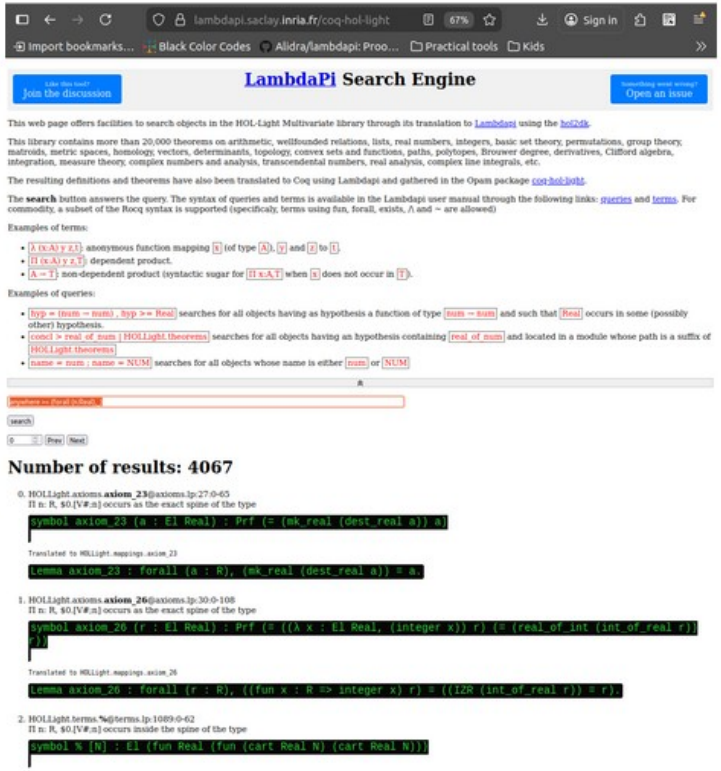
The HOL-Light library of Multivariate Real analysis in Rocq

Abdelghani Alidra
Claudio Sacerdoti Coen
Frédéric Blanqui

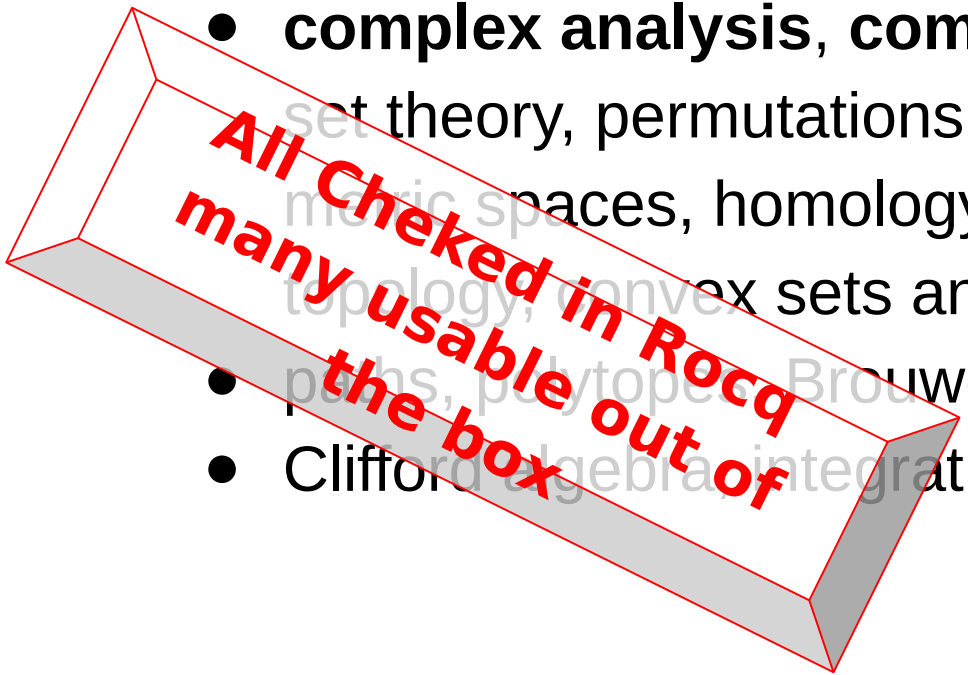
A search tool for Coq-Hol-Light

https://lambdapi.saclay.inria.fr/coq-hol-light



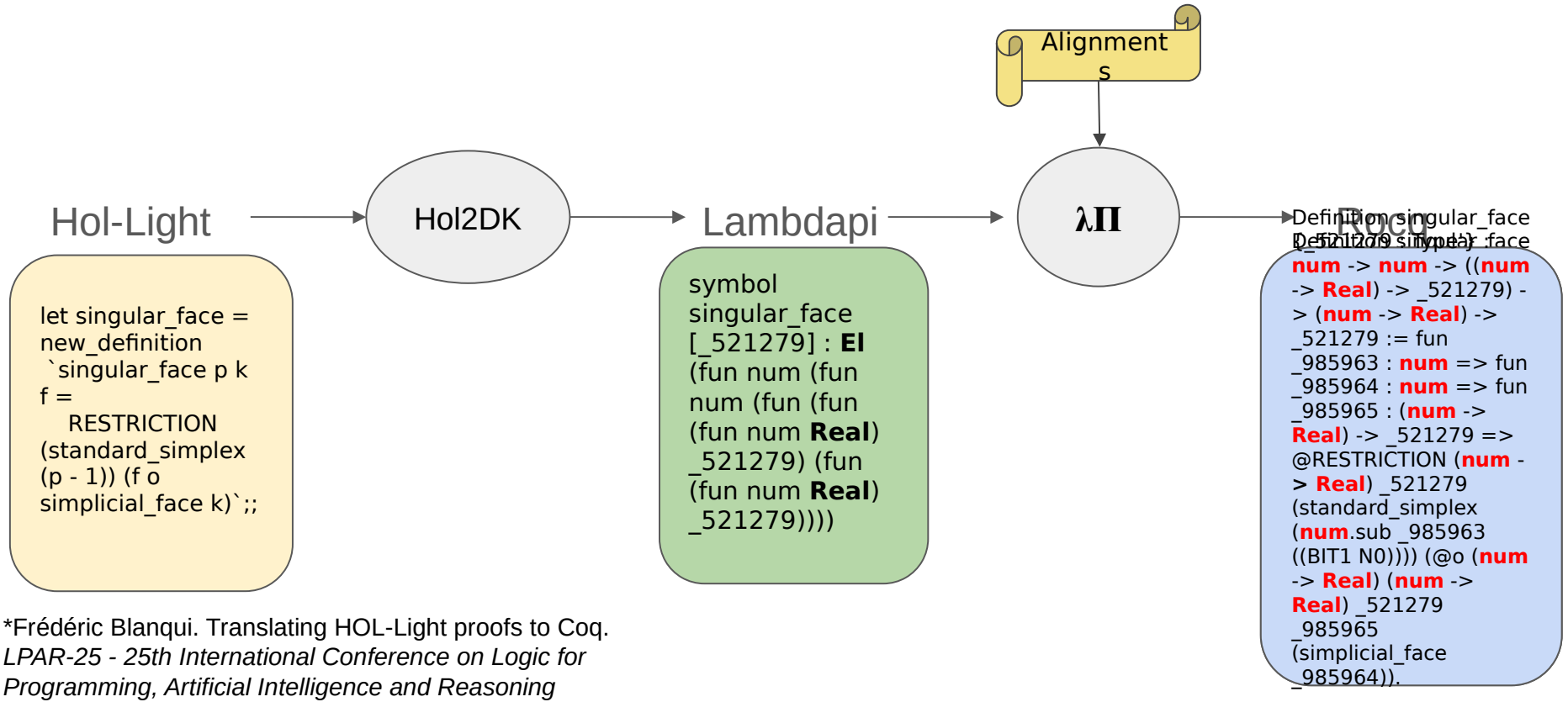
Coq-Hol-Light: 20,000+ theorems

- **Real analysis,**
- **complex analysis**, **complex line integrals**,Basic set theory, permutations, **group theory**, matroids, metric spaces, homology, **vectors**, determinants, topology, convex sets and functions,
- paths, polytopes, Brouwer degree, derivatives,
- Clifford algebra, integration, measure theory, etc.



https://github.com/Deducteam/coq-hol-light

Alignments to achieve true interoperability*



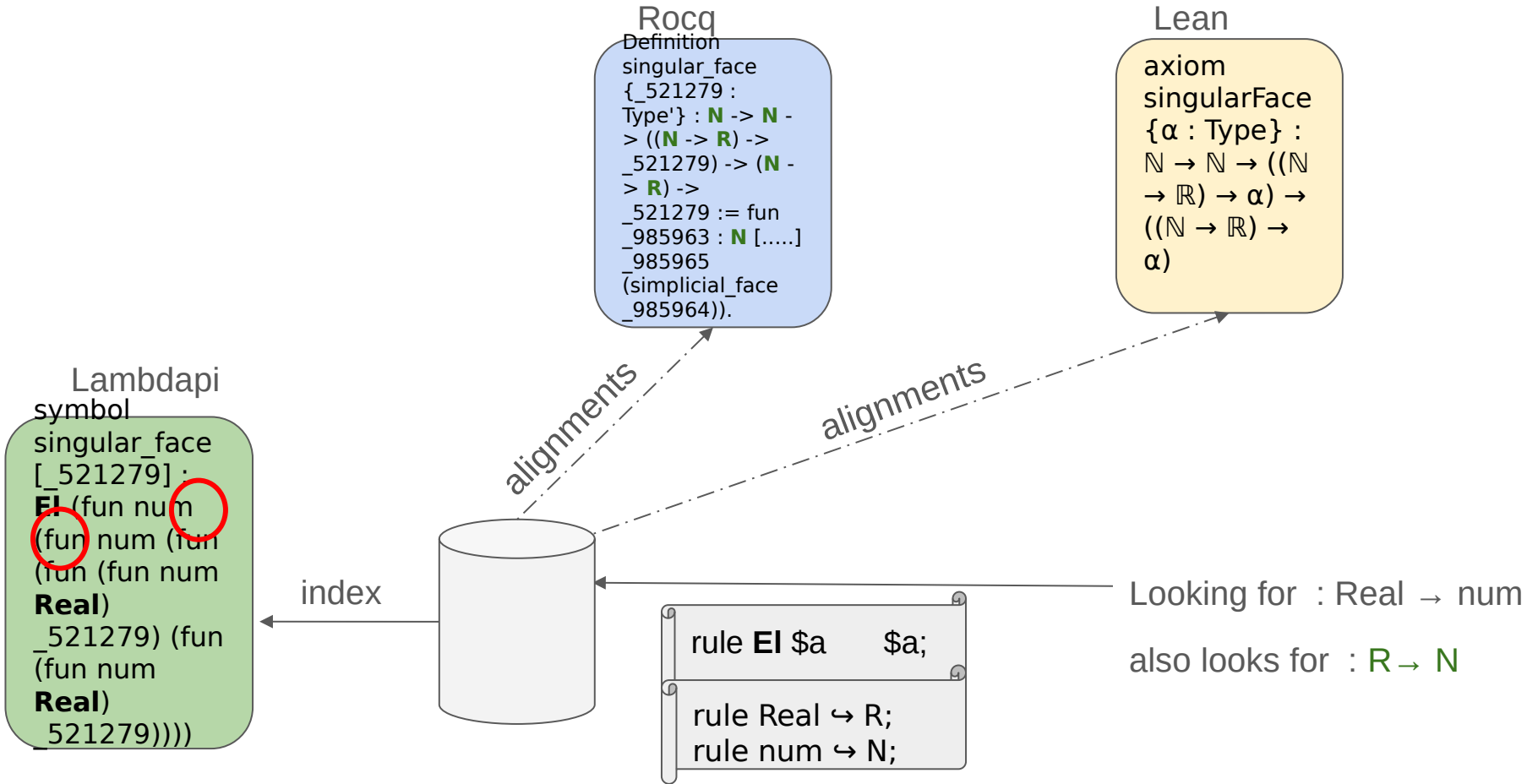
*Frédéric Blanqui. Translating HOL-Light proofs to Coq. *LPAR-25 - 25th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*

Alignments : bridge between libraries

- Alignment : a mapping between an item in the source logic and its counterpart in the target logic
  - **types:** unit, prod, list, option, sum, ascii, **N, R, Z, cart**
  - functions on N: pred, **add, mul, pow,** le, lt, ge, gt, max, min, sub, div, modulo, even, **odd**, **factorial**
  - functions on Z: IZR, le, lt, ge, gt, opp, **add, sub, mul**, abs, sgn, max, min, **pow,** div, rem, divide, coprime, **gcd, lcm**
  - functions on list: app, rev, map, removelast, In, hd, tl
  - functions on R: Rle, **Rplus, Rmult, Rinv**, Ropp, Rabs, Rdiv, Rminus, Rge, Rgt, Rlt, Rmax, Rmin, IZR, Rsgn, Rmod_eq, Rpow, **sqrt**
  - functions on cart: dot, vector_norm, distance, **vsum, sums**

- In Hol-Light, types and functions are **axiomatized**
- Alignments include proofs that target logic concepts **satisfy the corresponding axioms**

A tool to search them all



The query language

```
Q  ::=  B  |  Q,Q  |  Q;Q  |  Q|PATH
B  ::=  WHERE      HOW      GENERALIZE?      PATTERN
PATH  ::=  <<  string  >>
WHERE  ::=  name
          |  anywhere
          |  rule  |  lhs  |  rhs
          |  type  |  concl  |  hyp  |  spine
HOW  ::=  >  |  =  |  >=
GENERALIZE  ::=  generalize
PATTERN  ::=  <<  term  possibly  containing
              placeholders  _  (for  terms)
              and  V#  (for  variables)  >>
```

Looking for theorems about rearranging elements of series of reals

hyp = (num → num), hyp >= R

search

0 Prev Next

hyp = (num → num), hyp >= R

Number of results: 22

0. HOLLight.theorems.**thm_ABSOLUTELY_INTEGRABLE_TWIZZLE_EQ**@theorems.lp:15821:0-599
   N → N occurs as the exact hypothesis of the type and
   R occurs inside the hypothesis of the type

```
symbol thm_ABSOLUTELY_INTEGRABLE_TWIZZLE_EQ [M N P] : Prf (∀ (λ f : El (fun (cart Real N) (cart Real P)), (∀ (λ s : El (fun (cart Real M) bool), (∀ (λ p : El (fun num num), (⇒ (∧ (= ((@dimindex M) (@UNIV M)) ((@dimindex N) (@UNIV N))) ((@permutes num) p (… (NUMERAL (BIT1 _0)) ((@dimindex N) (@UNIV N))))) (= ((@absolutely_integrable_on P N) f ((@IMAGE (cart Real M) (cart Real N)) (λ x : El (cart Real M), ((@lambda Real N) (λ i : El num, ((@$ Real M) x (p i))))) s)) ((@absolutely_integrable_on P M) (λ x : El (cart Real M), (f ((@lambda Real N) (λ i : El num, ((@$ Real M) x (p i)))))) s)))))))))
```

Translated to HOLLight.theorems.thm_ABSOLUTELY_INTEGRABLE_TWIZZLE_EQ

```
Axiom thm_ABSOLUTELY_INTEGRABLE_TWIZZLE_EQ : forall {M N' P : Type'}, forall f : (cart R N') -> cart R P, forall s : (cart R M) -> Prop, forall p : N -> N, (((@dimindex M (@UNIV M)) = (@dimindex N' (@UNIV N'))) /\ (@permutes N p (dotdot ((BIT1 N0)) (@dimindex N' (@UNIV N'))))) -> (@absolutely_integrable_on P N' f (@IMAGE (cart R M) (cart R N') (fun x : cart R M => @lambda R N' (fun i : N => @dollar R M x (p i))) s)) = (@absolutely_integrable_on P M (fun x : cart R M => f (@lambda R N' (fun i : N => @dollar R M x (p i)))) s).
```

1. HOLLight.theorems.**thm_CAUCHY_CONVERGENT_SUBSEQUENCE**@theorems.lp:10500:0-330
   N → N occurs as the exact hypothesis of the type and
   R occurs inside the hypothesis of the type

```
symbol thm_CAUCHY_CONVERGENT_SUBSEQUENCE [N] (l : El (cart Real N)) : Prf (∀ (λ x : El (fun num (cart Real N)), (∀ (λ r : El (fun num num), (⇒ (∧ ((@cauchy N) x) (∧ (∀ (λ m : El num, (∀ (λ n : El num, (⇒ (< m n) (< (r m) (r n)))))))) ((@→ N num) ((@o num num (cart Real N)) x r) l sequentially))) ((@→ N num) x l sequentially))))))
```

Translated to HOLLight.theorems.thm_CAUCHY_CONVERGENT_SUBSEQUENCE

```
Axiom thm_CAUCHY_CONVERGENT_SUBSEQUENCE : forall {N' : Type'} (l : cart R N'), forall x : N -> cart R N', forall r : N -> N, ((@cauchy N' x) /\ ((forall m : N, forall n : N, (N.lt m n) -> N.lt (r m) (r n)) /\ (@FImp N' N (@o N N (cart R N') x r) l sequentially))) -> @FImp N' N x l sequentially.
```

Looking for the mean value theorm

hyp >= absolutely_continuous_on ; (hyp >= continuous_on, hyp >= differentiable)

search

0 Prev Next

hyp >= absolutely_continuous_on ; (hyp >= continuous_on, hyp >= differentiable)

Number of results: 40

0. HOLLight.theorems.**thm_ABSOLUTELY_CONTINUOUS_COMPARISON**@theorems.lp:16167:0-587
   absolutely_continuous_on occurs inside the hypothesis of the type

```
symbol thm_ABSOLUTELY_CONTINUOUS_COMPARISON [M N] : Prf (∀ (λ f : El (fun (cart Real unit) (cart Real M)), (∀ (λ g : El (fun (cart Real unit) (cart Real N)), (∀ (λ s : El (fun (cart Real unit) bool), (⇒ (∧ ((@absolutely_continuous_on M) f s) (∀ (λ x : El (cart Real unit), (∀ (λ y : El (cart Real unit), (⇒ (∧ ((@IN (cart Real unit)) x s) (∧ ((@IN (cart Real unit)) y s) (real_lt (drop x) (drop y)))) (real_le ((@distance N) ((@, (cart Real N) (cart Real N)) (g x) (g y))) ((@distance M) ((@, (cart Real M) (cart Real M)) (f x) (f y)))))))))) ((@absolutely_continuous_on N) g s))))))))
```

Translated to HOLLight.theorems.thm_ABSOLUTELY_CONTINUOUS_COMPARISON

```
Axiom thm_ABSOLUTELY_CONTINUOUS_COMPARISON : forall {M N' : Type'}, forall f : (cart R unit) -> cart R M, forall g : (cart R unit) -> cart R N', forall s : (cart R unit) -> Prop, ((@absolutely_continuous_on M f s) /\ (forall x : cart R unit, forall y : cart R unit, ((@IN (cart R unit) x s) /\ ((@IN (cart R unit) y s) /\ (Rlt (drop x) (drop y)))) -> Rle (@distance N' (@pair (cart R N') (cart R N') (g x) (g y))) (@distance M (@pair (cart R M) (cart R M) (f x) (f y))))) -> @absolutely_continuous_on N' g s.
```

1. HOLLight.theorems.**thm_ABSOLUTELY_CONTINUOUS_DIFFERENTIABLE_BV_GEN**@theorems.lp:16908:0-579
   continuous_on occurs inside the hypothesis of the type and
   differentiable occurs inside the hypothesis of the type

```
symbol thm_ABSOLUTELY_CONTINUOUS_DIFFERENTIABLE_BV_GEN [N] : Prf (∀ (λ f : El (fun (cart Real unit) (cart Real N)), (∀ (λ s : El (fun (cart Real unit) bool), (∀ (λ t : El (fun (cart Real unit) bool), (⇒ (∧ ((@is_interval unit) s) (∧ ((@bounded unit) s) (∧ ((@continuous_on unit N) f s) (∧ ((@has_bounded_variation_on N) f s) (∧ ((@COUNTABLE (cart Real unit)) t) (∀ (λ x : El (cart Real unit), (⇒ ((@IN (cart Real unit)) x ((@DIFF (cart Real unit)) s t)) ((@differentiable N unit) f ((@within (cart Real unit)) ((@at unit) x) s)))))))))) ((@absolutely_continuous_on N) f s))))))))
```

Translated to HOLLight.theorems.thm_ABSOLUTELY_CONTINUOUS_DIFFERENTIABLE_BV_GEN

```
Axiom thm_ABSOLUTELY_CONTINUOUS_DIFFERENTIABLE_BV_GEN : forall {N' : Type'}, forall f : (cart R unit) -> cart R N', forall s : (cart R unit) -> Prop, forall t : (cart R unit) -> Prop, ((@is_interval unit s) /\ ((@bounded unit s) /\ ((@continuous_on unit N' f s) /\ ((@has_bounded_variation_on N' f s) /\ ((@COUNTABLE (cart R unit) t) /\ (forall x : cart R unit, (@IN (cart R unit) x (@DIFF (cart R unit) s t)) -> @differentiable N' unit f (@within (cart R unit) (@_at unit x) s)))))))) -> @absolutely_continuous_on N' f s.
```

Existing searching tools

	Query	online availability	heterogeneous libraries
Integrated Rocq search commands	Scoped patterns and query language	no	no
Sherlorocq	regex	yes	no
Lemmasearch.com	regex	yes	no
LP websearch tool	**Scoped patterns and query language**	**yes**	**yes**

Future work

- Index more Hol-Light libraries
- Make the search tool available in VSCode, Emacs
- Extend to other provers (Lean)