Library mk_structure

mk_structure

```
Notation "∀ x .. y , P" := (forall x, .. (forall y, P) ..)
    (at level 200, x binder, y binder, right associativity,
    format "'[ ' ∀ x .. y ']' , P") : type_scope.

Notation "∃ x .. y , P" := (exists x, .. (exists y, P) ..)
    (at level 200, x binder, y binder, right associativity,
    format "'[ ' ∃ x .. y ']' , P") : type_scope.

Notation "' λ' x .. y , t" := (fun x => .. (fun y => t) ..)
    (at level 200, x binder, y binder, right associativity,
    format "'[ ' ' λ' x .. y ']' , t").

Axiom classic : ∀ P, P \/ ~P.

Proposition peirce : ∀ P, (~P -> P) -> P.

Proposition NNPP : ∀ P, ~~P <-> P.

Proposition notandor : ∀ P Q,
    (~(P /\ Q) <-> (~P) \/ (~Q)) /\ (~(P \/ Q) <-> (~P) /\ (~Q)).

Proposition inp : ∀ {P Q: Prop}, (P -> Q) -> (~Q) -> (~P).


Parameter Class : Type.
Parameter In : Class -> Class -> Prop.
Parameter Classifier : (Class -> Prop) -> Class.

Notation "x ∈ y" := (In x y) (at level 70).
Notation "\{ P \}" := (Classifier P) (at level 0).


Definition Ensemble x := ∃ y, x ∈ y.

Global Hint Unfold Ensemble : core.

Definition Union x y := \{ λ z, z ∈ x \/ z ∈ y \}.

Notation "x ∪ y" := (Union x y) (at level 65, right associativity).

Definition Intersection x y := \{ λ z, z ∈ x /\ z ∈ y \}.

Notation "x ∩ y" := (Intersection x y) (at level 60, right associativity).

Definition NotIn x y := ~ (x ∈ y).
Notation "x ∉ y" := (NotIn x y) (at level 70).

Definition Complement x := \{λ y, y ∉ x \}.
Notation "¬ x" := (Complement x) (at level 5, right associativity).

Definition Setminus x y := x ∩ (¬ y).
Notation "x ~ y" := (Setminus x y) (at level 50, left associativity).

Notation "x ≠ y" := (~ (x = y)) (at level 70).

Definition Φ := \{λ x, x ≠ x \}.

Definition μ := \{ λ x, x = x \}.

Definition Element_I x := \{ λ z, ∀ y, y ∈ x -> z ∈ y \}.
Notation "∩ x" := (Element_I x) (at level 66).
```

```
Definition Element_U x := \{ λ z, ∃ y, z ∈ y /\ y ∈ x \}.
Notation "∪ x" := (Element_U x) (at level 66).

Definition Included x y := ∀ z, z ∈ x -> z ∈ y.
Notation "x ⊂ y" := (Included x y) (at level 70).

Definition PowerClass x := \{ λ y, y ⊂ x \}.
Notation "pow( x )" := (PowerClass x)(at level 0, right associativity).

Definition Singleton x := \{ λ z, x ∈ μ -> z = x \}.

Notation "[ x ]" := (Singleton x) (at level 0, right associativity).

Definition Unordered x y := [x] ∪ [y].
Notation "[ x | y ]" := (Unordered x y) (at level 0).

Definition Ordered x y := [ [x] | [x|y]].
Notation "[ x , y ]" := (Ordered x y) (at level 0).

Definition First z := ∩∩z.

Definition Second z := (∩∪z)∪(∪∪z) ~ (∪∩z).

Definition Relation r := ∀ z, z ∈ r -> ∃ x y, z = [x,y].

Notation "\{\ P \}\" :=
    (\{ λ z, ∃ x y, z = [x,y] /\ P x y \}) (at level 0).

Definition Composition r s :=
    \{\ λ x z, ∃ y, [x,y] ∈ s /\ [y,z] ∈ r \}\.
Notation "r ∘ s" := (Composition r s) (at level 50).

Definition Inverse r := \{\ λ x y, [y,x] ∈ r \}\.
Notation "r ⁻¹" := (Inverse r) (at level 5).

Definition Function f :=
    Relation f /\ (∀ x y z, [x,y] ∈ f -> [x,z] ∈ f -> y = z).

Definition Domain f := \{ λ x, ∃ y, [x,y] ∈ f \}.
Notation "dom( f )" := (Domain f)(at level 5).

Definition Range f := \{ λ y, ∃ x, [x,y] ∈ f \}.
Notation "ran( f )" := (Range f)(at level 5).

Definition Value f x := ∩(\{ λ y, [x,y] ∈ f \}).

Notation "f [ x ]" := (Value f x)(at level 5).

Definition Cartesian x y := \{\ λ u v, u ∈ x /\ v ∈ y \}\.
Notation "x × y" := (Cartesian x y) (at level 2, right associativity).

Definition Exponent y x :=
    \{ λ f, Function f /\ dom( f ) = x /\ ran( f ) ⊂ y \}.

Definition On f x := Function f /\ dom(f) = x.

Definition To f y := Function f /\ ran(f) ⊂ y.

Definition Onto f y := Function f /\ ran(f) = y.

Definition Rrelation x r y := [x,y] ∈ r.

Definition Connect r x := ∀ u v, u ∈ x -> v ∈ x
    -> (Rrelation u r v) \/ (Rrelation v r u) \/ (u = v).

Definition Transitive r x := ∀ u v w, u ∈ x -> v ∈ x -> w ∈ x
    -> Rrelation u r v -> Rrelation v r w -> Rrelation u r w.

Definition Asymmetric r x := ∀ u v, u ∈ x -> v ∈ x
    -> Rrelation u r v -> ~ Rrelation v r u.
```

```
Definition FirstMember z r x :=
    z ∈ x /\ (∀ y, y ∈ x -> ~ Rrelation y r z).

Definition WellOrdered r x :=
    Connect r x /\ (∀ y, y ⊂ x -> y ≠ Φ -> ∃ z, FirstMember z r y).

Definition rSection y r x := y ⊂ x /\ WellOrdered r x
    /\ (∀ u v, u ∈ x -> v ∈ y -> Rrelation u r v -> u ∈ y).

Definition Order_Pr f r s := Function f
    /\ WellOrdered r dom(f) /\ WellOrdered s ran(f)
    /\ (∀ u v, u ∈ dom(f) -> v ∈ dom(f) -> Rrelation u r v
       -> Rrelation f[u] s f[v]).

Definition Function1_1 f := Function f /\ Function (f⁻¹).

Definition Order_PXY f x y r s := WellOrdered r x /\ WellOrdered s y
    /\ Order_Pr f r s /\ rSection dom(f) r x /\ rSection ran(f) s y.

Definition E := \{\ λ x y, x ∈ y \}\.

Definition Full x := ∀ m, m ∈ x -> m ⊂ x.

Definition Ordinal x := Connect E x /\ Full x.

Definition R := \{ λ x, Ordinal x \}.

Definition Ordinal_Number x := x ∈ R.

Definition Less x y := x ∈ y.
Notation "x ≺ y" := (Less x y) (at level 67, left associativity).

Definition LessEqual (x y: Class) := x ∈ y \/ x = y.
Notation "x ≼ y" := (LessEqual x y) (at level 67, left associativity).

Definition PlusOne x := x ∪ [x].

Definition Restriction f x := f ∩ (x × μ).
Notation "f | ( x )" := (Restriction f x) (at level 30).

Definition Integer x := Ordinal x /\ WellOrdered (E⁻¹) x.

Definition LastMember x E y := FirstMember x (E⁻¹) y.

Definition ω := \{ λ x, Integer x \}.

Definition ChoiceFunction c :=
    Function c /\ (∀ x, x ∈ dom(c) -> c[x] ∈ x).

Definition Nest n := ∀ x y, x ∈ n -> y ∈ n -> x ⊂ y \/ y ⊂ x.

Definition Equivalent x y :=
    ∃ f, Function1_1 f /\ dom(f) = x /\ ran(f) = y.
Notation "x ≈ y" := (Equivalent x y) (at level 70).

Definition Cardinal_Number x :=
    Ordinal_Number x /\ (∀ y, y ∈ R -> y ≺ x -> ~ (x ≈ y)).

Definition C := \{ λ x, Cardinal_Number x \}.

Definition P := \{\ λ x y, x ≈ y /\ y ∈ C \}\.

Definition Finite x := P[x] ∈ ω.

Definition Max x y := x ∪ y.

Definition LessLess := \{\ λ a b, ∃ u v x y, a = [u,v]
    /\ b = [x,y] /\ [u,v] ∈ (R × R) /\ [x,y] ∈ (R × R)
    /\ ((Max u v ≺ Max x y) \/ (Max u v = Max x y /\ u ≺ x)
       \/ (Max u v = Max x y /\ u = x /\ v ≺ y)) \}\.

Notation "≪" := (LessLess)(at level 0, no associativity).
```

```
Class MK_Axioms := {
    A_I : ∀ x y, x = y <-> (∀ z, z ∈ x <-> z ∈ y);
    A_II : ∀ b P, b ∈ \{ P \} <-> Ensemble b /\ (P b);
    A_III : ∀ {x}, Ensemble x
        -> ∃ y, Ensemble y /\ (∀ z, z ⊂ x -> z ∈ y);
    A_IV : ∀ {x y}, Ensemble x -> Ensemble y -> Ensemble (x ∪ y);
    A_V : ∀ {f}, Function f -> Ensemble dom(f) -> Ensemble ran(f);
    A_VI : ∀ x, Ensemble x -> Ensemble (∪x);
    A_VII : ∀ x, x ≠ Φ -> ∃ y, y ∈ x /\ x ∩ y = Φ;
    A_VIII : ∃ y, Ensemble y /\ Φ ∈ y
        /\ (∀ x, x ∈ y -> (x ∪ [x]) ∈ y);
    }.

Parameter MK_Axiom : MK_Axioms.

Notation AxiomI := (@ A_I MK_Axiom).
Notation AxiomII := (@ A_II MK_Axiom).
Notation AxiomIII := (@ A_III MK_Axiom).
Notation AxiomIV := (@ A_IV MK_Axiom).
Notation AxiomV := (@ A_V MK_Axiom).
Notation AxiomVI := (@ A_VI MK_Axiom).
Notation AxiomVII := (@ A_VII MK_Axiom).
Notation AxiomVIII := (@ A_VIII MK_Axiom).

Axiom AxiomIX : ∃ c, ChoiceFunction c /\ dom(c) = μ ~ [Φ].

Ltac New H := pose proof H.

Ltac TF P := destruct (classic P).

Ltac Absurd := apply peirce; intros.


Ltac deand :=
    match goal with
      | H: ?a /\ ?b |- _ => destruct H; deand
      | _ => idtac
    end.

Ltac deor :=
    match goal with
      | H: ?a \/ ?b |- _ => destruct H; deor
      | _ => idtac
    end.

Ltac deandG :=
    match goal with
        |- ?a /\ ?b => split; deandG
        | _ => idtac
    end.

Ltac eqext := apply AxiomI; split; intros.

Ltac appA2G := apply AxiomII; split; eauto.

Ltac appA2H H := apply AxiomII in H as [].
```


Global Index A B C D E F G H I J K L M N O P Q R S T U V W X Y Z _ other (1 entry)
Library Index A B C D E F G H I J K L M N O P Q R S T U V W X Y Z _ other (1 entry)

Global Index

M

mk_structure [library]

Library Index

M

mk_structure
