Mathematical quotients and quotient types in Coq

Laurent Chicli[1,2], Loïc Pottier[1], Carlos Simpson[2]

[1] Projet Lemme, INRIA Sophia Antipolis
[2] Laboratoire J.A.Dieudonné, Université de Nice Sophia-Antipolis

Abstract. This note studies quotient types in the Calculus of Inductive Constructions (CIC), implemented in the proof assistant `Coq`, and compare their expressivity to that of mathematical quotients. In [Hof95], Martin Hofmann proposes an extension of the Calculus of Constructions (CC) with quotient types which he shows consistent, but notices that they are not sufficient to account for natural isomorphism θ which exists in set theory between functional spaces $E \to \frac{F}{R}$ and $\frac{E \to F}{S}$ where fSg iff $\forall x \in F,\ f(x)Rg(x)$. One can thus ask the question to know if it is possible to extend these quotient types to be able to show injectivity and surjectivity of this isomorphism. We show here that any extension of this type in Coq would be contradictory.

1 Introduction

Let us consider two sets E and F, R an equivalence relation on F and π_R the canonical surjection from F to $\frac{F}{R}$. We can define on $E \to F$ the following equivalence relation $S : fSg \Leftrightarrow \forall x \in E\ f(x)\ R\ g(x)$. By definition of the relation S, the map which composes any function $f : E \to F$ with π_R is compatible with S, and there is thus a natural morphism θ such that the following diagram commutes:

$$\begin{tikzcd} E \to F \arrow[rr, "\pi_R\ o\ ."] \arrow[d, "\pi_S"] & & E \to \dfrac{F}{R} \\ \dfrac{E \to F}{S} \arrow[urr, dashed, "\theta"] & & \end{tikzcd}$$

In set theory (where the axiom of choice and the extensionality of functions are given), this morphism is clearly injective (if $\pi_R\ o\ f = \pi_r\ o\ g$ i.e. $\forall x \in E,\ [f(x)]_R = [g(x)]_R$ one has fSg) and surjective (if f is a function from E to $\frac{F}{R}$, it is the image by θ of the function $\pi_S(g)$ where g sends x on a representative of the class of $f(x)$).

In the Calculus of Inductive Constructions implemented in `Coq` and extended by the quotient types of M.Hofmann [Hof95] the map θ can be defined (however we need the axiom of extensionality of functions) and can be shown injective, but not surjective. The goal of this paper is to show that one cannot extend these quotient types so as to obtain this surjectivity: the two types `E->F/R` and `(E->F)/S` are not and cannot be made isomorphic in the CIC of Coq.

The structure of this paper is the following. The problem is posed in details in section 2: one describes there the quotient types we consider, the difference in `Coq` between the proposition stating the existence of an object and the constructive data of it, finally we show the equivalence of the problem with the existence in general of a section for the canonical surjection of a quotient set (π_R).

In section 3, we show that the constructive data of a section is contradictory. We describe some results known on the quotients in CIC: the adaptation of the result of Diaconescu [Dia75] by Werner/Lacas [LacWer99] and results obtained by Maria Emilia Maietti [Mai98].

But our problem is equivalent to the *existence* of section for quotients and we show the inconsistency of this assumption in section 4. The first stage consists in showing the existence of a function `Proptobool` from $Prop$ towards *bool* sending the true propositions on `true` and the false propositions on `false`. The second stage proceeds as follows: in a context where $False$ (of type $Prop$) is the goal to prove, we can use elimination on the existence of `Proptobool` and thus use it to build an excluded middle in Set, which implies $False$ ([Pot00], [Geu01]).

All the proofs described here has been formalized in Coq, and we give in this paper the script corresponding to the definitions, lemmas and proofs each time they are introduced[3].

2 Quotient types in Coq

2.1 The quotient types

The quotient types which we consider were studied in [Hof95] , [Bar95] and [Bou97], they are a quasi-direct formalization of the mathematical practice of the quotients, to which we however cut off the concept of choice of a representative in an equivalence class.

For a type E and an equivalence relation R, they consist of:

- a type `quo` representing the quotient type,
- a function `class: E - > quo` for the canonical surjection,

[3] One can find the complete script including proofs at the following URL: `http://www-sop.inria.fr/lemme/personnel/Laurent.Chicli/types2002.zip`.

- two propositions characterizing the equality of classes,
- a constructive way of lifting in the quotient the functions compatible with R.
- Finally, because precisely one has no function of choice of a representative in the equivalence classes, we give the principle of induction `quo_ind` which makes it possible to show that a predicate is true for every element of the quotient type `quo` iff it is true for every equivalent class.

More precisely, in Coq, we formalize these quotient types like this :

```
Record type_quotient [E : Type; R : (Relation E);
                      p : (Equivalence R)] : Type := {
   quo:> Type;
   class:> E->quo;
   quo_comp: (x,y:E)(R x y)->(class x) == (class y);
   quo_comp_rev: (x,y:E)(class x)==(class y)->(R x y);
   quo_lift: (F:Type)(f:E->F)(compatible R f)->quo->F;
   quo_lift_prop:
     (F:Type)(f:E->F)(H:(compatible R f))
     (x:E)((comp (quo_lift F f H) class) x)== (f x);
   quo_ind: (P:quo->Prop)((x:E)(P (class x)))->(y:quo)(P y)}.

Axiom quotient :
   (E:Type) (R:(Relation E))(p:(Equivalence R))(type_quotient p).
```

Provided with these quotient types, we can define the equivalence relation S and the morphism θ in `Coq` (the extentionality of functions is necessary for that but is consistent with the quotient types):

```
Variable E,F:Type.
Variable R:(Relation E); p:(Equivalence R).

Definition S:=[f,g:F->E](x:F)(R (f x) (g x)).

Lemma Sequiv: (Equivalence S).

Axiom Extensionality :
   (F,G:Type)(f,g:F->G)((x:F)(f x) == (g x))->f == g.

Lemma ps:(compatible S [f:F->E](comp (class (quotient p)) f)).

Definition theta:=(!quo_lift ? ? ? (quotient Sequiv) ? ? ps).
```

2.2 *Prop* and *Set*.

The bottom of the hierarchy of universes of `Coq` consists of two kinds: *Prop* and *Set*. The first, *Prop*, is intended to represent the type of the logical propositions:

a proposition in Coq is a type of type *Prop* . The second, *Set*, is intended for the *objects of the speech*: the set of natural numbers, the lists, the programs are generally built in *Set*.

The elimination of (or the reasoning by cases on) a term of kind *Prop* is authorized in Coq to build a term of kind *Prop* but generally not to build a term of kind *Set*.

The existence of an object x verifying a property P being thus defined in *Prop*:

```
Inductive exT [A : Type; P : A->Prop]  : Prop :=
      exT_intro : (x:A)(P x)->(ExT P)
```

we will be able to make an elimination on the existence of x (and thus to have x in our context) to build a term of kind *Prop* (the proof of any proposition) but not to build a term of kind *Set* (and thus an object). To affirm the existence of an object is thus weaker than to give it in *Set*: in the first case we will be able to make use of it only to show propositions, in the second case we will be able to go further and to build new objects with it.

In the continuation of this paper the difference between the two concepts is fundamental and each reference to the existence of an object will indicate only the proposition (in *Prop*) stating its existence.

2.3 The existence of a section for a quotient type

Let us return to our initial problem. In fact, for given F and R, the surjectivity of the morphisms θ for all E is equivalent to the existence of a section of π_R, i.e. a function s such as

$$\exists s : \frac{F}{R} \to F \mid \pi_R \circ s = Id_{\frac{F}{R}}.$$

Indeed, if θ is surjective in the case where $E = \frac{F}{R}$, the proposition above results from

$$\exists s : \frac{F}{R} \to F \mid \pi_R \circ s = \epsilon(g) \text{ ou } g : \frac{\frac{F}{R} \to F}{S}.$$

By applying then the principle of induction `quo_ind` we are brought back to the goal

$$\forall h : \frac{F}{R} \to F,\ \exists s : \frac{F}{R} \to F \mid \pi_R \circ s = \theta(\pi_S(h)) \ ,$$

which is trivialy true since $\theta(\pi_S(h)) = \pi_R \circ h$ according to the property `quo_lift_prop`.

Conversely, if there is a function $s : \frac{F}{R} \to F$ such as $\pi_R \circ s = Id_{\frac{F}{R}}$, then for all E the corresponding map θ is surjective: for all $F : E \to \frac{F}{R}$ we have:

$$\theta(\pi_S(s \circ f)) = \pi_R \circ (s \circ f) = (\pi_R \circ s) \circ f = f.$$

In the continuation, only the first implication will interest us, and will give the following code in `Coq` :

```
Hypothesis theta_surj : (f : F ->  (quotient p))
  (!exT ? [g : (quotient Sequiv)](theta g) == f).
```

one shows:

```
Lemma step1: (!exT ? [s:(quo (quotient p))->E]
(comp (class (quotient p)) s) == (Id (quotient p))).
```

The surjectivity of the morphism θ is thus brought back to the question of the possibility of extending the quotient types by the existence of a section. In the next part, we will briefly recall results of [Pot00] where such a section is given constructively, and we will show the inconsistency of this assumption. It is only in the fourth section that we will treat the case where only the *existence* of the section is given in assumption.

3 To give constructively a function of choice leads to a contradiction.

We are here within a framework where, in addition to our quotient types, a function of choice of representative in equivalence classes `s: quo -> E` is given, such that:

$$E \underset{s}{\overset{c}{\rightleftarrows}} quo \qquad \forall x : quo,\ (c(s\ x)) == x.$$

(where c is the function `(class quo)`).

3.1 The trick of Diaconescu

B.Werner and S.Lacas showed in [LacWer99] (by adapting the trick of Diaconescu [Dia75]) that in this case, for a given proposition P, one can use the quotients to build a term of type `{P}+{~P}` deciding if P is true or not. They proceed in the following way.

Suppose $E = bool$ and let us define the relation R by:

$$x\ R\ y \Leftrightarrow x = y \vee P,$$

R is indeed an equivalence relation, and if we consider the quotient E/R (it depends of P), we have:

$$\begin{aligned}(s(c\ true)) =_{bool} (s(c\ false)) &\Leftrightarrow (c\ true) == (c\ false)\\ &\Leftrightarrow P\end{aligned}$$

And, the equality in *bool* being decidable, we obtain the decidability of P.

3.2 Inconsistency of the excluded midle in *Set*

By generalizing the Werner-Lacas construction for any proposition P one can thus build a term of type `(P:Prop){P}+{~P}`, i.e. the strong excluded middle, in the kind *Set*.

To conclude, the fact is precisely used that the excluded middle in the kind `Set` is contradictory in `Coq`.

The interested reader will be able to refer to [Pot00] [4] for a demonstration of this inconsistency. This one takes as a starting point the result shown by Barbanera-Berrardi in [BarBer96] (the axiom of choice with the excluded middle implies proof-irrelevance) and adapts it to *Set* (what is possible since *Set* is impredicative) in `Coq` showing that the strong excluded middle implies the equality of all the terms in a same type of kind *Set*, which is false (for example one can show that $0 \neq 1$ in the type `Nat` of the natural numbers of `Coq`).

Another demonstration, more complicated, can be found in [Geu01], based on an adaptation of the Burali-Forti paradox (by Girard Coquand and Hurkens). This proof also uses in a strong way the impredicativity of *Set*.

3.3 Related work

We can quote the work of Maria Emilia Maietti. In [Mai98], she shows that extending Martin-Löf type theory with quotient types and with the rule of uniqueness of equality proofs implies the excluded middle.

If we try to put this result in the CIC, we prove in `Coq`[5](with our quotient types) that the axiom of choice with the rule of uniqueness of equality proofs implies the excluded middle in *Prop*. However if we try to adapt this proof to get strong excluded middle, we cannot do it without adding a strong axiom of choice (and inconsistent with our quotient).

[4] This formalization suppose, in addition to the strong excluded middle, the axiom *eqT_rec* which makes it possible to rewrite on objects of kind *Set* if one has two equal terms for the Leibnitz equality in kind *Type*. Although noncontradictory with the CIC, this axiom was not included in `Coq` to preserve the mechanism of extraction. It is not however any more the case since the version 7.3 of `Coq` where *eqT_rec* was definitively integrated into the system.

[5] The Coq script can be found at this url : `http://www-sop.inria.fr/lemme/personnel/Laurent.Chicli/t02_rel_work.zip`

4 Even the existence of a section leads to a contradiction.

This time, the section `s: quo -> E` is not given any more, but we have only the proposition ensuring us its existence:

$$E \underset{s}{\overset{c}{\rightleftarrows}} quo \qquad \exists s : quo \to E,\ \forall x : quo,\ (c\ (s\ x)) == x.$$

4.1 From *Prop* towards *bool*

The goal of this section is to show the existence of a function `Proptobool: Prop -> bool` sending the true propositions on `true` and false propositions on `false`, i.e. such as:

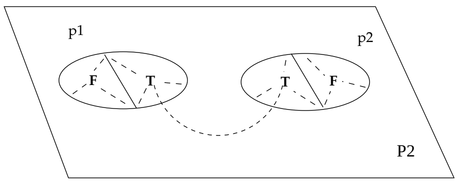
$$\forall P : Prop,\ P \leftrightarrow (Proptobool\ P) == true. \qquad (i)$$

A particular quotient With this intention, we will use the existence of the section for the following particular quotient.

Thus let us consider the inductive type $P2$, constituted of two copies of *Prop*

```
Inductive P2 : Type :=
   p1: Prop ->  P2
 | p2: Prop ->  P2 .
```

We define the following relation on the elements of $P2$: all the equivalent propositions of the same copy of *Prop* are equivalent, as well as the true propositions of $p1$ and $p2$.



In `Coq` , that gives:

```
Inductive R : P2 -> P2 ->  Prop :=
   R_12: (x, y : Prop) x -> y ->  (R (p1 x) (p2 y))
 | R_21: (x, y : Prop) x -> y ->  (R (p2 x) (p1 y))
 | R_11: (x, y : Prop) (iff x y) -> (R (p1 x) (p1 y))
 | R_22: (x, y : Prop) (iff x y) -> (R (p2 x) (p2 y)) .

Lemma Requiv: (Equivalence R).
```

and we define thus the quotient type `QP2`:

```
Definition QP2 := (quotient Requiv).
```

Some definitions and properties Since we want to define a term of kind `Prop` (we want to prove *the existence* of the function `Proptobool`), we can eliminate the existential quantifier on s and have s in our context.

Let us consider the function $f = s \circ c$, from $P2$ to itself. It has the two following properties:

– $\forall x : P2,\ xRf(x)$: (1)

$$\begin{aligned}\forall x : P2,\ xR(fx) &\Leftrightarrow (c\ x) == (c\ (f\ x))\\ &\Leftrightarrow (c\ x) == (c\ (s\ (c\ x)))\\ &\Leftrightarrow (c\ x) == (c\ x)\end{aligned}$$

– $\forall x, y : P2,\ xRy \Rightarrow f(x) == f(y)$: (2)

$$\begin{aligned}\forall x, y : P2,\ xRy &\Leftrightarrow (c\ x) == (c\ y)\\ &\Rightarrow (s(c\ x)) == (s(c\ y))\end{aligned}$$

Let us define the following functions `in_p1` and `in_p2`:

```
Definition in_p1: P2 ->  bool :=
   [x:P2]Cases x of (p1 p)=>true  | (p2 p)=> false end.

Definition in_p2 : P2 ->  bool :=
   [x:P2]Cases x of (p1 p)=>false | (p2 p)=> true end.
```

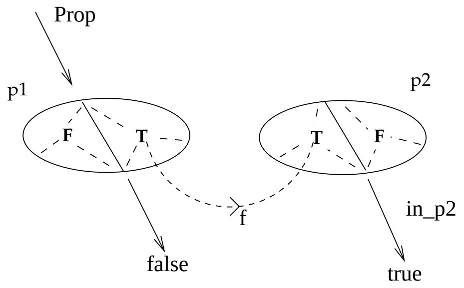
An element of $P2$ in relation with $p_1(False)$ cannot have the the form $(p_2\ y)$:

$$\forall x \in P2,\ x\ R\ p_1(False) \rightarrow in\_p_2(x) = false. \tag{3}$$

The proof is done easily in `Coq`, by elimination on x, and then an inversion on R.

The existence of the function `Proptobool`. We now have all bricks necessary to show the existence of the function `Proptobool`: we proceed by case analysis on `(in_p1 (f (p1 True)))`:

- if `(in_p2 (f (p1 True))) = true` then the function $in\_p2 \circ f \circ p1\ :\ Prop \rightarrow bool$ has the desired property i):



Indeed, let us suppose that a proposition P is true. We have

$$P \leftrightarrow True$$

and thus

$$p1(P)\ R\ p1(True).$$

By (2), it is followed from there that

$$f(p1(P)) == f(p1(True))$$

and thus:

$$in\_p2(f(p1(P))) == in\_p2(f(p1(True))) == true.$$

Conversely, let us suppose now that $\tilde{}P$ is true. We have

$$P \leftrightarrow False,$$

from where

$$p1(P)\ R\ p1(False),$$

then, by (2):

$$f(p_1(P)) == f(p_1(False)),$$

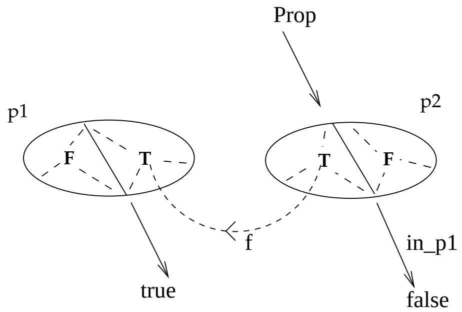
by applying (1):

$$f(p_1(P))\ R\ p_1(False),$$

finally, by (3):

$$in\_p_2(f(p_1(P))) == false.$$

- if `(in_p2 (f (p1 True))) = false` one sends this time `Prop` in `bool` through $in\_p1 \circ f \circ p2$, which we show in a similar way that it verifies the property i).



We show thus in Coq , without using any form of excluded middle:

```
Lemma existence_decision:
(!exT ? [Proptobool:Prop->bool]
        (P:Prop)( P <-> (Proptobool P) = true)).
```

4.2 Inconsistency

The idea to show the inconsistency in this case is the same one as before: use `Proptobool` to show the excluded middle in `Set`, which is contradictory. However, we cannot show here `(P:Prop){P}+{~P}` using `Proptobool` since we do not have *a priori* the function itself but only its existence, and that the elimination of an object of `Prop` (the existential quantifier) is not possible when the kind of goal is `Set` what is the case here.

We circumvent the difficulty by noticing that what we want to show, at the end, is a proposition: `False`. We thus can, to build a term of the type `False`, make an elimination on the existence of `Proptobool` and thus introduce it in our context.

Then we just make a cut with `(P:Prop)({P}+{~P})`, which we easily prove for all `P` by case analysis on `(Proptobool P)`.

```
Lemma incoherence: False.
Generalize existence_decision.
Intros yH.
Elim yH; Intros Proptobool H; Clear yH.
Cut (P:Prop){P}+{~P}.
Exact paradoxe.
Intros P.
Generalize (refl_equal ? (Proptobool P)).
Pattern -1 (Proptobool P); Case (Proptobool P).
Intros H0.
Case (H P).
Intros H1 H2.
Left; Exact (H2 H0).
```

```
Intros hyp; Right.
Unfold not;Intros p.
Case (H P).
Intros H0 H1.
Rewrite (H0 p) in hyp; Inversion hyp.
Qed.
```

4.3 Back to the results of the section 3

We can wonder whether the trick consisting in eliminating the existential quantifiers whereas the goal to prove is `False` makes useless any distinction between the constructive data of an object and the simple data of its existence.

Could we use, in our case, the constructions of Werner/Lacas/Diaconescu to show the inconsistency of the existence of the section? It seems that the answer is negative.

Indeed the construction of the quotient of the section 3.1 depends on a given proposition `P`, we thus build a term of type

$$H_0 : \quad \forall P : Prop,\ \forall s : E/R_P \rightarrow E,\ \forall x\ c(s(x)) == x \Rightarrow \{P\} + \{\tilde{}P\}.$$

In addition, we have in our context a proposition ensuring us, for all P, the existence of a section :

$$H_1 : \forall P : Prop,\ \exists s : E/R_P \rightarrow E \ \mid\ \forall x\ c(s(x)) == x.$$

So there is two possibilities to continue the proof. Either we cut with `(P:Prop){P}+{~P}`, and then introduce `P`; but the current goal (`{P}+{~P}`) is now in *Set* and we cannot eliminate the existential quantifier of `(H1 P)` any more. Or we try first (before any cut) to eliminate the existential quantifier of `H1`, but in order to prove later `(P:Prop){P}+{~P}` we have to do that for all propositions, which is impossible.

As we see, constructions of section 4 make it possible to go further (and so prove the inconsistency) because they treat all the propositions with only one quotient set.

5 Inconsistency of the axiom of choice with the excluded middle in *Prop*

Independently of the concepts of the quotient type, the proofs of section 4 can be used to show that the axiom of choice with values in `Prop` is contradictory with the excluded middle (in `Prop`).

The axiom of (unique) choice which we consider here is as follows:

```
Axiom choice :
   (A,B : Type)(R:A->B->Prop)
   ((x : A)  (exT B [y : B]  (R x y))) ->
   ((x : A) (y1, y2 : B) (R x y1) -> (R x y2) ->  y1 == y2) ->
   (exT A ->  B [f : A ->  B] (x : A)  (R x (f x))).
```

It affirms, for two types A and B, that if for all x of A there is single y such as $(R\ x\ y)$ (where R is a relation between elements of A and B) then there exists a function $f : A \to B$ such that for all x we have $(R\ x\ f(x))$.

Provided with this axiom and the excluded middle in `Prop`, we can prove once again the existence of the function `Proptobool`. With this intention, we apply the axiom of choice to the relation

$$\begin{aligned} R :\ & \ Prop \to bool \to Prop \\ & := [P : Prop][b : bool] P \leftrightarrow b = true \end{aligned}$$

And we just have to prove

$$\forall P : Prop,\ \exists! b : bool,\ P \leftrightarrow b = true$$

what is done easily with the excluded middle: it suffices to distinguish between P or $\tilde{}P$ and to choose `true` or `false` consequently.

```
Lemma existence_decision:
 (!exT ? [f : Prop ->  bool] (P : Prop)  (iff P (f P) == true)).
Apply (!AC ? ? [P : Prop] [b : bool]  (iff P b == true)).
Intros P; Elim (EM P).
Intros p; Exists true; Split; Intuition.
Intros p; Exists false; Split; Intuition.
Inversion H.
Intros x y1 y2.
Elim (EM x).
Intros xx H0 H;Elim H0;Elim H;Intros;Transitivity true;Intuition.
Case y1; Case y2; Intuition.
Elim (H (H3 (refl_eqT bool true))).
Qed.
```

We build then a term of type `False` starting from the existence of `Proptobool` by taking again the proof of section 4.2.

6 Conclusion

We thus showed by using a single quotient type, that even the proposition stating the existence of a section for the canonical surjection of a quotient, leads to a contradiction in `Coq` , and thus that the morphism θ is not, in general, surjective in the CIC with quotient types. In this direction the expressivity of the quotient types in `Coq` seems definitively weaker than that of the mathematical quotients.

It is the kind *Set* which is problematic, and more precisely its impredicativity: indeed, the impredicativity of *Set* is essential in the proof of the inconsistency of `(P:Prop){P}+{~P}` and without it, the classical boolean model makes possible to show that the surjectivity of θ is not contradictory any more. One of the solutions to allow `Coq` to have quotient types with the same properties on functional spaces as the mathematical quotients would be thus to extinguish the impredicativity of *Set*.

Let us finish by giving two alternatives to quotient types to represent quotient sets in `Coq` . The first one consists in representing sets not as a type with Leibnitz equality but as a type with an equivalence relation : a Setoïd. So a quotient set is simply a setoid with 'another' equivalence relation as equality. It is currently the most used solution, although it's very tiresome to work with. The second possibility is to use the work of Pierre Courtieu [Cou01] : quotients with computable "canonical form", to have quotient types with constructive choice, but the class of the quotients we can thus formalize is restricted.

References

[BarBer96] F. Barbanera and S. Berardi. *Proof-Irrelevance out of Exluded-Middle and Choice in the Calculus of Constructions.* Journal of Functional Programming, pages 519–525, 1996.

[Bar95] G. Barthe. *Extensions of pure type systems.* In M. Dezani-Ciancaglini and G. Plotkin, editors, Proceedings of TLCA'95, volume 902, pages 16–31. Springer-Verlag, 1995.

[Bou97] S. Boutin. *Réflexions sur les quotients.* PhD thesis, Thèse de doctorat, Université de Paris 7, 1997.

[Cou01] P. Courtieu. *Représentation d'algèbres non libres en théorie des types.* PhD thesis, Thèse de doctorat, Université Paris-Sud, 2001.

[Dia75] R. Diaconescu. *Axiom of choice and complementation.* in Proceedings of A.M.S., vol. 51, pages 176-178, 1975.

[Geu01] H. Geuvers. *Inconsistency of classical logic in type theory.* Note. `http://www.cs.kun.nl/ herman/note.ps.gz`

[Hof95] M. Hofmann. *Extensional concepts in intensional type theory.* PhD thesis, LFCS Edinburgh, 1995.

[LacWer99] S. Lacas and B. Werner. *Which choices imply the Excluded Middle?* preprint, 1999.

[Mai98] Maria Emilia Maietti. *About Effective Quotients in Constructive Type Theory.* TYPES'98, LNCS 1657,pp 164-178,1999.

[Pot00] Loïc Pottier. *Quotients dans le CCI.* Rapport de Recherche INRIA RR-4053. `http://www-sop.inria.fr/rapports/sophia/RR-4053.html`