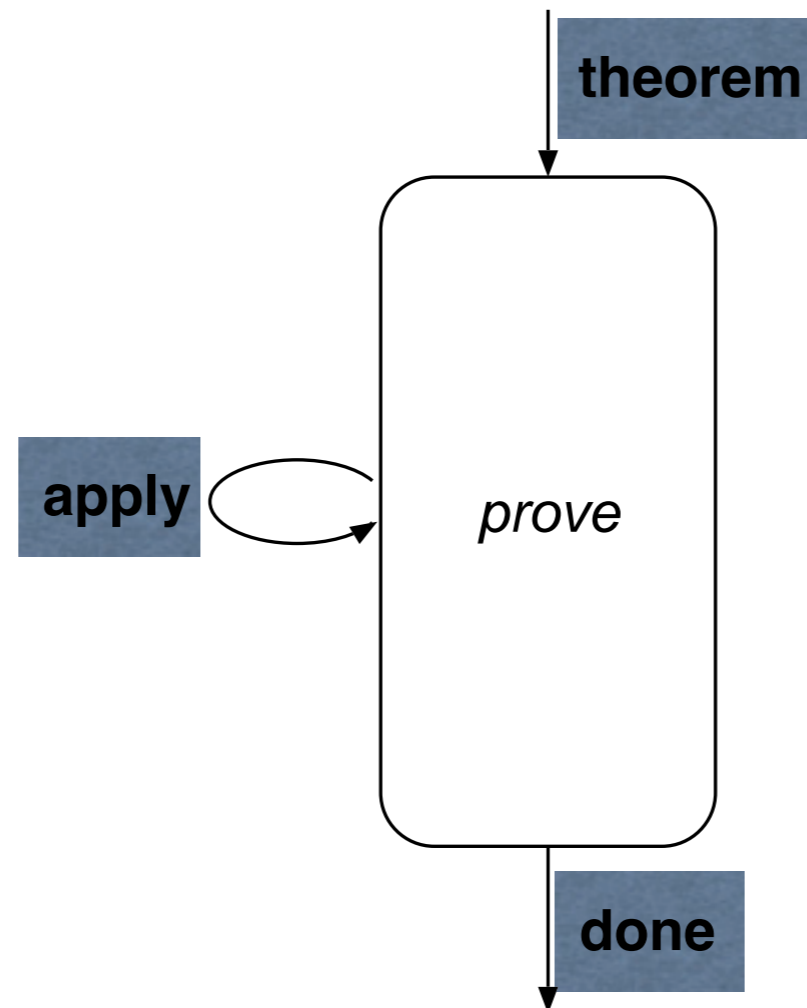
In order to show that  $\bigcap \mathcal{A}$  is a lower bound of  $\mathcal{A}$ , let  $H \in \mathcal{A}$ .  
Show  $\bigcap \mathcal{A} \leq H$ .

$\bigcap \mathcal{A} \subseteq H$  follows from  $H \in \mathcal{A}$ .

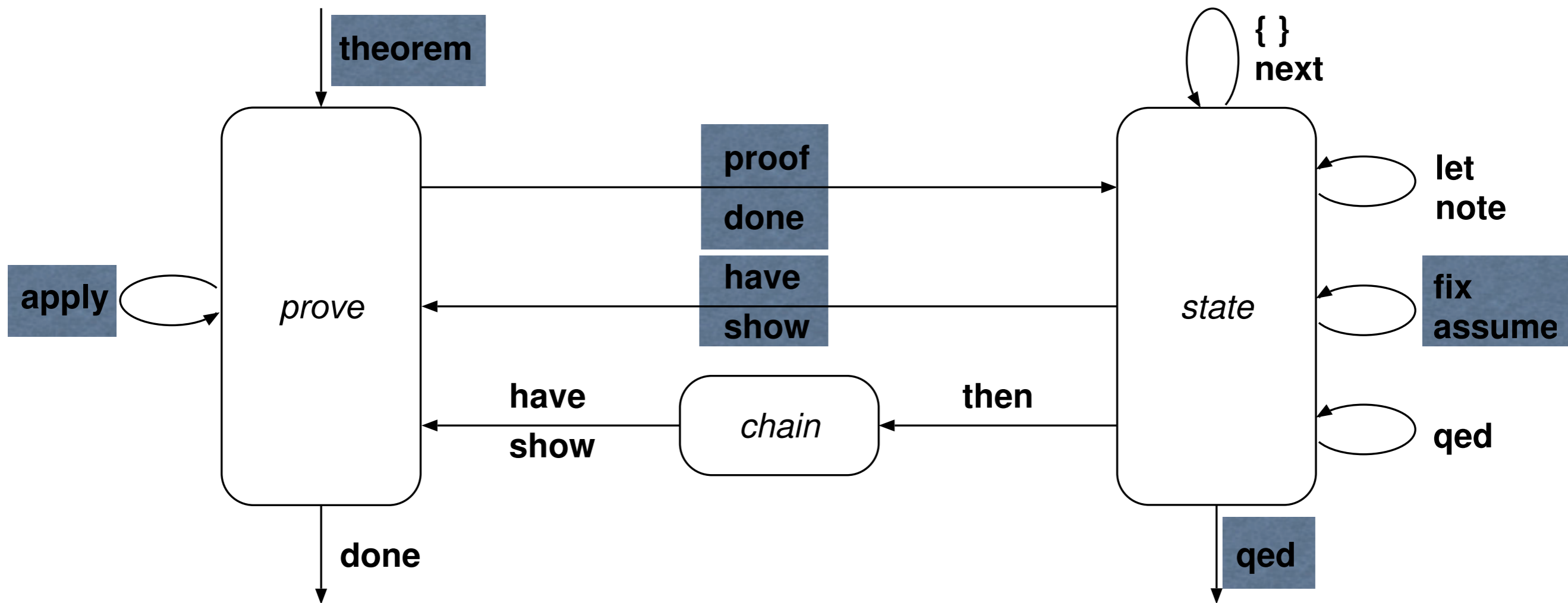
$\bigcap \mathcal{A}$  is closed wrt. operations of  $G$ , because  
it is a subgroup of  $G$ .

...

# Tactic Proof Processing



# Isar Proof Processing



# Isar Proof

**theorem**

complete\_lattice ( $\mid$  carrier =  $\{H. \text{subgroup } H \ G\}$ , le = subgrp  $G$ )  
(is complete\_lattice ? $L$ )

**proof** (rule partial\_order.complete\_lattice\_criterion1)

...

**fix**  $A$

**assume** L:  $A \subseteq \text{carrier } ?L$  **and** non\_empty:  $A \neq \{\}$

**then have** Int\_subgroup: subgroup  $(\bigcap A) \ G$

...

**have** greatest ? $L$   $(\bigcap A)$  (Lower ? $L$   $A$ )

**proof** (rule greatest\_LowerI)

**fix**  $H$

**assume** H:  $H \in A$

...

**show** le ? $L$   $(\bigcap A) \ H$

...

# Isar Proof

theorem

complete\_lattice ( $\mid$  carrier :  
(is complete\_lattice ? $L$ )

Fixed variables

proof (rule partial\_order.complete\_lattice\_criterion1)

...

**fix**  $A$

assume  $L$ :  $A \subseteq \text{carrier } ?L$  and non\_empty:  $A \neq \{\}$

then have Int\_subgroup: subgroup  $(\bigcap A) G$

...

have greatest ? $L$   $(\bigcap A)$  (Lower ? $L$   $A$ )

proof (rule greatest\_LowerI)

**fix**  $H$

assume  $H$ :  $H \in A$

...

show le ? $L$   $(\bigcap A) H$

...

# Isar Proof

**theorem**

complete\_lattice ( $\mid$  carrier =  
(is complete\_lattice ? $L$ )

**Local assumptions**

**proof** (rule partial\_order.complete\_lattice, ...

...

**fix**  $A$

**assume**  $L: A \subseteq \text{carrier } ?L$  and  $\text{non\_empty}: A \neq \{\}$

**then have** Int\_subgroup: subgroup  $(\bigcap A) G$

...

**have** greatest ? $L$   $(\bigcap A)$  (Lower ? $L$   $A$ )

**proof** (rule greatest\_LowerI)

**fix**  $H$

**assume**  $H: H \in A$

...

**show** le ? $L$   $(\bigcap A) H$

...

# Isar Proof

**theorem**

complete\_lattice ( $\mid$  carrier = {  
(is complete\_lattice ? $L$ )

**proof** (rule partial\_order.comp

...

**fix**  $A$

**assume**  $L$ :  $A \subseteq \text{carrier } ?L$  and non\_empty:  $A \neq \{\}$

**then have** Int\_subgroup: subgroup  $(\bigcap A) G$

...

**have** greatest ? $L$   $(\bigcap A)$  (Lower ? $L$   $A$ )

**proof** (rule greatest\_LowerI)

**fix**  $H$

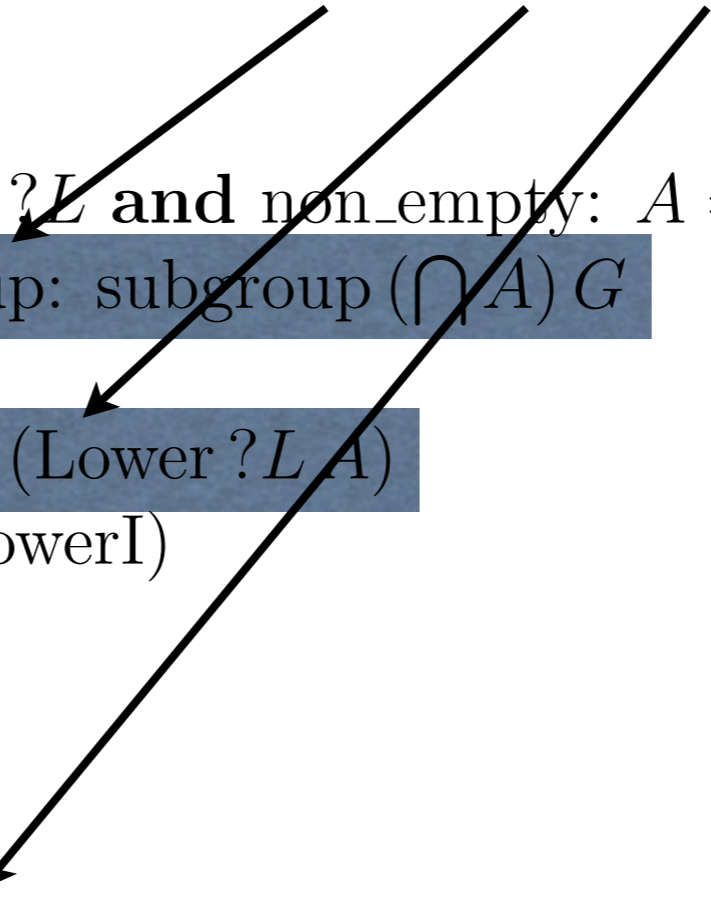
**assume**  $H$ :  $H \in A$

...

**show** le ? $L$   $(\bigcap A) H$

...

**Local theorems**



# Contexts

- Isar maintains stack of contexts.
- Defined by locally fixed variables and assumptions.
- Local lemmas are relative to context.
- Goals are exported to surrounding context.

# Locales

- Specifications, represented as contexts.
- Locales are named.
- Contain theorems relative to context.
- Structural inheritance (rename and merge).

# Subgroups as Locales

**locale** submagma = var  $H$  + struct  $G$  +  
 **assumes** subset [intro, simp]:  $H \subseteq \text{carrier } G$   
 **and** m\_closed [intro, simp]:  $\llbracket x \in H; y \in H \rrbracket \implies x \otimes y \in H$

**locale** submonoid = submagma +  
 **assumes** one\_closed [intro, simp]:  $1 \in H$

**locale** subgroup = submonoid +  
 **assumes** m\_inv\_closed [intro, simp]:  $x \in H \implies \text{inv } x \in H$

# Subgroups as Locales

Import sections

**locale** submagma =  $\text{var } H + \text{struct } G +$   
**assumes** subset [intro, simp]:  $H \subseteq \text{carrier } G$   
**and** m\_closed [intro, simp]:  $\llbracket x \in H; y \in H \rrbracket \implies x \otimes y \in H$

**locale** submonoid = submagma +  
**assumes** one\_closed [intro, simp]:  $1 \in H$

**locale** subgroup = submonoid +  
**assumes** m\_inv\_closed [intro, simp]:  $x \in H \implies \text{inv } x \in H$

# Subgroups as Locales

Specifications in locale  
bodies

**locale** submagma = var  $H$  + struct  $G$  +  
**assumes** subset [intro, simp]:  $H \subseteq \text{carrier } G$   
**and** m\_closed [intro, simp]:  $\llbracket x \in H; y \in H \rrbracket \implies x \otimes y \in H$

**locale** submonoid = submagma +  
**assumes** one\_closed [intro, simp]:  $1 \in H$

**locale** subgroup = submonoid +  
**assumes** m\_inv\_closed [intro, simp]:  $x \in H \implies \text{inv } x \in H$

# Subgroups as Locales

Hints for automation

**locale** submagma = var  $H$  + struct  $G$  +  
 **assumes** subset [intro, simp]:  $H \subseteq \text{carrier } G$   
 **and** m\_closed [intro, simp]:  $\llbracket x \in H; y \in H \rrbracket \implies x \otimes y \in H$

**locale** submonoid = submagma +  
 **assumes** one\_closed [intro, simp]:  $1 \in H$

**locale** subgroup = submonoid +  
 **assumes** m\_inv\_closed [intro, simp]:  $x \in H \implies \text{inv } x \in H$

# Adding Theorems

**lemma** (in subgroup) is\_submagma [intro, simp]:  
submonoid  $H G$   
**by** (rule submonoid.intro)

- Specifies context
- Lemma is added to the locale
- Also exported version available

# The Proof Continued

**show**  $le ?L (\cap A) H$

**proof** (simp, rule\_tac subgrpI)

...

**show**  $\wedge xy. \llbracket x \in \cap A; y \in \cap A \rrbracket \implies x \otimes y \in \cap A$

**apply** (rule submagma.m\_closed)

**apply** (rule subgroup.is\_submagma)

**apply** (rule Int\_subgroup)

**apply** assumption

**apply** assumption

**done**

...

# Critique

- Instantiation of locale parameters manual, on a per-lemma basis.
- Manual reasoning about locale hierarchy.
- Setup for automation not inherited.

# Locale Instantiation

- Like theory interpretation, but for locale contexts, not theories.

# Locale Instantiation

Instantiate locale  $\mathcal{L}$  in proof context  $\mathcal{C}$ .

$$\Phi : \text{Parm}_{\mathcal{L}} \longrightarrow \text{Term}_{\mathcal{C}}$$

$$\overline{\Phi} : \text{Prop}_{\mathcal{L}} \longrightarrow \text{Prop}_{\mathcal{C}}$$

If  $\mathcal{C} \Longrightarrow \overline{\Phi}(A)$  for all assumptions  $A$  of locale  $\mathcal{L}$ , then  
 $\mathcal{C} \Longrightarrow \overline{\Phi}(T)$  for all theorems  $T$  of  $\mathcal{L}$ .

# The Proof Improved

**from** Int\_subgroup **instantiate** Int: subgroup

...

**show**  $1 \in ?L(\bigcap A) H$

**proof** (simp, rule\_tac subgrpI)

...

**show**  $\bigwedge x y. [x \in \bigcap A; y \in \bigcap A] \implies x \otimes y \in \bigcap A ..$

**next**

**show**  $1 \in \bigcap A ..$

**next**

**show**  $\bigwedge x. x \in \bigcap A \implies \text{inv } x \in \bigcap A ..$

**qed**

# Conclusion

- Instantiation simplifies proofs involving modules.
- Conceptually simple since both Isar and Locales are based on contexts.
- Implementation not so simple ...
- Concrete syntax (e.g. infix) in instantiated propositions?