Interactive Theorem Proving in HOL4

Course 11: More Tactics

Dr Chun TIAN
chun.tian@anu.edu.au

10 ottobre 2024

Australian National University

Acknowledgement of Country

We acknowledge and celebrate the First Australians on whose traditional lands we meet, and pay our respect to the elders past and present.

More information about Acknowledgement of Country can be found here and here

More tactics: `irule` and `drule` (1)

`irule` as a better `MATCH_MP_TAC`

When applied to a theorem of the form

$$A \vdash \forall x_1 \ldots x_n.\, s \Rightarrow \forall y_1 \ldots y_m.t \Rightarrow \forall z_1 \ldots z_k.\, u,$$

`irule` produces a tactic that reduces a goal whose conclusion u' is a substitution and/or type instance of u to the corresponding instances of s and of t. Any variables free in s or t but not in u will be existentially quantified in the resulting subgoal.

$$\frac{A \vdash u'}{A \vdash (\exists z.s') \wedge (\exists w.\, t')}\ \texttt{irule}\ (\forall x.\, s \Rightarrow \forall y.\, t \Rightarrow u)$$

If without `irule`

One has to (repeatly) call `MP_TAC (Q.SPECL [...] th)` to instantiate $x_1 \ldots x_n$ and then push the remaining part $(s' \Rightarrow \forall y_1 \ldots y_m.t' \Rightarrow \forall z_1 \ldots z_k.\, u')$ into the goal, etc.

More tactics: irule and drule (2)

drule

If theorem th is of the form $A \vdash t$, where t is of the form

$$\forall x_1 \ldots x_n .\, P \wedge \cdots \Rightarrow Q \quad \text{or} \quad \forall x_1 \ldots x_n .\, P \Rightarrow Q$$

then drule th looks for an assumption that matches the pattern P in t. It then performs instantiation of th's universally quantified and free variables, transforms any conjunctions on the left into a minimal number of chained implications, and calls MP once to generate a consequent theorem $A \vdash t'$. This theorem is then passed to MP_TAC, turning the goal from g to $t' \Rightarrow g$.

$\vdash \forall n\ d.\ 0 < n \wedge 1 < d \Rightarrow n\ \mathtt{DIV}\ d < n$ [arithmeticTheory.DIV_LESS]

```
 0.  1 < x                             0.  1 < x
 1.  0 < y        drule DIV_LESS       1.  0 < y
---------------  ----------------->  ----------------------------------
     P                                (!d. 1 < d ==> y DIV d < y) ==> P
```