The ProoVer Competition

A Proof Checker Competition

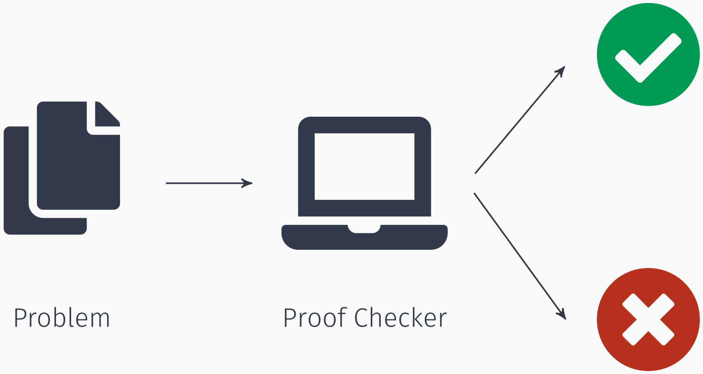
Julie Cailler & Simon Guilloud

August 2, 2025

The 14th TPTP Tea Party

Proof Verifier

Proof Verifier

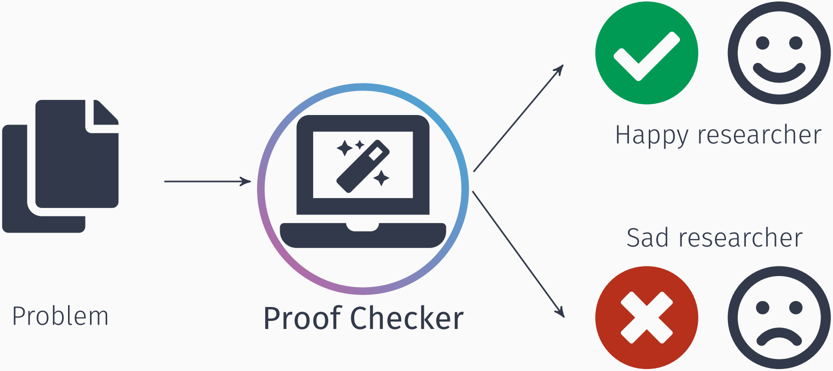


Problem

Proof Checker

ProoVer

Proof Verifier



Problem

Proof Checker

Happy researcher

Sad researcher

F.A.Q.

F.A.Q.

- Why?
    - Trust ATP proofs
    - Smaller CASC next year → no more budget for Vampire's trophies
    - Fun :)

F.A.Q.

- Why?
  - Trust ATP proofs
  - Smaller CASC next year → no more budget for Vampire's trophies
  - Fun :)
- When?
  - IJCAR 2025 (part of FLoC)

F.A.Q.

- Why?
  - Trust ATP proofs
  - Smaller CASC next year → no more budget for Vampire's trophies
  - Fun :)
- When?
  - IJCAR 2025 (part of FLoC)
- Is it free?
  - ~~Yes, it's subsidized by Geoff's gambling profits!~~ Actually, nope.

F.A.Q.

- Why?
  - Trust ATP proofs
  - Smaller CASC next year → no more budget for Vampire's trophies
  - Fun :)
- When?
  - IJCAR 2025 (part of FLoC)
- Is it free?
  - ~~Yes, it's subsidized by Geoff's gambling profits!~~ Actually, nope.
- Who can participate?
  - Anyone with a proof checker!

F.A.Q.

- Why?
    - Trust ATP proofs
    - Smaller CASC next year → no more budget for Vampire's trophies
    - Fun :)
- When?
    - IJCAR 2025 (part of FLoC)
- Is it free?
    - ~~Yes, it's subsidized by Geoff's gambling profits!~~ Actually, nope.
- Who can participate?
    - Anyone with a proof checker!
- What do I win?
    - ...Fun? :)

Proofs Structure & Language

- 1 problem file and its corresponding solution file
- Proofs in TSTP format
- `FOF` problems with axioms and a conjecture
- Sequence of inference steps
- Roles: `axiom`, `conjecture`, `negated_conjecture`, or `plain`
- Status: `thm`, `esa`, or `cth`
- “reasonable” granularity
- Unspecified inference rules can be checked external tool (`thm` only)
- Specified inference rules must be checked by the proof checker (Skolemization, negated conjecture, ...)
- No restrictions on the order of the proof steps

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

`problem.p` and `solution.p`

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

problem.p and solution.p

```
fof(a1, axiom, p(a)).
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f1, plain, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [a1, f1])).
cnf(f3, plain, ~(p(a)) | ~(! [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(X0))) ,
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(X0))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, p(f(X0)) ,
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f2])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Internal checker

External prover

Evilization!

- Remove some hypothesis and/or axioms
- Wrong references
- Skolemization not done correctly
- Non-logical consequences
- ...

Evil Proof ☹

```
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y))))))).
```

```
fof(f0, negated_conjecture, ~~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f0_bis, plain, $false, inference(res, [status(thm), 0, 0], [myproblem, f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [f1])).
cnf(f1, negated_conjecture, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f3, plain, ~(p(a)) | ~(? [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(a))),
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(a))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, q(f(X0)),
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f7])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Evil Proof ☹

```
fof(myproblem, conjecture, p(a) & (? [X] : (p(X) => (! [Y] : (p(Y)))))).
```

```
fof(f0, negated_conjecture, ~~(p(a) & ? [X] : (p(X) => ! [Y] : (p(Y)))),
inference(negated_conjecture, [status(cth)], [myproblem])).
cnf(f0_bis, plain, $false, inference(res, [status(thm), 0, 0], [myproblem, f0])).
cnf(f2, plain, p(X0), inference(res, [status(thm), 0, 0], [f1])).
cnf(f1, negated_conjecture, ~p(a) | p(X0), inference(cnf, [status(thm)], [f0])).
cnf(f3, plain, ~(p(a)) | ~(? [Y] : (p(Y))), inference(cnf, [status(thm)], [f0])).
cnf(f4, plain, ~(p(a)) | ~(p(f(a))),
inference(skolemization, [status(esa), 1, $fot(Y), $fot(f(a))], [f3])).
cnf(f5, plain, ~(p(f(X0))), inference(res, [status(thm), 0, 0], [a1, f4])).
cnf(f6, plain, q(f(X0)),
inference(instantiate, [status(thm), 0, $fot(X0), $fot(f(X0))], [f7])).
cnf(f7, plain, $false, inference(res, [status(thm), 0, 0], [f5, f6])).
```

Competition Set-Up

- 100 proofs (50 good/50 bad)
- xxx seconds timeout (?)
- Expected result: `checked` or `proof incorrect` (+ where the error occurs)
- Grading scheme (proposal):
    - Saying a Bad proof is Bad = +2
    - Saying a Good proof is Good = +1
    - Saying Nothing = 0
    - Saying a Good proof is Bad = -1 (todo: increase)
    - Saying a Bad proof is Good = -2 (or disqualify, todo: increase)

Preliminary Schedule

- Now: Interested?
- October: Website, rules and benchmarks available
- End of June 2026: System registration
- End of July 2026: ProoVer Competition at IJCAR

Questions?

- Comments?
- Other specified rules?
- List of exhaustive rules?
- Time limit?
- Tiebreaker?
- Online tool on the TPTP website

Questions?

- Comments?
- Other specified rules?
- List of exhaustive rules?
- Time limit?
- Tiebreaker?
- Online tool on the TPTP website

Thank you for your attention! ☺