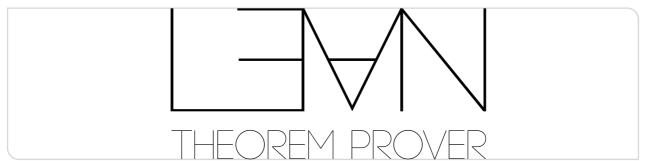




Theorembeweiserpraktikum

Even More Tactics

Jakob von Raumer, Sebastian Ullrich | SS 2021



focus vs. •



A correction: focus does not actually force the first goal to be closed

```
example (hp : p) (hq : q) : p \land q := by
  constructor
  focus -- 'r p'
    skip
  focus
   -- still `+ p`
   exact ha -- tupe mismatch
```

We introduced • (input as \centerdot) as a (prettier) way to do just that:

```
constructor
· skip -- unsolved goals: ... ⊢ p
· exact hg -- is still checked
```

This is basically an unnamed case

While focus is not strictly structuring given this insight, we'll allow both this semester



What's Up with Those Dots Anyway

We've seen · before: in terms, it can be used as a nameless lambda

```
set_option pp.binder_types false  
#check (\cdot :: \cdot) -- fun a a_1 => a :: a_1 : ...  
#check (1 + 2 * \cdot) -- fun a => 1 + 2 * a : Nat \rightarrow Nat  
#check [\theta, 1].map (\cdot.succ) -- List.map (fun a => Nat.succ a) [\theta, 1] : List Nat
```

All occurrences of · are bound to the nearest surrounding parentheses, from left to right

refine



We already know we can arbitrarily nest terms and tactics:

```
example (hr : r) (hrp : r \rightarrow p) (hq : q) : p \land q := by exact \langle (by simp_all), hq\rangle
```

We can use refine to move out nested tactic blocks

```
refine <?p, hq>
case p => -- + p
simp_all
```

apply e where e: $(h:p) \rightarrow \dots \rightarrow q$ can be thought of as a special case of refine: refine e?h...