

Comparison of two theorem provers: Isabelle & Coq

A. Yushkovskiy S. Tripakis

Department of Computer Science
School of Science
Aalto University

CS-E4000: Seminar in Computer Science
autumn 2017

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Outline

Foundations of Formal Approach

- A Formal System

- Properties of a Formal System

- Classical and Intuitionistic Logics

Two Theorem Provers

- Isabelle

- Coq

Comparison of the Theorem Provers

- Common Features

- Major Differences

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Elements of a Formal System

- ▶ A formula (judgement, statement) $\phi \in \Phi$:

$$\begin{aligned}\phi &:= p \mid q \mid \dots \\ &\mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \neg \phi_1 \mid \phi_1 \rightarrow \phi_2 \\ &\mid \textit{true} \mid \textit{false} \\ &\mid \dots\end{aligned}$$

- ▶ Propositional variables: — $p, q, \dots \in V$
- ▶ An axiom — $\phi_A \in A$
- ▶ An inference rule τ — a transition function $\tau : \Phi \rightarrow \Phi$
- ▶ A formula ϕ provable from Φ — $\Phi \vdash \phi$
- ▶ A tautology \top — $\vdash \phi$
- ▶ A contradiction \perp — $\vdash \neg \phi$

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle
Coq

Comparison of the
Theorem Provers

Common Features
Major Differences

Summary

Definition of the Formal System

A *formal system* is a quadruple $\Gamma = \langle A, V, \Omega, R \rangle$, where

- ▶ A – set of axioms
- ▶ V – set of propositional variables
- ▶ Ω – set of logical operators
- ▶ R – set of inference rules

A *formal proof* of the formula ϕ is a finite sequence of judgements

$$\psi_1 \xrightarrow{\tau_1} \psi_2 \xrightarrow{\tau_2} \dots \xrightarrow{\tau_n} \psi_n$$

where each ψ_i is either an axiom ϕ_{A_i} , or a formula inferred from the set of previously derived formulas according the rules of inference.

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Properties of a Formal System

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

A formal system Γ is called:

- ▶ *consistent*, if $\nexists \phi \in \Gamma : \Gamma \vdash \phi \wedge \Gamma \vdash \neg \phi \Leftrightarrow \Gamma \not\vdash \perp$;
- ▶ *complete*, if $\forall \phi \in U : A \vdash \phi \vee A \vdash \neg \phi$;
- ▶ *independent*, if $\nexists a \in A : A \vdash a$.

Classical Logic

example: The Hilbert System

Set of axioms:

$$A \rightarrow (B \rightarrow A) \quad (\text{A1})$$

$$(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \quad (\text{A2})$$

$$A \vee \neg A \quad (\text{EM})$$

Single inference rule (*Modus Ponens*)

$$\llbracket A, A \rightarrow B \rrbracket \longrightarrow B \quad (\text{MP})$$

Some provable tautologies:

$$\neg\neg(A \vee \neg A) \quad (\text{nnEM})$$

$$A \rightarrow \neg\neg A \quad (\text{DNi})$$

$$\neg\neg A \rightarrow A \quad (\text{DNe})$$

$$((A \rightarrow B) \rightarrow A) \implies B \quad (\text{PL})$$

$$\neg(A \wedge B) \rightarrow \neg A \vee \neg B \quad (\text{DMdi})$$

$$\neg(A \vee B) \rightarrow \neg A \wedge \neg B \quad (\text{DMci})$$

$$\neg A \wedge \neg B \rightarrow \neg(A \vee B) \quad (\text{DMce})$$

$$\neg A \vee \neg B \rightarrow \neg(A \wedge B) \quad (\text{DMde})$$

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Intuitionistic Logic

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

**Classical and Intuitionistic
Logics**

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Isabelle: First Acquaintance

- ▶ a generic proof assistant
- ▶ a successor of HOL theorem prover //TODO: cite
- ▶ created in 1986 by
 - ▶ Larry Paulson @ University of Cambridge, and
 - ▶ Tobias Nipkow @ Technische Universität München
- ▶ based on classical higher-order logic
- ▶ uses powerful functional language HOL
- ▶ has large collection of formalised theories //TODO: HOL, ZF, CCL, ...

Example 1: Definition of basic datatypes

```
datatype bool =  
  True | False
```

```
datatype nat =  
  zero ("0") | Suc nat
```

Example 2: ???

???

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

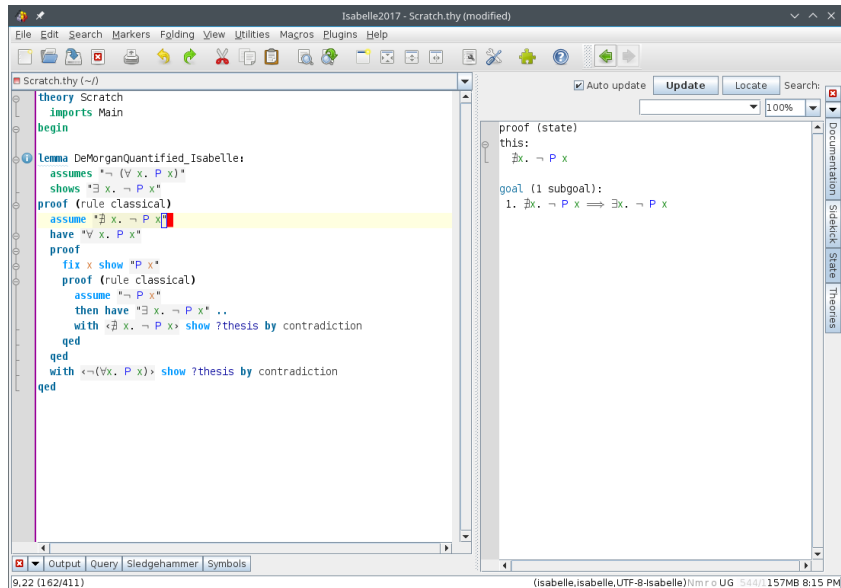
Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Isabelle: Native GUI



Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Coq: First Acquaintance

- ▶ a formal proof management system
- ▶ created at INRIA (Paris, France) in 1984
- ▶ based on Calculus of Inductive Constructions theory (an implementation of intuitionistic logic)
- ▶ uses powerful functional language `Gallina`
- ▶ has large collection of formalised theories //TODO
- ▶ widely used in software verification (proof code extraction)

Example 3: Definition of basic datatypes

```
Inductive False : Prop := .
```

```
Inductive True : Prop := I : True.
```

```
Inductive nat : Type :=  
  | O : nat  
  | S : nat -> nat.
```

Example 4: ???

???

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

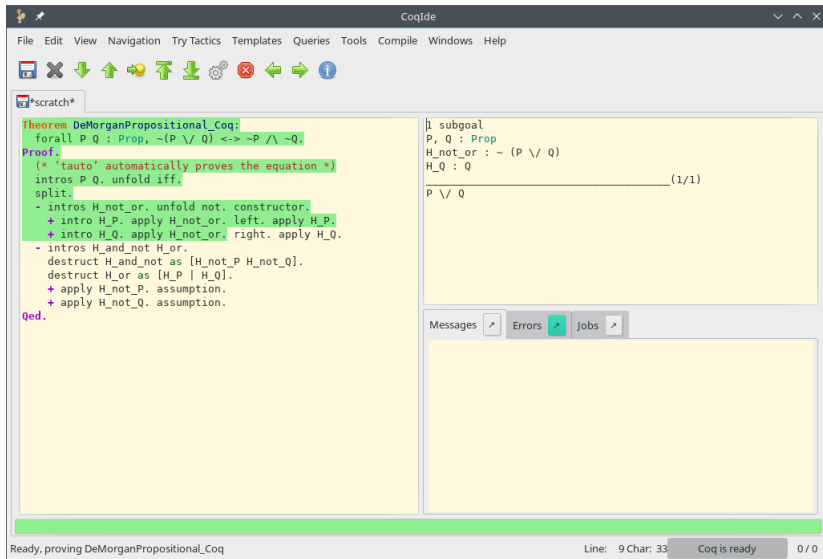
Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

The Coq theorem prover



The screenshot shows the CoqIDE interface. The top menu bar includes File, Edit, View, Navigation, Try Tactics, Templates, Queries, Tools, Compile, Windows, and Help. Below the menu is a toolbar with icons for saving, opening, undo, redo, and other editing functions. The main window is titled '*scratch*' and contains a Coq script. The script defines a theorem `DeMorganPropositional_Coq` and provides a proof. The proof uses tactics like `intros`, `split`, `destruct`, and `apply`. The right-hand pane shows the current proof state, including the subgoal, hypotheses, and the goal. The bottom status bar indicates 'Ready, proving DeMorganPropositional_Coq' and shows the current line and character position.

```
Theorem DeMorganPropositional_Coq:
  forall P Q : Prop, ~(P /\ Q) <-> ~P /\ ~Q.
Proof.
  (* 'tauto' automatically proves the equation *)
  intros P Q. unfold iff.
  split.
  - intros H_not_or. unfold not. constructor.
    + intro H_P. apply H_not_or. left. apply H_P.
    + intro H_Q. apply H_not_or. right. apply H_Q.
  - intros H_and_not H_or.
    destruct H_and_not as [H_not_P H_not_Q].
    destruct H_or as [H_P | H_Q].
    + apply H_not_P. assumption.
    + apply H_not_Q. assumption.
Qed.
```

Proof state (right pane):

```
| subgoal
P, Q : Prop
H_not_or : ~ (P /\ Q)
H_Q : Q
----- (1/1)
P /\ Q
```

Status bar: Ready, proving DeMorganPropositional_Coq Line: 9 Char: 33 Coq is ready 0 / 0

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

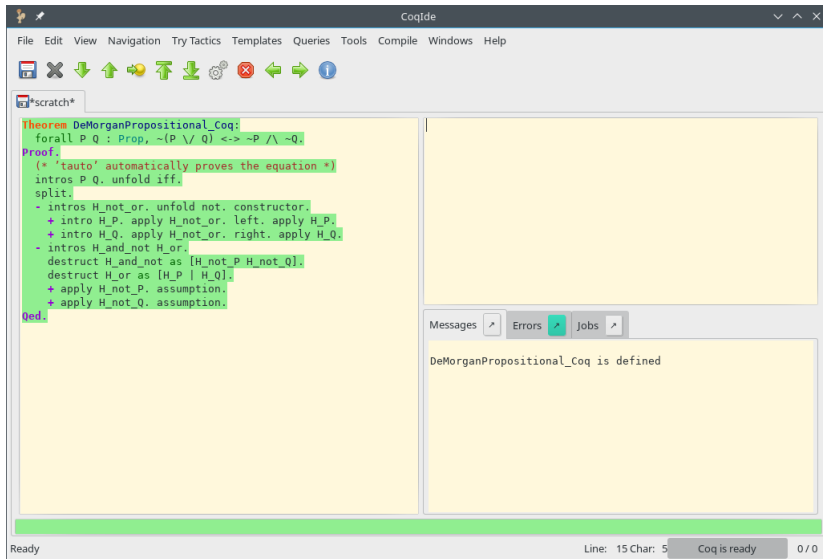
Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

The Coq theorem prover



Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary

Summary

- ▶ The **first main message** of your talk in one or two lines.
- ▶ The **second main message** of your talk in one or two lines.
- ▶ Perhaps a **third message**, but not more than that.
- ▶ Outlook
 - ▶ Something you haven't solved.
 - ▶ Something else you haven't solved.

Comparison of two
theorem provers:
Isabelle & Coq

A. Yushkovskiy,
S. Tripakis

Foundations of
Formal Approach

A Formal System

Properties of a Formal
System

Classical and Intuitionistic
Logics

Two Theorem
Provers

Isabelle

Coq

Comparison of the
Theorem Provers

Common Features

Major Differences

Summary