Example

Kevin Buzzard Johan Commelin Patrick Massot

July 4, 2019

Contents

1 Huber_pair.lean

Huber_pair
A Huber pair consists of a Huber ring and a so-call ring of integral elements: an integrally closed, power bounded, open subring. (Huber called such objects “affinoid rings”.)

2 Huber_ring/basic.lean

3 Huber_ring/localization.lean

Huber_ring.away
The localization at `s`, endowed with a topology that depends on `T`

Huber_ring.away.D.aux
An auxiliary subring, used to define the topology on `away T s`

4 Spa.lean

spa.rational_open_data.s_inv_aux
This awful function produces r1.s as a unit in localization r2

spa.rational_open_data.localization_map
The map A(T1/s1) -> A(T2/s2) coming from the inequality r1 ≤ r2

spa.r_o_d_completion
A < T/s>, the functions on D(T,s). A topological ring

spa.presheaf_value
The underlying type of 𝒪_X(U), the structure presheaf on Spa(A)

5 Tate_ring.lean

6 adic_space.lean

PreValuedRingedSpace
A convenient auxiliary category whose objects are topological spaces equipped with a presheaf of topological rings and on each stalk (considered as abstract ring) an equivalence class of valuations. The point of this category is that the local isomorphism between a general adic space and an affinoid model Spa(A) can be checked in this category.

PreValuedRingedSpace.has_coe_to_sort
coercion from a PreValuedRingedSpace to the underlying topological space

PreValuedRingedSpace.topological_space
Adding the fact that the underlying space of a PreValuedRingedSpace is a topological space, to the type class inference system

stalk_map
The map on stalks induced from an f-map

CLVRS
Category of topological spaces endowed with a sheaf of complete topological rings and (an equivalence class of) valuations on the stalks (which are required to be local rings; moreover the support of the valuation must be the maximal ideal of the stalk). Wedhorn calls this category $\mathcal{V}$.

Spa
The adic spectrum of a Huber pair.

7 continuous_valuations.lean

valuation.is_continuous
Continuity of a valuation (Wedhorn 7.7).

8 for_mathlib/algebra.lean

9 for_mathlib/data/set/basic.lean

10 for_mathlib/data/set/finite.lean

11 for_mathlib/division_ring.lean

12 for_mathlib/equiv.lean

13 for_mathlib/filter.lean

14 for_mathlib/filtered.lean

15 for_mathlib/finset.lean

16 for_mathlib/finsupp_prod_inv.lean

17 for_mathlib/function.lean

18 for_mathlib/group.lean

units.unit_of_mul_left_eq_unit
produces a unit s from a proof that s divides a unit

19 for_mathlib/is_cover.lean

20 for_mathlib/lc_algebra.lean

21 for_mathlib/linear_ordered_comm_group.lean

linear_ordered_comm_group.eq_one_of_pow_eq_one
Wedhorn Remark 1.6 (3)

actual_ordered_comm_monoid
An ordered commutative monoid is a commutative monoid with a partial order such that mulition is an order embedding, i.e. `a * b ≤ a * c ↔ b ≤ c` .

22 for_mathlib/localization.lean

23 for_mathlib/logic.lean

24 for_mathlib/monotone.lean

strict_mono
A function between preorders is strictly monotone if `a < b` implies `f a < f b`.

25 for_mathlib/nonarchimedean/adic_topology.lean

26 for_mathlib/nonarchimedean/alt.lean

27 for_mathlib/nonarchimedean/basic.lean

topological_group.nonarchimedean
A topological group is non-archimedean if every neighborhood of 1 contains an open subgroup.

topological_add_group.nonarchimedean
A topological additive group is non-archimedean if every neighborhood of 0 contains an open subgroup.

28 for_mathlib/nonarchimedean/is_subgroups_basis.lean

29 for_mathlib/nonarchimedean/open_subgroup.lean

30 for_mathlib/open_embeddings.lean

31 for_mathlib/open_nhds.lean

32 for_mathlib/opens.lean

33 for_mathlib/option_inj.lean

34 for_mathlib/order.lean

preorder.lift'
version of preorder.lift which doesn't use type class inference

35 for_mathlib/pointwise.lean

36 for_mathlib/prime.lean

37 for_mathlib/quotient.lean

38 for_mathlib/quotient_group.lean

39 for_mathlib/rings.lean

40 for_mathlib/sheaves/covering.lean

41 for_mathlib/sheaves/opens.lean

42 for_mathlib/sheaves/presheaf.lean

43 for_mathlib/sheaves/presheaf_maps.lean

44 for_mathlib/sheaves/presheaf_of_rings.lean

45 for_mathlib/sheaves/presheaf_of_rings_maps.lean

46 for_mathlib/sheaves/presheaf_of_topological_rings.lean

47 for_mathlib/sheaves/sheaf.lean

48 for_mathlib/sheaves/sheaf_of_rings.lean

49 for_mathlib/sheaves/sheaf_of_topological_rings.lean

50 for_mathlib/sheaves/stalk.lean

51 for_mathlib/sheaves/stalk_of_rings.lean

52 for_mathlib/sheaves_of_types.lean

53 for_mathlib/subgroup.lean

54 for_mathlib/submodule.lean

55 for_mathlib/submonoid.lean

56 for_mathlib/subrel.lean

57 for_mathlib/subtype.lean

58 for_mathlib/topological_field.lean

topological_division_ring
A topological division ring is a division ring with a topology where all operations are continuous, including inversion.

59 for_mathlib/topological_groups.lean

exists_limit_of_ultimately_const
If a function is constant on some set of a proper filter then it converges along this filter

60 for_mathlib/topological_rings.lean

61 for_mathlib/topology.lean

62 for_mathlib/uniform_space/basic.lean

63 for_mathlib/uniform_space/group.lean

64 for_mathlib/uniform_space/pi.lean

65 for_mathlib/uniform_space/ring.lean

66 for_mathlib/uniform_space/separation.lean

sep_quot
separation space

67 for_mathlib/uniform_space/uniform_field.lean

zero_not_adh
Zero is not adherent to F

hat_star_is_units
homeomorphim between non-zero elements of hat K and units of hat K

68 for_mathlib/with_zero.lean

tactic.interactive.with_zero_cases
Case bashing for with_zero. If `x₁, ... x_n` have type `with_zero α` then `with_zero cases x₁ ... x_n` will split according to whether each `x_i` is zero or coerced from `α` then run

norm_cast at * , try to simplify using the simp rules with_zero_simp , and try to get a contradiction.

69 perfectoid_space.lean

is_perfectoid
Condition for an object of CLVRS to be perfectoid: every point should have an open neighbourhood isomorphic to Spa(A) for some perfectoid ring A.

PerfectoidSpace
The category of perfectoid spaces.

70 power_bounded.lean

is_topologically_nilpotent
Wedhorn Definition 5.25 page 36

is_bounded
Wedhorn Definition 5.27 page 36

71 prime.lean

72 r_o_d_completion.lean

spa.r_o_d_completion.to_valuation_field_commutes
the maps from rationals opens to completions commute with allowable restriction maps

spa.presheaf.to_valuation_field_completion
The map from F(U) to K_v for v ∈ U

spa.presheaf.to_valuation_field_completion_commutes
If v ∈ U then the map from 𝒪_X(U) to completion (valuation_field v) commutes with restriction (so we can get a map from the stalk at v)

73 relation_example.lean

74 relation_on_colimit.lean

75 relation_on_stalk.lean

76 scratch.lean

77 stalk_valuation.lean

78 the_category_C.lean

79 valuation/basic.lean

valuation
Γ-valued valuations on R

valuation.to_preorder
a valuation gives a preorder on the underlying ring

valuation.comap
f : S → R induces map valuation R Γ → valuation S Γ

valuation.on_quot
Extension of valuation v on R to valuation on R/J if J ⊆ supp v

valuation.supp_quot_supp
quotient valuation on R/J has support supp(v)/J if J ⊆ supp v

valuation.on_frac
Extension of valuation on R with supp 0 to valuation on field of fractions

80 valuation/canonical.lean

valuation.value_group
The value group of the canonical valuation.

valuation.valuation_field.canonical_valuation
The canonical valuation on Frac(R/supp(v))

valuation.quotient.canonical_valuation
The canonical valuation on R/supp(v)

valuation.canonical_valuation
The canonical valuation on R

valuation.canonical_valuation_is_equiv
A valuation is equivalent to its canonical valuation

valuation.is_equiv.supp_eq
Wedhorm 1.27 iii -> ii (part a)

81 valuation/field.lean

valuation.topological_division_ring
The topology coming from a valuation on a division rings make it a topological division ring [BouAC, VI.5.1 middle of Proposition 1]

continuous_unit_map
The valuation map restricted to units of a field endowed with the valuation toplogy is continuous if we endow the target with discrete topology. [BouAC, VI.5.1 end of Proposition 1]

82 valuation/localization.lean

valuation.localization
Extension of a valuation to a localization

valuation.localization_comap
the extension of a valuation pulls back to the valuation

valuation.eq_localization_of_comap
if a valuation on a localisation pulls back to v then it's the localization of v

83 valuation/localization_Huber.lean

Huber_pair.rational_open_data.to_valuation_field_cts
If v : spa A is in D(T,s) then the map A(T/s) -> K_v is continuous

84 valuation/topology.lean

85 valuation_spectrum.lean

Spv
Valuation spectrum of a ring.

Spv.mk
The constructor for a term of type Spv R given an arbitrary valuation

Spv.out_Γ
The value group attached to a term of type Spv R

Spv.out
An explicit valuation attached to a term of type Spv R

Spv.lift_eq
The computation principle for Spv

Spv.lift_eq'
Prop-valued version of computation principle for Spv

Spv.basic_open
The open sets generating the topology of Spv R, see Wedhorn 4.1.