universität
innsbruck

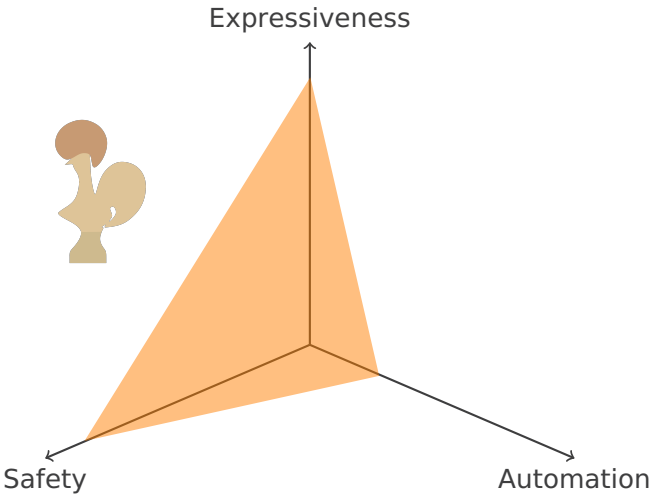
SMTCoq and decision procedures

Jamie Hochrainer

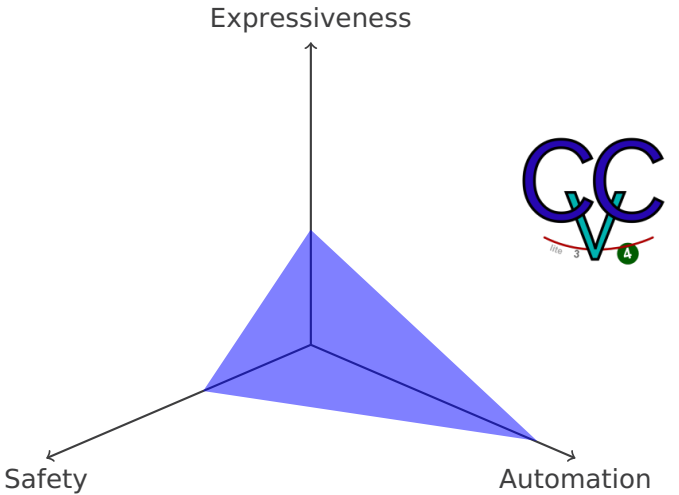
Outline

- **Motivation**
- **Introduction**
- **Architecture**
- **Coq Checker**
- **"Demo"**
- **Partial proofs with holes**
- **"Demo"**

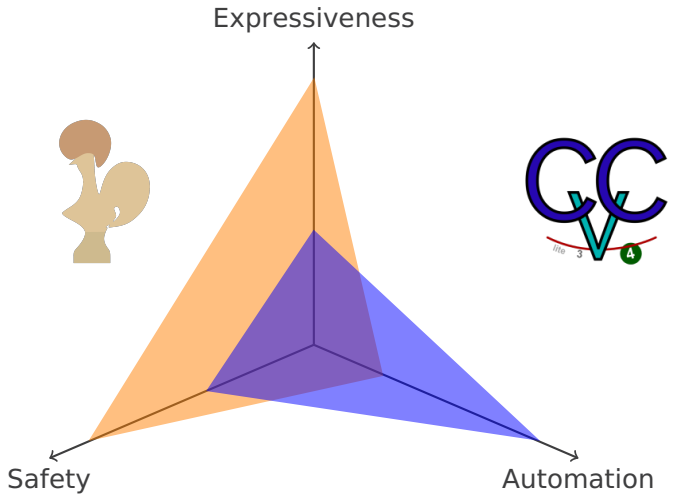
SMTCoq: Motivation



SMTCoq: Motivation



SMTCoq: Motivation



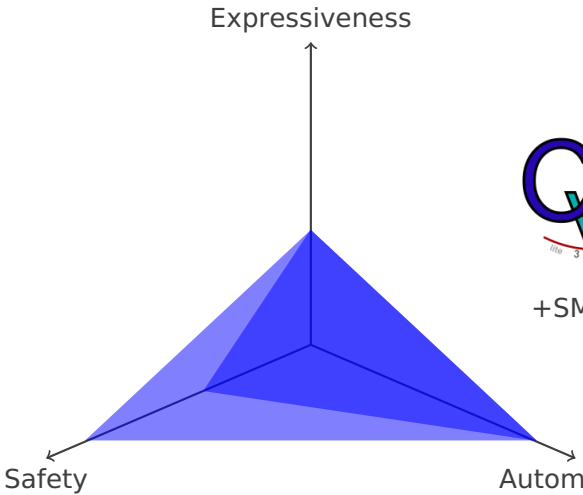
SMTCoq: Introduction

- SMTCoq: A plug-in for integrating SMT solvers into Coq

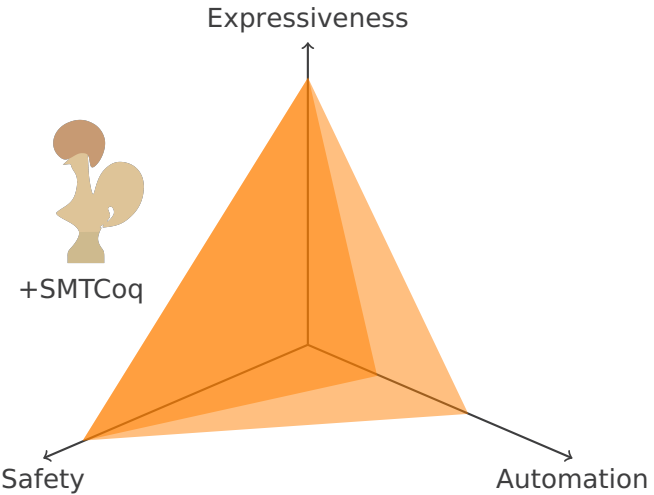
- Goal: Increase level of automation in Coq
  Increase confidence in SAT and SMT solvers

	SAT/SMT solvers	Coq
Automation	**Automatic proof**	User-guided proof
Expressiveness	First-order logic	**CIC**
Safety	Trust the whole software	**Kernel for proof checking**

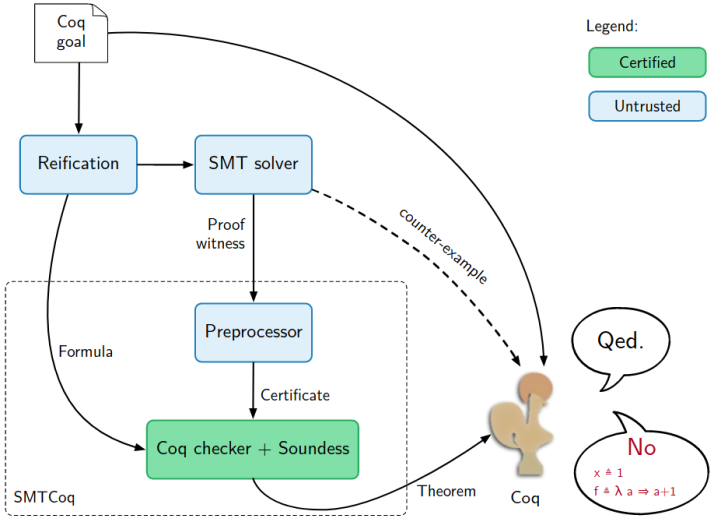
SMTCoq: Introduction



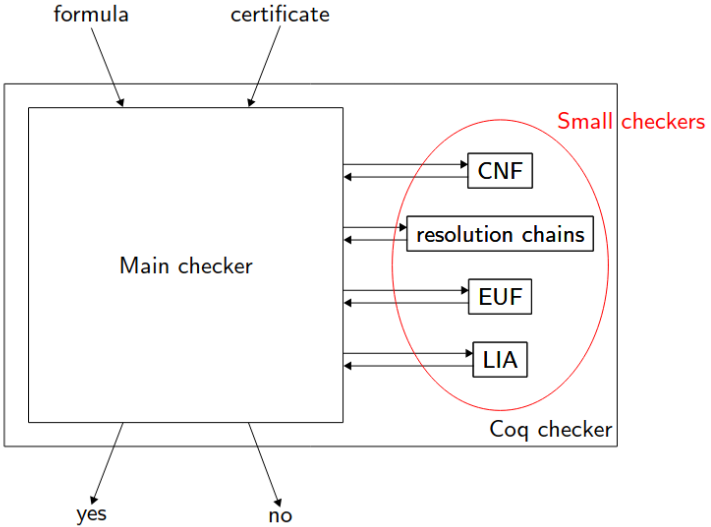
SMTCoq: Introduction



SMTCoq Architecture



Coq Checker


formula
certificate
Small checkers
CNF
resolution chains
EUF
LIA
Main checker
Coq checker
yes
no

Features

Theroy	zChaff	veriT	CVC4
Propositional logic	✓	✓	✓
EUF		✓	✓
LIA		✓	✓
Bit vectors			✓
Arrays			✓

- Add solver: write preprocessor
- Add theories: write certified small checkers

Demo

```
Section Group.
Variable G : Type.
Variable e : G.
Variable op: G -> G -> G.
Variable inv : G -> G.

Hypothesis associative:
  forall a b c : G, op a (op b c) = op (op a b) c.
Hypothesis identity:
  forall a : G, (op e a = a) /\ (op a e = a).
Hypothesis inverse:
  forall a : G, (op a (inv a) = e) /\ (op (inv a) a = e).

Lemma unique_identit e':
  (forall z, op e' z = z) -> e' = e.
Proof.
  intros pe'.
  rewrite <- (pe' e).
  destruct (identity e') as [_ H].
  now rewrite H.
(*smt (associative, identity, inverse) *)
Qed.
End Group.
```

```
No more subgoals.
```

Partial proofs with holes

- Proof certificates may contain holes.
- Introduce a cut / subgoal.
- Recognize unproven lemma, lift them to Coq subgoals.

Demo

```
Goal
 ∀ (a b: farray Z Z) (v w x y: Z)
   (r s: bitvector 4)
   (f: Z → Z)
   (g: farray Z Z → Z)
   (h: bitvector 4 → Z),
   a[x ← v] = b ∧ a[y ← w] = b →
   r = s ∧ h r = v ∧ h s = y →
   v < x + 1 ∧ v > x - 1 →
   f (h r) = f (h s) ∨ g a = g b.
Proof.
  cvc4.
  verit.
Qed.
```

```
1 subgoal, subgoal 1 (ID 10899)
  a, b : farray Z Z
  v, w, x, y : Z
  r, s : bitvector 4
  f : Z → Z
  g : farray Z Z → Z
  h : bitvector 4 → Z
  ====================================
  (a[x ← v] = b ∧ a[y ← w] = b →
   r = s ∧ h r = v ∧ h s = y →
   v < x + 1 ∧ x - 1 < v →
   f (h r) = f (h s) ∨ g a = g b)
  ↔ (a[x ← v] = b ∧ a[y ← w] = b →
     r = s ∧ h r = v ∧ h s = y →
     1 ≤ v + -1 * x ∧ 0 ≤ v + -1 * x →
     f (h r) = f (h s) ∨ g a = g b)
```

Literature

Content and images are taken from:

Ekici, B., Mebsout, A., Tinelli, C., Keller, C., Katz, G., Reynolds, A., & Barrett, C. (2017, July). SMTCoq: A plug-in for integrating SMT solvers into Coq. In International Conference on Computer Aided Verification (pp. 126-133). Springer, Cham.

Ekici, B., Katz, G., Keller, C., Mebsout, A., Reynolds, A. J., & Tinelli, C. (2016). Extending SMTCoq, a certified checker for SMT. arXiv preprint arXiv:1606.05947.

`https://smtcoq.github.io/`

Thank you for your attention!

Jamie Hochrainer