

# Project overview/update

Krzysztof Drewniak

HPAC

1 Dec 2017

# Project goal

- ▶ Automatic generation of {expression equality checker, normalizer, property inference}
- ▶ Current code is fragile and hard to extend
- ▶ Something easier to add to would be nice

## Initial approach: rewrite rules

- ▶ Use completion algorithm to make a confluent rewrite system
- ▶ Confluence: apply rules in any order and get some results
  - ▶ Consider  $y + x + -x \rightarrow y$  on  $x + -x + -(-x)$ .
  - ▶ Can get  $-(-x)$  or  $x$  as final result
  - ▶ Need to add  $-(-x) = x$  to fix
- ▶ (Done) Base Knuth-Bendix algorithm and syntactic unification
- ▶ (Near done) Associative-commutative completion and unification
  - ▶ One small bug in there somewhere maybe
- ▶ **(Failed)** Rewrite rules conditioned on properties
  - ▶ Trying to represent these as unconditional equations gets unwieldy and prevents sensible results
  - ▶ Also, associativity only sort of works ( $((AB)C \rightarrow A(BC)$  does not cover all cases)

# The future: theorem provers?

- ▶ Property inference and equality checking are all first-order logic things
- ▶ We could delegate that to a theorem prover, potentially
- ▶ Could also enable more stuff like bandwidth properties (generalization of diagonal, triangular, etc.)
- ▶ Potential tools
  - ▶ Prolog (might not be expressive enough)
  - ▶ E (first order logic, but nothing else — we'd need to roll our own  $\mathbb{N}$ )
  - ▶ SMT solvers like z3 (seem to be very well developed, may not solve our problem)
  - ▶ Coq *et al* don't fit our problem type