Deciding intuitionistic propositional tautologies
Proof Assistants

Sébastien Patte

February 2022

1 Previous experience with Coq

J'ai suivi le cours 2.7.1 (Foundations of Proof Systems) cette année, on y a fait quelques execices pratiques en Coq. Je n'avais jamais fait de Coq avant, mais ai suivi un cours d'introduction à l'assistant de preuve Isabelle.

2 Implementing the decision procedure

La règle la plus compliquée a implémenter est celle d'elimination de l'implication

$$\frac{\Delta, B \vdash C \qquad \Delta \vdash A}{\Delta, A \Rightarrow B \vdash C}\ (\Rightarrow\text{-E})$$

Il n'existe pas de tactic en Coq permettant de faire $\Rightarrow$-E directement, mais on peut le faire en combinant un cut et $\Rightarrow$-I :

$$(\Rightarrow\text{-I})\ \frac{\dfrac{\dfrac{\Delta, B \vdash C}{\Delta, A, A \Rightarrow B \vdash C}\ (\text{clear})}{\Delta, A \Rightarrow B \vdash A \Rightarrow C} \qquad \Delta \vdash A}{\Delta, A \Rightarrow B \vdash C}\ (\text{cut-A})$$

3 Formalizing the tactic

3.1 Step

3.2 Size criterion

We choose the number of operators in a formula to be the size criterion. The rules Ax, $\bot$-E, and $\top$-I don't have premises, in the other rules there is on more operator in the conclusion than in each premise. Therefore, for each rule, the size of each premise is smaller than the size of the conclusion.

3.3 Termination

We need to prove that any sequent is smaller than each of the subgoals reachable in one step. i.e. :

$$\forall(\Delta' \vdash C') \in \text{concat}(\text{step}(\Delta \vdash C)), \quad |\Delta' \vdash C'| < |\Delta \vdash C|$$

We prove it by induction on Δ and C.

3.4 Soundness

3.4.1 Semantics

```
Fixpoint sem (f:form) : Prop :=
  match f with
    | f_true    => True
    | f_false   => False
    | f_var x   => prop_of_nat x
    | f_or A B  => (sem A) /\ (sem B)
    | f_and A B => (sem A) \/ (sem B)
    | f_imp A B => (sem A) -> (sem B)
  end.
```

Où **prop_of_nat** est n'importe quelle fonction renvoyant une **Prop** à partir d'un **nat**.

3.4.2 leaf case

On prouve **is_leaf_prop** $(H \vdash C) \implies$ **valid_seq** $(H \vdash C)$, par induction sur C.