A TPTP Formalization of the Unified Foundational Ontology

Daniele Porelo, João Paulo A. Almeida, Giancarlo Guizzardi,
Claudenir M. Fonseca, Tiago Prince Sales

October 7, 2021

Abstract

This document presents a formalization of the Unified Foundation Ontology (UFO) expressed in first-order logics through the TPTP syntax. This formalization is intended to support verification of UFO's theory through automated provers and consistency checkers.

1 Introduction

This document presents a formalization of the Unified Foundation Ontology (UFO) expressed in first-order logics through the TPTP syntax. This formalization is intended to support verification of UFO's theory through automated provers and consistency checkers.

2 UFO's TPTP Specification

2.1 UFO Taxonomy

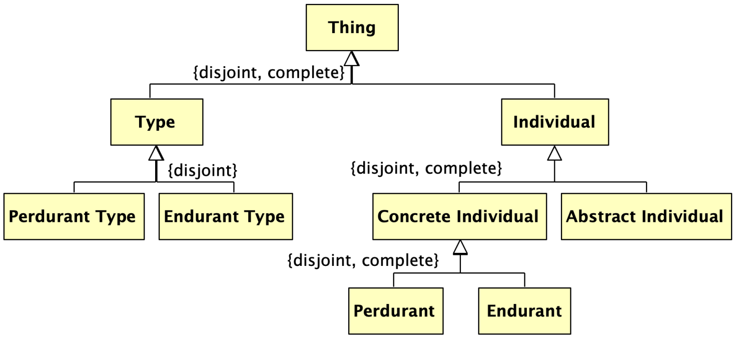

Figure 1: Partial Taxonomy of UFO – Thing.

```
% Thing

fof(ax_thing_taxonomy, axiom, (
  ![X]: ((type(X) | individual(X)) <=> (thing(X)))
)).

fof(ax_thing_partition, axiom, (
  ~?[X]: (type(X) & individual(X))
)).

% Individual

fof(ax_individual_taxonomy, axiom, (
  ![X]: ((concreteIndividual(X) | abstractIndividual(X)) <=> (
    individual(X)))
)).

fof(ax_individual_partition, axiom, (
  ~?[X]: (concreteIndividual(X) & abstractIndividual(X))
)).

% Concrete Individual

fof(ax_concreteIndividual_taxonomy, axiom, (
  ![X]: ((endurant(X) | perdurant(X)) <=> (concreteIndividual(X)))
)).

fof(ax_concreteIndividual_partition, axiom, (
  ~?[X]: (endurant(X) & perdurant(X))
)).

% Type

fof(ax_type_taxonomy, axiom, (
  ![X]: ((endurantType(X) | perdurantType(X)) <=> (type(X)))
)).

fof(ax_type_partition, axiom, (
  ~?[X]: (endurantType(X) & perdurantType(X))
)).

% Thing partial taxonomy instances

fof(ax_thing_instances, axiom, (
  type(type1) & individual(individual1) & concreteIndividual(
    concreteIndividual1) & abstractIndividual(abstractIndividual1)
    & endurant(endurant1) & perdurant(perdurant1) & endurantType(
    endurantType1) & perdurantType(perdurantType1)
)).
```

```
% Abstract Individual

fof(ax_abstractIndividual_taxonomy_quale, axiom, (
  ![X]: (quale(X) => (abstractIndividual(X)))
)).

fof(ax_abstractIndividual_taxonomy_set, axiom, (
  ![X]: (set(X) => (abstractIndividual(X)))
```

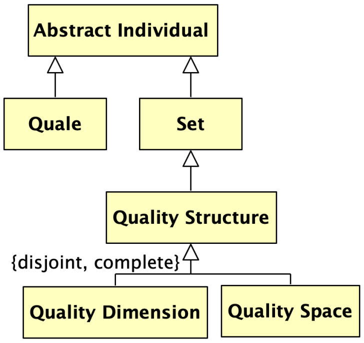

Figure 2: Partial Taxonomy of UFO – Abstract Individual.

```
)).

% Set

fof(ax_set_taxonomy_qualityStructure, axiom, (
  ![X]: (qualityStructure(X) => (set(X)))
)).

% Quality Structure

fof(ax_qualityStructure_taxonomy, axiom, (
  ![X]: ((qualityDimension(X) | qualitySpace(X)) <=> (
    qualityStructure(X)))
)).

fof(ax_qualityStructure_partition, axiom, (
  ~?[X]: (qualityDimension(X) & qualitySpace(X))
)).

% TODO: review the definition of "world" as a subtype of "
    qualityStructure"

fof(ax_qualityStructure_taxonomy_world, axiom, (
  ![X]: (world(X) => (qualityStructure(X)))
)).

% Abstract Individual partial taxonomy instances

fof(ax_abstractIndividual_instances, axiom, (
  set(set1) & quale(quale1) & qualityStructure(qualityStructure1) &
     qualityDimension(qualityDimension1) & qualitySpace(
    qualitySpace1) & world(world1)
)).
```

```
% Endurant

fof(ax_endurant_taxonomy, axiom, (
  ![X]: ((substantial(X) | moment(X)) <=> (endurant(X)))
)).
```

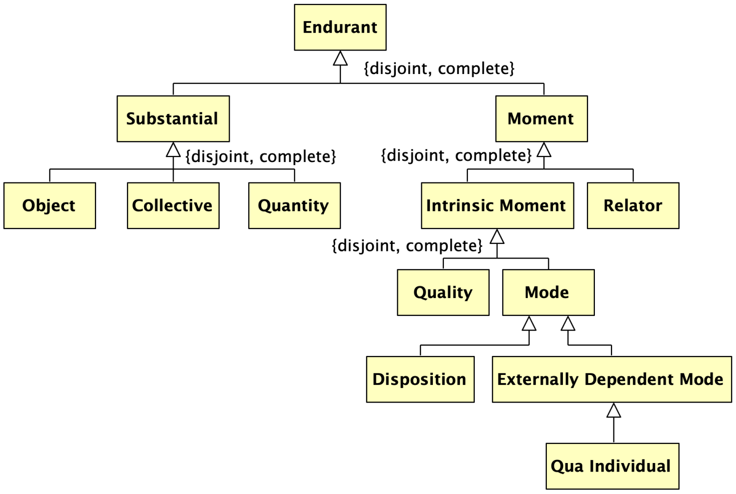

Figure 3: Partial Taxonomy of UFO – Endurant.

```

fof(ax_endurant_partition, axiom, (
  ~?[X]: (substantial(X) & moment(X))
)).

% Substantial

fof(ax_substantial_taxonomy, axiom, (
  ![X]: ((object(X) | collective(X) | quantity(X)) <=> (substantial(X)))
)).

fof(ax_substantial_partition, axiom, (
  ~?[X]: ((object(X) & collective(X)) | (object(X) & quantity(X)) | (collective(X) & quantity(X)))
)).

% Moment

fof(ax_moment_taxonomy, axiom, (
  ![X]: ((intrinsicMoment(X) | relator(X)) <=> (moment(X)))
)).

fof(ax_moment_partition, axiom, (
  ~?[X]: (intrinsicMoment(X) & relator(X))
)).

% Intrinsic Moment

fof(ax_intrinsicMoment_taxonomy, axiom, (
  ![X]: ((quality(X) | mode(X)) <=> (intrinsicMoment(X)))
)).

```

```
fof(ax_intrinsicMoment_partition, axiom, (
  ~?[X]: (quality(X) & mode(X))
)).

% Mode

fof(ax_mode_taxonomy_externallyDependentMode, axiom, (
  ![X]: (externallyDependentMode(X) => (mode(X)))
)).

% Externally Dependent Mode

fof(ax_externallyDependentMode_taxonomy_quaIndividual, axiom, (
  ![X]: (quaIndividual(X) => (externallyDependentMode(X)))
)).

% Endurant partial taxonomy instances

fof(ax_endurant_instances, axiom, (
  substantial(substantial1) & moment(moment1) & object(object1) &
    collective(collective1) & quantity(quantity1) & intrinsicMoment
    (intrinsicMoment1) & relator(relator1) & quality(quality1) &
    mode(mode1) & disposition(disposition1) &
    externallyDependentMode(externallyDependentMode1) &
    quaIndividual(quaIndividual1)
)).
```

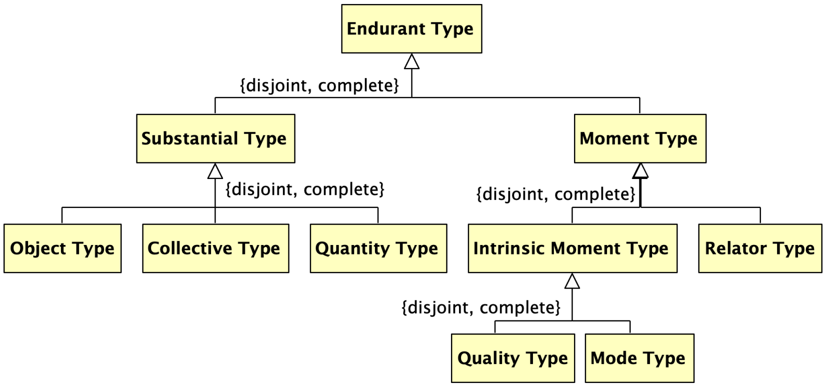

Figure 4: Partial Taxonomy of UFO – Endurant Types (by ontological nature).

```
% Endurant Type (by ontological nature)

fof(ax_endurantType_taxonomy, axiom, (
  ![X]: ((substantialType(X) | momentType(X)) <=> (endurantType(X))
    )
)).

fof(ax_endurantType_partition, axiom, (
  ~?[X]: (substantialType(X) & momentType(X))
)).
```

```

% Substantial Type

fof(ax_substantialType_taxonomy, axiom, (
  ![X]: ((objectType(X) | collectiveType(X) | quantityType(X)) <=>
    (substantialType(X)))
)).

fof(ax_substantialType_partition, axiom, (
  ~?[X]: ((objectType(X) & collectiveType(X)) | (objectType(X) &
    quantityType(X)) | (collectiveType(X) & quantityType(X)))
)).

% Moment Type

fof(ax_momentType_taxonomy, axiom, (
  ![X]: ((intrinsicMomentType(X) | relatorType(X)) <=> (momentType(
    X)))
)).

fof(ax_momentType_partition, axiom, (
  ~?[X]: (intrinsicMomentType(X) & relatorType(X))
)).

% Intrinsic Moment Type

fof(ax_intrinsicMomentType_taxonomy, axiom, (
  ![X]: ((qualityType(X) | modeType(X)) <=> (intrinsicMomentType(X)
    ))
)).

fof(ax_intrinsicMomentType_partition, axiom, (
  ~?[X]: (qualityType(X) & modeType(X))
)).

% Endurant Type (by ontological nature) partial taxonomy instances

fof(ax_endurantType_instances, axiom, (
  substantialType(substantialType1) & momentType(momentType1) &
    objectType(objectType1) & collectiveType(collectiveType1) &
    quantityType(quantityType1) & intrinsicMomentType(
    intrinsicMomentType1) & relatorType(relatorType1) & qualityType
    (qualityType1) & modeType(modeType1) &
    externallyDependentModeType(externallyDependentModeType1) &
    quaIndividualType(quaIndividualType1)
)).
```

```
% Endurant Type (by modal properties of types)

fof(ax_endurantType_taxonomy, axiom, (
  ![X]: ((sortal(X) | nonSortal(X)) <=> (endurantType(X)))
)).

fof(ax_endurantType_partition, axiom, (
  ~?[X]: (sortal(X) & nonSortal(X))
)).

% Sortal
```

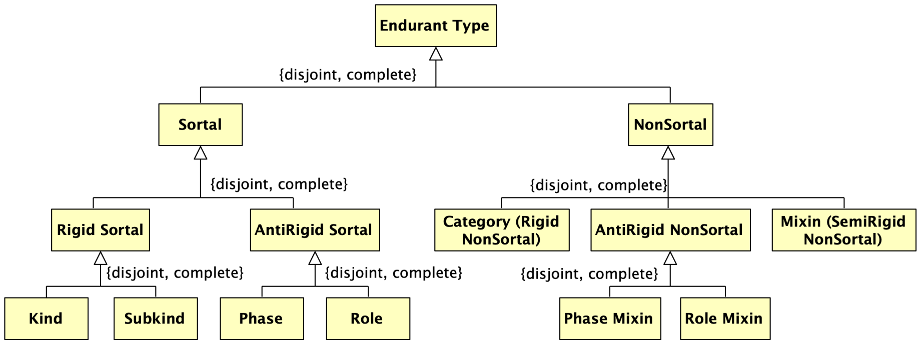

Figure 5: Partial Taxonomy of UFO – Endurant Types (by modal properties of types).

```

fof(ax_sortal_taxonomy, axiom, (
  ![X]: ((rigidSortal(X) | antiRigidSortal(X)) <=> (sortal(X)))
)).

fof(ax_sortal_partition, axiom, (
  ~?[X]: (rigidSortal(X) & antiRigidSortal(X))
)).

% Rigid Sortal

fof(ax_rigidSortal_taxonomy, axiom, (
  ![X]: ((kind(X) | subkind(X)) <=> (rigidSortal(X)))
)).

fof(ax_rigidSortal_partition, axiom, (
  ~?[X]: (kind(X) & subkind(X))
)).

% Anti-Rigid Sortal

fof(ax_antiRigidSortal_taxonomy, axiom, (
  ![X]: ((phase(X) | role(X)) <=> (antiRigidSortal(X)))
)).

fof(ax_antiRigidSortal_partition, axiom, (
  ~?[X]: (phase(X) & role(X))
)).

% Non-Sortal

fof(ax_nonSortal_taxonomy, axiom, (
  ![X]: ((rigidNonSortal(X) | semiRigidNonSortal(X) |
    antiRigidNonSortal(X)) <=> (nonSortal(X)))
)).

fof(ax_nonSortal_partition, axiom, (
  ~?[X]: ((rigidNonSortal(X) & semiRigidNonSortal(X)) | (
    rigidNonSortal(X) & antiRigidNonSortal(X)) | (
    semiRigidNonSortal(X) & antiRigidNonSortal(X)))
)).

% Category

fof(ax_rigidNonSortal_taxonomy, axiom, (
  ![X]: (rigidNonSortal(X) <=> (category(X)))
)).

% Mixin

fof(ax_semiRigidNonSortal_taxonomy, axiom, (
  ![X]: (semiRigidNonSortal(X) <=> (mixin(X)))
)).

% Anti-Rigid Non-Sortal

fof(ax_antiRigidNonSortal_taxonomy, axiom, (
  ![X]: ((phaseMixin(X) | roleMixin(X)) <=> (antiRigidNonSortal(X))
    )
)).

fof(ax_antiRigidNonSortal_partition, axiom, (
  ~?[X]: (phaseMixin(X) & roleMixin(X))
)).

% Endurant Type (by modal properties of types) partial taxonomy
    instances

fof(ax_endurantType_instances, axiom, (
  sortal(sortal1) & nonSortal(nonSortal1) & rigidSortal(
    rigidSortal1) & antiRigidSortal(antiRigidSortal1) & kind(kind1)
     & subkind(subkind1) & phase(phase1) & role(role1) & category(
    category1) & mixin(mixin1) & antiRigidNonSortal(
    antiRigidNonSortal1) & phaseMixin(phaseMixin1) & roleMixin(
    roleMixin1)
)).
```

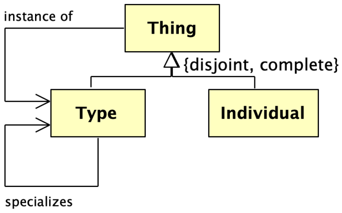

Figure 6: Types, individuals, instantiation, and specialization.

```
%%%%%%%%%%% Instance of, Types, and Individuals %%%%%%%%%%

fof(ax_dIof, axiom, (
  ![X,Y,W]: (iof(X,Y,W) => (individual(X) & type(Y) & world(W)))
)).
```

```

fof(ax_dType_a1 , axiom , (
  ![X]: (type(X) <=> (?[Y,W]: iof(Y,X,W)))
)).

fof(ax_dIndividual_a2 , axiom , (
  ![X]: (individual(X) <=> (~?[Y,W]: iof(Y,X,W)))
)).

% TODO: confirm whether we are including second-order types in this formalization

fof(ax_multiLevel_a3 , axiom , (
  ![X,Y,W]: (iof(X,Y,W) => (type(X) | individual(X)))
)).

fof(ax_twoLevelConstrained_a4 , axiom , (
  ~?[X,Y,Z,W]: (type(X) & iof(X,Y,W) & iof(Y,Z,W))
)).

fof(ax_iofInUse , axiom , (
  type(t2) & individual(i2) & wolrd(w2) & iof(i2,t2,w2)
)).

% TODO: (t1) and (t2) do not follow from the previous axioms; we might need an axiom constraining our quantification domain to be limited to things (i.e., "![X]: (thing(X))").

% fof(th_everythingIsAThing_t1 , conjecture , (
%   ![X]: (type(x) | individual(x))
% )).

% fof(th_thingPartition_t2 , conjecture , (
%   ~?[X]: (type(x) & individual(x))
% )).

%%%%%%%% Specialization and Proper Specialization %%%%%%%%



% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
% %Specialization
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5


% fof(ax_dspecialization_a5 , axiom , (
%   ![T1,T2] :  (specializes(T1,T2) <=> (type(T1) & type(T2) &
%                        ![W]: (world(W) => ![E]:(iof(E,T1,W) => iof(E,T2,W))))))).
```