Two Set-based Implementations of Quotients in Type Theory

Niccolò Veltri

Institute of Cybernetics, Tallinn

Theory Days in Rogosi, 08 January 2015
In TT (a framework for constructive mathematics) there are no quotient types.

Quotients are typically represented by setoids.

A setoid is a pair \((A, R)\) where \(A\) is a set and \(R\) is an equivalence relation on \(A\).

A map between setoids \((A, R)\) and \((B, S)\) is a map \(f : A \rightarrow B\) compatible with the relations, i.e. if \(R a b\) then \(S (f a) (f b)\).

The implementation of quotients as setoids forces us to lift every type former to setoids, e.g. the type formers of lists and trees must become setoid transformers.

In several applications it is preferable to work with sets instead of setoids, e.g. formalisation of category theory.
We present two different frameworks for reasoning about set-based quotients, i.e. quotients as types:

1. Quotients as particular inductive types, inspired by quotient types in Hofmann’s Ph.D. thesis.
2. An impredicative encoding of quotients, reminiscent of Church numerals and more generally of encodings of inductive types in Calculus of Constructions.

We will use the notation of the dependently typed programming language Agda.
Let $A$ be a set and let $R$ be an equivalence relation on $A$ (i.e. reflexive, symmetric, transitive binary relation).

Equivalence class of $a \in A$:

$$[a] = \{x \in A \mid R \ a \ x\}$$

Quotient set:

$$A/R = \{[a] \mid a \in A\}$$

One can extract a representative from any equivalence class.

For all $a, a' \in A$ such that $[a] = [a']$ we have $R \ a \ a'$. 
Universal property of quotients

Let $A$ be a set and let $R$ be an equivalence relation on $A$.

Let $B$ be a set and $f: A \to B$ map compatible with $R$, i.e. $f \ x \cong f \ y$ if $R \ x \ y$.

There exists a unique map $g := \text{qelim} \ (\lambda \rightarrow B) \ f \ p$ such that the following triangle commutes:

$$
\begin{array}{c}
A \\
\downarrow \\
A/R
\end{array}
\xymatrix{
A \ar[r]^f \ar[d]_-{[\cdot]} & B \\
A/R & \\
& B \\
\downarrow_{g} & & \\
\end{array}
$$

where $p$ is a proof of compatibility of $f$. 
Induction principle for quotients

- Let $A$ be a set and let $R$ be an equivalence relation on $A$.
- Let $P$ be a predicate on $A/R$.
- If for all $a : A$ we can construct a proof of $P \ [a]$, then for all $x : A/R$ we have a proof of $P \ x$. 
Quotients as inductive types ($i$)

- Quotient types:

```lean
record Quotient (A : Set)(R : EqR A) : Set where
  field
  Q : Set
  abs : A → Q

  compat : (B : Q → Set)
    (f : (a : A) → B (abs a)) → Set
  compat B f = {x y : A} → R x y → f x ≃ f y
```
Quotients as inductive types (ii)

field

sound : compat (λ _ → Q) abs
qelim : (B : Q → Set)
  (f : (a : A) → B (abs a))
  (p : compat B f)
  (q : Q) → B q
qcomp : (B : Q → Set)
  (f : (a : A) → B (abs a))
  (p : compat B f)
  (a : A) → qelim B f p (abs a) ≃ f a

We postulate the existence of Quotient A R for any type A and equivalence relation R on A.
Derived results: uniqueness

- \( \text{qelim } B \ f \ p \) is the only map of type \( Q \rightarrow B \) that satisfies the equation in \( \text{qcomp} \).

\[
\text{qelimUniq} : (B : Q \rightarrow \text{Set}) \\
(f : (a : A) \rightarrow B \ (\text{abs } a)) \\
(p : \text{compat } B \ f) \\
(g : (q : Q) \rightarrow B \ q) \\
(r : (a : A) \rightarrow g \ (\text{abs } a) \equiv f \ a) \\
(q : Q) \rightarrow \text{qelim } B \ f \ p \ q \equiv g \ q
\]
Comparison with classical quotients (i)

- In general we cannot prove:

  \[
  \text{complete} : \{x y : A\} \rightarrow \text{abs } x \equiv \text{abs } y \rightarrow R x y
  \]

  \[
  \text{surjAbs} : (y : Q) \rightarrow \Sigma A (\lambda x \rightarrow \text{abs } x \equiv y)
  \]

  \[
  \text{isSectionAbs} : \Sigma (Q \rightarrow A) (\lambda f \rightarrow (x : Q) \rightarrow \text{abs } (f x) \equiv x)
  \]

- Note that \text{surjAbs} \rightarrow \text{isSectionAbs} holds, it is a consequence of the axiom of choice.
Comparison with classical quotients (\textit{ii})

- Suppose we have squash types in our theory (simulating the kind \textit{Prop} in Coq).
- For all $p r : ||A||$, $p \cong r$.
- Intuition: $||A||$ is inhabited iff $A$ is provable.
- One can prove:

$\text{surjAbs}' : (y : Q) \rightarrow ||\Sigma A (\lambda x \rightarrow \text{abs } x \cong y)||$

- But not:

$\text{isSectionAbs}' : ||\Sigma (Q \rightarrow A) (\lambda f \rightarrow (x : Q) \rightarrow \text{abs } (f x) \cong x)||$

- $\text{surjAbs}' \rightarrow \text{isSectionAbs}'$ would be a consequence of “internal” axiom of choice, which does not hold in \textit{TT}.
Comparison with classical quotients (iii)

- Let \( S \) relation on \( B \to A \) be the pointwise extension of \( R \), i.e. \( S \not\sim g \) iff for all \( x : B, R \ (fx) \ (gx) \).
- We can always define a map \( \theta : (B \to A)/S \to (B \to A/R) \) such that:

\[
\begin{array}{ccc}
B \to A & \xrightarrow{\text{abs}_R \circ -} & B \to A/R \\
\downarrow{\text{abs}_S} & & \downarrow{\theta} \\
(B \to A)/S & & \\
\end{array}
\]

- Classically \( \theta \) is surjective.
- In TT it is equivalent to \text{isSectionAbs}.
- Question: can we postulate it?
Comparison with classical quotients (iv)

- Postulating completeness for all sets and equivalence relations leads to classical logic, excluded middle is derivable (Maietti '99).

  \[
  \text{complete} \colon \{x, y : A\} \rightarrow \text{abs } x \simeq \text{abs } y \rightarrow R x y
  \]

- Fundamental ingredient in the proof is uniqueness of propositional equality proofs (tacitly assumed and present in Agda).
Natural numbers in Calculus of Constructions (CoC) $(i)$

- In impredicative systems (like CoC) no inductive types.
- Such types can be encoded as (weakly) initial algebras.

\[
\begin{array}{c}
\begin{array}{ccc}
1 + N & \xrightarrow{\mathrm{[zero},\mathrm{suc},\mathrm{c}]} & N \\
\downarrow & & \downarrow \\
1 + X & \xrightarrow{f} & X \\
\end{array}
\end{array}
\]

with $f = [z, s]$, $z : 1 \to X$, $s : X \to X$. 
Natural numbers in Calculus of Constructions (CoC) (ii)

- **Church numerals:**

  \[ N : \text{Set} \]
  \[ N = (X : \text{Set}) \rightarrow X \rightarrow (X \rightarrow X) \rightarrow X \]

  \[ \text{zero} : N \]
  \[ \text{zero} = \lambda X z s \rightarrow z \]

  \[ \text{succ} : N \rightarrow N \]
  \[ \text{succ} n = \lambda X z s \rightarrow s (n X z s) \]

  \[ \text{fold} : (X : \text{Set}) \rightarrow X \rightarrow (X \rightarrow X) \rightarrow N \rightarrow X \]
  \[ \text{fold} X z s n = n X z s \]
Let $A$ and $R$ be a set and an equivalence relation on it. We first define (non-dependent) compatibility.

\[
\text{compat} : (B : \text{Set}) (f : A \to B) \to \text{Set}
\]

\[
\text{compat } B \ f = \{x \ y : A\} \to R \ x \ y \to f \ x \sim f \ y
\]

We define the quotient of $A$ over $R$ as the following type.

\[
Q : \text{Set}
\]

\[
Q = (B : \text{Set}) (f : A \to B) \to \text{compat } B \ f \to B
\]

In other words $Q$ is a polymorphic function which assigns, to every type $B$ equipped with a compatible function $f : A \to B$, an element of $B$. 

Quotients via impredicative polymorphism (i)
Quotients via impredicative polymorphism (ii)

- One can then define the constructor abs.
  
  \[
  \text{abs} : A \rightarrow Q \\
  \text{abs } a = \lambda B f p \rightarrow f \ a
  \]

- One can prove the non-dependent elimination rule for \( Q \), which satisfies its computation rule definitionally.
  
  \[
  \text{qelim} : (B : \text{Set})(f : A \rightarrow B)(p : \text{compat } B \ f) \rightarrow Q \rightarrow B \\
  \text{qelim } B f p q = q B f p
  \]

  \[
  \text{qcomp} : (B : \text{Set})(f : A \rightarrow B)(p : \text{compat } B \ f)(a : A) \rightarrow \\
  \text{qelim } B f p (\text{abs } a) \cong f \ a \\
  \text{qcomp } B f p a = \text{refl}
  \]
Comparison with classical quotients

▶ It is impossible to derive a dependent elimination principle but it is safe to postulate one (Geuvers ’01).

▶ Completeness is provable (for $R : A \to A \to \text{Prop}$).

$$\text{complete} : \{x \, y : A\} \to \text{abs } x \equiv \text{abs } y \to R \, x \, y$$

▶ Postulating $\text{isSectionAbs}$ (in Prop) for all sets and equivalence relations leads to excluded middle in $\text{Set}$ (Chicli et al. ’03).

$$\text{isSectionAbs} : \Sigma (Q \to A) (\lambda f \to (x : Q) \to \text{abs } (f \, x) \equiv x)$$

▶ Impredicative $\text{Set} +$ excluded middle in $\text{Set}$ lead to inconsistency (Hurkens ’95, Geuvers ’01), i.e. equality of all terms in a same type.
Conclusions

▶ We presented two different implementation of quotient types in TT.
▶ Both set-based and therefore different from the setoid approach.
▶ Hofmann’s extension of CoC with quotient types is consistent, therefore the same holds for our first implementation.
▶ Postulating dependent elimination for impredicative encodings of inductive types is safe. Similarly this can be extended to our impredicative encoding of quotients.
▶ The analysis of other metaproperties is work in progress.