

# Semantic Web Technologies II

## SS 2008

07.05.2008

## Qualität von Ontologien II

**Dr. Peter Haase**  
**PD Dr. Pascal Hitzler**  
**Dr. Steffen Lamparter**  
**Denny Vrandečić**



Content licensed under Creative Commons  
<http://creativecommons.org/licenses/by/2.0/de/>

# Rückblick: Kriterien und Aspekte

- Accuracy
- Adaptability
- Clarity
- Completeness
- Computational efficiency
- Conciseness
- Consistency
- Organizational fitness
- Vokabular
- Syntax
- Struktur
- Semantik
- Repräsentation
- Kontext

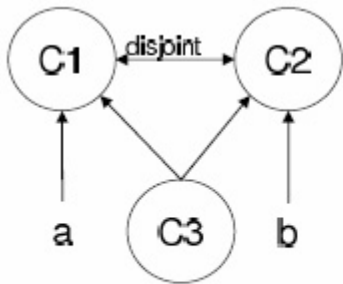
# Übersicht

- Reparieren von Ontologien
- OntoClean
- Muster üblicher Fehler
- Bewerten von Ontologien

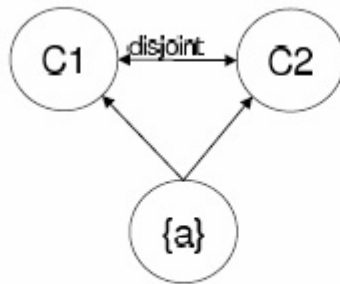
# Reparieren von Ontologien

- **Inkonsistenz:** Kein Modell kann die Ontologie erfüllen (die Axiome sind widersprüchlich)
- **Unerfüllbar:** Eine Klasse kann keine Instanzen haben
- **Inkohärenz:** Die Ontologie hat unerfüllbare Klassen
- **Reparieren:** Entfernen von Inkonsistenz und Inkohärenz

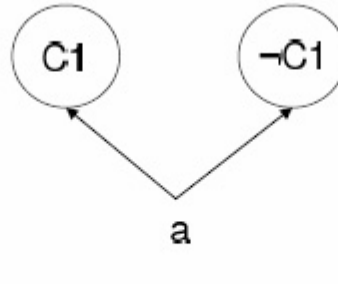
# Inkonsistenz und Inkohärenz



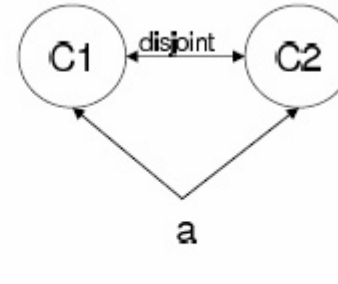
(1)



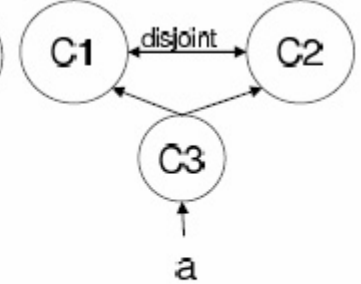
(2)



(3)



(4)



(5)

*konsistent,  
inkohärent*

*inkonsistent,  
inkohärent*

*inkonsistent,  
kohärent*

*inkonsistent,  
kohärent*

*inkonsistent,  
inkohärent*

# Motivation

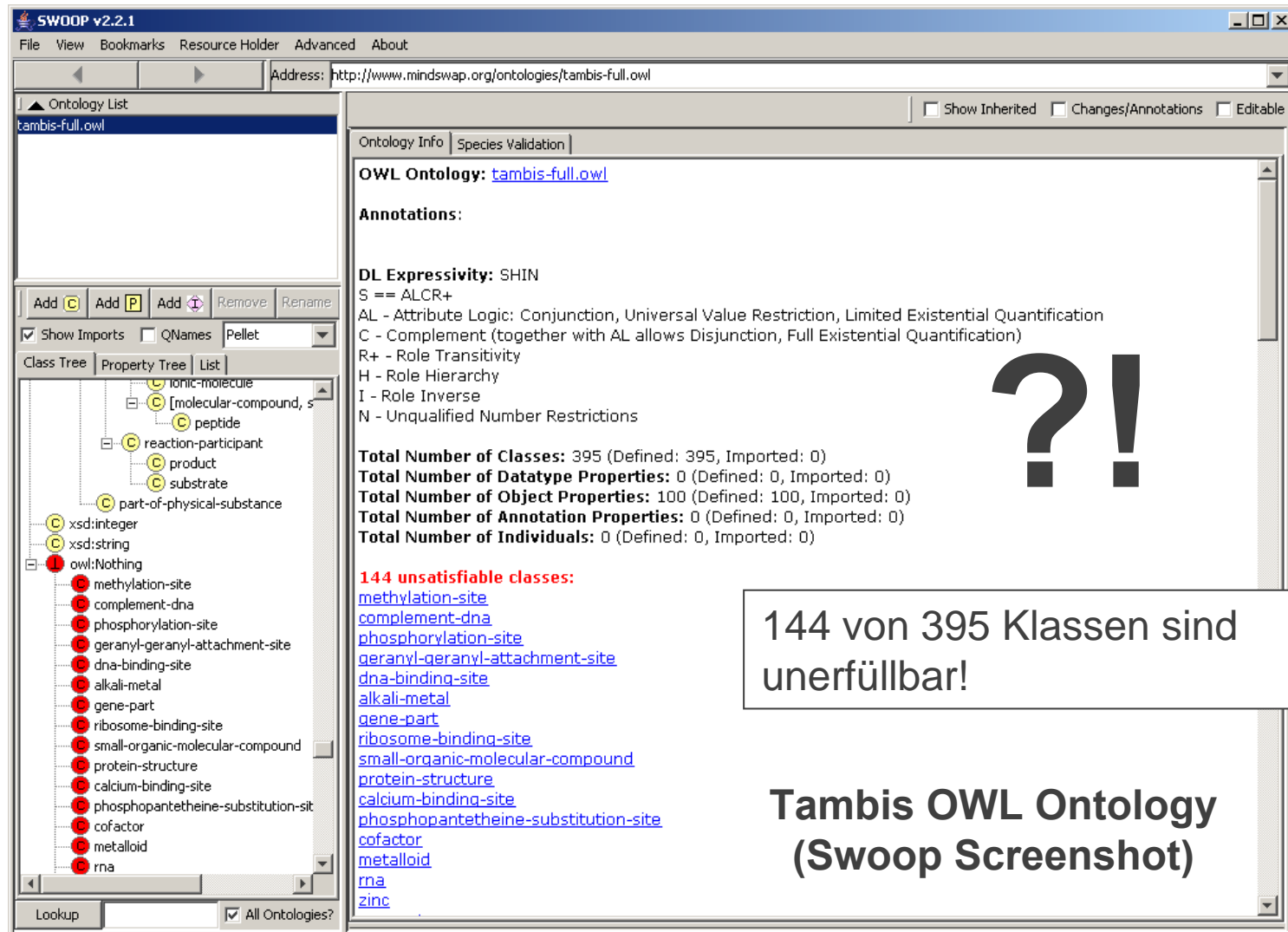
- Logische Fehler haben verschiedene Gründe
  - Modellierungsfehler: der Benutzer missversteht die komplexen DL Axiome und deren Zusammenhang
  - Vereinigung von mehreren Ontologien
  - Ontologieevolution
- Logische Fehler beeinträchtigen die Benutzbarkeit
  - Ergebnisse von Standard-reasonern sind bedeutungslos
  - Aus Falschem folgt beliebiges
- Verschiedene Methoden, um mit Fehlern umzugehen
  - Schlussfolgern mit fehlerhaften Ontologien
  - **Fehler beheben**

# Aufgabenstellung beim Reparieren

1. Relevante Axiome identifizieren
  - Mitsamt der **Begründung** für den Fehler
2. Fehlerwurzel identifizieren
  - **Abgeleitete** Fehler unterscheiden von den Wurzeln der Fehler
3. Reparatur von Fehlern
  - Automatisches Erstellen und Bewerten von Reparaturvorschlägen



# Fehler in Ontologien



Address: <http://www.mindswap.org/ontologies/tambis-full.owl>

Ontology List  
tambis-full.owl

Ontology Info | Species Validation

OWL Ontology: [tambis-full.owl](#)

Annotations:

DL Expressivity: SHIN  
S == ALCR+

AL - Attribute Logic: Conjunction, Universal Value Restriction, Limited Existential Quantification  
C - Complement (together with AL allows Disjunction, Full Existential Quantification)  
R+ - Role Transitivity  
H - Role Hierarchy  
I - Role Inverse  
N - Unqualified Number Restrictions

Total Number of Classes: 395 (Defined: 395, Imported: 0)  
Total Number of Datatype Properties: 0 (Defined: 0, Imported: 0)  
Total Number of Object Properties: 100 (Defined: 100, Imported: 0)  
Total Number of Annotation Properties: 0 (Defined: 0, Imported: 0)  
Total Number of Individuals: 0 (Defined: 0, Imported: 0)

144 unsatisfiable classes:

- [methylation-site](#)
- [complement-dna](#)
- [phosphorylation-site](#)
- [geranyl-geranyl-attachment-site](#)
- [dna-binding-site](#)
- [alkali-metal](#)
- [gene-part](#)
- [ribosome-binding-site](#)
- [small-organic-molecular-compound](#)
- [protein-structure](#)
- [calcium-binding-site](#)
- [phosphopantetheine-substitution-site](#)
- [cofactor](#)
- [metalloid](#)
- [rna](#)
- [zinc](#)

144 von 395 Klassen sind unerfüllbar!

# Relevante Axiome identifizieren

## Zwei Herangehensweisen

- Glass Box (Reasoner-abhängig): Reasoner modifizieren, um die relevanten Axiome per Analyse zu identifizieren
- Black Box (Reasoner-unabhängig): Reasoner als „Orakel“ verwenden und relevante Axiome per test zu entdecken

- Für den Fehler relevante Axiome identifizieren
  - MUPS: Minimal unsatisfiability-preserving sub-TBox
- Brut Force Black Box Methode
  - Entferne Axiome aus  $O$  bis das nicht mehr möglich ist, ohne dass die Ontologie konsistent wird
  - Das Ergebnis ist ein MUPS von  $O$
- Eigenschaften von MUPS
  - Submenge von  $O$
  - Egal welches Axiom entfernt wird, der MUPS ist danach konsistent

# Kritische Fehler identifizieren

Wenn es viele Fehler gibt:

- Einen Fehler nach den anderen zu verarbeiten dauert lange, kann sinnlos sein, und manchmal sogar schädlich
- Man muss die kritischen Fehler (Wurzel) identifizieren, welche die sonstigen (Abgeleiteten) Fehler verursachen
- Beispiele:
  - $C = (A \text{ und } B)$  ist unerfüllbar wenn B unerfüllbar ist
  - $A = \exists R.B$  ist unerfüllbar wenn B unerfüllbar ist

# Kritische Fehler identifizieren

**SWOOP v2.3 beta 3 (Jan 2006)**

File View Bookmarks Resource Holder Advanced About

Address: <http://www.mindswap.org/ontologies/tambis-full.owl>

Ontology List: **tambis-full.owl\***

Ontology Info | Species Validation

OWL Ontology: [tambis-full.owl](#) (Edit URI)

Annotations: (Add)

Imports: (Add)

**Root/Derived Debugging Information:**

**66 unsatisfiable classes:**

**3 kritische Fehler korrigieren beseitigt 78 Fehler**

**Neue Menge von kritischen Fehlern berechnet (8)**

**Diese zu korrigieren beseitigte alle restlichen Fehler**

root unsat. classes (8)
<a href="#">enzyme</a>
<a href="#">hydrolase</a>
<a href="#">lyase</a>
<a href="#">oxidoreductase</a>
<a href="#">peptidase</a>
<a href="#">phosphatase</a>
<a href="#">ribosomal-rna</a>
<a href="#">small-nuclear-rna</a>

derived unsat. classes (58)	parent dependencies
<a href="#">acetylation-site</a>	<a href="#">modification-site</a> , <a href="#">protein-part</a> ,
<a href="#">active-site</a>	<a href="#">protein</a> , <a href="#">site</a> , <a href="#">protein-part</a> ,
<a href="#">amidation-site</a>	<a href="#">modification-site</a> , <a href="#">protein-part</a> ,

Class Tree | Property Tree | List

- owl:Nothing
- methylation-site
- complement-dna
- phosphorylation-site
- dna-binding-site
- geranyl-geranyl-attachment
- myristoylation-site
- protein
- ribosome-binding-site
- glycosaminoglycan-substitu
- lyase
- phosphopantetheine-subst
- calcium-binding-site
- hydroxylation-site
- amidation-site
- coenzyme-requiring-holoen
- n-acyl-diglyceride-attachme
- prosthetic-group
- rna

# Fehler beseitigen

- Bringe die Axiome in ein Ranking
  - Wie viele Fehler verursacht ein Axiom?
  - Welche Auswirkung hat ein Entfernen?
  - Wie stark ist das Axiom verbunden?
  - Wie wichtig ist das Axiom (für den Benutzer, das Tool...)?
- Iteratives Entfernen oder Umschreiben einzelner Axiome unter Anleitung des Benutzers
  - Damit der Benutzer anleiten kann, muss er verstehen was passiert
  - Informierte Entscheidungen

# Ontologie reparieren

Repairing Ontology university.owl

Weights: (Note: Rank = -W1 \* arity + W2 \* impact + W3 \* usage) W1: 0.9 W2: 0.7 W3: 0.1 Recompute Ranks  View Axioms Globally

Erroneous Axioms	Arity	Impact	Usage	Rank	Status
1) (Lecturer $\sqsubseteq$ $\neg$ AssistantProfessor)	2	0	0	-1.8	[R] [Undo]
2) $\perp$ (Lecturer $\sqsubseteq$ (TeachingFaculty $\sqcap$ ( $\exists$ hasTenure . {"false"^^<xsd:boolean>})))	2	4	4	1.4	[R] [K]
3) (AssistantProfessor $\sqsubseteq$ (TeachingFaculty $\sqcap$ ( $\exists$ hasTenure . {"false"^^<xsd:boolean>})))	2	4	4	1.4	[R] [K]

**Axioms causing the problem: AISTudent**

Erroneous Axioms	Arity	Impact	Usage	Rank	Status
1) (AISTudent $\sqsubseteq$ $\neg$ HCISTudent)	2	0	1	-1.7	[R] [Undo]
2) (AISTudent $\sqsubseteq$ ( $\exists$ hasAdvisor . ProfessorInHCIorAI))	1	1	4	0.19	[R] [K]
3) $\perp$ (ProfessorInHCIorAI $\sqsubseteq$ ( $\forall$ advisorOf . HCISTudent))	1	1	4	0.19	[R] [K]
4) $\perp$ ( $\_$ (advisorOf inverse hasAdvisor))	2	1	5	-0.6	[R] [Undo]

**Axioms causing the problem: HCISTudent**

Kept Axioms (4)    Removed Axioms (0)     Extended Impact

Generate Plan    Repair All Unsatisfiable     Include Rewrites     Auto Recompute Plan

Main Plan

[X] [Remove] [Keep] (CS\_Library  $\sqsubseteq$  ( $\exists$ affiliatedWith . EE\_Library)) (Arity:1 Impact: 1 Usage: 3)

[X] [Remove] [Keep] (HCISTudent  $\sqsubseteq$  ( $\exists$ hasAdvisor . ProfessorInHCIorAI)) (Arity:1 Impact: 1 Usage: 4)

[X] [Remove] [Keep] (ProfessorInHCIorAI  $\sqsubseteq$  ( $\forall$ advisorOf . HCISTudent)) (Arity:1 Impact: 1 Usage: 4)

[X] [Remove] [Keep] (AssistantProfessor  $\sqsubseteq$  (TeachingFaculty  $\sqcap$  ( $\exists$ hasTenure . {"false"^^<xsd:boolean>}))) (Arity:2 Impact: 4 Usage: 4)

$\perp$ [Rewrite?] (AssistantProfessor  $\sqsubseteq$  (TeachingFaculty  $\sqcap$  ( $\exists$ hasTenure . {"false"^^<xsd:boolean>})))

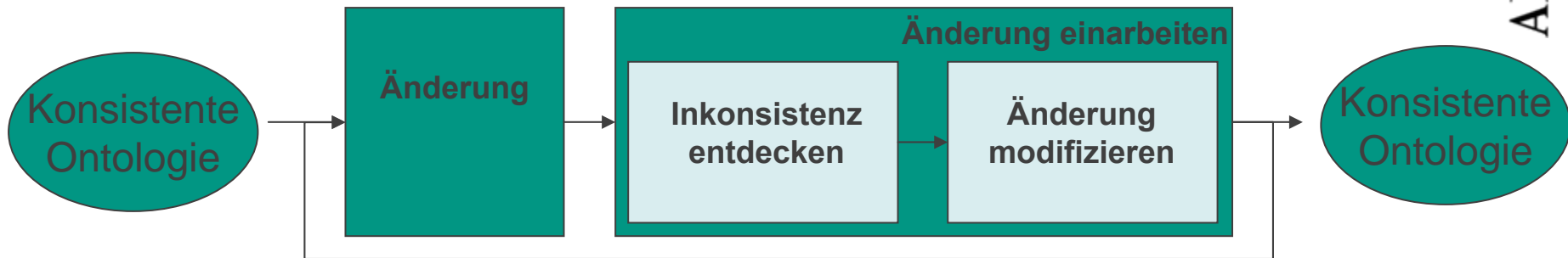
PREVIEW:  
**Unsatisfiable** Fixed:8 Remaining:0  
**Entailments** Lost: 3 Retained: 13

Clear Save Preview Execute

Analyse  
der  
Axiome

Vorschläge

# Konsistente Ontologie-Evolution



- Vorteil: man kennt die Änderung, die zur Inkonsistenz geführt hat
  - Effizientere Fehlererkennung
  - Effektivere Reparaturvorschläge

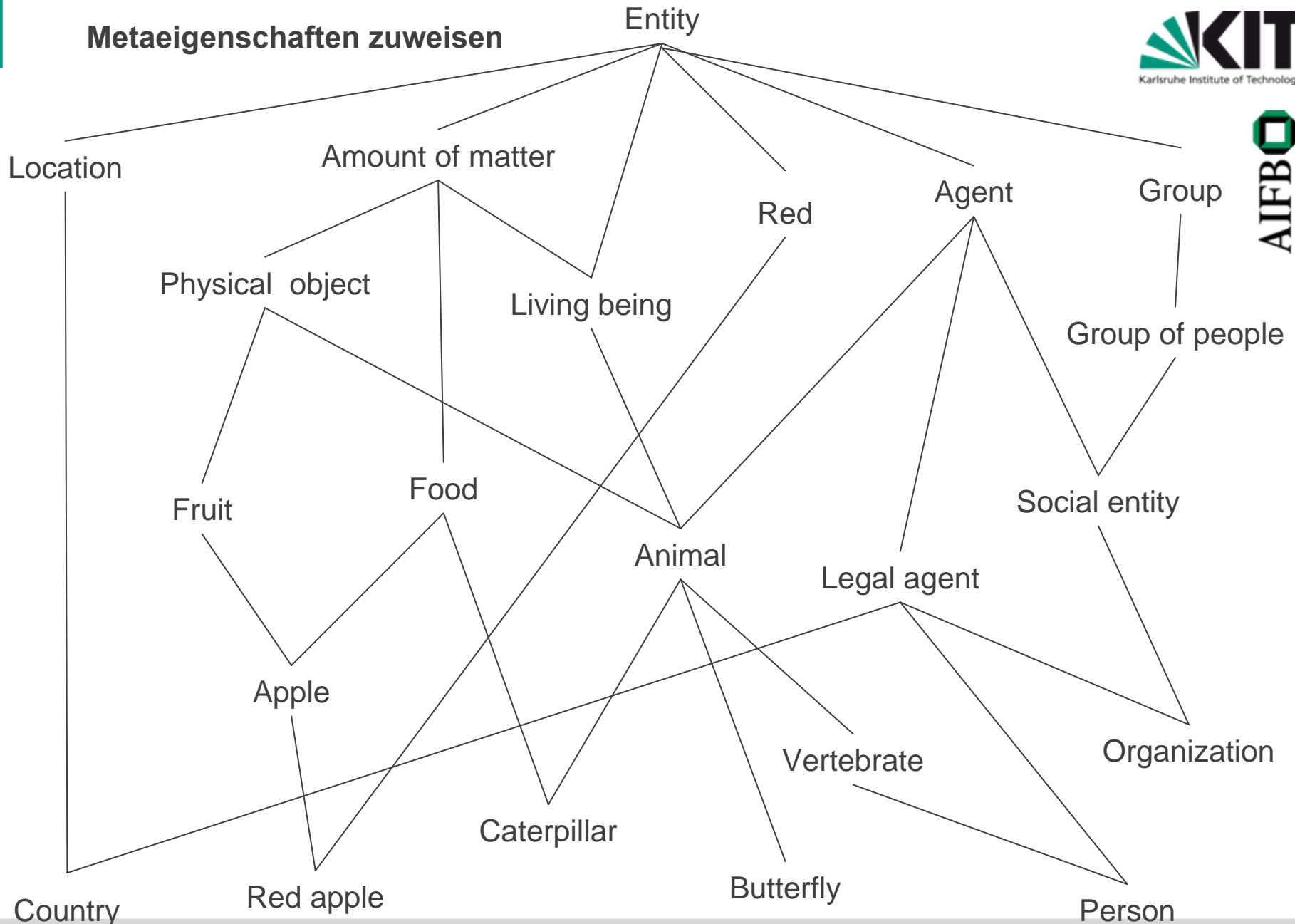


# OntoClean

# Idee von OntoClean

- Ziel: Klassenhierarchie analysieren, Bedeutung der Klassen klären
- Klassen mit Metaeigenschaften taggen
  - Rigidity
  - Identity
  - Dependence
  - Unity
- Klassenhierarchie auf Einschränkungen durch die Metaeigenschaften prüfen

# Metaeigenschaften zuweisen



Country

Red apple

Butterfly

Person

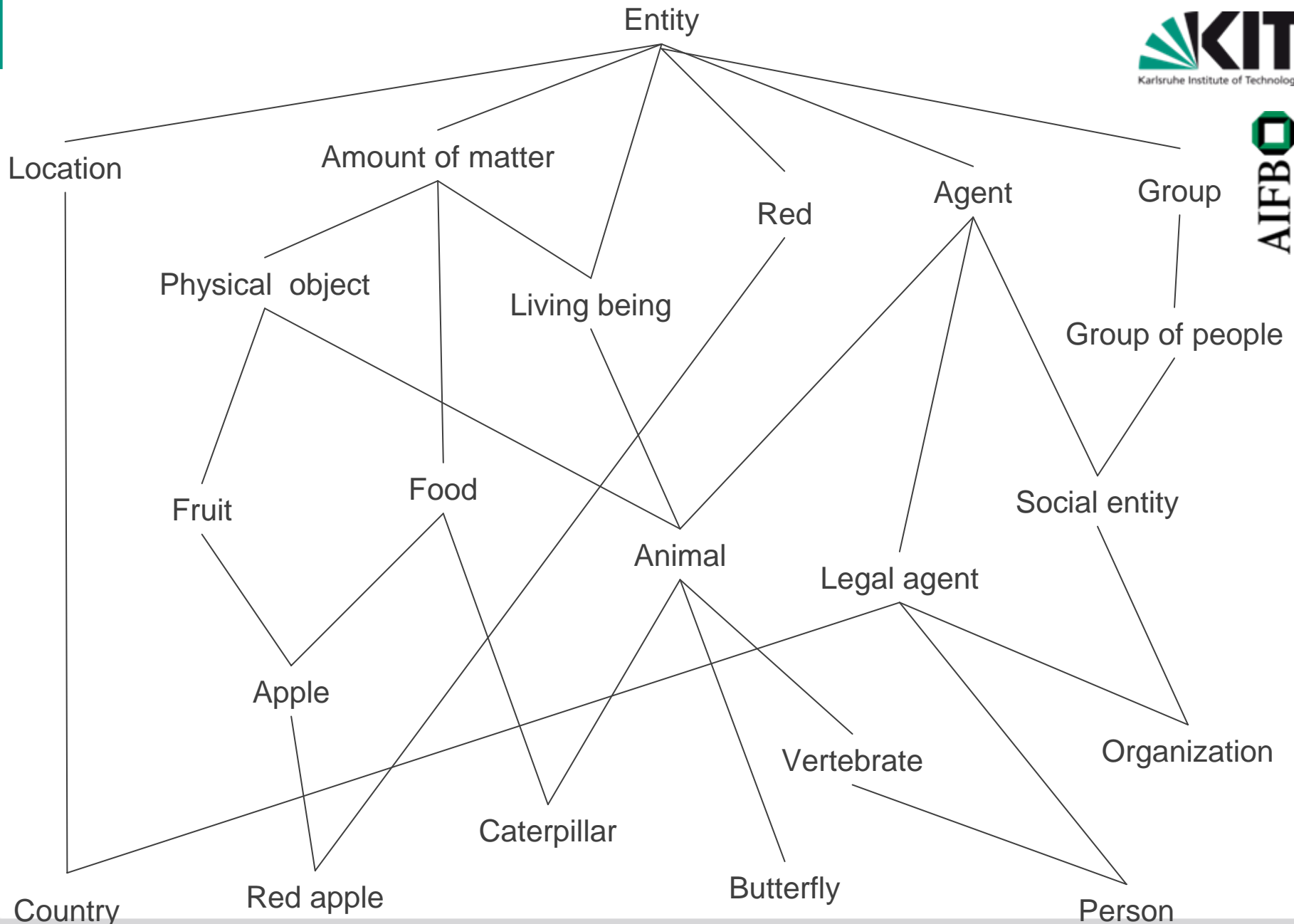
- **Essence** – a class is essential for an individual *iff.* it necessarily holds for that individual
- **Rigidity**
  - A class is rigid (+R) *iff.* it is necessarily essential for all its individuals
  - A class is non-rigid (-R) *iff.* it is not essential for some of its individuals
  - A class is anti-rigid ( $\sim$ R) *iff.* it is not essential for all its individuals
  - $\neg$ R (semi-rigid) = -R without  $\sim$ R
- **Example: Student vs Person**

- Does the property carry an Identity Criterion (IC)?
- A property supplies an IC (+O) *iff.*, by virtue of this property, every instance is identifiable
- -O = does not supply an IC
- +I = carries an IC but does not necessarily supply one
- -I = does not carry an IC

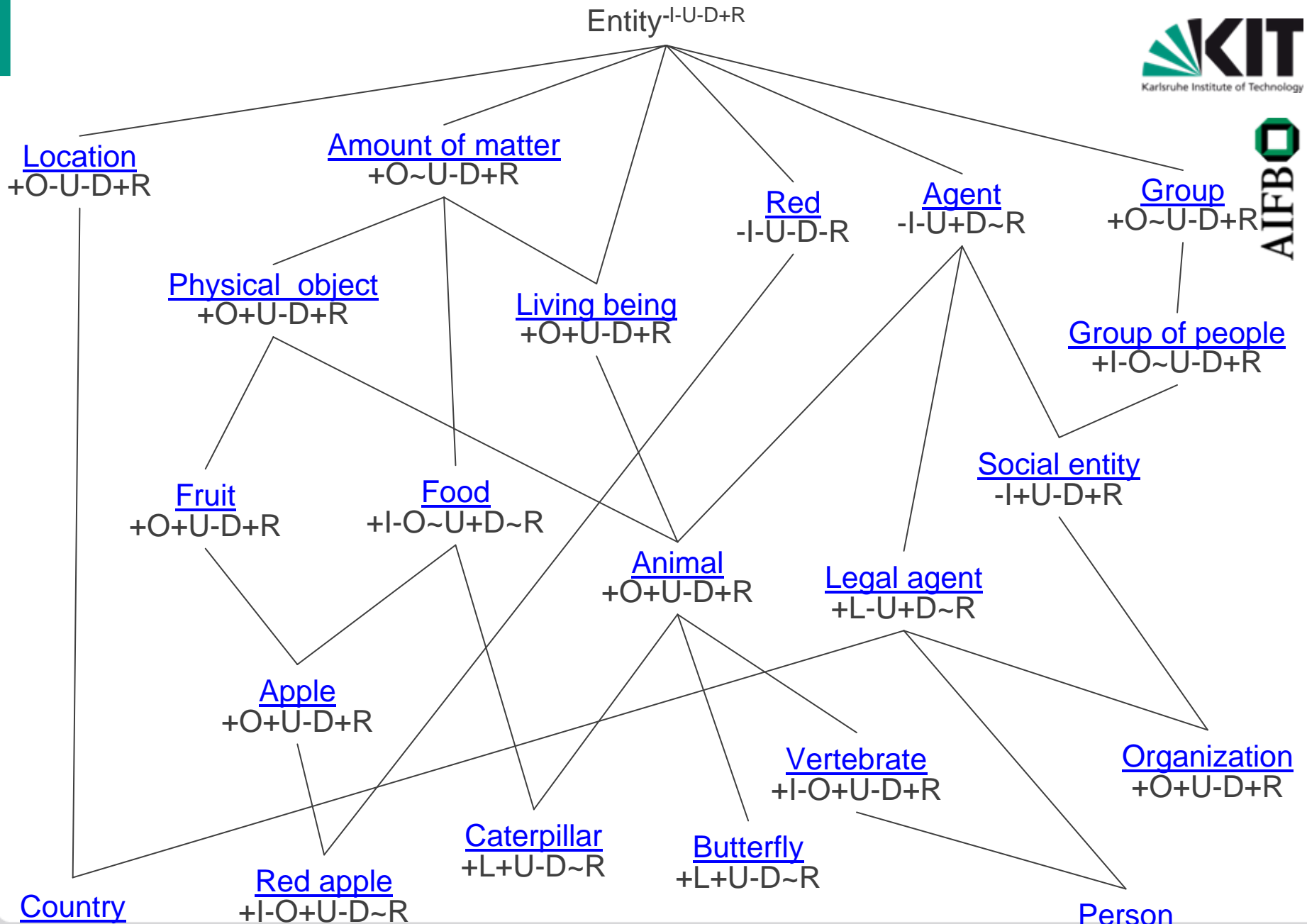
- Does every instance have a Unity Criterion (UC)?
- +U = all instances of the property are wholes with a common UC
- -U (carries no unity) = all instances are wholes but without a common UC
- $\sim$ U (carries anti-unity) = instances of the property are not always wholes

# Dependence

- A property is dependent (+D) *iff.* for every instance of C necessarily some instance of D must exist, that is neither a part nor a constituent of C
- A property is not dependent otherwise







# Metaklassen

+O	+I	+R	+D	Type	Sortal	
			-D			
-O	+I	+R	+D	Quasi-type		
			-D			
-O	+I	~R	+D	Material role		
-O	+I	~R	-D	Phased sortal		
-O	+I	¬R	+D	Mixin		
			-D			
-O	-I	+R	+D	Category		Non-sortal
			-D			
-O	-I	~R	+D	Formal Role		
-O	-I	~R	-D	Attribution		
		¬R	+D			
+O	-I			incoherent		
	+I	~R				
		-R				

# Architektur der Ontologie

Non-sortals

Backbone Taxonomy

Categories

Attributions

Formal  
Roles

Sortals

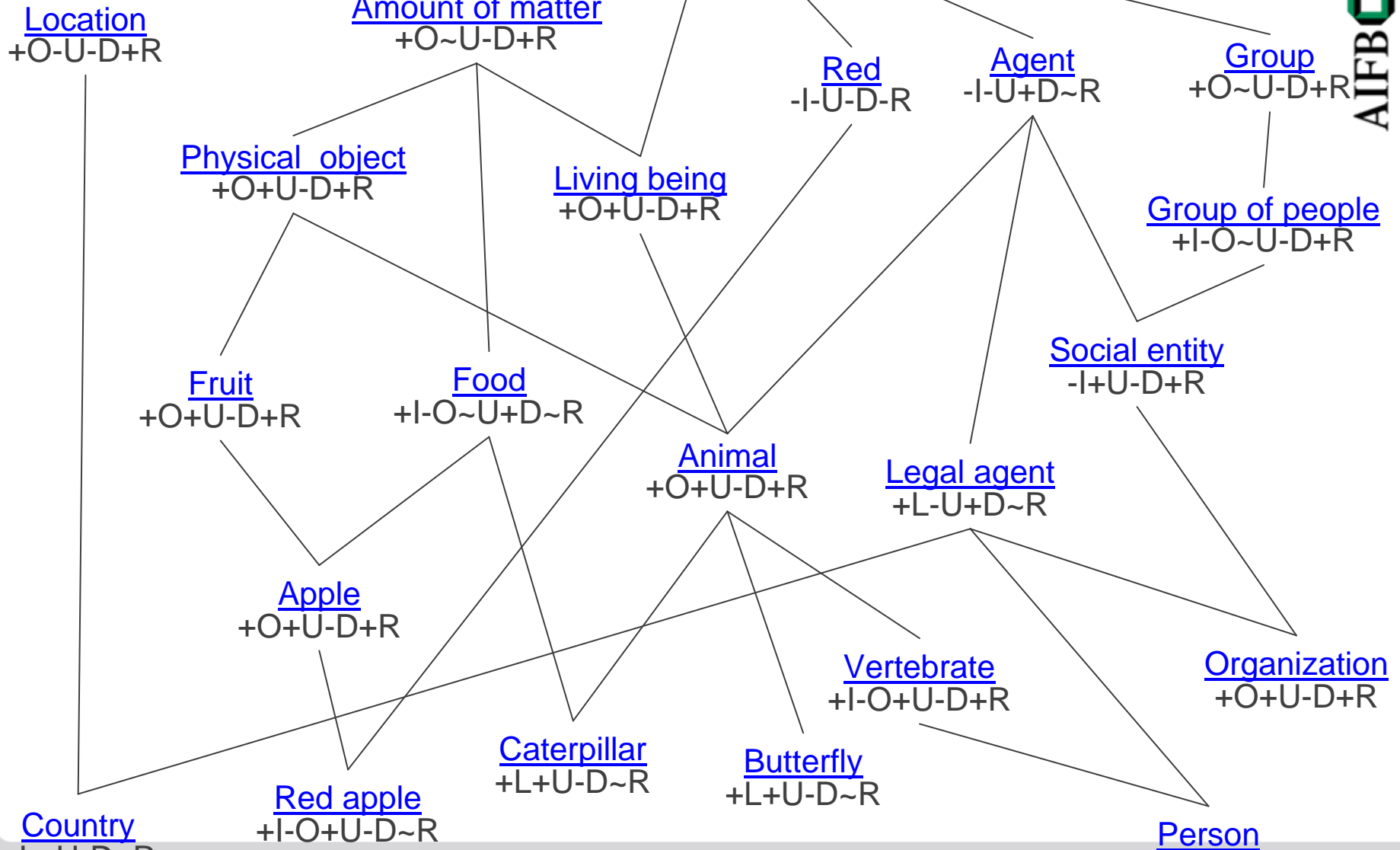
Top Types

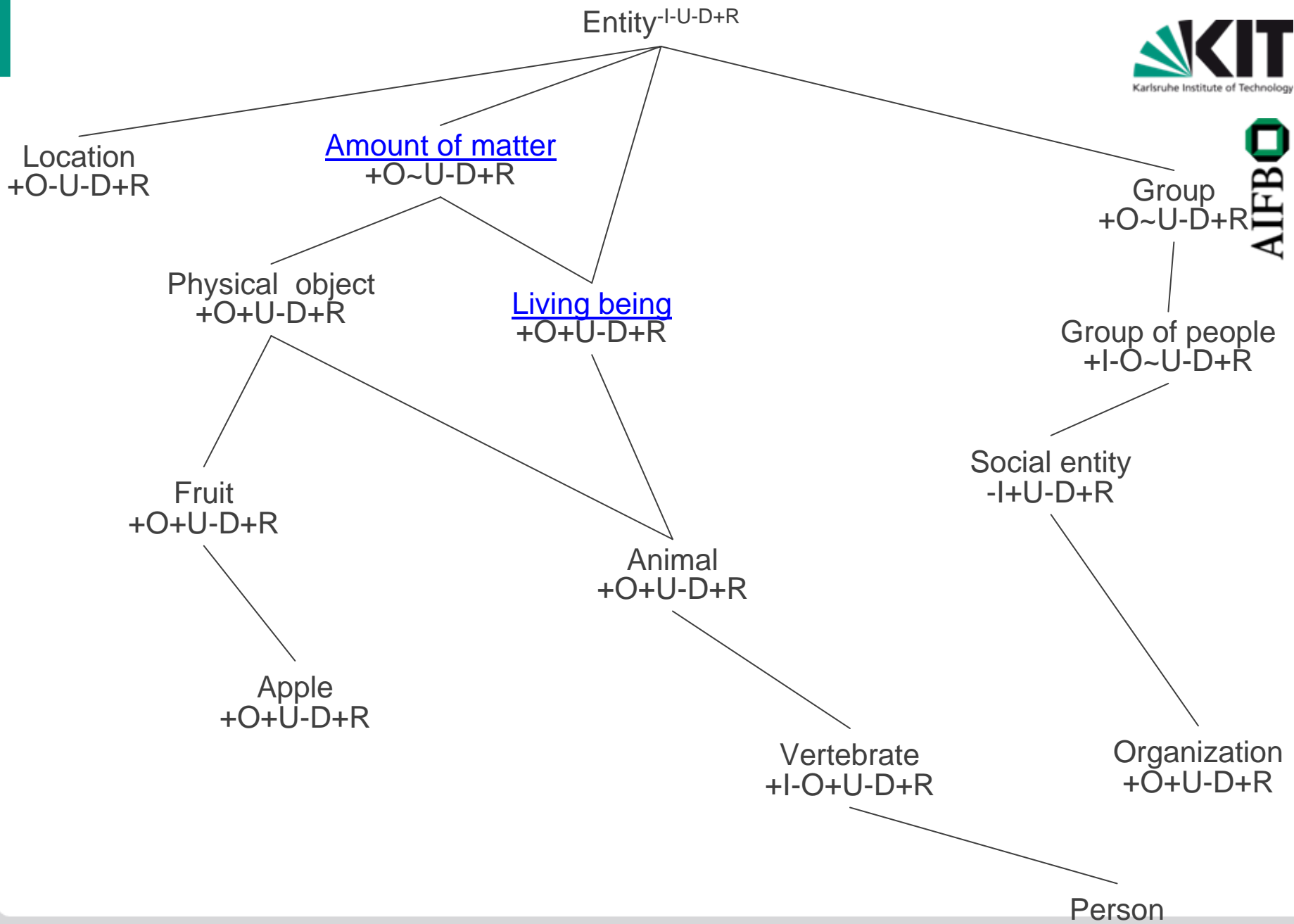
Types &  
Quasi-Types

Mixins

Material  
Roles

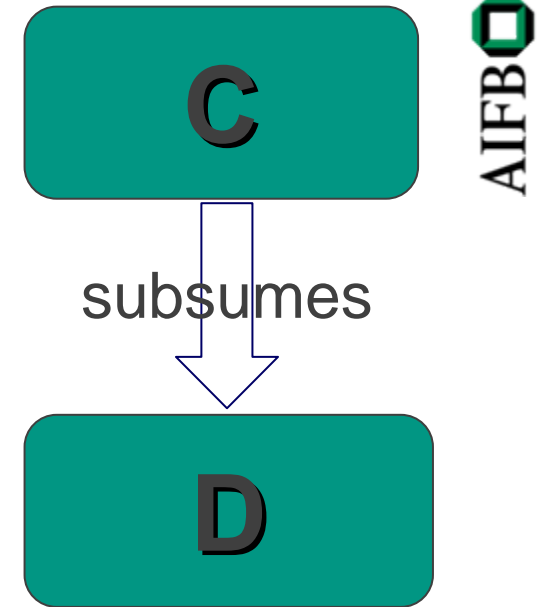
Phased Sortals



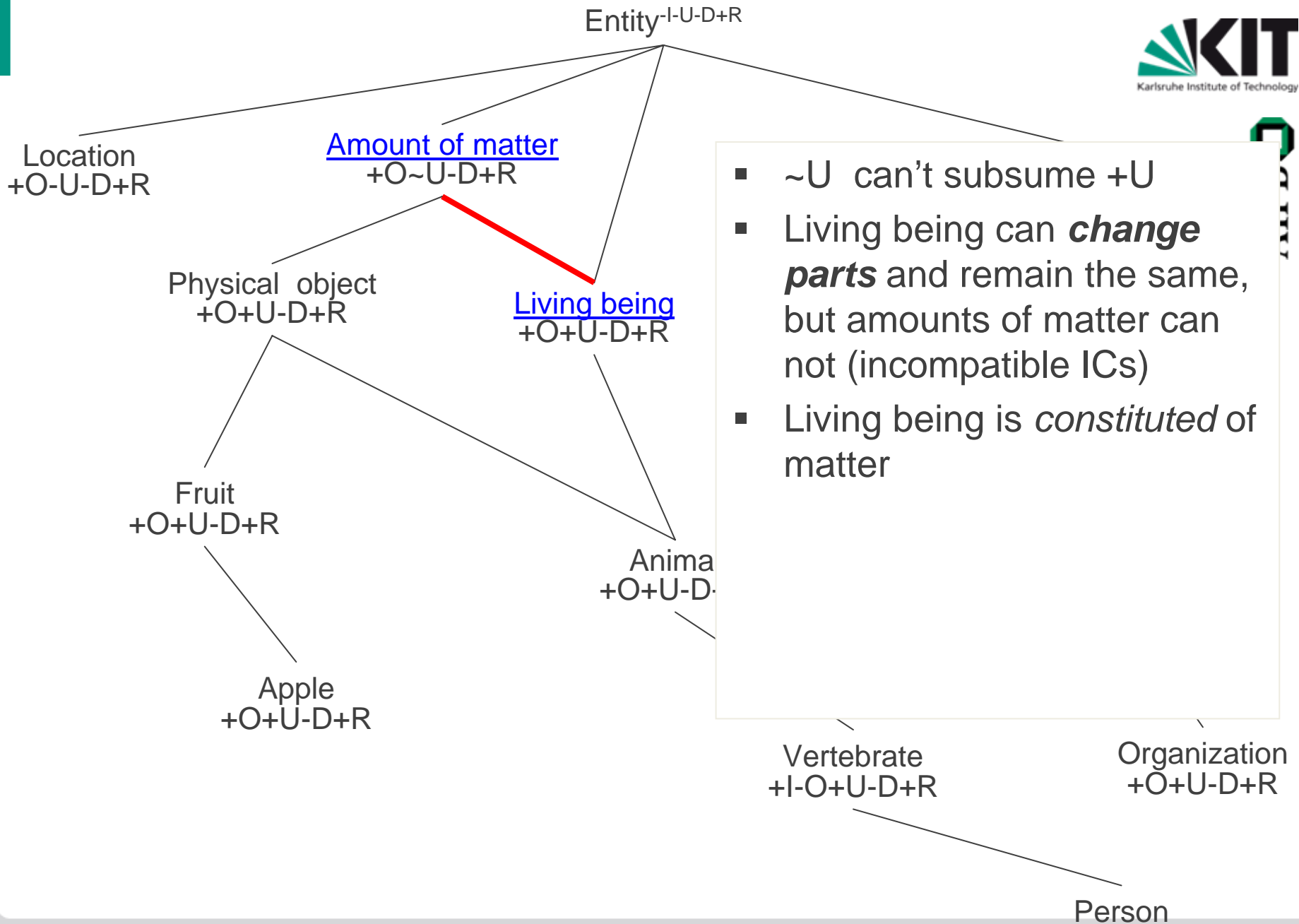


# Consistency rules

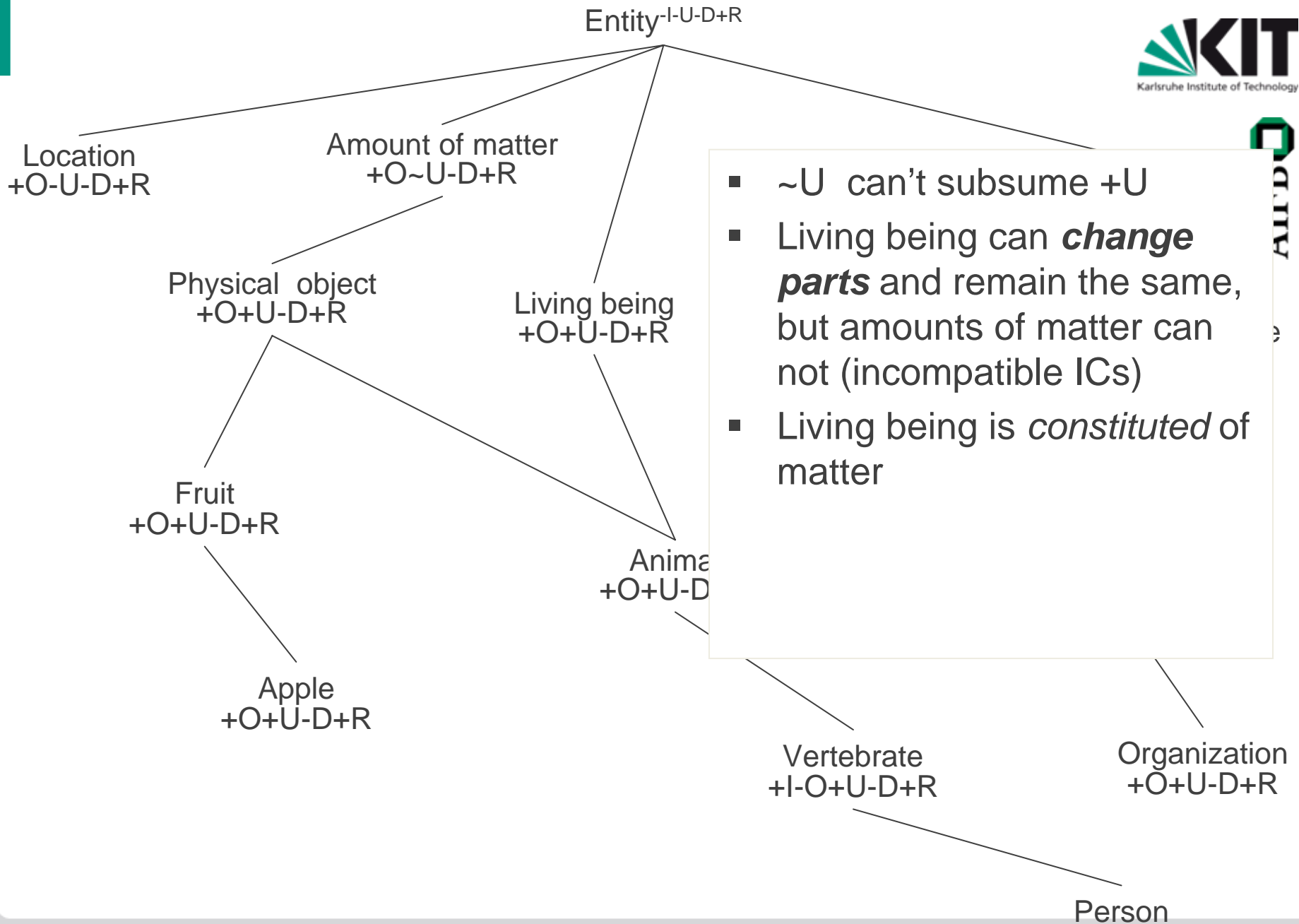
- $+D(C) \rightarrow +D(D)$
- $\sim R(C) \rightarrow \sim R(D)$
- $+O(C) \rightarrow +I(D)$
- $+O(C) \rightarrow +I(C), +R(C)$
- ... and so on



But: these rules are not meant for faster tagging of Ontologies, but for consistency checking!

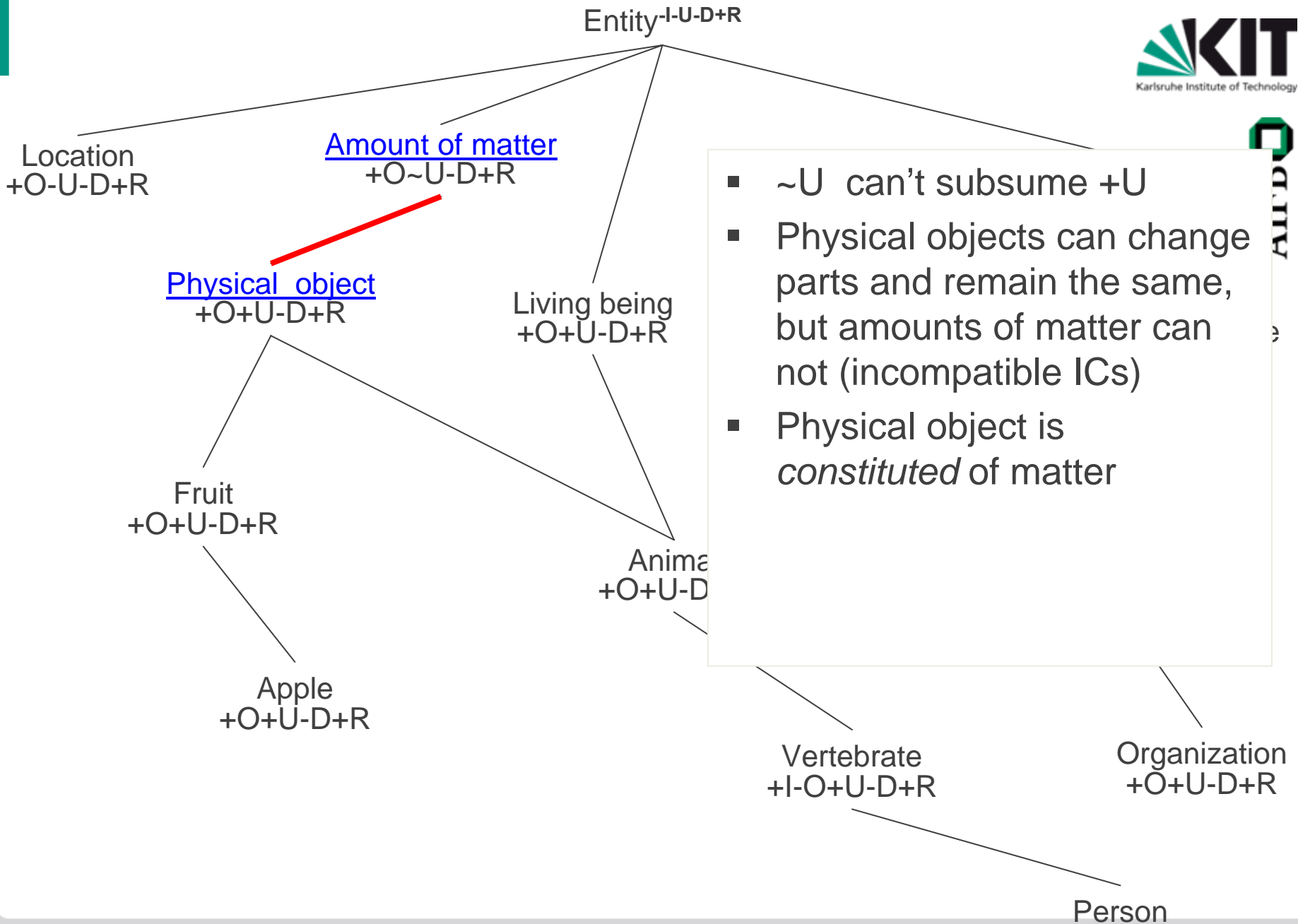


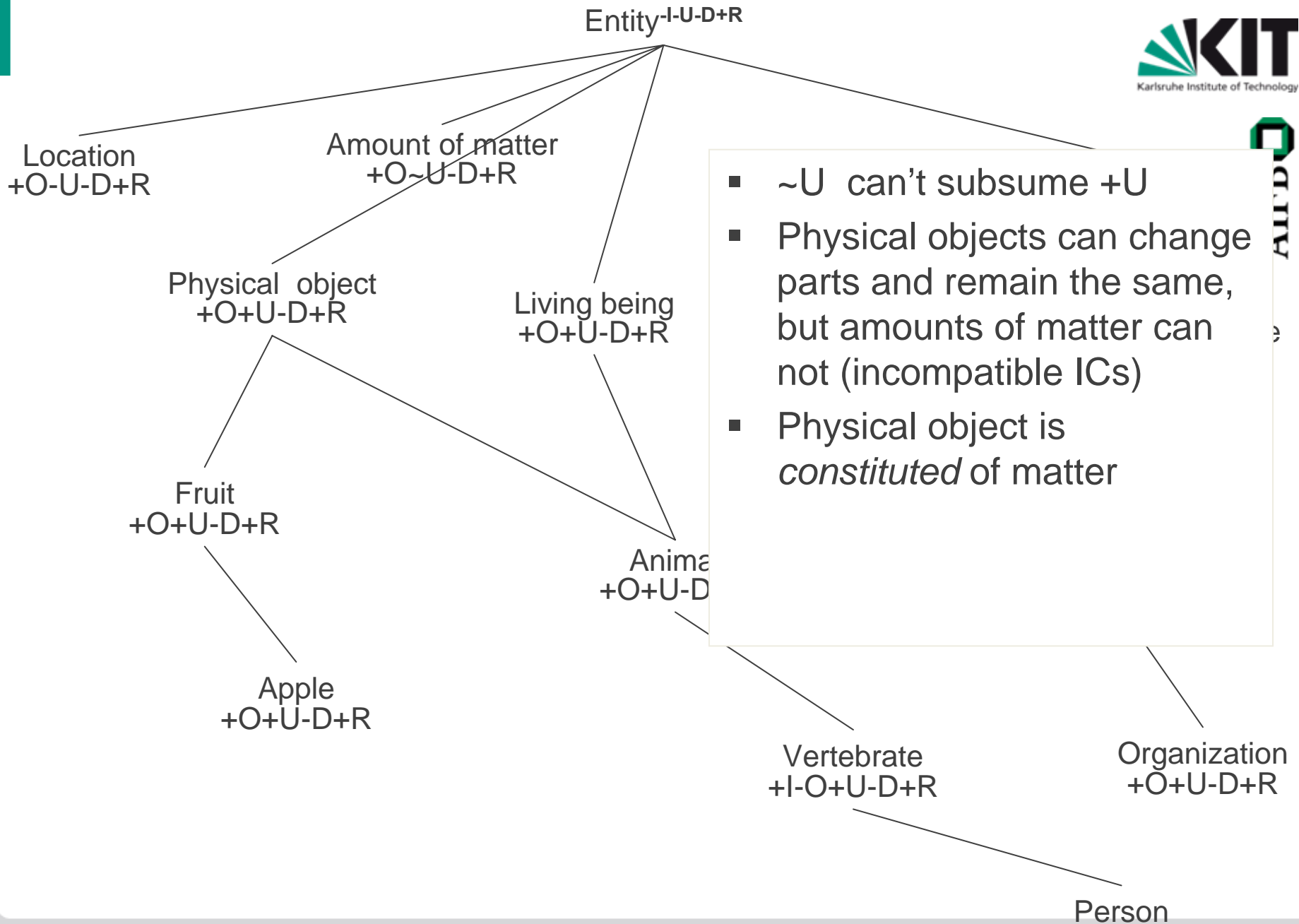
- ~U can't subsume +U
- Living being can **change parts** and remain the same, but amounts of matter can not (incompatible ICs)
- Living being is *constituted* of matter



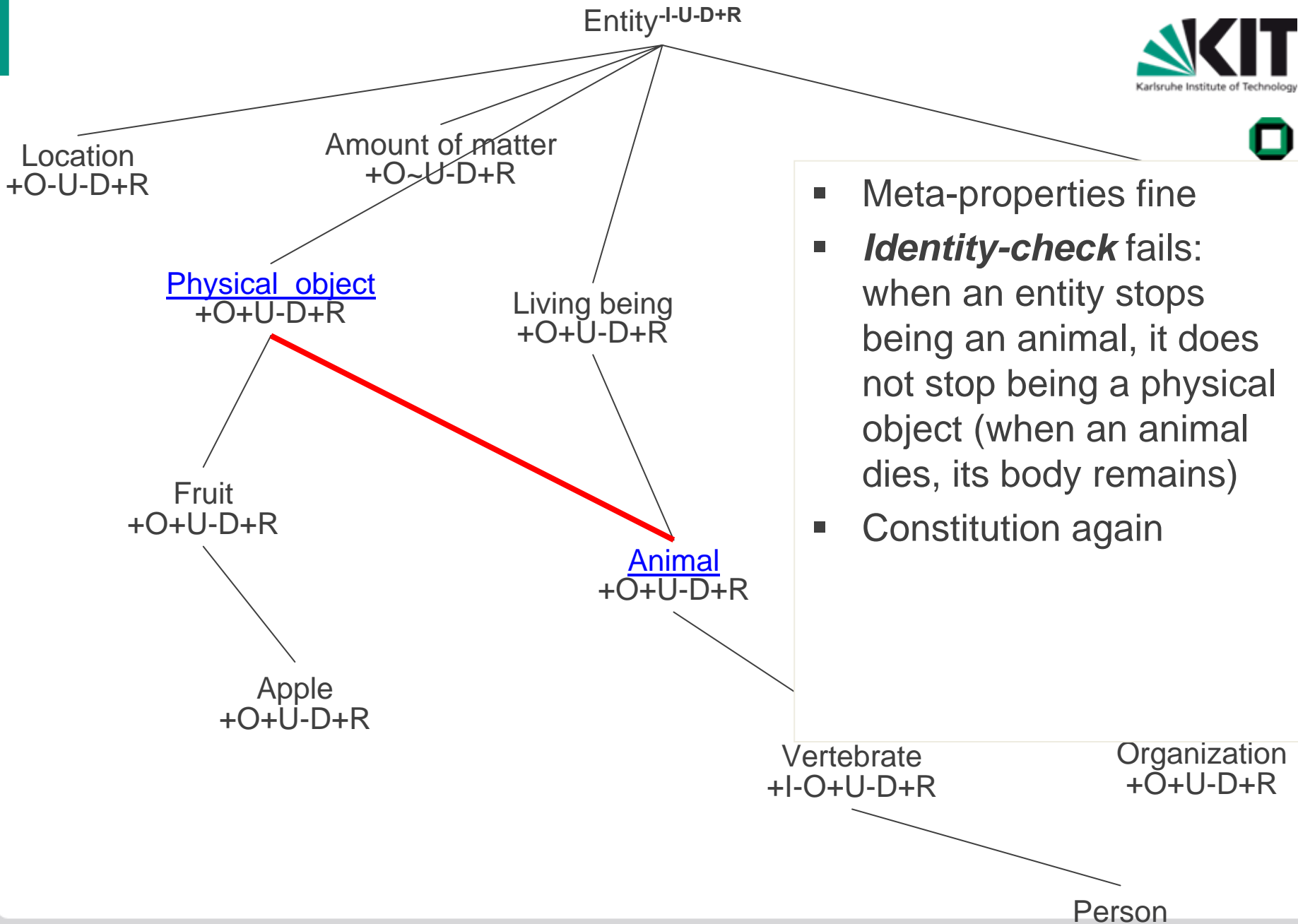
- ~U can't subsume +U
- Living being can **change parts** and remain the same, but amounts of matter can not (incompatible ICs)
- Living being is *constituted* of matter



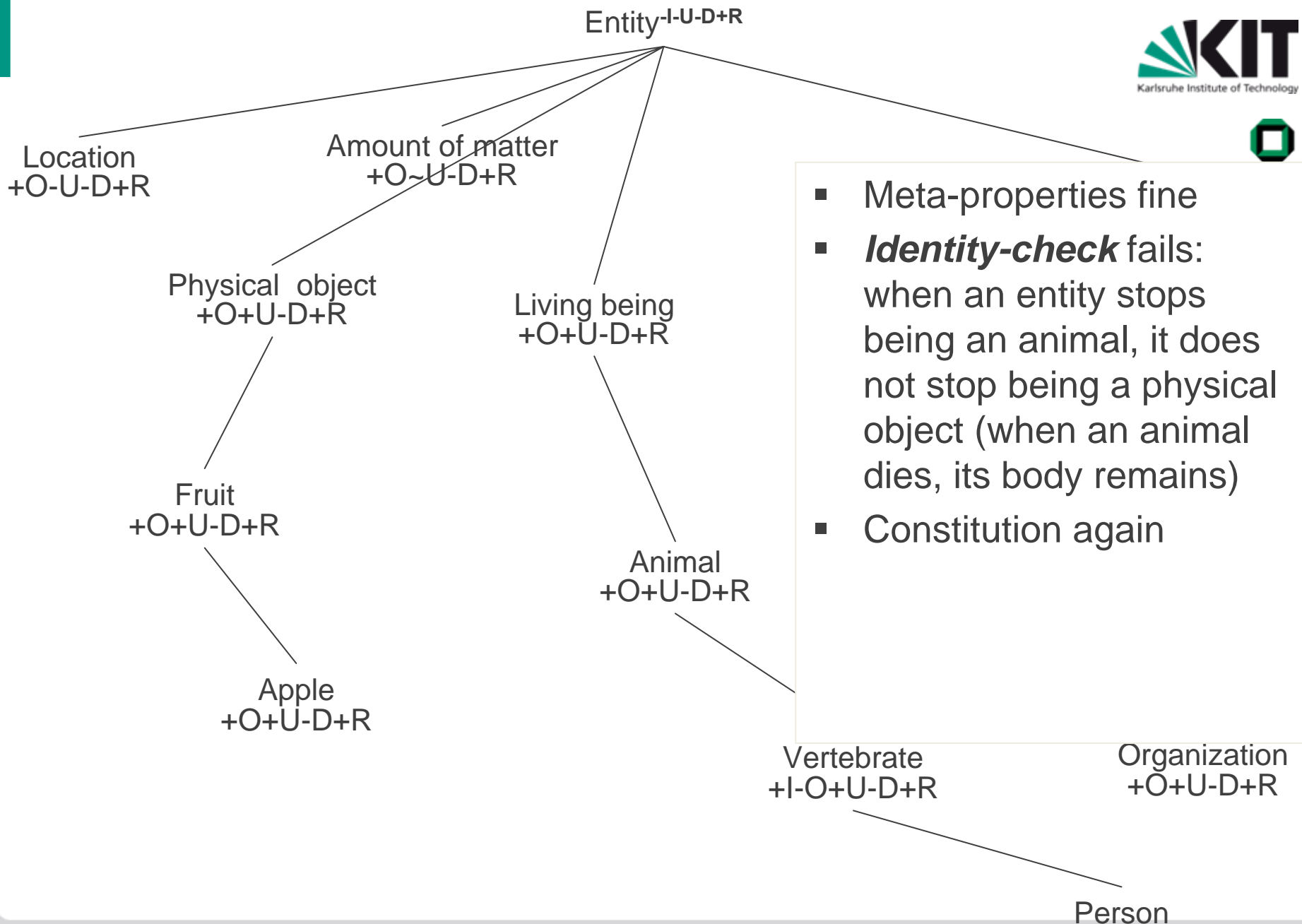




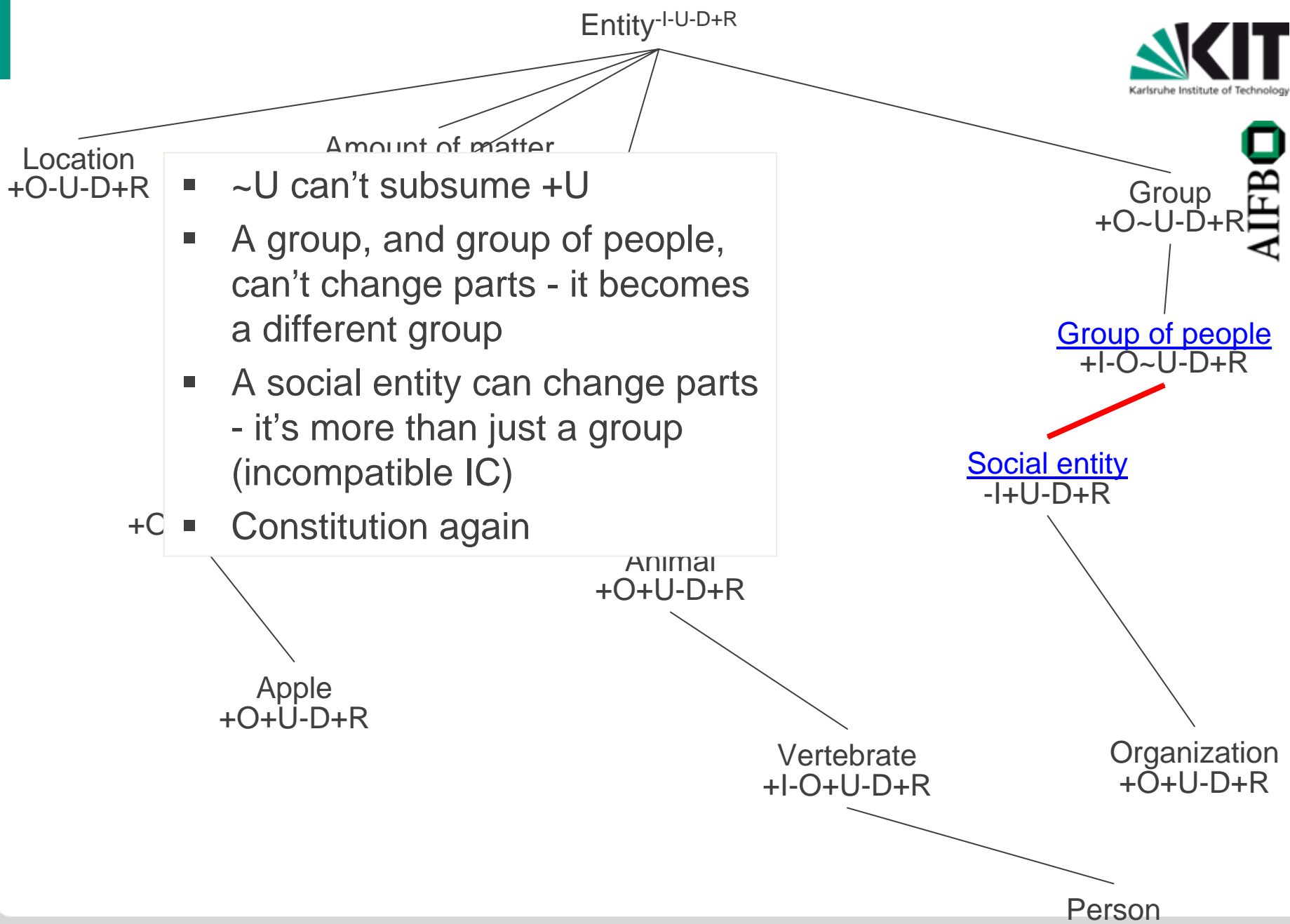
- ~U can't subsume +U
- Physical objects can change parts and remain the same, but amounts of matter can not (incompatible ICs)
- Physical object is *constituted* of matter



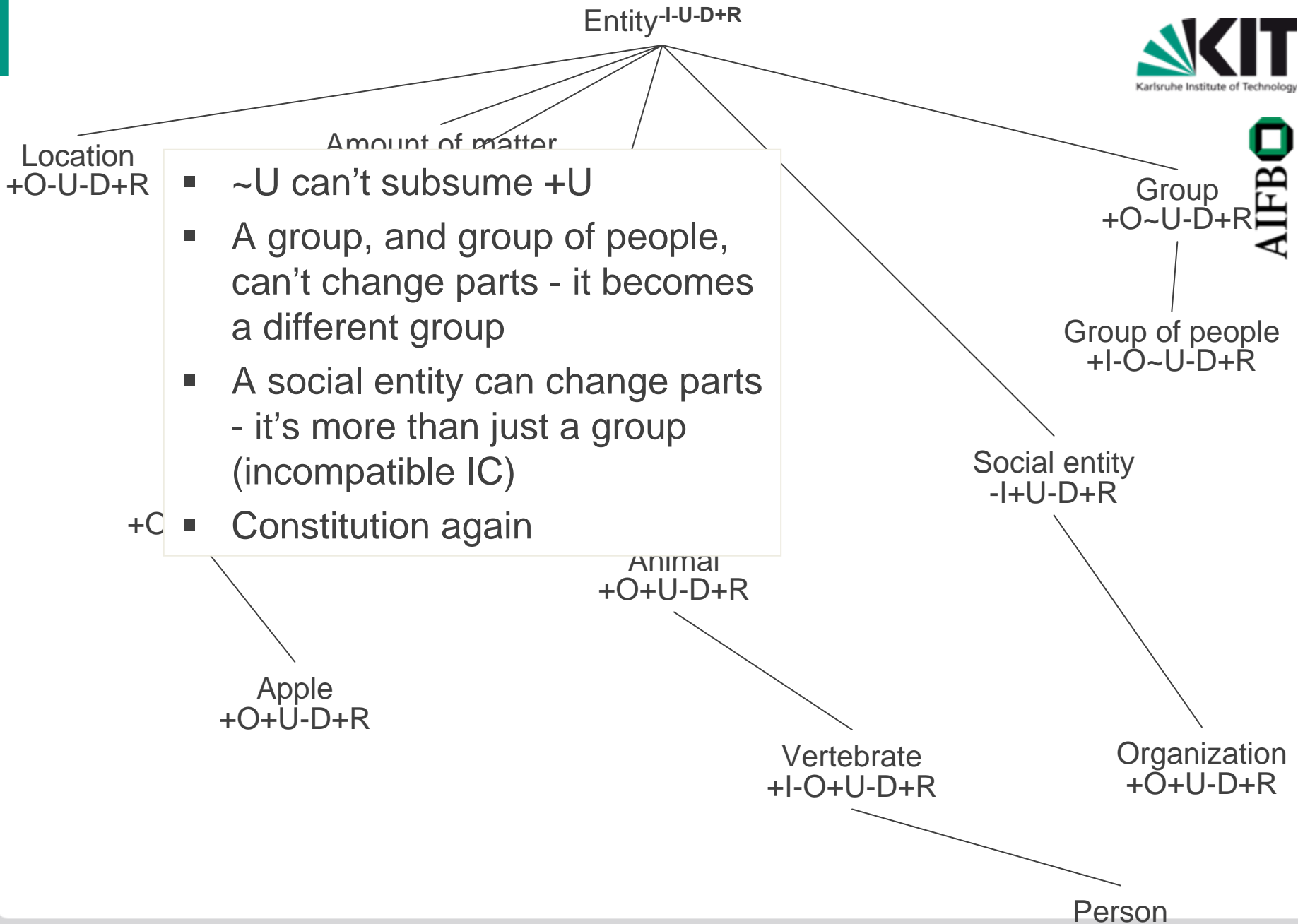
- Meta-properties fine
- **Identity-check** fails:  
when an entity stops  
being an animal, it does  
not stop being a physical  
object (when an animal  
dies, its body remains)
- Constitution again

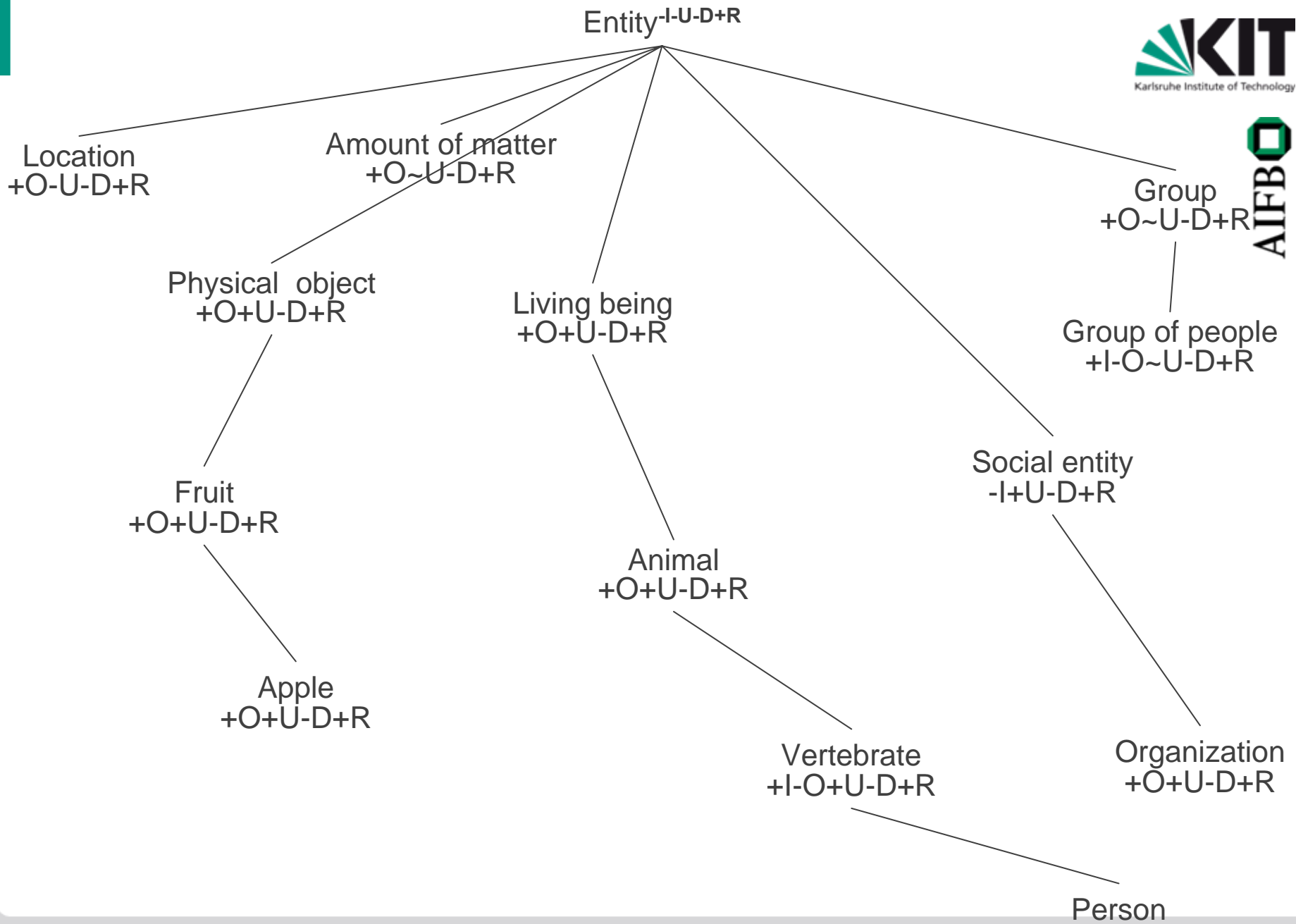


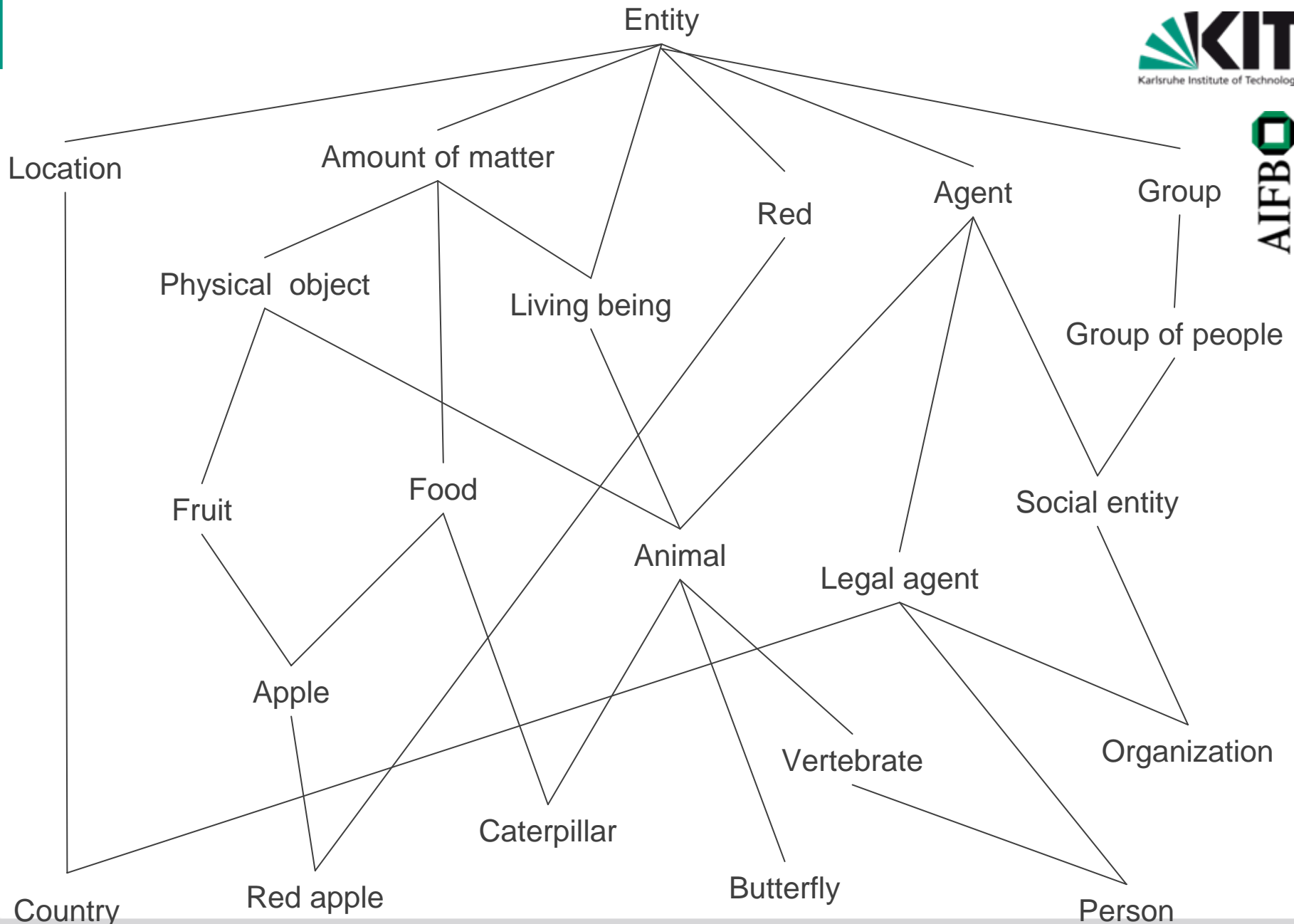
- Meta-properties fine
- **Identity-check** fails:  
when an entity stops being an animal, it does not stop being a physical object (when an animal dies, its body remains)
- Constitution again



- ~U can't subsume +U
- A group, and group of people, can't change parts - it becomes a different group
- A social entity can change parts - it's more than just a group (incompatible IC)
- Constitution again









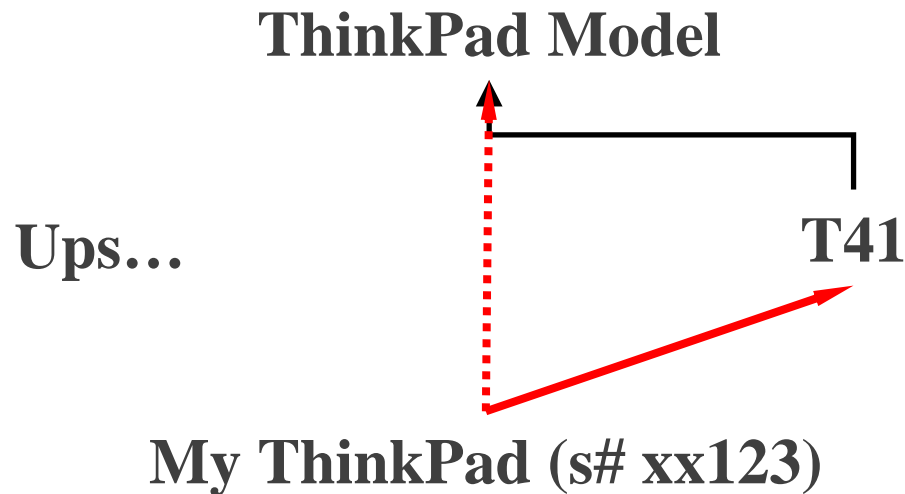
# Probleme

- Ergebnisse erscheinen oft trivial
- Metaeigenschaften sind hart zu verstehen
  - Teures Tagging
  - AEON – Automatisches Tagging aufgrund von Sprachmustern
- Wenig Beispiele vorhanden

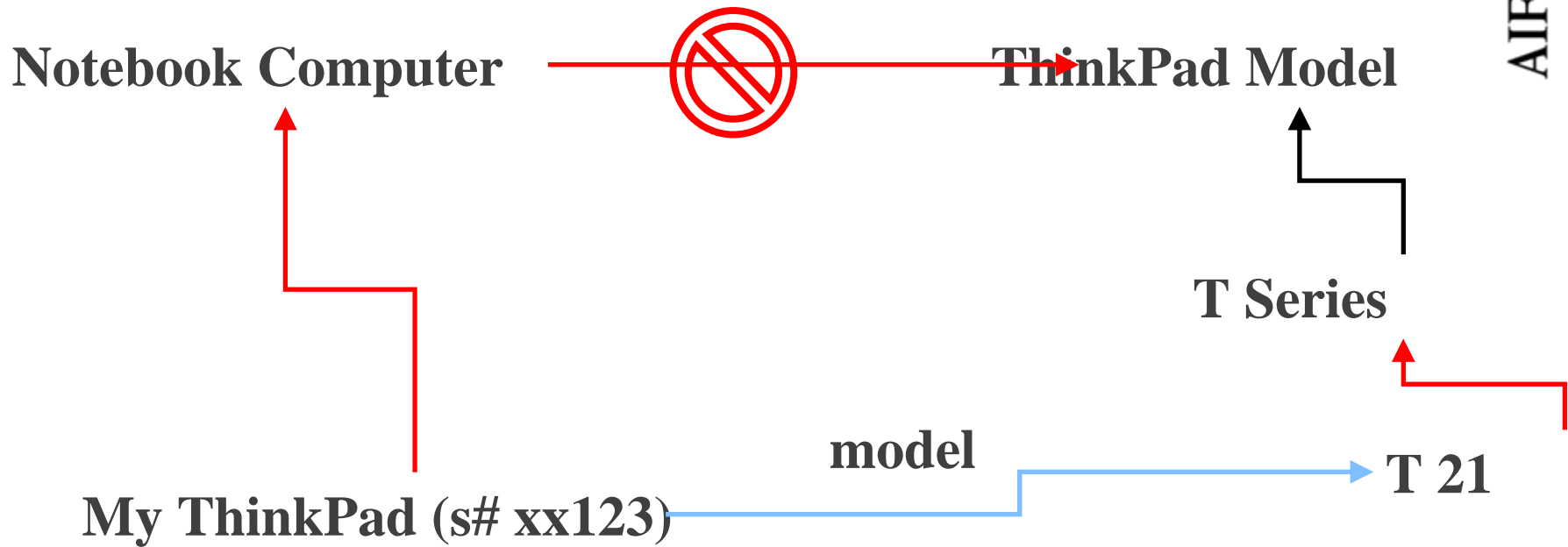
# Häufige Fehler

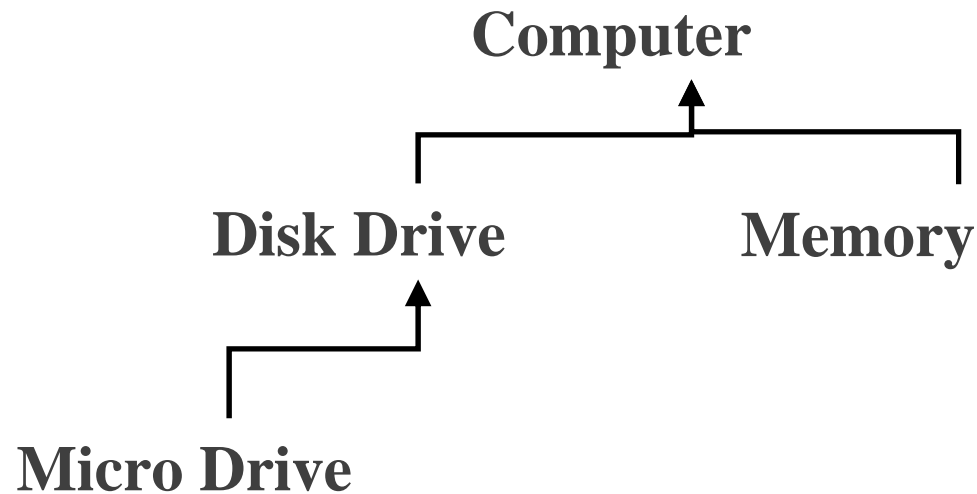
# Instanziierung

Bedeutet diese Ontologie, dass mein ThinkPad ein ThinkPad Modell ist?



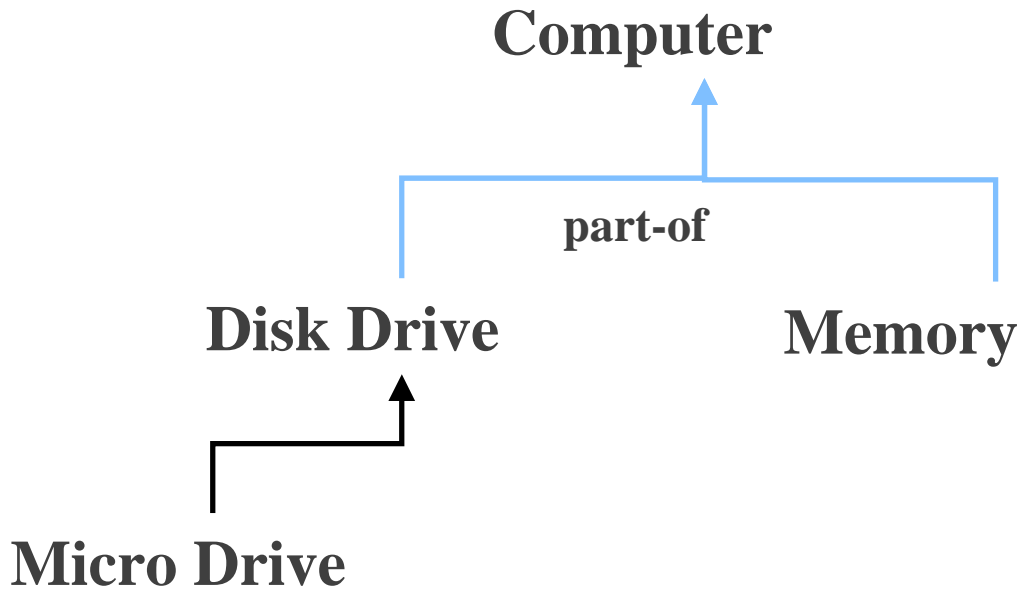
**Frage: Welche ThinkPad Modelle verkaufen wir?**  
**Die Antwort sollte nicht mein ThinkPad einschließen!**





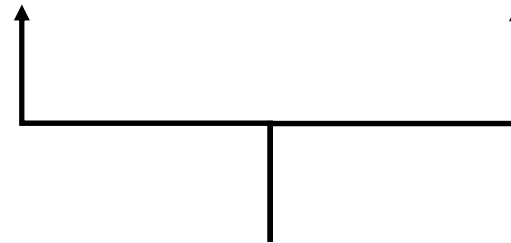
**Frage: Welche Computern werden verkauft?**  
**Die Antwort sollte nicht die Laufwerke beinhalten!**

# Komposition



# Polysemie

Physical Object    Abstract Entity



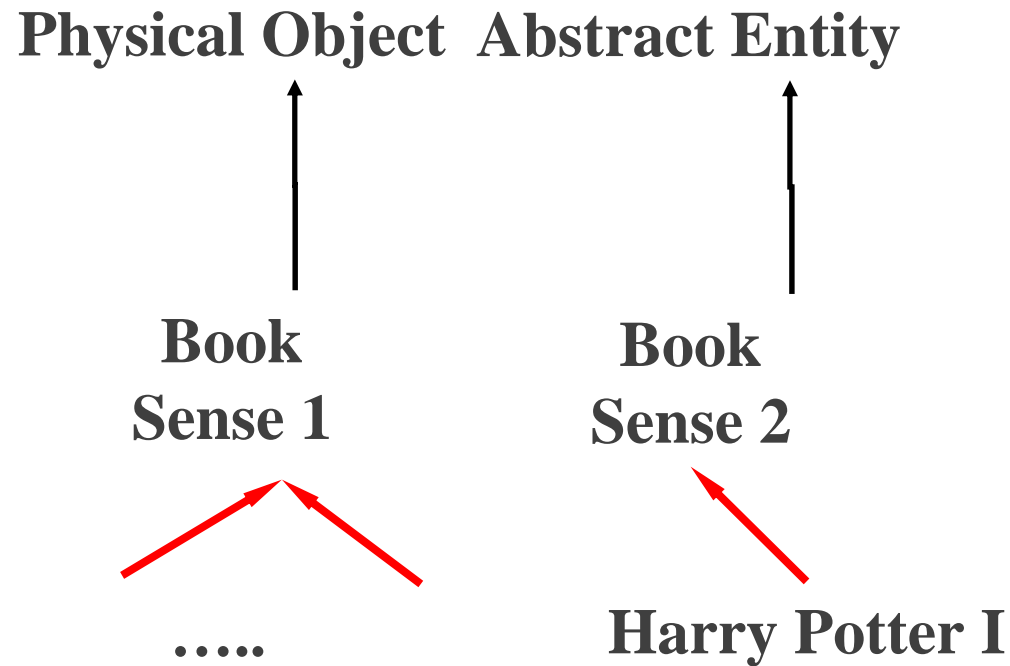
Book



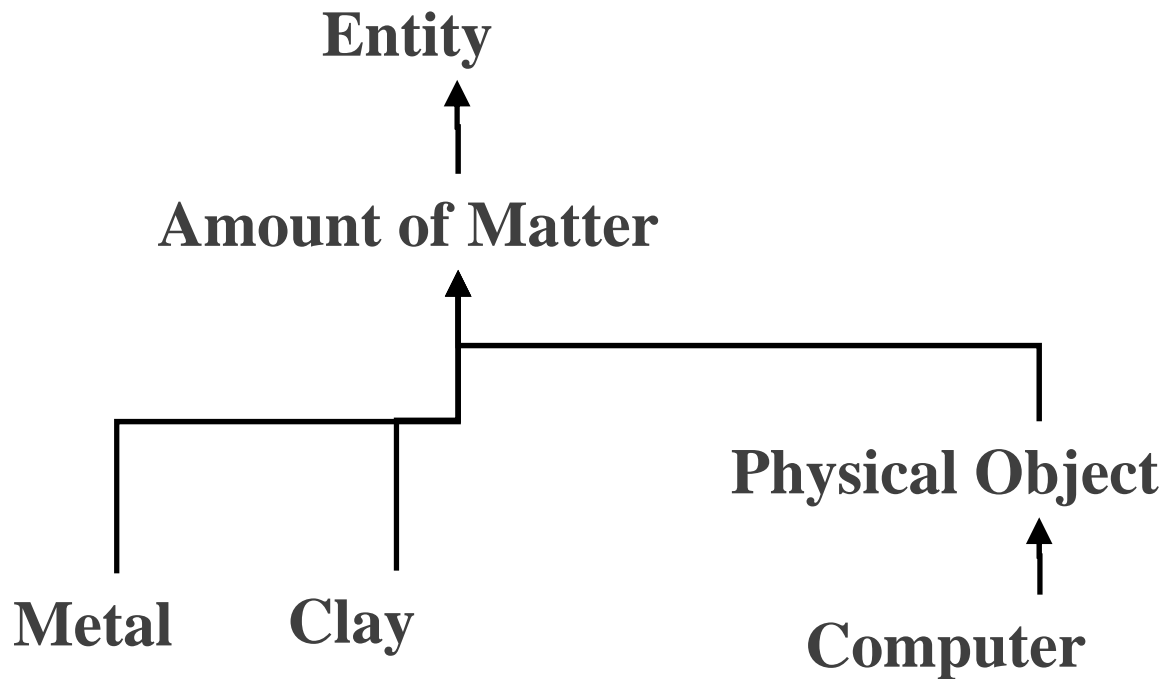
.....

**Frage: Wieviele Bände von Harry Potter verkauft Amazon?**  
**Antwort: Einige Millionen**

# Polysemie

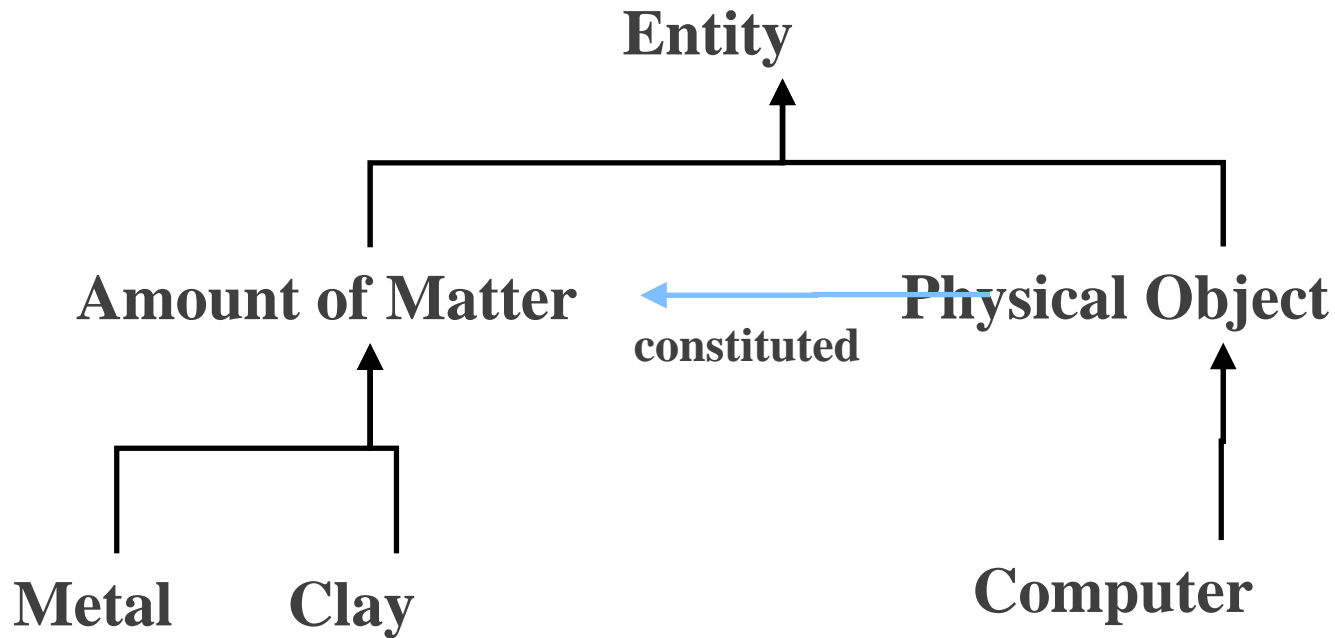




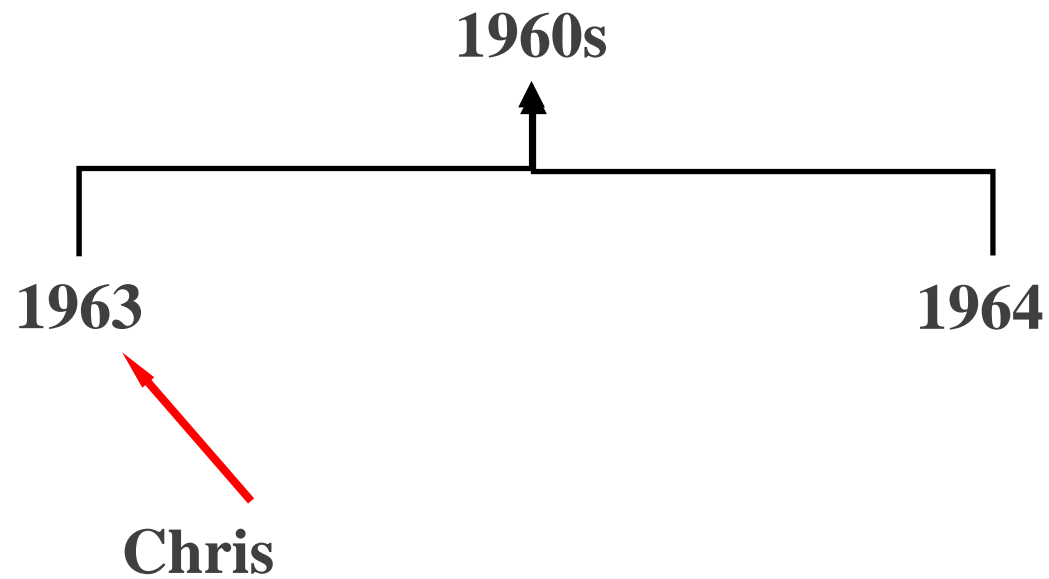


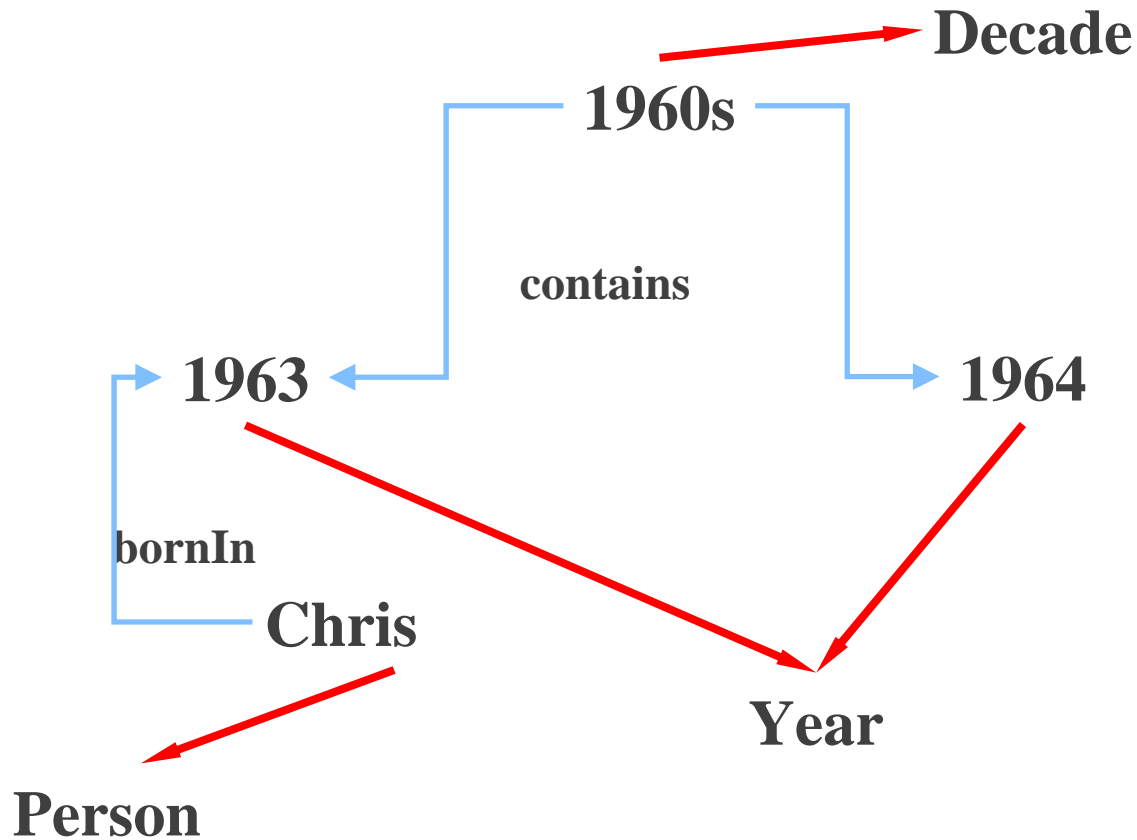
**Frage: Welche Materialien leiten Strom?**  
**Antwort sollte nicht Computer sein!**

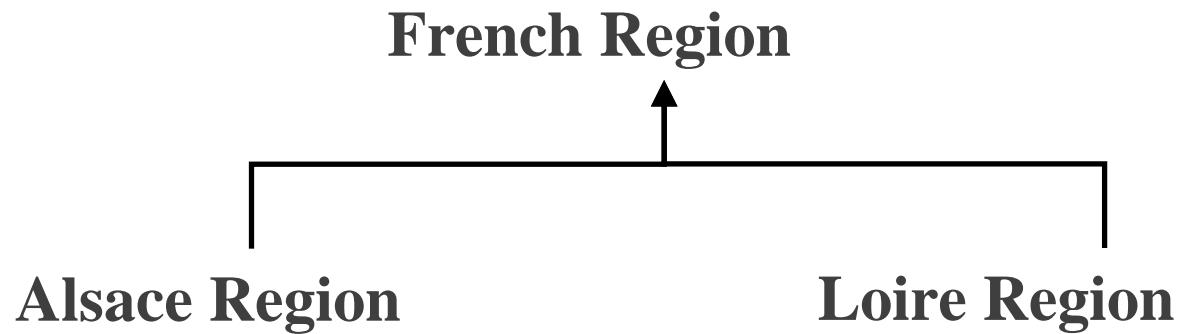
# Konstitution

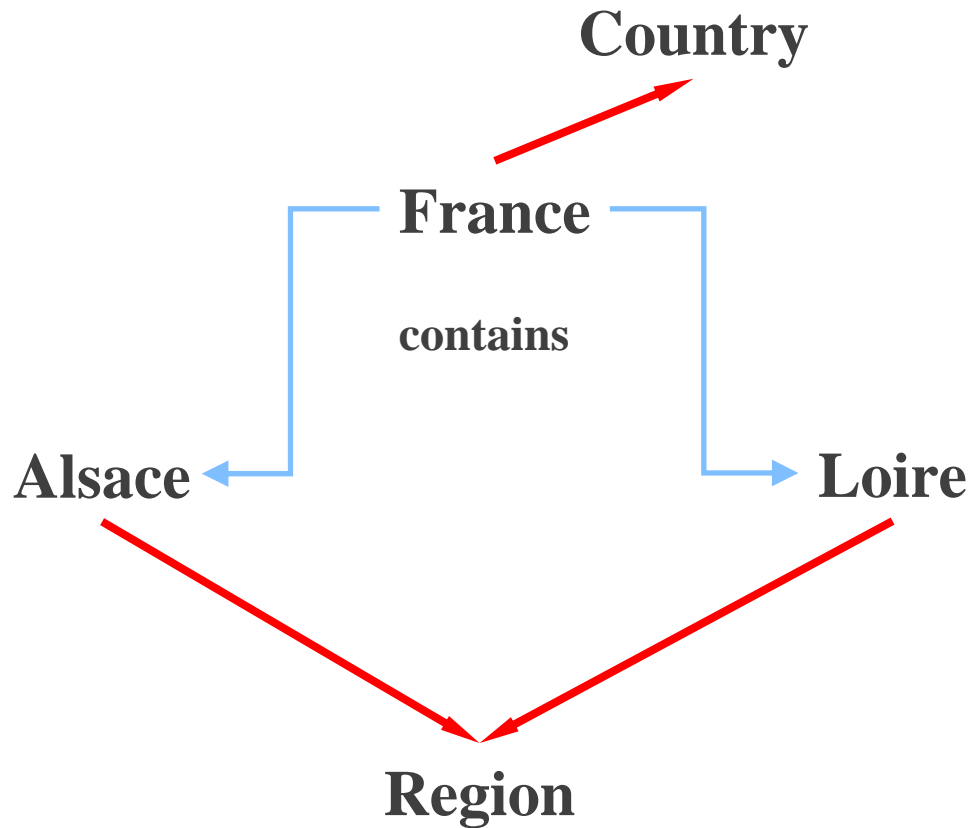


# Zeit (Wikipedia)









# Bewerten von Ontologien

# Open Rating Systeme

**Object** → [Book cover image]

**Rating** → [Average Customer Review: ★★★★★]

**User** → [Customer Review text]

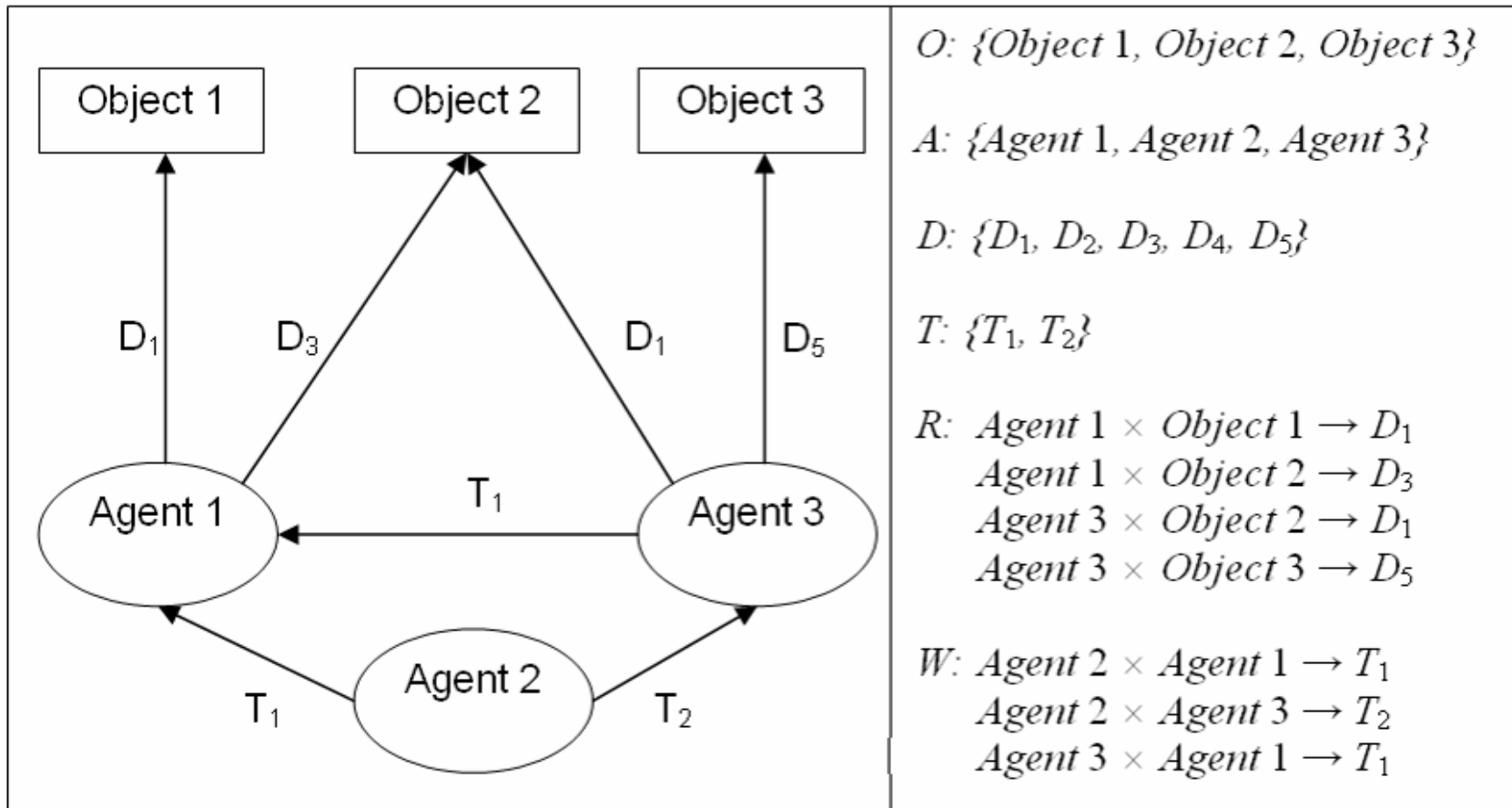
**Meta-Rating (trusted by other users?)** → [36 of 38 people found the following review helpful:]

**Review** → [A must for the beginner or the advance phenomenologist...]

Annotations include: 'SEARCH INSIDE!', 'Write an online review', '36 of 38 people found the following review helpful:', '★★★★★', 'A must for the beginner or the advance phenomenologist.', 'Monadsense "monadsense"', and 'Was this review helpful to you? Yes No Report this'.



# Traditionelle Open Rating Systeme



Guha, R.: “Open Rating Systems”. Technical report, Stanford University, CA. 2003

# Topic-Specific Trust ORS

- Ermöglicht das Bewerten einzelner Eigenschaften, nicht nur des ganzen Objekts
- Ermöglicht Trust bezüglich bestimmter Eigenschaften bzw. Themen
- Trust wird entsprechend feingranular propagiert

Lewen, H., Supekar, K., Noy, N., Musen, M.: „**Topic-Specific Trust and Open Rating Systems: An Approach for Ontology Evaluation**“ In *Proceedings of the 4th International Workshop on Evaluation of Ontologies for the Web (EON2006) at the 15th International World Wide Web Conference (WWW 2006)*. Edinburgh, UK, May 2006.

Sabou, M., Angeletou, S., d’Aquin, M., Barrasa, J., Dellschaft, K., Gangemi, A., Lehmann, J., Lewen, H., Maynard, D., Mladenic, D., Nissim, M., Peters, W., Presutti, V., Villazon, B.: „**D2.2.1 Methods for Selection and Integration of Reusable Components from Formal or Informal User Specifications**“ NeOn Project Deliverable D2.2.1, The Open University. May 2007.

# Kollaboratives Bewerten von Ontologien

- Jeder Benutzer entscheidet, was und wie viel er bewerten will
  - Nur eine bestimmte Eigenschaft
  - Mehrere Eigenschaften
- Ein paar schlechte Reviews stören nicht, weil sie durch Meta-Reviews rausgefiltert werden
  - Nur die Top-N bewerteten Reviews werden berücksichtigt
- Personalisierung durch Trust und individuelles Gewichten der Eigenschaften

Ontology Navigator

- NewOntologyProject
  - >ontology12
    - Concepts
      - Animal
        - Amphibian
        - Bird
        - Fish
          - FreshwaterFish
          - MarineFish
        - Invertebrate
          - Arthropod
          - Insect
          - Mollusk
          - Tarantula
          - Worm
        - Mammal
          - Dog
          - cat
          - human
        - Reptile
  - Attributes
  - Relations
  - .ontologies
  - .project

Entity Properties View

Name:

Namespace:

Attributes:

Attribute	Range	Min	Max

Properties

son **range**

attends **range**

**comment** "Modern man, the only remaining species of the Homo genus."

<http://kmi-web05.open.ac.uk:81/cache/1/f5d/857a/f0cae/8b96c6b3ac/1e19b0c5cf7f2a849#Human> ★★★★★

<http://kmi-web05.open.ac.uk:81/cache/0/339/c2ff/21d76/1013cd189c/557c6d296bdc6957c#Human> ★★★★★

<http://onto.cs.yale.edu:8080/umls/UMLSinDAML/NET/SRDEF#Human> ★★★★★

**subClassOf** Mammal

<http://kmi-web05.open.ac.uk:81/cache/9/2b9/d184/eecdc/efe52ea651/d0b87f986918a81c4#Human> ★★★★★

<http://kmi-web05.open.ac.uk:81/cache/8/6e8/368c/58d32/e93217cae8/23bc158d81684b28c#Human> ★★★★★

<http://edge.cs.drexel.edu/assemblies/ontologies/woolly/2004/06/flows.owl#Human> ★★★★★

**subClassOf** Material

**disjointWith** Liquid

# Zusammenfassung

- Reparieren von Ontologien
- OntoClean
- Häufige Fehler
- Bewerten von Ontologien

# Ausblick

- Montag keine Übung
- Mittwoch Vorlesung von Peter Haase zu Informationsintegration und Semantischer Suche

*Folien mit Dank an Holger Lewen, Peter Haase, Guilin Qi, Aldo Gangemi, Chris Welty, Aditya Kalyanpur*