Master Parisien de Recherche en Informatique 2010–2011
Proof Assistants

TD/TP 6 – Reflection

2011-01-26

1 Encoding Propositions

1. Define an inductive type `formula` representing logical formulas ($\wedge$, $\vee$, $\neg$, $\Rightarrow$) whose atoms are relations ($=$, $\leq$, $<$) between natural numbers.

2. Define a recursive function `interp_formula: formula -> Prop` that converts an inductive formula to the corresponding logical proposition.

3. Define a recursive tactic `reify_formula` that takes a logical proposition and returns an inductive formula whose application to `interp_formula` is β-convertible to the given proposition.

   For instance, the following piece of script has to work fine:

```
Goal forall m n : nat, m + n <= n -> m = 0 /\ 0 <= n.
intros m n.
match goal with
| |- ?g => let f := reify_formula g in change (interp_formula f)
end.
```

2 Small Scale Reflection

1. Define the three functions `eq_bool`, `le_bool`, `lt_bool` on natural numbers and prove that they are equivalent to the corresponding relations. For instance, the lemma for $\leq$ is:

```
Lemma le_bool_correct : forall m n : nat,
  le_bool m n = true <-> m <= n.
```

2. Define four functions `and_bool`, `or_bool`, `not_bool`, `imp_bool` and prove that they are equivalent to the corresponding connectors. For instance, the lemma for $\Rightarrow$ is:

```
Lemma imp_bool_correct : forall p q : bool,
  imp_bool p q = true <-> (p = true -> q = true).
```

3. Define a function `bool_formula` such that the following theorem holds:

```
Theorem bool_formula_correct : forall f : formula,
  bool_formula f = true <-> interp_formula f.
```

3 Classical Logic in a Decidable World

1. Define a tactic that replaces the goal G by the goal $\neg G \Rightarrow G$ (assuming it can be reified).

2. Define a tactic that removes all the $\neg$ and $\Rightarrow$ connectors of the goal (assuming it can be reified). The negation $\neg$ might still appear in front of equalities.