# Proving your proofs

Guillaume Bury

September 22, 2017

Université Paris Diderot; Inria; LSV, ENS Cachan

# Introduction

- Automated theorem proving
    - Usually blackboxes
    - Yes/No answer
    - Cannot verify the answer
    - Very complex algorithms and heuristics $\rightarrow$ potentially some bugs
- Proof certificates
    - Easily verifiable
    - Very detailed
    - Small trusted core which does simple verifications
    - Tedious to do by hand

# Sat Solving and Resolution Proofs

# Simplified control flow

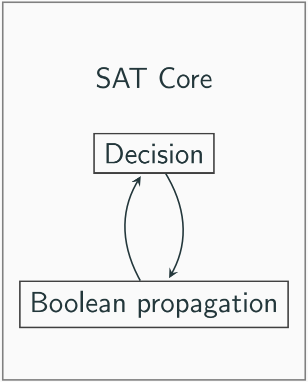



**Figure 1:** Simplified SAT Solver architecture

# SAT Solving Algorithm

- Maintain a partial propositional model
- Propagation
  - If there exists a clause $C = a \vee c_1 \vee \ldots \vee c_n$, where every $c_i \rightsquigarrow \bot$ in the current partial model, then add $a \rightsquigarrow_C \top$ to the model
  - Record the clause $C$ as the **reason** for the propagation of $a$
- Decision
  - When no propagation is possible
  - Choose an unassigned litteral $a$
  - Add $a \mapsto \top$ to the model

# SAT Solving Algorithm

- When there is a clause $C = c_1 \vee \ldots \vee c_n$, where every $c_i \mapsto \bot$, begin analyzing with current clause $C$
- Walk back the propagations/decisions from most recent
- If the currently looked at atom is:
  - Not part of the current clause, continue
  - part of the current clause, and propagated by a clause $D$, perform a resolution between the current clause and $D$:

$$\frac{C \vee p \qquad \neg p \vee D}{C \vee D}$$

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false

## SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \leadsto_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied

# SAT Solving - Example sat

- $C_1 = \neg p(a) \lor p(b)$, $C_2 = \neg p(a) \lor \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \lor p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \lor \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \lor \neg p(b)$ and $C_1 = \neg p(a) \lor p(b)$

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- New clause : $C_3 = \neg p(a)$, backtrack to before decision.

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b)$, $C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- New clause : $C_3 = \neg p(a)$, backtrack to before decision.
- Propagation: $p(a) \rightsquigarrow_{C_3} \bot$

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- New clause : $C_3 = \neg p(a)$, backtrack to before decision.
- Propagation: $p(a) \rightsquigarrow_{C_3} \bot$
- Decision: $p(b) \mapsto \top$

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- New clause : $C_3 = \neg p(a)$, backtrack to before decision.
- Propagation: $p(a) \rightsquigarrow_{C_3} \bot$
- Decision: $p(b) \mapsto \top$
- Propagation (nothing to do)

# SAT Solving - Example sat

- $C_1 = \neg p(a) \vee p(b), C_2 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Decision: $p(a) \mapsto \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- New clause : $C_3 = \neg p(a)$, backtrack to before decision.
- Propagation: $p(a) \rightsquigarrow_{C_3} \bot$
- Decision: $p(b) \mapsto \top$
- Propagation (nothing to do)
- Model Found !

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- Resolution between $T_1 = \neg p(a)$ and $C_0 = p(a)$

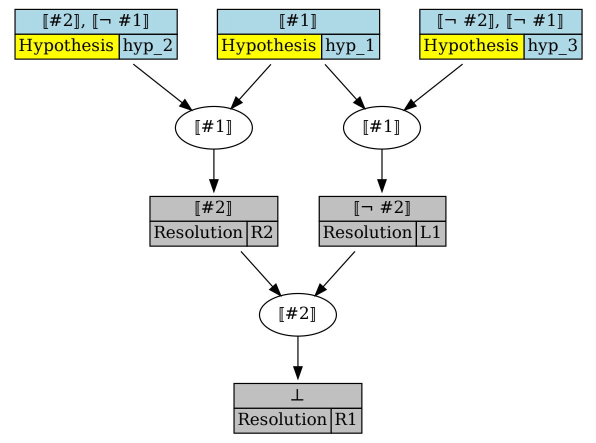
# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- Resolution between $T_1 = \neg p(a)$ and $C_0 = p(a)$
- Empty clause $C_4 = \bot$ reached

# SAT Solving - Example unsat

- $C_0 = p(a), C_1 = \neg p(a) \vee p(b), C_3 = \neg p(a) \vee \neg p(b)$
- Problem: find a model or a proof of false
- Propagation: $p(a) \mapsto_{C_0} \top$
- Propagation in $C_1 = \neg p(a) \vee p(b)$: $p(b) \rightsquigarrow_{C_1} \top$
- Conflict: $C_2 = \neg p(a) \vee \neg p(b)$ not satisfied
- Resolution between $C_2 = \neg p(a) \vee \neg p(b)$ and $C_1 = \neg p(a) \vee p(b)$
- Resolution between $T_1 = \neg p(a)$ and $C_0 = p(a)$
- Empty clause $C_4 = \bot$ reached
- Input problem is unsatisfiable

# SAT Solving - proofs



# Resolution proofs in Coq

- Disjunctions are not easy to work with
  - Ordering matters
  - Need to manually apply commutativity and associativty lemmas
- Solution: use a weak form of clauses, as implications:

$$c_1 \vee \ldots \vee c_n \mapsto \neg c_1 \to \ldots \to \neg c_n \to \bot$$

- Resolution on weak clauses:

$$\begin{aligned}&\mathrm{Res}(c_1 \vee \ldots \vee \neg p \vee \ldots \vee c_n,\\ &\qquad d_1 \vee \ldots \vee p \vee \ldots \vee d_m) \mapsto\\ &\mathrm{Res}(\neg c_1 \to \ldots \to \neg\neg p \to \ldots \to c_n \to \bot,\\ &\qquad \neg d_1 \to \ldots \to \neg p \to \ldots \to d_m \to \bot)\end{aligned}$$

# SMT solving and proofs for first-order

# Simplified control flow

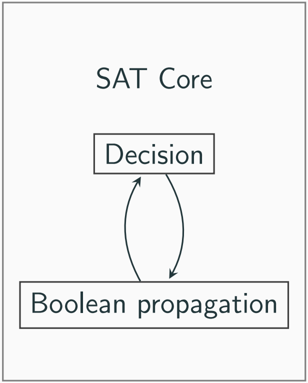



**Figure 2:** Simplified SAT/SMT Solver architecture

# Simplified control flow

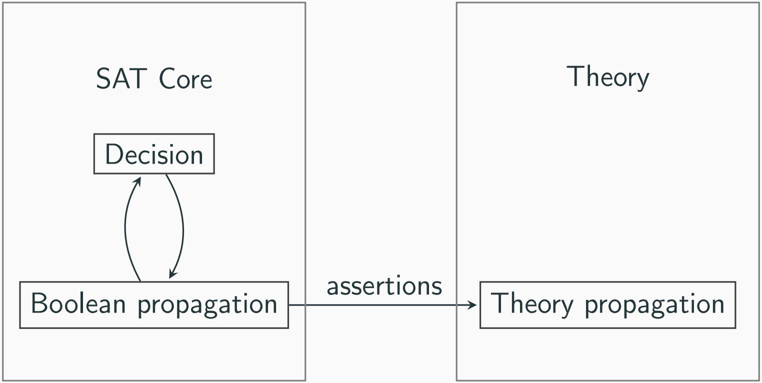
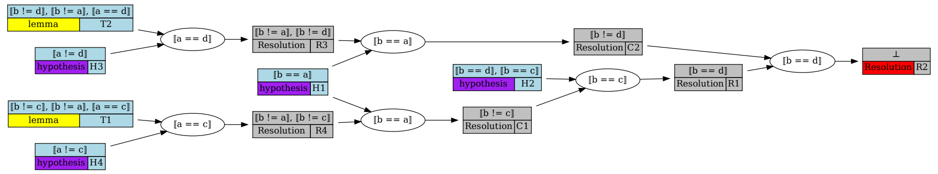



**Figure 2:** Simplified SAT/SMT Solver architecture

# SMT proofs vs SAT proofs

- Leafs can be either:
  - A Hypothesis
  - A Theory lemma
- A theory lemma is a tautology in the theory, for instance:
  - Equality reflexivity: Lemma $= (a = a)$
  - Equality transitivty: Lemma $= \neg(a = b) \vee \neg(b = c) \vee (a = c)$
  - Equality substitution: Lemma $= \neg(a = b) \vee (f(a) = f(b))$

# SMT proofs



⟦b != d⟧, ⟦b != a⟧, ⟦a == d⟧
lemma
T2
⟦a != d⟧
hypothesis
H3
⟦a == d⟧
⟦b != a⟧, ⟦b != d⟧
Resolution
R3
⟦b == a⟧
⟦b != d⟧
Resolution
C2
⟦b == d⟧
⊥
Resolution
R2
⟦b == a⟧
hypothesis
H1
⟦b == d⟧, ⟦b == c⟧
hypothesis
H2
⟦b == c⟧
⟦b == d⟧
Resolution
R1
⟦b != c⟧, ⟦b != a⟧, ⟦a == c⟧
lemma
T1
⟦a == c⟧
⟦b != a⟧, ⟦b != c⟧
Resolution
R4
⟦b == a⟧
⟦b != c⟧
Resolution
C1
⟦a != c⟧
hypothesis
H4

# Lazy CNF conversion

- Add clauses while solving
- Distinguish clausal calculus (SAT) from logic connectors $(\vee, \wedge, \Rightarrow \ldots)$

| Clauses | Assumed atoms |
|---|---|
| • $\neg[(A \wedge B) \Rightarrow A]$ | |

# Lazy CNF conversion

- Add clauses while solving
- Distinguish clausal calculus (SAT) from logic connectors $(\vee, \wedge, \Rightarrow \ldots)$

| Clauses | Assumed atoms |
|---|---|
| • $\neg[(A \wedge B) \Rightarrow A]$ | • $P \equiv (A \wedge B) \Rightarrow A \mapsto \bot$ |

# Lazy CNF conversion

- Add clauses while solving
- Distinguish clausal calculus (SAT) from logic connectors $(\vee, \wedge, \Rightarrow \ldots)$

| Clauses | Assumed atoms |
|---|---|
| • $\neg[(A \wedge B) \Rightarrow A]$<br>• $[P], [A \wedge B]$<br>• $[P], \neg[A]$ | • $P \equiv (A \wedge B) \Rightarrow A \mapsto \bot$ |

# Lazy CNF conversion

- Add clauses while solving
- Distinguish clausal calculus (SAT) from logic connectors $(\vee, \wedge, \Rightarrow \ldots)$

| Clauses | Assumed atoms |
|---|---|
| • $\neg[(A \wedge B) \Rightarrow A]$ | • $P \equiv (A \wedge B) \Rightarrow A \mapsto \bot$ |
| • $[P], [A \wedge B]$ | • $Q \equiv A \wedge B \mapsto \top$ |
| • $[P], \neg[A]$ | • $A \mapsto \bot$ |

# Lazy CNF conversion

- Add clauses while solving
- Distinguish clausal calculus (SAT) from logic connectors $(\vee, \wedge, \Rightarrow \ldots)$

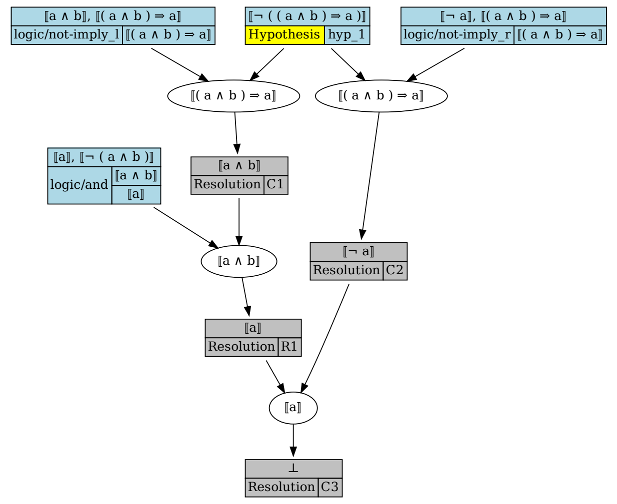
| Clauses | Assumed atoms |
|---|---|
| • $\neg[(A \wedge B) \Rightarrow A]$ | • $P \equiv (A \wedge B) \Rightarrow A \mapsto \bot$ |
| • $[P], [A \wedge B]$ | • $Q \equiv A \wedge B \mapsto \top$ |
| • $[P], \neg[A]$ | • $A \mapsto \bot$ |
| • $\neg[Q], [A]$ | |
| • $\neg[Q], [B]$ | |

# Lazy CNF conversion

- Add clauses while solving
- Distinguish clausal calculus (SAT) from logic connectors $(\vee, \wedge, \Rightarrow \ldots)$

| Clauses | Assumed atoms |
|---|---|
| • $\neg[(A \wedge B) \Rightarrow A]$ | • $P \equiv (A \wedge B) \Rightarrow A \mapsto \bot$ |
| • $[P], [A \wedge B]$ | • $Q \equiv A \wedge B \mapsto \top$ |
| • $[P], \neg[A]$ | • $A \mapsto \bot$ |
| • $\neg[Q], [A]$ | • $B \mapsto \top$ |
| • $\neg[Q], [B]$ | • $\rightarrow$ conflict ! |

# Lazy CNF conversion - proof graph



# Proofs in action

Demo Coq

# Technical subtleties

- Proper naming and escaping
- Keep information on formula order and parentheses:
  - equality: $a = b \not\equiv b = a$
  - logical connectives: $p \wedge (q \wedge r) \not\equiv (p \wedge q) \wedge r$
- First-order implicit assumptions vs actual hypotheses

# Conclusion

- Fully checkable proof output
- Increased trust in results
- Future work:
    - Extend to other proof assistants
    - Faster proofs