

(Quick) Proposal: SZS dubious

Michael Rawson, Martin Suda

dubious /dooˈbē-əs, dyooˈ-/



adjective

1. Fraught with uncertainty or doubt; undecided.

2. Arousing doubt; doubtful.

"a dubious distinction."

3. Of questionable character.

"dubious profits."

Proposal In Brief

- ATPs **self-flag** some (SZS) output as **dodgy**
 - Not necessarily **wrong**
 - Just in need of closer inspection
- **Users** are notified of questionable behaviour
- **Developers** have something to grep for in logs
- **Competition organisers** know what to look for in solutions



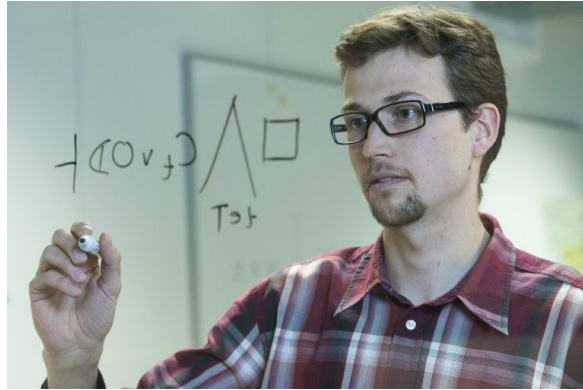
Disclaimer: no bugs here!

- Vampire is a sound system with **zero bugs**.
 - No bugs here, no sir.
 - Even those that lurk in the bug tracker.
- But: you can never be too careful!
- Sometimes a pre-release Vampire escapes.

Motivation

- Sometimes detect **bad states** internally.
- But: have **already printed** a proof.
- Or: not **sure** that it's wrong, just dubious.

Would like to indicate to the user that something is off,
without necessarily retracting a possibly-already-printed proof.



In Conversation with Martin Suda



Martin Suda

Hey, what is our preferred way to abort in release mode (like an assertion violation would)?

EDITED

I am currently moving the two sanity checks for SaturationAlgorithm's `handleEmptyClause` to `UIHelper::outputResult` to make it cover also refutations from AVATAR, GlobalSubsumption and even FMB, so I thought I would make them nicer.



Michael Rawson

`Debug::Tracer::printStack`, then `System::terminateImmediately`? Not sure, we don't often want that!

Or if just want to be sure that it crashes for experimenting you could `__builtin_trap()`.



Martin Suda

Hmm, I was thinking something more polite to the user ... Like "Sorry, this is probably our bug."

So ideally this would stay the assertion (no need for stacktrace), but also in release.

Giles' default was to go silently into `REFUTATION_NOT_FOUND`, but in my version, I have already printed "Refutation found. Thanks to.."

The idea being:

- 1) we say we got the proof (and want to say it quickly)
- 2) we minimize the proof (which was until now only done during proof printing)
- 3) use 2) as the opportunity to collect the most precise statistics about the empty clause and its derivation
- 4) Do those sanity checks.

EDITED



Michael Rawson

Now that I look at the sanity checks - would it perhaps make sense to actually print the proof, just with a disclaimer that we think it's buggy? That way when we get the inevitable report we know how we got it?



Michael Rawson

Martin Suda said:

So, I should from now on start grepping for `"viola\|SIG\|Error\|Invalid"`

Maybe we should have a more common interface for this - I think we could (ab)use SZS for this, and we should definitely report `SZS TimeOut` and `SZS Incomplete` which we don't at the moment and is very annoying.



Martin Suda

Good points. Will you keep them on your big TODO list? ;)

We could invent SZS status Bug for this use-case.

Use Cases

- Proof sanity checks:
 - DAG does not contain an input clause
 - Only uses theory clauses
 - AVATAR empty clause contains no splits
 - Doesn't match status in TPTP header
 - 50/50 our problem or Geoff's problem!
 - ...
- Crashes - if a proof printed it's **probably** OK.
- Assertion violations that aren't bad enough to crash for.
 - Less "assert", more "I would really feel better if ..."

Concretely

- At any time, systems **may** indicate

```
% SZS dubious
```

- **May** inflect with a code

```
% SZS dubious PureTheoryProof
```

```
% SZS dubious FailedAssertion
```

```
% SZS dubious Crashed
```

- **Should** explain themselves with a helpful message

```
% Vampire crashed after printing a proof: the proof is very likely OK,
```

```
% but give it a once-over just to be sure.
```



Action!

- Do you like this?
- Do you hate it?
- Are you indifferent to it?
- Would you rather be **eating cheese**?
- If no love for this, **forget it**.
- If there **is** some support:
 - Bikedshed about the name/syntax
 - What codes would you use?
 - Other use-cases?
 - I'll write it up in Geoff-approved HTML as an addendum to SZS.



And now
for something
completely different...



What do you need from our finite models?

Are saturations **ever** useful?