

total functionality

[/Reference manual](#)/[Z-related commands](#)/[Proof rule commands](#)

The *total functionality* command is applicable to a name in a goal, where the name refers to an injection of a free type. It makes the total functionality constraint implied by the free type available as an antecedent in the sub-goal.

Given the general form of a free type,

$$\begin{aligned} f_1 &::= h_{1,1} \dots h_{1,m_1} \mid g_{1,1} \langle\langle e_{1,1} \rangle\rangle \dots g_{1,n} \langle\langle e_{1,n} \rangle\rangle \\ &\&\dots\& \\ f_r &::= h_{r,1} \dots h_{r,m_r} \mid g_{r,1} \langle\langle e_{r,1} \rangle\rangle \dots g_{r,n_r} \langle\langle e_{r,n_r} \rangle\rangle \end{aligned}$$

the total functionality constraint for an injection is

$$\forall x : e_{i,k} \bullet \exists_1 y : g_{i,k} \bullet y.1 = x$$

1. Tactic example

“total functionality” $e_1 \ e_2$

This example applies the *total functionality* command to expressions e_1 and e_2 .

IT 19-Feb-1999