

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

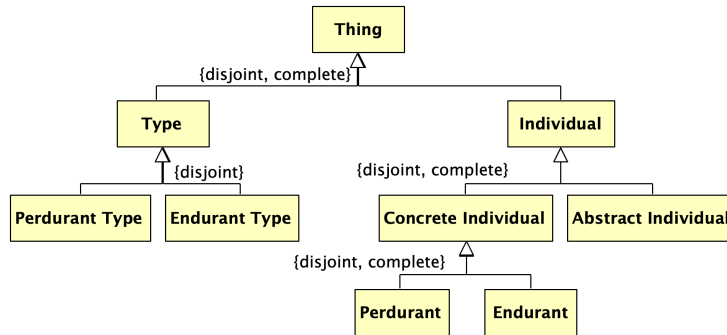


Figure 1: Partial Taxonomy of UFO – Thing.

```

4 % Thing
5
6 fof(ax_thing_taxonomy, axiom, (
7   ![X]: ((type(X) | individual(X)) <=> (thing(X)))
8 )).
9
10 fof(ax_thing_partition, axiom, (
11   ~?[X]: (type(X) & individual(X))
12 )).
13
14 % Individual
15
16 fof(ax_individual_taxonomy, axiom, (
17   ![X]: ((concreteIndividual(X) | abstractIndividual(X)) <=> (
18     individual(X)))
19 )).
20
21 fof(ax_individual_partition, axiom, (
22   ~?[X]: (concreteIndividual(X) & abstractIndividual(X))
23 )).
24
25 % Concrete Individual
26
27 fof(ax_concreteIndividual_taxonomy, axiom, (
28   ![X]: ((endurant(X) | perdurant(X)) <=> (concreteIndividual(X)))
29 )).
30
31 fof(ax_concreteIndividual_partition, axiom, (
32   ~?[X]: (endurant(X) & perdurant(X))
33 )).
34
35 % Type
36
37 fof(ax_type_taxonomy, axiom, (
38   ![X]: ((endurantType(X) | perdurantType(X)) <=> (type(X)))
39 )).
40
41 fof(ax_type_partition, axiom, (
42   ~?[X]: (endurantType(X) & perdurantType(X))
43 )).
44
45 % Thing partial taxonomy instances
46
47 fof(ax_thing_instances, axiom, (
48   type(type1) & individual(individual1) & concreteIndividual(
49     concreteIndividual1) & abstractIndividual(abstractIndividual1)
50     & endurant(endurant1) & perdurant(perdurant1) & endurantType(
51     endurantType1) & perdurantType(perdurantType1)
52 )).
53
54 % Abstract Individual
55
56 fof(ax_abstractIndividual_taxonomy_quale, axiom, (
57   ![X]: (quale(X) => (abstractIndividual(X)))
58 )).
59
60 fof(ax_abstractIndividual_taxonomy_set, axiom, (
61   ![X]: (set(X) => (abstractIndividual(X)))
62 )).

```

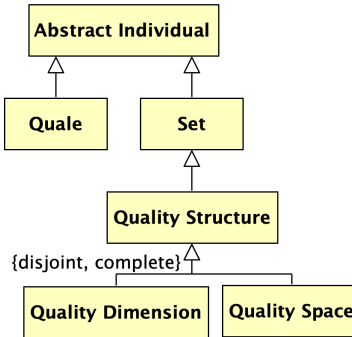


Figure 2: Partial Taxonomy of UFO – Abstract Individual.

```

58 ))).
59
60 % Set
61
62 fof(ax_set_taxonomy_qualityStructure, axiom, (
63   ![X]: (qualityStructure(X) => (set(X)))
64 ))).
65
66 % Quality Structure
67
68 fof(ax_qualityStructure_taxonomy, axiom, (
69   ![X]: ((qualityDimension(X) | qualitySpace(X)) <=> (
70     qualityStructure(X)))
71 ))).
72
73 fof(ax_qualityStructure_partition, axiom, (
74   ~?[X]: (qualityDimension(X) & qualitySpace(X))
75 ))).
76
77 % TODO: review the definition of "world" as a subtype of "
78   qualityStructure"
79
80 fof(ax_qualityStructure_taxonomy_world, axiom, (
81   ![X]: (world(X) => (qualityStructure(X)))
82 ))).
83
84 % Abstract Individual partial taxonomy instances
85
86 fof(ax_abstractIndividual_instances, axiom, (
87   set(set1) & quale(quale1) & qualityStructure(qualityStructure1) &
88   qualityDimension(qualityDimension1) & qualitySpace(
89     qualitySpace1) & world(world1)
90 ))).
91
92 % Endurant
93
94 fof(ax_endurant_taxonomy, axiom, (
95   ![X]: ((substantial(X) | moment(X)) <=> (endurant(X)))
96 ))).

```

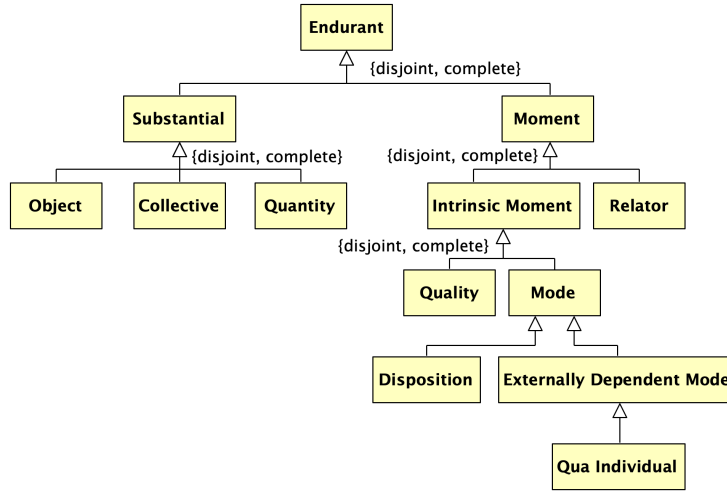


Figure 3: Partial Taxonomy of UFO – Endurant.

```

93
94 fof(ax_endurant_partition, axiom, (
95   ~?[X]: (substantial(X) & moment(X))
96 ))).
97
98 % Substantial
99
100 fof(ax_substantial_taxonomy, axiom, (
101   ![X]: ((object(X) | collective(X) | quantity(X)) <=> (substantial
102     (X)))
103 ))).
104
105 fof(ax_substantial_partition, axiom, (
106   ~?[X]: ((object(X) & collective(X)) | (object(X) & quantity(X)) |
107     (collective(X) & quantity(X)))
108 ))).
109
110 % Moment
111
112 fof(ax_moment_taxonomy, axiom, (
113   ![X]: ((intrinsicMoment(X) | relator(X)) <=> (moment(X)))
114 ))).
115
116 fof(ax_moment_partition, axiom, (
117   ~?[X]: (intrinsicMoment(X) & relator(X))
118 ))).
119
120 % Intrinsic Moment
121
122 fof(ax_intrinsicMoment_taxonomy, axiom, (
123   ![X]: ((quality(X) | mode(X)) <=> (intrinsicMoment(X)))
124 ))).

```

```

124 fof(ax_intrinsicMoment_partition, axiom, (
125   ~?[X]: (quality(X) & mode(X))
126 )).
127
128 % Mode
129
130 fof(ax_mode_taxonomy_externallyDependentMode, axiom, (
131   ![X]: (externallyDependentMode(X) => (mode(X)))
132 )).
133
134 % Externally Dependent Mode
135
136 fof(ax_externallyDependentMode_taxonomy_quaIndividual, axiom, (
137   ![X]: (quaIndividual(X) => (externallyDependentMode(X)))
138 )).
139
140 % Endurant partial taxonomy instances
141
142 fof(ax_endurant_instances, axiom, (
143   substantial(substantial1) & moment(moment1) & object(object1) &
144   collective(collective1) & quantity(quantity1) & intrinsicMoment
145   (intrinsicMoment1) & relator(relator1) & quality(quality1) &
146   mode(mode1) & disposition(disposition1) &
147   externallyDependentMode(externallyDependentMode1) &
148   quaIndividual(quaIndividual1)
149 )).

```

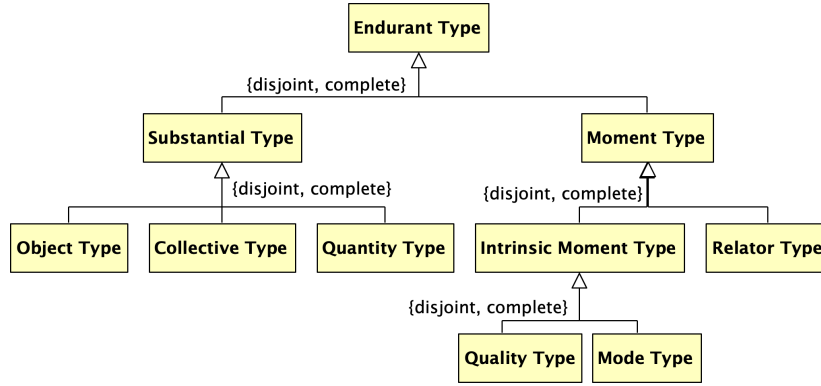


Figure 4: Partial Taxonomy of UFO – Endurant Types (by ontological nature).

```

146 % Endurant Type (by ontological nature)
147
148 fof(ax_endurantType_taxonomy, axiom, (
149   ![X]: ((substantialType(X) | momentType(X)) <=> (endurantType(X))
150  )
151 )).
152
153 fof(ax_endurantType_partition, axiom, (
154   ~?[X]: (substantialType(X) & momentType(X))
155 )).

```

```

155
156 % Substantial Type
157
158 fof(ax_substantialType_taxonomy, axiom, (
159   ![X]: ((objectType(X) | collectiveType(X) | quantityType(X)) <=>
160     (substantialType(X)))
161 ))).
162
163 fof(ax_substantialType_partition, axiom, (
164   ~?[X]: ((objectType(X) & collectiveType(X)) | (objectType(X) &
165     quantityType(X)) | (collectiveType(X) & quantityType(X)))
166 ))).
167
168 % Moment Type
169
170 fof(ax_momentType_taxonomy, axiom, (
171   ![X]: ((intrinsicMomentType(X) | relatorType(X)) <=> (momentType(X)))
172 ))).
173
174 fof(ax_momentType_partition, axiom, (
175   ~?[X]: (intrinsicMomentType(X) & relatorType(X))
176 ))).
177
178 % Intrinsic Moment Type
179
180 fof(ax_intrinsicMomentType_taxonomy, axiom, (
181   ![X]: ((qualityType(X) | modeType(X)) <=> (intrinsicMomentType(X)))
182 ))).
183
184 fof(ax_intrinsicMomentType_partition, axiom, (
185   ~?[X]: (qualityType(X) & modeType(X))
186 ))).
187
188 % Endurant Type (by ontological nature) partial taxonomy instances
189
190 fof(ax_endurantType_instances, axiom, (
191   substantialType(substantialType1) & momentType(momentType1) &
192   objectType(objectType1) & collectiveType(collectiveType1) &
193   quantityType(quantityType1) & intrinsicMomentType(
194     intrinsicMomentType1) & relatorType(relatorType1) & qualityType(
195     qualityType1) & modeType(modeType1) &
196   externallyDependentModeType(externallyDependentModeType1) &
197   quaIndividualType(quaIndividualType1)
198 ))).
199
200 % Endurant Type (by modal properties of types)
201
202 fof(ax_endurantType_taxonomy, axiom, (
203   ![X]: ((sortal(X) | nonSortal(X)) <=> (endurantType(X)))
204 ))).
205
206 fof(ax_endurantType_partition, axiom, (
207   ~?[X]: (sortal(X) & nonSortal(X))
208 ))).
209
210 % Sortal

```

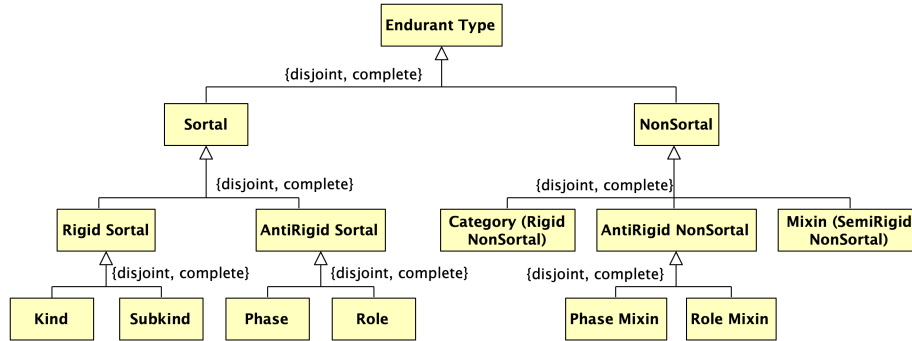


Figure 5: Partial Taxonomy of UFO – Endurant Types (by modal properties of types).

```

203
204 fof(ax_sortal_taxonomy, axiom, (
205   ![X]: ((rigidSortal(X) | antiRigidSortal(X)) <=> (sortal(X)))
206 )).
207
208 fof(ax_sortal_partition, axiom, (
209   ~?[X]: (rigidSortal(X) & antiRigidSortal(X))
210 )).
211
212 % Rigid Sortal
213
214 fof(ax_rigidSortal_taxonomy, axiom, (
215   ![X]: ((kind(X) | subkind(X)) <=> (rigidSortal(X)))
216 )).
217
218 fof(ax_rigidSortal_partition, axiom, (
219   ~?[X]: (kind(X) & subkind(X))
220 )).
221
222 % Anti-Rigid Sortal
223
224 fof(ax_antiRigidSortal_taxonomy, axiom, (
225   ![X]: ((phase(X) | role(X)) <=> (antiRigidSortal(X)))
226 )).
227
228 fof(ax_antiRigidSortal_partition, axiom, (
229   ~?[X]: (phase(X) & role(X))
230 )).
231
232 % Non-Sortal
233
234 fof(ax_nonSortal_taxonomy, axiom, (
235   ![X]: ((rigidNonSortal(X) | semiRigidNonSortal(X) |
236     antiRigidNonSortal(X)) <=> (nonSortal(X)))
237 )).
238
239 fof(ax_nonSortal_partition, axiom, (
240   ~?[X]: ((rigidNonSortal(X) & semiRigidNonSortal(X)) | (

```

```

    rigidNonSortal(X) & antiRigidNonSortal(X)) | (
    semiRigidNonSortal(X) & antiRigidNonSortal(X)))
240)).
241
242% Category
243
244fof(ax_rigidNonSortal_taxonomy, axiom, (
245    ![X]: (rigidNonSortal(X) <=> (category(X)))
246)).
247
248% Mixin
249
250fof(ax_semiRigidNonSortal_taxonomy, axiom, (
251    ![X]: (semiRigidNonSortal(X) <=> (mixin(X)))
252)).
253
254% Anti-Rigid Non-Sortal
255
256fof(ax_antiRigidNonSortal_taxonomy, axiom, (
257    ![X]: ((phaseMixin(X) | roleMixin(X)) <=> (antiRigidNonSortal(X))
258    ))).
259
260fof(ax_antiRigidNonSortal_partition, axiom, (
261    ~?[X]: (phaseMixin(X) & roleMixin(X))
262)).
263
264% Endurant Type (by modal properties of types) partial taxonomy
    instances
265
266fof(ax_endurantType_instances, axiom, (
267    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)
268)).

```

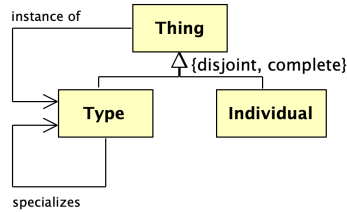


Figure 6: Types, individuals, instantiation, and specialization.

```

270 %%%%%%%%% Instance of, Types, and Individuals %%%%%%%%%
271
272fof(ax_dIof, axiom, (
273    ![X,Y,W]: (iof(X,Y,W) => (individual(X) & type(Y) & world(W)))
274)).

```



```

275
276 fof(ax_dType_a1, axiom, (
277   ![X]: (type(X) <=> (?[Y,W]: iof(Y,X,W)))
278 )).
279
280 fof(ax_dIndividual_a2, axiom, (
281   ![X]: (individual(X) <=> (~?[Y,W]: iof(Y,X,W)))
282 )).
283
284 % TODO: confirm whether we are including second-order types in this
      formalization
285
286 fof(ax_multiLevel_a3, axiom, (
287   ![X,Y,W]: (iof(X,Y,W) => (type(X) | individual(X)))
288 )).
289
290 fof(ax_twoLevelConstrained_a4, axiom, (
291   ~?[X,Y,Z,W]: (type(X) & iof(X,Y,W) & iof(Y,Z,W))
292 )).
293
294 fof(ax_iofInUse, axiom, (
295   type(t2) & individual(i2) & wolrd(w2) & iof(i2,t2,w2)
296 )).
297
298 % 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))").
299
300 % fof(th_everythingIsAThing_t1, conjecture, (
301 %   ![X]: (type(x) | individual(x))
302 % )).
303
304 % fof(th_thingPartition_t2, conjecture, (
305 %   ~?[X]: (type(x) & individual(x))
306 % )).
307
308 %%%%%%%%% Specialization and Proper Specialization %%%%%%%%%
309
310
311
312 % %%%%%%%%%%5
313 % %Specialization
314 % %%%%%%%%%%5
315
316
317 % fof(ax_dspecialization_a5, axiom, (
318 %   ![T1,T2] : (specializes(T1,T2) <=> (type(T1) & type(T2) &
319 %   ![W]: (world(W) => ![E]: (iof(E,T1,W) => iof(E
      ,T2,W)))))).

```