Auto Sledgehammer

Qiyuan Xu

November 5, 2024

theory *Auto-Sledgehammer*
  imports *HOL.Sledgehammer*
begin

named-theorems *φsledgehammer-simps ‹Simplification rules used before applying slegehammer automation›*

ML-file *‹library/helpers0.ML›*
ML-file *‹library/Hasher.ML›*
ML-file *‹library/Phi-ID.ML›*
ML-file *‹library/cache-file.ML›*
ML-file *‹library/sledgehammer-solver.ML›*

end
theory *Auto-Sledgehammer-Doc*
  imports *Auto-Sledgehammer*
begin

1 Installation

This is a standard Isabelle package, following the standard installation instructions (We refer readers to Isabelle's *system* document). Basically, you could run the following commands

- `isabelle components <THE BASE DIR OF OUR CODE>`
- `isabelle build Auto_Sledgehammer`

2 Usage

Method *auto-sledgehammer* is a slightly smart wrapper of Sledgehammer, allowing users to call Sledgehammer using a normal tactic like *auto*

lemma *foo*: *‹(1::nat) + 2 = 3›*
  by *auto-sledgehammer*

This tactics first applies *auto* or *clarsimp*; ((*rule conj*)+)*?* to simplify and split the target proof goal into several subgoals. For each obtained subgoal, it applies Sledgehammer successively.

The proofs obtained by Sledgehammer are cached by the hash of the proof goal. When Isabelle's evaluation reaches the last *end* command of a theory file, the cache will be stored into a file named "<theory-name>.proof-cache". This cache will be loaded when later Isabelle re-evaluates the same theory, so that *auto-sledgehammer* can reuse the cached proofs without rerunning Sledgehammer again.

Sledgehammer can return multiple tactics, while not all of them can terminate in a short time. Our *auto-sledgehammer* will replay each tactic within a time limit (by default 20 seconds). You could configure this timeout as follows.

declare [[*auto-sledgehammer-preplay-timeout* = *10*]] — always in the unit of seconds

The first successfully replayed tactic will be returned immediately, killing all other working replays and the Sledgehammer process. It speeds up the proof search a lot. When Sledgehammer typically spends one minute, this strategy can allow our *auto-sledgehammer* to terminate within few seconds.

Before applying Sledgehammer, *auto-sledgehammer* applies *auto* to split a big goal into small subgoals. According to our experience, it could improve the success rate of Sledgehammering a lot. However, *auto* can non-terminate for complex goals. Thus we impose a time limit (by default, 3 seconds). If *auto* timeouts, the tactic tries *clarsimp*; ((*rule conj*)+)*?* instead. If it still timeouts, the tactic tries plain Sledgehammer without any prelude.

The Sledgehammer parameters used by this tactic are configurable as follows.

declare [[*auto-sledgehammer-params* = *isar-proofs* = *false*, *timeout* = *666*, *minimize* = *false*]]

You can also add additional simplification rules that will be used by our prelude *auto* that splits goals.

declare *foo* [[*φsledgehammer-simps*]]

end