Brackets and blast

Kyndylan

March 5, 2015

Contents

Some dummy definitions.

definition *foo* :: *bool* **where**
  foo = *True*

definition *bar* :: *bool* **where**
  bar = *foo*

definition *baz* :: *bool* **where**
  baz = *bar*

A conventional elim rule for bar.

lemma *barE*:
  assumes *bar*
  obtains *foo*
using *assms*
unfolding *bar-def*
by *auto*

Version 1 of an elim rule for baz.

lemma *bazE1*:
  assumes *baz*
  obtains *bar* ∧ *True* ∧ (∀ *x*. *True*)
using *assms*
unfolding *baz-def*
by *auto*

Blast is not able to use that version.

lemma
  assumes *baz*
  shows *foo*
using *assms*
— apply (blast elim: bazE1 barE) fails
oops

Version 2. Note that the brackets are the only difference with version 1.

lemma *bazE2*:
  assumes *baz*
  obtains (*bar* ∧ *True*) ∧ (∀ *x*. *True*)
using *assms*
unfolding *baz-def*
by *auto*

Now blast is able to prove the lemma.

lemma
  assumes *baz*
  shows *foo*
using *assms*
by (*blast elim*: *bazE2 barE*)

If the quantifier is removed, then the version without the brackets can be used by blast to prove the lemma.

lemma *bazE3*:
  assumes *baz*
  obtains *bar* ∧ *True* ∧ *True*
using *assms*
unfolding *baz-def*
by *auto*

lemma
  assumes *baz*
  shows *foo*
using *assms*
by (*blast elim*: *bazE3 barE*)

Also, if there is only one layer of definitions (baz - bar), instead of two (baz - bar - foo), then the version without the brackets works as well.

lemma *bazE4*:
  assumes *baz*
  obtains *bar* ∧ *True* ∧ (∀ *x*. *True*)
using *assms*
unfolding *baz-def*
by *auto*

lemma
  assumes *baz*
  shows *bar*
using *assms*
by (*blast elim*: *bazE4*)

Why is the combination of no brackets, the quantifier, and the extra layer of definitions preventing blast from finding a proof?