

A Coq Tactic for One Step Conditional Rewriting

Nijmegen, 1st November 2004

Claudio Sacerdoti Coen*

<sacerdot@lix.polytechnique.fr>

* Project PCRI, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

Overview

1. Motivations
2. Theory and implementation
3. Applications (with demos): later on demand
4. Conclusions and future works

Overview

1. Motivations
2. Theory and implementation
3. Applications (with demos): later on demand
4. Conclusions and future works

Rewriting in Coq [1/2]

Tactics for multi-steps rewriting

- **autorewrite**

- computes the normal form given a (stratified) TRS
- a stratified TRS is an ordered list of sets of oriented equations (one step conditional rewriting rules)

$$L_i : \forall \bar{x} : \bar{T}, E_1 R E_2$$

s.t. “rewrite \rightarrow ” can be applied to L_i

- **Alvarado's Elan trace replayer**

- the TRS is given as a set of first order equalities

$$L_i : \forall \bar{x} : \bar{T}, E_1 = E_2 \quad (= \text{ is Leibniz equality})$$

- the first order equalities are compiled in meta-level rewrite functions of type $Term\ s \rightarrow Term\ s$ (rewrite is not used)

Rewriting in Coq [2/2]

Tactics for one-step rewriting

- `rewrite`
 - applies lemmas of type $\forall \bar{x} : \bar{T}, E_1 = E_2$
 - provided that the goal satisfies a particular (typing) condition
- `[setoid_]rewrite` (by Clement Renard)
 - applies lemmas of type $[\forall \bar{x} : \bar{T},]E_1 \equiv E_2$
where \equiv is an **equivalence relation**
 - provided that the goal satisfies several syntactic and typing conditions
- **Can we generalize more?**
 - lemmas of type $\forall \bar{x} : \bar{T}, E_1 R E_2$, R not an equivalence relation

Dropping symmetry

From equivalence relations \equiv to reductions \triangleright

E.g.: β -reducing sub-terms in a development about λ -calculus

$$(\lambda y.(f(\lambda x.x y))) \triangleright_{\beta} (\lambda y.(f y))$$

(the usual approach requires three lemmas applications; no support for multi-steps rewriting)

Not only reductions.

E.g.: (large) inequalities \leq, \geq

$$\epsilon \geq 0 \rightarrow \epsilon/2 \geq 0/2$$

but $\epsilon \geq 0 \rightarrow -2 * 0 \geq -2 * \epsilon$

(contexts as monotone increasing/decreasing functions)

Dropping transitivity

Every equivalence/reduction relation is transitive ...

...but the tactic does not use this hypothesis ...

...except to prove that the relation is itself a valid context.

Dropping reflexivity

From equivalence relations \equiv to PERs \approx

Partial setoids are a useful tool that captures at once subtyping and partitioning.

An element $x : T$ is **proper** iff $x \approx x$ (also written $x \in T$).

The type $\{x : T \mid x \approx x\}$ is a subtype of T .

Functions are totally defined over T but they “make sense” only on the domain of proper elements. (cfr. $(1/0)\%R$ in Coq).

Much easier to use (but strictly less expressive) than total setoids on sigma types $\Sigma x : T. P x$

E.g. $E_1 \approx E_2 \rightarrow (f x E_1) \approx (f x E_2)$ iff $x \approx x$

the tactic can generate side conditions that must be proven

Overview

1. Motivations
2. Theory and implementation
3. Applications (with demos): later on demand
4. Conclusions and future works

Generalizing the rewrite tactic

$$\boxed{\frac{H:E_1=E_2}{C[E_1]} \Longrightarrow \frac{H:E_1=E_2}{C[E_2]}}$$

by $(\text{eq_ind_r}' \text{ - } \lambda x.C[x] \ E_2 \ E_1 \ H) : C[E_2] \rightarrow C[E_1]$

provided that the context $\lambda x.C[x]$ is well typed

$\text{eq_ind_r}'$:

$$\forall (T : Type)(C : T \rightarrow Prop)(x \ y : T), y =_T x \rightarrow (C \ x \rightarrow C \ y)$$

Generalizing the rewrite tactic

$$\boxed{\frac{H:E_1=E_2}{C[E_1]} \implies \frac{H:E_1=E_2}{C[E_2]}}$$

by `(eq_ind_r' - [[C]] [[E2]] [[E1]] H) : C[E2] → C[E1]`

provided that `[[C]]` is well typed

`eq_ind_r'`:

$\forall (T : Type)(C : \text{context } T)(x \ y : \text{term } T), y \equiv_T x \rightarrow (C[x] \rightarrow C[y])$

`term T := T`

`≡T := =`

`context T := T → Prop`

`C[E] := C E`

Property: `[[C]] [[E]] ≅ C[E]`

Generalizing the rewrite tactic

$$\boxed{\frac{H:E_1 \equiv_T E_2}{C[E_1]} \implies \frac{H:E_1 \equiv_T E_2}{C[E_2]}}$$

by (eq_ind_r' $T \equiv_T \llbracket C \rrbracket \llbracket E_2 \rrbracket \llbracket E_1 \rrbracket H$) : $C[E_2] \rightarrow C[E_1]$

provided that $\llbracket C \rrbracket$ is well typed

eq_ind_r':

$\forall (T : Type) (\equiv_T : T \rightarrow T \rightarrow Prop) (C : context T) (x y : term T), y \equiv_T x \rightarrow (C[x] \rightarrow C[y])$

term $T : Type$

$C[E] : context T \rightarrow term T \rightarrow Prop$

context $T : Type$

Property: $\llbracket C \rrbracket \llbracket \llbracket E \rrbracket \rrbracket \cong C[E]$

Generalizing the rewrite tactic

$$\boxed{\frac{H:E_1 R_T E_2}{C[E_1]} \implies \frac{H:E_1 R_T E_2}{C[E_2]}}$$

by $(\text{eq_ind_r}' T R_T Prop (\rightarrow) \llbracket C \rrbracket \llbracket E_2 \rrbracket \llbracket E_1 \rrbracket H) : C[E_2] \rightarrow C[E_1]$

provided that $\llbracket C \rrbracket$ is well typed

$\text{eq_ind_r}'$:

$$\forall (T : Type) (R_T : T \rightarrow T \rightarrow Prop) (T' : Type) (R_{T'} : T' \rightarrow T' \rightarrow Prop) \\ (C : \text{morphism } R_T R_{T'}) (x y : \text{term } T), y R_T x \rightarrow C[x] R_{T'} C[y]$$

term $T : Type$

morphism $R R_{T'} : Type$

$C[E] : \text{morphism } R_T R_{T'} \rightarrow \text{term } T \rightarrow T'$ Property: $\llbracket C \rrbracket \llbracket \llbracket E \rrbracket \rrbracket \cong C[E]$

Generalizing the rewrite tactic

$$\boxed{\frac{H:E_1 R_T E_2}{C[E_1]} \Longrightarrow \frac{H:E_1 R_T E_2}{C[E_2]}}$$

by (monotony $T R_T Prop \triangleleft (\rightarrow) \llbracket C \rrbracket \llbracket E_2 \rrbracket \llbracket E_1 \rrbracket H$) : $C[E_2] \rightarrow C[E_1]$

provided that $\llbracket C \rrbracket$ is well typed

monotony:

$\forall (T : Type)(R_T : T \rightarrow T \rightarrow Prop)(T' : Type)(R_{T'} : T' \rightarrow T' \rightarrow Prop)$
 $(d : \mathbb{B})(C : \text{morphism } R_T^d R_{T'})(x y : \text{term } T), x R_T^d x \rightarrow C[x] R_{T'} C[y]$

term $T : Type$

morphism $R_T^d R_{T'} : Type$

$C[E] : \text{morphism } R_T^d R_{T'} \rightarrow \text{term } T \rightarrow T'$ Property: $\llbracket C \rrbracket \llbracket E \rrbracket \cong C[E]$

Generalizing the rewrite tactic

$$\boxed{\frac{H:E_1 R_T E_2}{C[E_1]} \Longrightarrow \frac{H:E_1 R_T E_2}{C[E_2]}}$$

by $(\text{monotony } T \ R_T \ Prop \triangleleft \ (\rightarrow) \ \llbracket C \rrbracket \ \llbracket E_2 \rrbracket \ \llbracket E_1 \rrbracket \ H) : C[E_2] \rightarrow C[E_1]$

provided that $\llbracket C \rrbracket$ is well typed

monotony:

$$\forall (T : Type) (R_T : T \rightarrow T \rightarrow Prop) (T' : Type) (R_{T'} : T' \rightarrow T' \rightarrow Prop) \\ (d : \mathbb{B}) (C : \text{morphism } R_T^d \ R_{T'}) (x \ y : \text{term } T), x \ R_T^d \ y \rightarrow C[x] R_{T'} C[y]$$

term $T : Type$

morphism $R_T^d \ R_{T'} : Type$

$C[E] : \text{morphism } R_T^d \ R_{T'} \rightarrow \text{term } T \rightarrow T'$ Property: $\llbracket C \rrbracket \llbracket E \rrbracket \cong C[E]$

Morphisms and signatures

Given $(T_1, R_1), \dots, (T_n, R_n), (T, R)$, a morphism of signature

$$R_1 \Rightarrow^{d_1} \dots \Rightarrow^{d_{n-1}} R_n \Rightarrow^{d_n} R$$

is a function

$$f : T_1 \rightarrow \dots \rightarrow T_n \rightarrow T$$

that is monotone in its arguments:

$$\forall x_1 x'_1, x_1 R_1^{d_1} x'_1 \rightarrow \dots \rightarrow \forall x_n x'_n, x_n R_n^{d_n} x'_n \rightarrow (f x_1 \dots x_n) R (f x'_1 \dots x'_n)$$

Lemma: if R_i is symmetric and f is a morphism covariant in R_i , then it is also contravariant in R_i .

Morphisms composition

Given

$$f : R_1 \Rightarrow^{d_1} \dots \Rightarrow^{d_{n-1}} R_n \Rightarrow^{d_n} R$$

and

$$f_i : R_1^i \Rightarrow^{d_1^i} \dots \Rightarrow^{d_{n_i}^i} R_{n_i} \Rightarrow^{d_{n_i}^i} R_i \text{ for } i = 1, \dots, n$$

the compound function

$$f \circ \langle f_1, \dots, f_n \rangle$$

is a morphism of signature

$$R_1^1 \Rightarrow^{d_1^1=d_1} \dots \Rightarrow^{d_{n_1}^1=d_1} R_{n_1}^1 \Rightarrow^{d_{n_1}^1=d_1}$$

...

$$R_1^n \Rightarrow^{d_1^n=d_n} \dots \Rightarrow^{d_{n_n}^n=d_n} R_{n_n}^n \Rightarrow^{d_{n_n}^n=d_n} R$$

The identity and 0-ary morphisms

Given (T, R) :

- the identity function $\lambda x : T.T$ is a morphism of signature $R \Rightarrow^{\triangleright} R$;
- the 0-ary constant function t is a morphism of signature R if and only if tRt .

Lemma: if R is a reflexive relation than every 0-ary function of type T is a morphism of signature T .

Observation: if (T, R) is a partial setoid then the 0-ary functions of type T are the proper elements of the setoid.

Contraction (simplified)

Given a morphism

$$f : R \Rightarrow^{\triangleright} \dots \Rightarrow^{\triangleright} R \Rightarrow^{\triangleleft} R \Rightarrow^{\triangleleft} \dots R \Rightarrow^{\triangleleft} R'$$

the function $\lambda x.\lambda y.(f\ x\ \dots\ x\ y\ \dots\ y)$ has signature

$$R \Rightarrow^{\triangleright} R \Rightarrow^{\triangleleft} R'$$

Extension to multiple arguments to be contracted and to arbitrary permutation of the arguments are left to the reader.

Observation: we cannot contract arguments that occur in covariant position with arguments that occur in contravariant position

The quoting function

Given the current goal G and a term E to rewrite it must build identify a context C such that $G \cong C[E]$ and $\llbracket G \rrbracket$ is a well typed morphism.

$$\frac{H:x < y}{x * 2 < -x + x - x / x}$$

$$\begin{aligned} *, + : & \quad < \Rightarrow^{\triangleright} =_{\mathbb{R}} \Rightarrow^{\triangleright} < \\ < : & \quad < \Rightarrow^{\triangleleft} =_{\mathbb{R}} \Rightarrow^{\triangleleft} < \\ / : & \quad < \Rightarrow^{\triangleright} <_0 \Rightarrow^{\triangleleft} < \\ & \quad =_{\mathbb{R}} \Rightarrow^{\triangleright} =_{\mathbb{R}_0} \Rightarrow^{\triangleleft} < \end{aligned}$$

$$\begin{aligned} & =_{\mathbb{R}} \text{ is reflexive (???)} \\ & x =_{\mathbb{R}_0} x \text{ iff } x \neq_{\mathbb{R}} 0 \end{aligned}$$

- $C_1 := [] * 2 < -[] + x - [] / x : \quad < \Rightarrow^{\triangleleft} < \quad \text{iff } x \in \mathbb{R}_0 \text{ iff } x =_{\mathbb{R}_0} x \text{ iff } x \neq_{\mathbb{R}} 0$
- $C_2 := x * 2 < -[] + x - x / x : \quad < \Rightarrow^{\triangleleft} <$
- $C_3 := x * 2 < -x + [] - x / x : \quad < \Rightarrow^{\triangleright} < \quad \text{(but not the right variance)}$
- $C_4 := x * 2 < -x + x - x / [] : \quad <_0 \Rightarrow^{\triangleright} < \quad \text{(but not the right relation)}$
- $C_5 := [] * 2 < -x + [] + 1 / x : \quad \perp \quad \text{(cannot contract, different types)}$
- $C_6 := x * 2 < -x + [] - x / [] : \quad \perp \quad \text{(cannot contract, different variances)}$

A syntactic restriction [1/2]

Given a set S of user-registered quantified morphisms and a term t that must be rewritten, the quoting function $\llbracket _ \rrbracket$ handles only the set of terms that:

1. contains fully applied occurrences all the identity and 0-ary morphisms s.t. t does not occur in T ; the proof of reflexivity of the 0-ary morphism is synthesized as a fresh metavariable (a new goal), unless the relation is known to be reflexive
2. contains fully applied occurrences of all the morphisms in S with the restriction that t does not occur in the arguments that instantiate the quantifiers

A syntactic restriction [2/2]

3 contains all the morphisms of signature

$=_{T_1} \Rightarrow^{\langle \triangleright \rangle} \dots \Rightarrow^{\langle \triangleright \rangle} =_{T_n} \Rightarrow^{\langle \triangleright \rangle} =_T$ such that $=_{T_1}, \dots, =_{T_n}, =_T$ are all

Leibniz equalities and such that the function is obtained by

β -abstracting a term

4 is closed by syntactic composition

The roles of symmetry and transitivity

Lemma: if (T, R) is such that R is **transitive**, then R is a morphism of signature $R \Rightarrow^{\triangleleft} R \Rightarrow^{\triangleright} Prop$ (where $Prop$ is the carrier of $(Prop, (\rightarrow))$)

If (T, R) is such that R is **reflexive**, then every morphism covariant in an argument of type T is also contravariant. Thus the user must register an (exponentially) lower number of morphisms.

Overview

1. Motivations
2. Theory and implementation
3. Applications (with demos): later on demand
4. Conclusions and future works

Overview

1. Motivations
2. Theory and implementation
3. Applications (with demos): later on demand
4. Conclusions and future works

Conclusions

1. a tactic for generalized (restricted) conditional rewriting
2. that generalizes setoid rewriting
3. that generalizes reasoning on partial setoids
4. that simulates reduction
5. great reduction in proof script sizes
6. a reflexive implementation (proof size reduction; easier to recognize and render in MKM tools; easier to port the tactic to others proof assistants based on CIC)
7. the same ideas can be ported to other proof assistants
8. compatible w.r.t. the Coq module system (not trivial!)

Questions/future work [1/2]

1. find a suitable name for the tactic
2. try to generate an informative error message when the tactic fails
3. should subrelations be implemented?
4. (Bas Spitters): should “morphisms” in the context be used?