1 Tactics

This section presents some of the tactics that I used to automize some of the proofs of the Splay Tree algorithm. Reduction tactics try to reduce hypothesis/conclusions without creating any extra hypothesis. One example of a tactic that creates new hypothesis is Modus Ponens: If we have P and P $\rightarrow$ Q then Q is added to the context. Lets go trough all of these tactics one by one to see how they behave in a generalized context Γ.

1.1 Logical

Lets start by enumerating some of the logical reductions that we used for our proofs and may use for future proofs. Lets start with a simple one: If something in our context evaluates to True, such as $a = a$ by reflexivity, then we can discard it since it will not be of any use to have it in our context. This tactic will be useful to clean the proof context so we can see which hypothesis we should really be looking at.

$$\frac{\Gamma \vdash \mathrm{C}}{\{\mathrm{True}\ ,\ \Gamma\} \vdash \mathrm{C}}$$

A logical tactic that is useful whenever we have an hypothesis in our context which is a conjunction of propositions such as H1 $\wedge$ H2 is coqs "destruct" tactic. However, whenever we have too many conjunctions it may be tedious to destroy them all. Therefore, we can create an iterative rule that performs this automatically, however the name given to all of the hypothesis are, for the meantime, attributed by Coq. This applies for disjunction as well.

$$\frac{\{\mathrm{H1}\ ,\ \mathrm{H2}\ ,\ \Gamma\} \vdash \mathrm{C}}{\{\mathrm{H1} \wedge \mathrm{H2}\ ,\ \Gamma\} \vdash \mathrm{C}}$$

$$\frac{\{\mathrm{H1}\ ,\ \Gamma\} \vdash \mathrm{C} \qquad \{\mathrm{H2}\ ,\ \Gamma\} \vdash \mathrm{C}}{\{\mathrm{H1} \vee \mathrm{H2}\ ,\ \Gamma\} \vdash \mathrm{C}}$$

2 Proven Theorems

The first part of this project was to prove a simple functional implementation of the Splay Tree algorithm. This implementation is the same as Nipkow, but it was written in Gallina and proved with Coq proof assistant instead of Isabelle. We have used in most of these proofs, functional induction and we have proved important theorems that assure us that the implementation is correct.