© CL 2001 Darmstadt

Dr. Christoph Lübbert
Viktoriastraße 36
D-64293 Darmstadt

Tel: 06151 422298, T-Mob: 0171 2045811,
christoph.luebbert@t-online.de ,
www.cl-diesunddas.de

Darmstadt, 23.06.2014,V4f

Änderungen gegenüber V2, V3: Kürzungen für den Vortrag

„Verfeinerung" der Ontology-Sprache durch die FBA-Sprache

Danksagung:

Peter Zahn, *Thomas Zeh*, *Karl Erich Wolff* und *Hermann Bense* danke ich herzlich für Kommentare, Hinweise und Korrekturen bei der Ausarbeitung.

Inhalt

1 Vorbemerkung

Als ich vor etwa drei Jahren zu diesem erlauchten **„Ontologie-Arbeitskreis“** stieß, glaubte ich, man werde sich in diesem Kreis u.a. ausführlich mit den derzeit kursierenden **Strukturansätzen zum Konzept „informatische Ontology“** beschäftigen; und da der Ontologie-Arbeitskreis der **h_da** glücklicherweise eng mit dem **FBA-Seminar** (Formale Begriffsanalyse) der **TUD** zusammenarbeitet, hoffte ich, Informatiker und Mathematiker würden sich im Lauf der Zeit in diesem Kreis auch fachsprachlich ein bisschen näher kommen.

Davon habe ich in den letzten 2-3 Jahren aber recht wenig verspürt: Eine Annäherung zwischen „Ontology-Sprache“ und „FBA-Sprache“ hat nach meinem Empfinden bei „uns“ noch nicht so recht stattgefunden. Ich empfand viele Diskussionen so, als „redete man in Randbezirken um den heißen Brei herum“!

Ich versuche in dieser Notiz ***zu vermitteln*** zwischen der ***Informatiker-Sprache*** (zur Beschreibung eines Konzepts **Ontology**) und der ***Mathematiker-Sprache*** (insbes. in der **Formalen Begriffsanalyse** – **FBA**). Die im Titel angegebene „Verfeinerung“ besteht darin, dass man bei Anwendung der FBA-Sprache **Strukturen** im Konzept „Ontology“ erkennt, über welche die mit „Ontology“ beschäftigten Informatiker sich m.E. bislang wohl noch nicht ganz einig sind.

In meiner Notiz unterscheide ich zwischen der **Struktur** eines Konzepts „Ontology“ und ihren vielfachen **Anwendungsmöglichkeiten** („Deutungen“). In Dieser Notiz geht es mir hauptsächlich um die ***Struktur***. Deutungen ziehe ich nur in einfachen Beispielen als Anschauungsmaterial heran.

Das scheint für manche Ontologen erstaunlicherweise ein ungewohntes Vorgehen zu sein: *H. Herre* / Leipzig, [19], stellt in seiner *„General Formal Ontology“* die mathematischen, also ***strukturbestimmenden*** Hilfsmittel, auf gleiche Stufe mit ihren vielen Deutungsmöglichkeiten. Z.B. stellt er die **Mengennotation**, zusammen mit seinen ominösen „Urelementen", auf die gleiche Stufe „*Entity*", obwohl mit der Mengennotation auf sog. „*Entities*" ja höchstens *in einer Deutung referiert* werden kann, wogegen „Urelemente" etwas aus einer gewissen außermathematischen *Deutung* der Mengenlehre sind, die man in den Anfängen der Mengenlehre zu benötigen glaubte, die aber heute als obsolet und hinderlich für die Mengennotation gilt, weil sie den ***sehr nützlichen*** Gebrauch des Terminus „leere Menge" behindert. Das ergibt für mein Empfinden in *Herre's* System ein ***unerträgliches Durcheinander!***

Dies sage ich auch als **WARNUNG** für die Nicht-Mathematiker und die Nicht-Informatiker in unserem erlauchten Kreis: Es geht mir heute ***<u>nicht</u>*** um *schon wieder und noch einmal eine weitere* „philosophische Betrachtung" zu dem, was man „Ontologie" oder eines ihrer Teilaspekte nennen könnte, sondern es geht mir um einen **Vermittlungsversuch zwischen zwei Fachsprachen,** welche beide die Beschreibung sehr ähnlicher Informationsstrukturen zum Ziel haben.

1.1 Was ist für mich der Zweck eines Konzepts „Informatic Ontology“?

Sie implementiert ein **Wissensgebiet** (*knowledge domain*), so dass sie dem menschlichen User oder auch einem IT-System (das dazu in der Lage ist) etwas intelligentere Fragen als nur in Form von Stichworten à la Google verständlich und konsistent beantworten kann.

Eine **Ontology** besteht nach meiner Anschauung aus zwei Teilen:

den „<u>**Systeminterna**</u>“ und

der „<u>**Userschnittstelle**</u>“.

Auf die „Userschnittstelle“ (ich halte sie für den ***anspruchsvolleren*** Teil) will ich hier ***nicht*** eingehen; ihr Kern kann mit den Hilfsmitteln der **FBA** (Formale Begriffs-analyse) aber leicht dargestellt werden durch zwei „**Formale Kontexte**“: einen für die umgangssprachlichen Ausdrücke der „Objekte“ des Wissensgebiets, einen für die umgangssprachlichen Ausdrücke der „Beziehungen“ (zwischen Objekten) im Wissensgebiet; vgl. auch den etwas unvollständigen Ansatz von *Pickert* [2].
Die ***Systeminterna*** – auf die beschränke ich mich hier – bestehen aus zwei Ebenen: (A) dem **Ontology-Schema** (O-Schema) und (B) der **Instanziierung**.

Zunächst (Kap.2 - Kap.3) gehe ich auf die mir bekannte ***Informatiker-Sprache*** für die *Systeminterna* einer „**Ontology**“ ein. Zur metasprachlichen Ontology-Beschreibung benutze ich, falls angebracht, die sehr hilfreiche Notation von ***Hermann Bense***:

Präfix „**^**“ für den Namen einer „**Klasse**“,

Präfix „**.**“ für den Namen eines „**Attributs**“

Präfix „<>“ für den Namen eines „**Relationship-Type**“

Präfix „>“ für den Namen einer „**Instanz**“ oder eines „**Attribut*wertes***“.

Danach (ab Kap.4) führe ich die Grundtermini der **FBA** (**Formale Begriffsanalyse**) wiederholt bei einigen Anwendungsmöglichkeiten der FBA auf die Ontology-Sprache ein und zeige, dass die FBA-Sprache die diversen Gedanken der Ontology-Informatiker präzisieren kann. Damit aber sind die Anwendungsmöglichkeiten von FBA im Konzept „Ontology“ noch längst nicht ausgeschöpft.

1.2 Anschauungsbeispiel aus Wikipedia

Zur Einstimmung die **Abb.1** aus Wikipedia unter dem Strichwort „Ontologie (Informatik)“ [10].

Die obere Ebene: **„O-Schema“** (in Abb.1 als „Ontologie“ bezeichnet).
Die untere Ebene: „**Instanziierung**“ (in Abb.1 als „Instanzen“ bezeichnet).

Im O-**Schema** der Abb.1 sehen wir:

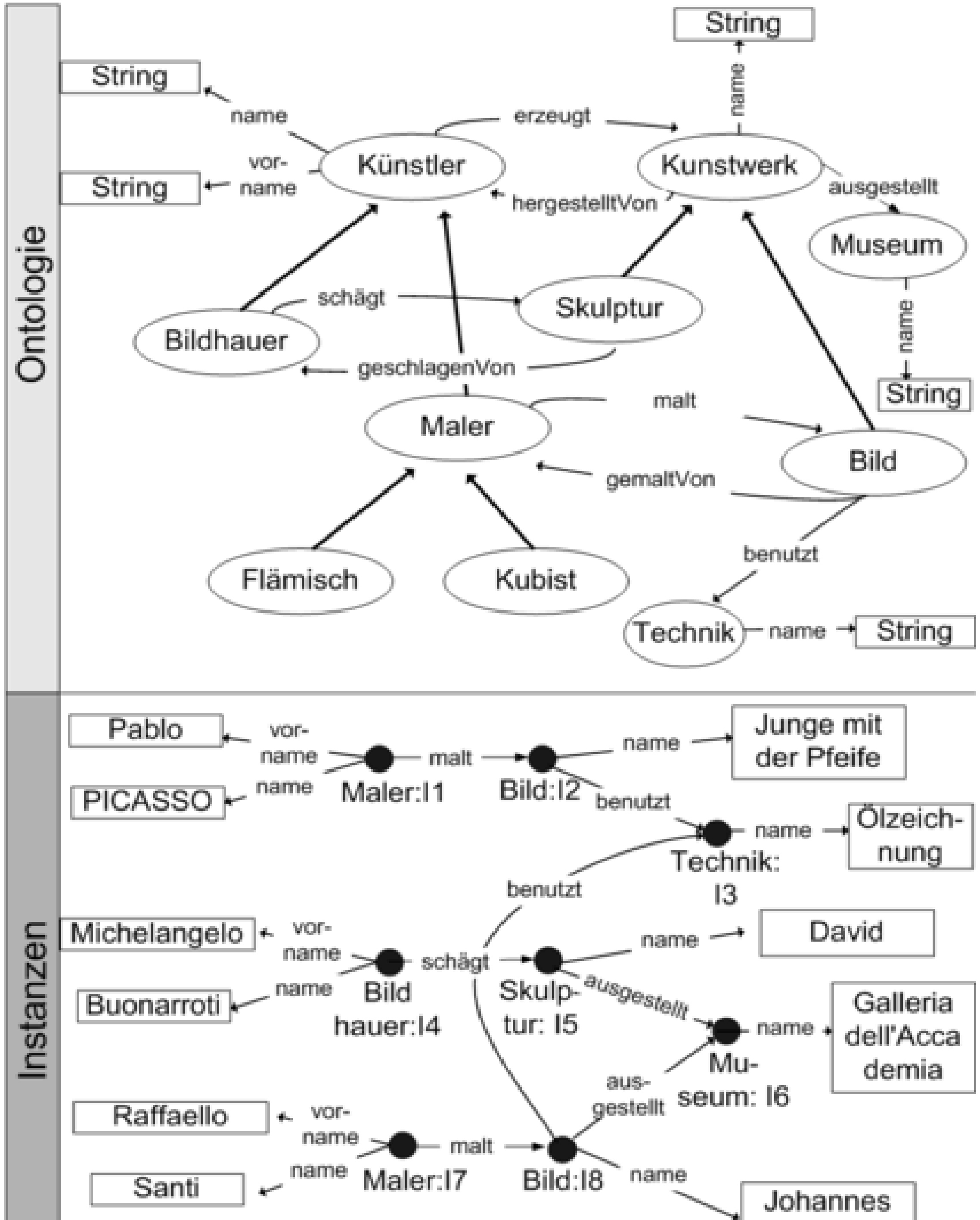
„**Klassen**“ als *Ovale* gezeichnet; „**Attribute**“, als *rechteckige Kästchen* gezeichnet, in denen das Wort „String“ steht. Klassen-Ovale sind mit Attribut-Kästchen durch dünnere gerade und u.a. mit „name“, „vorname“ beschriftete Linien verbunden.

Das soll heißen, dass den Klassen gewisse Attribute zugeordnet sind. „String“ bedeutet, dass bei *Instanziierung* in das betreffende Leerfeld ein Attribut***wert*** (Merkmal einer gewissen Instanz) im Format „String“ einzutragen ist (es könnte bei Bedarf auch ein numerisches Format sein).

Einige Klassen sind verbunden durch dicke gerade Pfeile **von unten nach oben**. Zum Beispiel ist die Klasse ^Maler pfeilverbunden mit der Klasse ^Künstler. Das soll heißen, dass z.B. ^Maler eine *Unterklasse* von ^Künstler sein soll. Zugleich soll es heißen, dass die Oberklasse ihre **Attribute** an die Unterklassen **vererbt** (also von oben nach unten, *entgegen* der Pfeilrichtung). Damit wird auch die **O-Schema-Konvention** klar, dass das Attribut-Kästchen einer Oberklasse nicht noch einmal bei ihren Unterklassen eingetragen werden muss, weil es ja nach unten vererbt wird. Die dicken Pfeile von unten nach oben visualisieren die sog. „**Klassentaxonomie**“, auch „**Klassenhierarchie**“ genannt. Mathematisch wird diese „dicke-Pfeile-Struktur“ schlicht eine „**Ordnung**“ genannt.

Schließlich sind einige Klassen paarweise durch geschwungene Pfeile verbunden. Zum Beispiel ist ^Künstler verbunden mit ^Kunstwerk, und der Pfeil ist beschriftet mit <>erzeugt. „Invers“ dazu ist auch ein Pfeil <>hergestelltVon zu sehen. Das soll einfach den allgemeinen Satz „Künstler erzeugen Kunstwerke“ (bzw. invers dazu: „Kunstwerke werden von Künstlern hergestellt“) wiedergeben. Die Elemente „<>erzeugt“, „<>hergestelltVon“ nenne ich (binäre) **„Relationschip-Types“** (im Wikipedia-Text wird dafür das Wort „Relation“ verwendet, dieses Wort vermeide ich zunächst.)

Abb.1 (aus Wikipedia) ➔



Kritische Anmerkung zur O-Schema-Darstellung in Abb.1:

Die Abbildung aus Wikipedia ist meinem Empfinden nach ***mit Vorsicht zu genießen***.

a) Attributevererbung und Identifizierbarkeit bei Klassen: Die Konvention ist, dass ein von oben nach unten vererbtes Attribut nicht noch einmal bei der Unterklasse als Attribut-Kästchen angehängt werden muss. Z.B. **erbt** die Unterklasse ^Maler von ihrer Oberklasse ^Künstler die Attribute „**.**name:….“ und „**.**vorname: …“; dasselbe erfolgt bei der Unterklasse ^Bildhauer. Diese Konvention wird in Abb.1 ***richtig*** wiedergegeben.
Aber: ^Maler hat, außer den geerbten, keine weiteren Attribute. In was unterscheidet sich die Klasse ^Maler von der Klasse ^Künstler denn nun systemintern? – Nur in der Beschriftung der Ovale? – Ist diese Beschriftung Teil der Systeminterna des O-Schemas?? Das bleibt im Wikipediabeispiel unklar! Ich meine: An das Oval ^Maler müsste mindestens ein weiteres **Attribut**-Kästchen (etwa mit dem Attributnamen „**.**Malereigenschaft:….“ oder „**.**MalerInfo:….“) angehängt werden, das von den ans Oval ^Künstler angehängten ***verschieden*** sein sollte. Andernfalls hätte ^Maler ***genau dieselben*** Attribute wie ^Künstler, und man sähe nicht den ***Grund*** ein, warum in diesem O-Schema das ^Maler-Oval formal überhaupt vom ^Künstler-Oval unterschieden werden sollte! Denn die Worte „Künstler“ *im Oval* für die Klasse ^Künstler und „Maler“ *im Oval* für die Klasse ^Maler sind ja nur Beschriftungen zur Lesbarkeit der graphischen Darstellung des O-Schemas. Soll damit aber eine für die Klasse *spezifische* ***Information*** gemeint sein, so ist ein dazu geeignetes **Attribut**-Kästchen anzulegen; ***das ist ja der Sinn und Zweck von Attributen!***

b) In Abb.1 bleibt auch die Struktur **Relationship-Types** und deren **Vererbung** unklar. Die Gesamtheit der in Abb.1 benutzten Relationship-Types ist die Menge {<>erzeugt, <>hergestelltVon, <>ausgestellt, <>schlägt, <>geschlagenVon, <>malt, <>gemaltVon, <>benutzt}.

Aber: Die Relationship-Types <>malt, <>gemaltVon, <>schlägt, <>geschlagenVon haben einen anderen Charakter als die Relationship-Types <>erzeugt, <>hergestelltVon, <>ausgestellt und <>benutzt. Diesen Unterschied erläutern wir wieder an einem Beispiel: <>malt ist hier offensichtlich als eine *Spezialisierung*, als ein „***Unter-****Relationship-Type“* von <>erzeugt gemeint. In der Interpretation der gemeinten Objekte heißt das: „*Jeder Maler ist ein Künstler, jedes Bild ist ein Kunstwerk*“. Die instanziierende Aussage *„Der Maler1 malt das Bild1“* ist eine Spezialisierung der Aussage *„Der Künstler1 erzeugt das Kunstwerk1“*, nämlich: *„Der Künstler1 (welcher der Maler1 ist) erzeugt (also malt) das Kunstwerk1 (welches das Bild1 ist)“*.

Andererseits ist „<>erzeugt“ in der Abb.1 kein Unter-Relationship-Type eines allgemeineren Relationship-Types. <>erzeugt ist vielmehr ein, wie ich vorläufig sagen will, „***maximaler***“ Relationship-Type, von dem die Unter-Relationship-Types <>malt und <>schlägt – und einige andere in Abb.1 nicht genannte – durch **Vererbung** *abgeleitet* sind. Würde man die ***Vererbung*** entlang der ***Klassentaxonomie*** ernst nehmen und entsprechend der Konvention verfahren, dass vererbte Elemente *nicht* noch einmal bei der Unterklasse eingezeichnet werden müssen, so wäre die Erwähnung der abgeleiteten Unter-Relationship-Types <>malt, <>schlägt im O-Schema der Abb.1 ***überflüssig***. Wollte man andererseits ***alle*** möglichen Unter-Relationship-Types von <>erzeugt im O-Schema vermerken, so müsste man dies in Abb.1 bei ***allen*** zusammengehörigen Klassenpaaren unterhalb ^Künstler und ^Kunstwerk tun und zum Beispiel auch einen Unter-Relationship-Type zwischen ^Kubist und ^Bild – in das O-Schema mit aufnehmen. So aber, wie es in Abb.1 dargestellt ist, erscheint das System der Relationship-Types ziemlich ***willkürlich und unsystematisch.***

Die Instanziierungsebene in Abb.1:

Die Beispiel-Eintragungen in der Instanziierungsebene (untere Ebene der Ontology) verstehen sich von selbst; ich gehe hier nicht besonders darauf ein.

Nach dieser Einführung am Beispiel der Abb.1 sind m.E. im Folgenden nun doch ein paar strukturerläuternde und präzisierende Worte zu sagen. Ich gehe aus von den Termini „**Klasse**“, „**Klassentaxonomie**“, „**Attribut**“, „**Relationship-Type**“ und „**Vererbung**“. Für die *Instanziierung* kommen noch die Termini „**Instanz**“ und „**Attribut*wert***“ hinzu.

Diese Termini seien im Folgenden etwas genauer untersucht, ohne dass ich mich in den *Jungle* der in der Informatik üblichen vielen Abkürzungen begeben will.

2 O-Schema

In einem O-Schema, wie ich es kenne, hat man:

(1) Ein geordnetes System ($\mathbb{C}$, $\leq_C$) von "**Klassen**". (Manchmal sagt man statt „Klasse“ abstrakter auch „Begriff“ (engl.: *„concept“*); ich verwende hier das Wort „***Klasse***“, um nicht mit dem später einzuführenden FBA-Terminus *„formaler Begriff“* durcheinander zu kommen).

Die Menge $\mathbb{C}$ ist die Gesamtheit der **Klassen**, die im O-Schema auftreten. Sie ist geordnet durch eine sog. **Klassen-Taxonomie** (Klassenhierarchie); Mathematiker sagen einfach „**Ordnung**“ dazu**.**

DEF.: Eine binäre Beziehung $\leq$ auf einer Menge M heißt eine (nicht-strikte) **Ordnung** genau dann, wenn sie die folgenden drei Eigenschaften hat:

Reflexivität:	$x \leq x$	für alle $x \in M$
Antisymmetrie:	wenn $x \leq y$ und $y \leq x$, dann $x=y$	für alle $x, y \in M$
Transitivität:	wenn $x \leq y$ und $y \leq z$, dann $x \leq z$	für alle $x, y, z \in M$.

Die zugehörige strikte Form „<“, definiert als $x<y$:gdw. $x \leq y$ und $x \neq y$, ist irreflexiv und transitiv; daraus folgt, dass sie auch azyklisch ist. Je nach Bedarf verwende ich für „Ordnung“ einmal die nicht-strikte Form „≤“, ein andermal die strikte Form „<“.

Gilt c1 $<_C$ c2 für zwei Klassen (eines O-Schemas), so heißt c1 eine (echte) **Unterklasse** von c2, bzw. c2 eine (echte) **Oberklasse** von c1.

Die Klassentaxonomie $<_C$ ist in Abb.1 ***monohierarchisch*** *(d.h. jede Klasse hat* *höchstens **einen** oberen Nachbarn).* Allgemein darf eine Klassentaxonomie aber auch ***polyhierarchisch*** sein, d.h. eine Unterklasse darf mehrere obere Nachbarn haben. Das geordnete System ($\mathbb{C}$, $<_C$) der Klassen ist sozusagen das ***Grundgerüst*** im O-Schema: An ihm wird alles andere angehängt und entlang der Klassentaxonomie nach unten vererbt.

(2) Ein System $\mathfrak{A}$ von "**Attributen**":

Jede Klasse c $\in \mathbb{C}$ ist zu beschreiben durch eine Menge **att**(c) von **Attributen,** att(c) $\subseteq \mathfrak{A}$**.** Der Operator **att**: $\mathbb{C} \rightarrow$Pot$\mathfrak{A}$ ordnet jeder Klasse c eindeutig ihre Attributemenge att(c) zu. Das Attribute***mengen***system [$\mathfrak{A}$] := {att(c) $\subseteq \mathfrak{A}$ | c$\in \mathbb{C}$} wird später eine zum Klassensystem $\mathbb{C}$ korrespondierende Rolle spielen.
Ein "Attribut" kann man sich vorstellen als eine Art "Formular".
Beispiele: **.**Familienname: .... ; **.**Vorname: ....; **.**geboren am: ....; **.**Geburtsort: .... sind z.B. Attribute einer Klasse ^Person. Bei Instanziierung wird in das Leerfeld „**:…..**“ ein gewisser "Wert" eingetragen. Man muss jedenfalls zwischen „Attribut“ und „Attribut*wert*“ unterscheiden! Hier geht es zunächst nur um die *Attribute selbst*; *nicht* um ihre Werte.

(3) **Vererbung**: Attribute **vererben** sich von oben nach unten entlang der Klassentaxonomie $<_C$, d.h.: ist c1 $<_C$ c2 und hat die Klasse c2 das Attribut a2 $\in \mathbb{A}$, so hat auch c1 das Attribut a2. Die Unterklasse ist „***spezieller***“ als die Oberklasse; c1 darf daher daneben auch noch ein Attribut a1 haben, das die Oberklasse c2 nicht hat. Die Attributemenge att(c) einer Klasse c kann man daher aufteilen in

(2.0) att(c) = att$_{\textbf{vererbt}}$ (c) $\cup$ att $_{\textbf{neu bei c}}$(c).

Im polyhierarchischen Fall dürfen zwei unvergleichbare Klassen auch eine gemeinsame Unterklasse c haben. Dann erbt c die Attributemengen ***beider*** Oberklassen. In diesem Fall muss c kein weiteres eigenes Attribut aufweisen, d.h. att $_{\text{neu bei c}}$(c) darf $=\emptyset$ sein, denn c hat ja auch in diesem Fall ***mehr*** Attribute als jede einzelne ihrer Oberklassen. Wenn aber c genau ***einen*** oberen Nachbarn hat, sollte att $_{\text{neu bei c}}$(c) $\neq\emptyset$ sein; andernfalls wäre c *nicht spezieller* als ihr oberer Nachbar und somit von dem oberen Nachbarn formal ***nicht unterscheidbar***! Es gibt m.E. ***keinen Grund***, im O-Schema zwei Klassen c, d zu unterscheiden, wenn ihre Attributmengen att(c), att(d) gleich sind. ***Das ist ja der Sinn und Zweck von Attributen.*** Wir können also formulieren:

(2.1) att(c) = att(d) genau dann, wenn c=d.

(2.2) c1 $<_C$ c2 genau dann, wenn att(c2) $\subset$ att(c1) att(c2) echte Teilm.von att(c1).

Anmerkung: Hier gibt es *Widerspruch* von Seiten einiger Informatiker. Sie meinen, die Regeln (2.1), (2.2) schränken die Freiheit des Ontology-Designers zu sehr ein. Sie wollen zulassen, dass eine ***echte*** Unterklasse c1 von c2 ***dieselbe*** Attributemenge wie die Oberklasse c2 haben darf: att(c1) = att(c2).
Da frage ich mich aber: durch welche systeminternen Parameter unterscheiden sich dann die Klassen im O-Schema, und warum sind diese verdeckt??

(4) **"Relationship-Types"**: Das O-Schema hat meist auch ein System **RELT** von binären **"Relationship-Types"** auf dem System $\mathbb{C}$ der Klassen.
– Ich wähle hier extra das Wort „Relationship-Type“ und ***nicht*** das Wort "Relation" (wie in Wikipedia), weil ich letzteren Terminus für ***„Relation im mathematischen Sinne“*** reservieren will. Hierzu erst Genaueres, wenn wir uns mit der Instanziierung beschäftigen.
– Außerdem beschränke ich mich hier der Einfachheit halber auf nur *<u>binäre</u>* Relationship-Types, – so verfahren auch *Meadche et.al.*[12] und Pickert [2], – obwohl man Ähnliches auch für mehr als 2-stellige Relationship-Types machen kann.
Zu jedem binären Relationship-Type **r** gehören zwei Klassen c, d und damit eine **relationale Teilkonfiguration c r d** des O-Schemas. Wir haben also einen Operator relt: **RELT** $\rightarrow \mathbb{C} \times \mathbb{C}$. Dabei kommt es auf die Reihenfolge an: c r d ist nicht dasselbe wie d r c! In Abb.1 ist zum Beispiel „^Maler <>malt ^Bild“ eine solche O-Schema-Teilkonfiguration, aber ^Bild <>gemaltVon ^Maler ist davon

zu unterscheiden: es ist die „inverse“ Teilkonfiguration. Es darf zwischen *dem-selben* Klassenpaar auch zwei verschiedene Relationship-Types geben: relt(r) = relt(s) = (c, d). Es darf aber auch eine Klasse *mit sich selbst* kombiniert werden: relt(t) = (c, c); Beispiel:

^Maler <>ist_befreundet_mit ^Maler. (In Abb.1 nicht vorgesehen.)

Auch Relationship-Types können **vererbt** werden: ist z.B. c1 $<_C$ c und d1 $<_C$ d, und besteht zwischen c, d der Relationship-Type r, also die Teilkonfiguration **c r d**, so ergibt sich daraus zwischen c1, d1 ein sog. „**Unter-Relationship-Type“** r1, der von r ***abgeleitet*** ist durch **Einschränkung** auf die Unterklassen. (Genaueres erst in **Kap.3.4**.)

Wir haben dann die relationale Teilkonfiguration **c1 r1 d1**. Man sollte r1 aber ***nicht*** mit demselben Namen wie r bezeichnen, denn r1 ist ja eine ***Spezialisie-rung*** von r.

Beispiel (vgl. Abb.1): ^Künstler <>erzeugt ^Kunstwerk; ^Maler <>malt ^Bild: Hier ist <>malt ein echter Unter-Relationship-Type des Relationship-Types <>erzeugt und nicht mit ihm zu identifi-zieren, weil er nur zwischen Malern und deren Bildern gemeint ist, wogegen <>erzeugt allgemeiner zwischen Künstlern und deren Kunstwerken gemeint ist!

Ist r1 ein Unter-Relationship-Type von r, so schreiben wir: r1 $<_{RELT}$ r.

Merkregel: Bei ***Vererbung*** entlang der Klassentaxonomie nach unten werden die Attri-butemengen att(c) „größer“, die abgeleiteten Relationship-Types „kleiner“ (spezieller).

Jedenfalls wird durch die **Vererbung** entlang der Klassentaxonomie $\leq_C$ von oben nach unten auch auf dem System **RELT** aller Relationship-Types eine **Ordnung** $\leq_{RELT}$ induziert: (**RELT**, $\leq_{RELT}$) ist wieder ein ***geordnetes*** System. Man könnte die Ordnung $\leq_{RELT}$ die „**Relationship-Taxonomie**“ nennen. Sie wird bei Informatikern nicht immer hervorgehoben, ist aber m.E. genauso relevant wie die Klassen-Taxonomie $\leq_C$, sofern man überhaupt Relationship-Types in der Ontology berücksichtigt.

Mathematisch Genaueres zur Relationship-Taxonomie $\leq_{RELT}$ lässt sich erst dann einfach ausdrücken, wenn wir die Instanziierung eingeführt haben. Daher sei auf **Kap.3.4** vertröstet.

3 Instanziierung

3.1 Instanziierung in Klassen

Die „**Instanziierung**“ ermittelt zu jeder Klasse c $\in \mathbb{C}$ die Menge <u>c</u> := inst(c) ihrer in der Ontology ermittelbaren **Ausprägungen** – „**Instanzen**“. Manche sagen statt „Instanz“ auch „Entity“, aber ich bleibe beim Wort „Instanz“, da das Wort „Entity“ in anderen Datenmodellierungen wie etwa ERM (Entity Relationship Modelling) benutzt wird, die ich hier nicht beschreibe.

Der Operator **inst**: $\mathbb{C} \rightarrow$ Pot**G** ordnet jeder Klasse c eindeutig die Menge inst(c) ihrer Ausprägungen (Instanzen) zu, wobei **G** die Gesamtheit aller *in der Ontology ermittelbaren* Instanzen sei.

Ähnlich wie bei den Attributen gehe ich von folgenden Regeln aus:

(3.1) c = d gdw. inst(c) = inst(d), u. dies genau dann, wenn att(c)=att(d).

(3.2) c $<_C$ d gdw. inst(c) $\subset$ inst(d), u. dies genau dann, wenn att(d)$\subset$att(c).

Merkregel: Bei ***Vererbung*** nach unten entlang der Klassentaxonomie $<_C$ werden die Klassen c spezieller, die Instanzenmengen inst(c) demnach *kleiner* und die Attributmengen att(c) demnach *größer*.

Begründung: Ist c eine echte Unterklasse von d (also c $<_C$ d), so sind alle Instanzen von c auch Instanzen von d: inst(c) $\subseteq$ inst(d). Wenn aber zwei Klassen c, d *dieselbe* Ausprägungsmenge hätten, inst(c) = inst(d), so hätten sie auch dieselben Attribute, att(c) = att(d); gemäß **(2.1)** wären sie also bei der Instanziierung *nicht unterscheidbar*, sollten also auch im O-Schema ***identifiziert*** werden: c=d ! Je „kleiner“ (im Sinne der Klassentaxonomie $<_C$) eine Klasse c ist, desto ***spezieller*** ist sie, und desto *größer* ist ihre Attributemenge att(c) (vgl. **(2.2)**); je *spezieller* aber die Klasse ist, desto weniger Instanzen kann sie (in der gemeinten Ontology) haben. Beispiel: *„Alle Maler sind Künstler, aber nicht alle Künstler sind Maler!“*

Anmerkung: Auch diese beiden Regeln werden – soviel ich gesehen habe – bei Darstellung einer Klassentaxonomie nicht immer berücksichtigt, und das kann zu Schwierigkeiten oder sogar zu Inkonsistenzen im O-Schema führen.

Anmerkung: Eine Instanz g kann zu mehreren Klassen gehören. Ist etwa c eine Unterklasse von d, so gehört eine Instanz der Klasse c, wie oben gesagt, auch zur Klasse d: g$\in$inst(c) und g$\in$inst(d). Sind andererseits c, d unvergleichbar im Sinne der Klassentaxonomie $\leq_C$, so kann es trotzdem eine Instanz g geben mit $g \in inst(c)$ und $g \in inst(d)$, d.h. die *Ausprägungsmengen* unvergleichbarer Klassen müssen zueinander i.allg. nicht fremd sein (es kann $inst(c) \cap inst(d) \neq \emptyset$ sein)!

Beispiel aus Abb.1: Die Instanz `>Pablo_Picasso` ist der Klasse ^Maler zugeordnet; da Picasso aber auch ein paar Skulpturen gemacht hat, könnte (sofern diese in die Ontology aufgenommen sind), `>Pablo_Picasso` auch als eine Instanz der Klasse ^Bildhauer gelten.

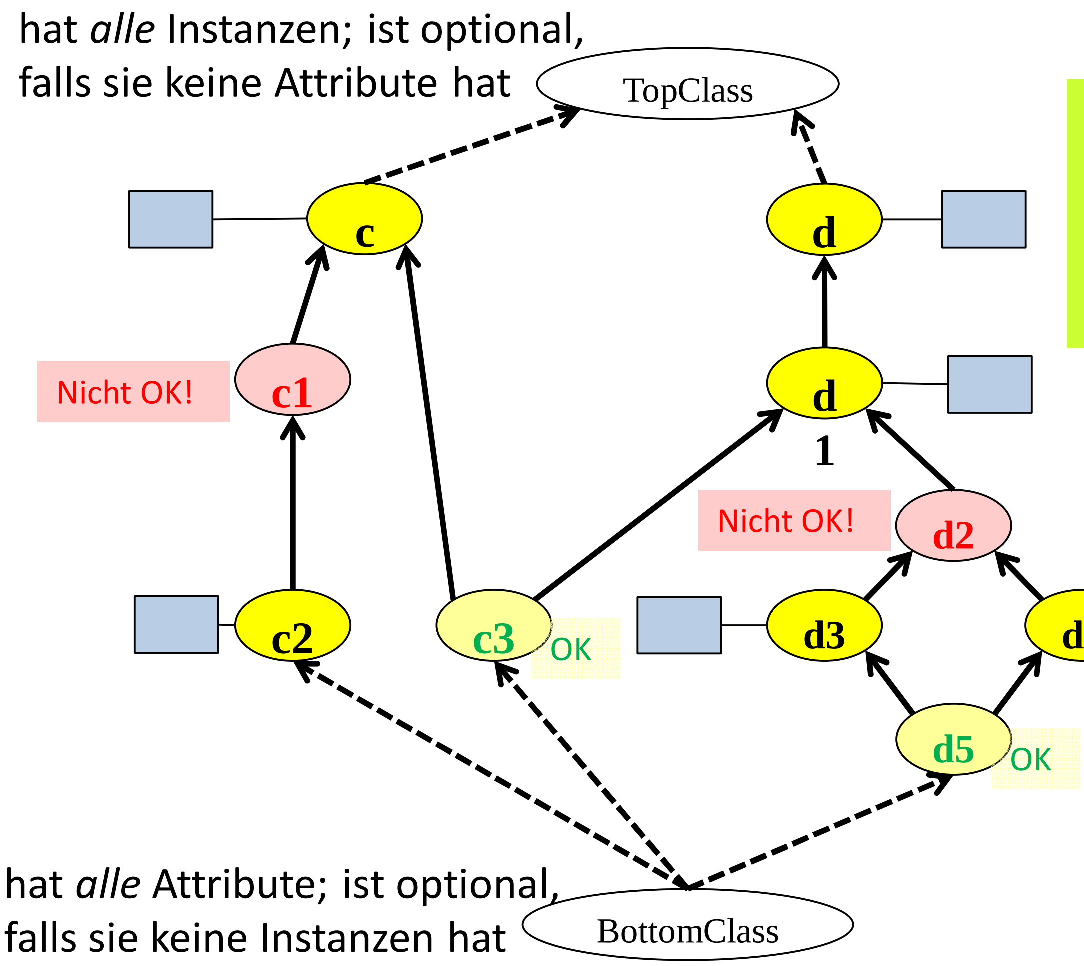
Anmerkung-#: Sind nun c, d zwei *unvergleichbare* Klassen mit $inst(c) \cap inst(d) \neq \emptyset$, und haben sie beide noch keine unteren Nachbarn, so besteht m.E. ***ein Grund dazu***, einen gemeinsamen unteren Nachbarn e in die Taxonomie $<_C$ einzufügen mit

$$inst(e) = inst(c) \cap inst(d) \text{ und } att_{vererbt}(e) = att(c) \cup att(d).$$

– Auch diese Regel wird in einem O-Schema oft nicht beachtet. Das führt zu Unvollständigkeiten der Klassentaxonomie $<_C$ bzw. der Gesamtheit $\mathbb{C}$ aller im O-Schema zu berücksichtigenden Klassen. Im Beispiel der Abb.1 aus Wikipedia kommt das aber nicht zum Tragen.

Abb.2 veranschaulicht noch einmal die Regeln (2.1), (2.2), (3.1) und (3.2), sowie die erwähnten Regelverletzungen, wobei Relationship-Types noch nicht berücksichtigt sind.

Abb.2: O-Schema ohne Relationship-Types



Vererbung:
* Attributemengen (von Klassen) nehmen von oben nach unten zu.
* Instanzenmengen (von Klassen) nehmen von oben nach unten ab.

Klassen c1, d2: Nicht OK, da sie nur je einen oberen Nachbarn, aber keine eigenen Attribute haben.

Klassen c3, d5: OK. Sie haben zwar keine eigenen Attribute, erben aber von je zwei oberen Nachbarn.

3.2 Was ist die volle Menge *aller* Attribute, einer Instanz g?

Alle Attribute einer Klasse c kommen auch allen Instanzen g∈inst(c) zu.

Nehmen wir eine bestimmte Instanz g∈inst(c) an; innerhalb der Klasse c werden der Instanz g – wie allen anderen Instanzen von c auch – nur die Attribute der Menge att(c) zugeordnet. Das müssen aber nicht notwendig schon alle der Instanz g insgesamt zugeordneten Attribute sein! Hat nämlich c eine echte Unterklasse d (d $<_C$ c), zu der g ebenfalls als Ausprägung gehört, so ist wegen (**2.2**) att(d) größer als att(c), also kommen der Instanz g mehr als nur die Attribute von c zu. Um *alle* der Instanz g zugeordneten Attribute zu bekommen, muss man die (im Sinne der Taxonomie $<_C$) ***kleinste*** Unterklasse unterhalb c wählen, die noch g als Instanz hat. Sei d_{min} die *kleinste* Unterklasse von c mit g∈inst($\underline{d}_{min}$). Dann ist att(d_{min}) die ***volle*** Menge *aller* der Instanz g zugeordneten Attribute, und zu *jedem* a∈att(d_{min}) bekommt die Instanz g je einen definierten und bestimmten Attribut-***wert*** a(g). Diese volle Attributemenge einer Instanz g will ich vorläufig mit **„att[**g**]“** bezeichnen:

$$\textbf{att[}g\textbf{]} := \{a \in \mathfrak{A} \mid \text{es gibt } c \in \mathbb{C} \text{ mit } a \in att(c) \text{ und } g \in inst(c)\} \subseteq \mathfrak{A}.$$

Anmerkung: Ist g eine bestimmte Instanz, so muss sie *nicht für jedes* beliebige Attribut a∈$\mathfrak{A}$ überhaupt einen definierten Attributwert a(g) haben. g hat nur Attribut*werte* für die Attribute der Menge att[g], also der Attribute aller derjenigen

Mengen att(c), für deren Klassen g∈inst(c) gilt. – Gegenbeispiel (vgl. Abb.1): Ein Attribut der Klasse ^Museum sei „gegründet_im_Jahr: ….“. Der Instanz `>Pablo_Picasso` wird man wohl nicht dieses Attribut zuschreiben; also gibt es auch keinen Wert dieses Attributs für `>Pablo_Picasso`.

3.3 Instanziierung in Relationship-Types

Beispiel: Sei c1:= ^Person, i1:= >Max ∈ inst(c1), c2:= ^Wohnort, i2:= >Darmstadt ∈ inst(c2), r := <>wohnt_in mit relt(r)=(c1, c2). Dann lautet die relationale

(i) **Teilkonfiguration c1 r c2**: ^Person <>wohnt_in ^Wohnort,

und eine Instanziierung ergibt z.B. die

(ii) **Aussage**: >Max °<>wohnt_in >Darmstadt („Max wohnt in Darmstadt“).

Darin unterscheide ich den **Relationship-Type** <>wohnt_in der Konfiguration (i) von der **Aussageform** °<>wohnt_in der **Aussage** (ii). Denn Relationship-Types sind Komponenten des **O-Schemas**, aber Aussagen sind Ausprägungen der **Instanziierung**. Den Terminus „Aussageform“ für °<>wohnt_in habe i ch aus der Mathematik entlehnt. Er vermittelt sozusagen zwischen Relationship-Type und Aussage.

3.4 Vererbung von Relationship-Types entlang der Klassentaxonomie

Hier wollen wir die Ordnungsbeziehung $\leq_{RELT}$ präzisieren. Besteht ein Relationship-Type **r** zwischen den Klassen c, d, also relt(r) = (c, d), und ist °r die zu r gehörige **Aussageform**, so ist

(3.1) $\underline{r} := \mathrm{instR}(r) := \{(g, h) \in G \times G \mid g\ °r\ h\}$

die Menge der **Ausprägungen** von **r** (also die Menge der Instanzenpaare (g, h) für welche die Aussage g °r h gilt) . $\underline{r}$ ist eine **„Relation im mathematischen Sinne“**, also eine gewisse **Teilmenge** des Mengenprodukts **G**×**G**, wobei **G** die Gesamtheit der in der Ontology ermittelbaren **Instanzen** sei.

Sind nun c1, d1 irgendwelche Unterklassen von d, c (d.h. $d1 \leq_C c$, $d1 \leq_C d$), so wird durch **r** ein Relationship-Type r1 zwischen c1 und d1 **induziert**, dessen Ausprägungsmenge die *Teilrelation*

(3.2) $\underline{r1} := \mathrm{instR}(r1) = \underline{r} \cap (\mathrm{inst}(c1) \times \mathrm{inst}(d1))$

ist. Für sie gilt dann natürlich:

(3.3) $\underline{r1} = \mathrm{instR}(r1) \subseteq \underline{r} = \mathrm{instR}(r)$ $\underline{r1}$ Teilmenge von $\underline{r}$.

Wir können sagen: Der Relationship-Type **r** wird entlang der Klassentaxonomie $\leq_C$ nach unten durch *Einschränkung* auf die Unterklassen c1, d1 **vererbt** auf einen Unter-Relationship-Type **r1**.

Vgl. das Beispiel in Abb.1: <>malt in der Teilkonfiguration „^Maler <>malt ^Bild“ ist ein Unter-Relationship-Type von ^erzeugt , welcher in der darüber liegenden Teilkonfiguration „^Künstler <>erzeugt ^Kunstwerk“ besteht.

Die Mengeninklusion in **(3.3) definiert** geradezu die Ordnungsbeziehung r1$\leq_{RELT}$ r zwischen Relationship-Types: **Dadurch wird das System (RELT, $\leq_{RELT}$) zu einer geordneten Menge.** Die strikte Form $<_{RELT}$ nenne ich die **Relationship-Taxonomie** im O-Schema. Sie ist m.E. ***genau so relevant*** wie die Klassen-Taxonomie $<_C$, sofern Relationship-Types im O-Schema auftreten.

Def.: Ein Relationship-Type **r** zwischen zwei Klassen c, d heiße „**maximal**“, wenn er ***keinen*** oberen Nachbarn (im Sinne der Ordnung $<_{RELT}$) hat, d.h. wenn er nicht von oben her (im Sinne der Klassentaxonomie $<_C$) vererbt ist.
Zwei verschiedene *maximale* Relationship-Types r, s sind folglich stets *unvergleichbar* im Sinne der Relationship-Taxonomie $\leq_{RELT}$, d.h. keiner ist ein Unter-Relationship-Type des anderen. Die ***maximalen*** Relationship-Types sind sozusagen die „***wichtigen***“ oder ***„relevanten“*** im O-Schema, denn alle anderen sind Unter-Relationship-Types von ihnen und können durch Vererbung aus ihnen abgeleitet werden. – Zum Stichwort „relevante Relation“ vgl. auch Lübbert [18].

Die Gesamtheit **RELT** aller Relationship-Types des O-Schemas besteht also aus zwei Teilen:

(3.4) $\mathbf{RELT = RELT_{max} \cup RELT_{vererbt}}$

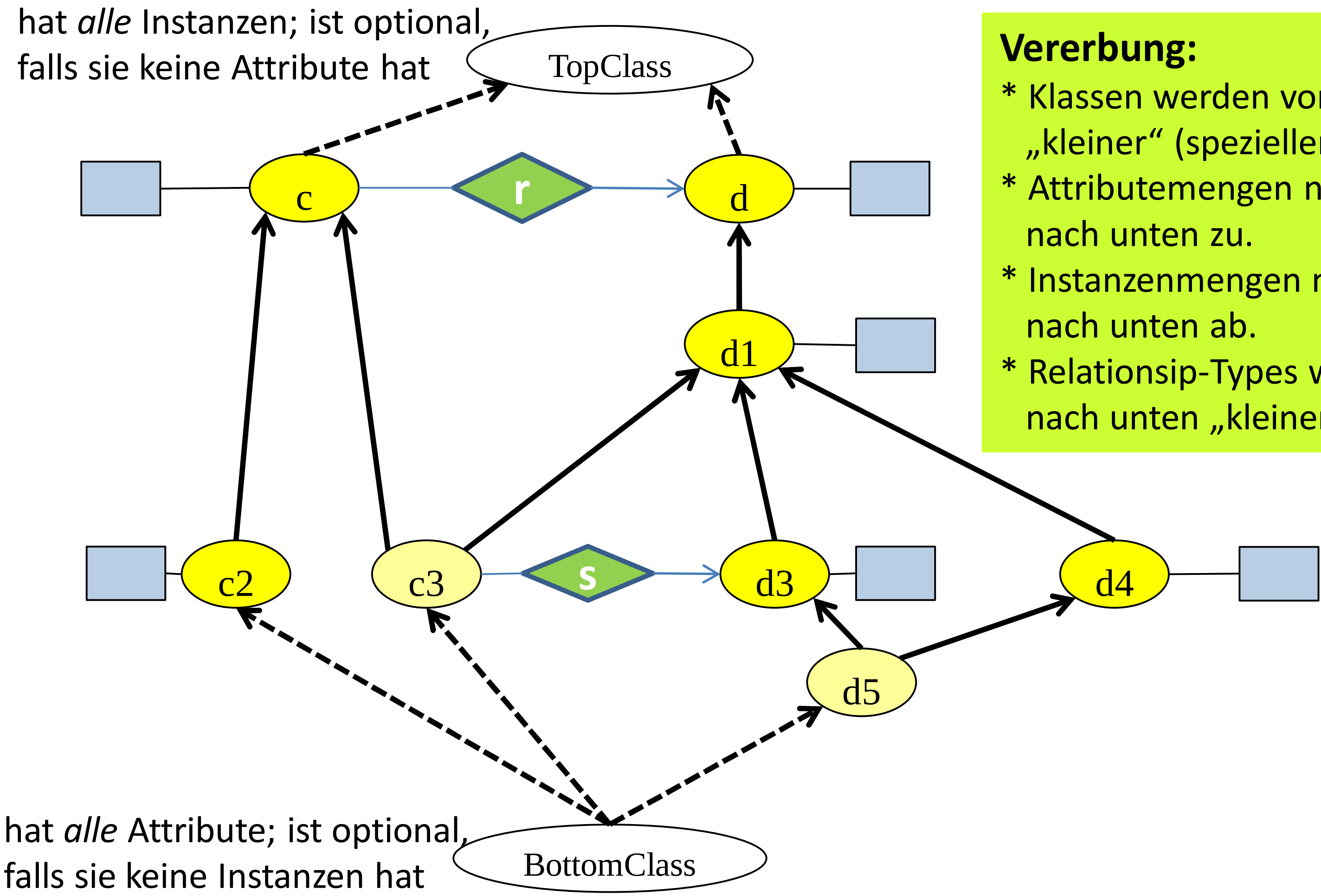
Vorschlag: Aus Gründen der Übersichtlichkeit würde ich in der graphischen Darstellung eines O-Schemas nur die ***maximalen*** Relationship-Types einzeichnen und ihre eventuell sehr vielen Vererbungen in der Graphik weglassen. Ganz analog macht man das ja auch bei den Attributen. In der **Abb.1** wäre der Eintrag des Relationship-Types <>malt (zwischen ^Maler und ^Bild) dann ***überflüssig***.

Dazu geben wir noch einmal eine Veranschaulichung in **Abb.3.** Dieses Bild ist entstanden aus Abb.2, indem wir die „Fehler“ der Abb.2 beseitigt haben, und danach zwei verschiedene ***maximale*** Relationship-Types <>r, <>s eingeführt haben mit den relationalen Teilkonfigurationen

^c <>r ^d und ^c3 <>s ^d3.

Abb.3: O-Schema

mit zwei **maximalen(!)** Relationship-Types <>r, <>s



Vererbung:

* Klassen werden von oben nach unten „kleiner“ (spezieller).
* Attributemengen nehmen von oben nach unten zu.
* Instanzenmengen nehmen von oben nach unten ab.
* Relationsip-Types werden von oben nach unten „kleiner“ (spezieller).

Beachte noch einmal:
(a) Gemäß unserem Vorschlag sind weder die bei Vererbung entstehenden Unter-Relationship-Types von <>r noch die von <>s im O-Schema der **Abb.3** eingezeichnet.

(b) Da <>r und <>s beide *maximal* (und damit in der Ordnung $\leq_{RELT}$ unvergleichbar) sein sollen ist <>s ***nicht*** etwa ein Unter-Relationship-Type von <>r; vielmehr bestehen zwischen ***denselben*** beiden Klassen c3, d3 *zwei* ***verschiedene*** *(eventuell sogar unvergleichbare*) Relationship-Types: der eine ist von <>r vererbt, der andere ist <>s.

3.5 Instanziierung in Attributen

Zu jedem Attribut $a \in \mathfrak{A}$ gehört die Menge $\underline{a}$:= val(a) seiner **Attribut*werte*** – statt „Attributwert“ sagen einige auch "*literal value*", vgl. *Maedche* [12]. Die Termini „Attribut“ und „Attribut*wert*“ muss man unterscheiden – ganz analog dazu, dass man „Klasse“ von „Instanz“ unterscheidet!

Ist g eine Instanz der Klasse c, $g \in$ inst(c), und a ein Attribut der Klasse c, $a \in$ att(c), so kommt g ein eindeutig bestimmter Attribut***wert*** von a zu, den ich mit a(g) bezeichne. Es ist also $a(g) \in \underline{a}$ = val(a).

3.6 Was in den mir bekannten O-Schema-Definitionen auffällt

(1) Klassen werden bei Informatikern ***anders*** behandelt als ihre Attribute. – Was ist der Grund für diese Asymmetrie? Insbesondere scheint das System $\mathbb{A}$ der Attribute keine Ordnung zu tragen, wie das im taxonomisch geordneten Klassensystem ($\mathbb{C}$, $<_C$) der Fall ist.

(2) Die Regeln (2.1), (2.2) und (3.1), (3.2) werden von manchen Informatikern als zu restriktiv angesehen und daher nicht immer eingehalten.

⇨ Die in (1) erwähnte Asymmetrie löst sich auf, und die in (2) erwähnten Regeln rechtfertigen sich, wenn man in die Sprache der **FBA (Formale Begriffsanalyse)** übersetzt. Das ist Thema des nächsten Kapitels.

4 Bezug zur FBA (Formale Begriffsanalyse)

In Kap.4 und Kap.5 führe ich die Grundkonzepte der **Formalen Begriffsanalyse** (**FBA**), vgl. *Ganter / Wille* [1], ein – nicht mathematisch-abstrakt wie in [1], sondern *wiederholt* an ein paar Anwendungsbeispielen im Bereich des bisher beschriebenen Konzepts „Ontology“. Durch die Wiederholungen hoffe ich, dass die Grundkonzepte der FBA auch für Informatiker eingängiger werden.

4.1 Formaler Kontext für die Beziehung zwischen Instanzen und Attributen

Betrachten wir vom O-Schema zunächst nur das Stück

$\mathbf{O}_{\text{Tax}} := (\mathbb{C}, <_C, \mathfrak{A}, \text{VEB})$

und lassen das System **RELT** der Relationship-Types noch weg. Der Vererbungsmechanismus VEB bezieht sich dann nur auf Attribute. Wir lassen hier auch die bei Instanziierung notwendige Zuordnung von Attribut***werten*** zu Instanzen noch weg, sondern beachten nur, wann bei gegebener Instanz g und gegebenem Attribut $a \in \mathfrak{A}$ ein Attributwert a(g) überhaupt definierbar ist und wann nicht.

In der **FBA** hat man für jene ganze – bisher im Kap.2 + 3 recht umständliche – Beschreibung des Stücks $\mathbf{O}_{\text{Tax}}$ eine etwas kompaktere Notation, die ich nun vorstelle, und die den Zusammenhang zwischen Klassen (bzw. deren Instanzen) und Attributen ***und deren Vererbung*** etwas präziser und vor allem ***„symmetrischer“*** darstellt, als es in der Ontology-Sprache zurzeit üblich ist:

Sei **G** die **Gesamtheit aller Instanzen**, die aus Klassen des O-Schemas bei der gewählten Instanziierung **inst** gewonnen werden können, und sei $\mathfrak{A}$ wieder die **Gesamtheit aller Attribute**.

I sei die **Aussageform**, die angibt, ob einer Instanz $g \in \mathbf{G}$ das Attribut $a \in \mathfrak{A}$ überhaupt zukommt. Für die Aussage „g **I** a“ wollen auch wir sagen: „die Instanz g hat das Attribut a“. Falls g **I** a zutrifft, ist der Attribut***wert*** a(g) eindeutig definiert und ermittelbar; der Wert a(g) selbst kommt hier aber in diesem FBA-Anwendungsschritt ***noch gar nicht ins Spiel***.

Im Jargon der Mathematiker gehört nun zur Aussageform **I** die "**Relation**" $\underline{\mathbf{I}}$ im *mathematischen* Sinne: das ist die Menge

$$\underline{\mathbf{I}} := \{(x, y) \in \mathbf{G} \times \mathfrak{A} \mid x \ \mathbf{I} \ \mathbf{y}\};$$

es ist eine gewisse **Teilmenge** des Mengenprodukts $\mathbf{G} \times \mathfrak{A}$. Die Aussage "g **I** a" kann nun auch in der Form „$(g, a) \in \underline{\mathbf{I}}$“ ausgedrückt werden.

Mit den Mengen **G**, $\mathfrak{A}$ und $\underline{\mathbf{I}}$ bekommen wir in der **FBA** den sogenannten **<u>formalen Kontext</u>**

(4.1) $K(\mathbf{I}) := (\mathbf{G}, \mathfrak{A}, \underline{\mathbf{I}})$

In der FBA-Sprache nennt man nun die Elemente von **G** (also die Instanzen) "**Gegenstände**", die Elemente von $\mathfrak{A}$ (also die Attribute) "**Merkmale**". Die **Inzidenzrelation $\underline{\mathbf{I}}$** induziert zwei Abbildungen $\uparrow\mathbf{I}$: Pot**G** $\rightarrow$Pot$\mathfrak{A}$ und $\downarrow\mathbf{I}$: Pot$\mathfrak{A}$ $\rightarrow$Pot**G** ("Galois-Verbindung zwischen **G** und $\mathfrak{A}$"), definiert für beliebige Teilmengen $X\subseteq\mathbf{G}$, $Y\subseteq\mathfrak{A}$ durch

(4.2) $X^{\uparrow I} := \{a\in\mathfrak{A} \mid \text{für alle } g\in X\text{: } g\ \mathbf{I}\ a\} = \bigcap_{g\in X} \{g\}^{\uparrow I}$, mit $\{g\}^{\uparrow I} := \{a\in\mathfrak{A} \mid g\ \mathbf{I}\ a\} = \mathrm{att}[g]$,

(4.3) $Y^{\downarrow I} := \{g\in\mathbf{G} \mid \text{für alle } a\in Y\text{: } g\ \mathbf{I}\ a\} = \bigcap_{a\in Y} \{a\}^{\downarrow I}$, mit $\{a\}^{\downarrow I} := \{g\in\mathbf{G} \mid g\ \mathbf{I}\ a\}$.

In Worten: **$X^{\uparrow I}$ ist die Menge der Attribute** (Merkmale), **die *allen* Instanzen** (Gegenständen) **von X gemeinsam sind; und $Y^{\downarrow I}$ ist die Menge der Instanzen** (Gegenstände)**, die *allen* Attributen** (Merkmalen) **von Y gemeinsam sind.**

$\{g\}^{\uparrow I}$ ist gerade die volle Attributemenge att[g] der Instanz g. Symmetrisch (oder „dual“) dazu ist $\{a\}^{\downarrow I}$ die Menge aller Instanzen $g\in\mathbf{G}$, die das Attribut a haben.

DEF.: Ein sog. **„formaler Begriff“** zum Kontext K(**I**) ist ein **Mengenpaar** (A, B) mit $A\subseteq\mathbf{G}$, $B\subseteq\mathfrak{A}$, das den beiden Bedingungen

(4.4) $A^{\uparrow I} = B$, $B^{\downarrow I} = A$

genügt. A heißt der **Umfang** ("Extension"), B der **Inhalt** ("Intension") des formalen Begriffs.

Zum formalen Kontext K(**I**) gehört die Menge $\underline{\mathfrak{B}}$(**I**) all dieser formalen Begriffe; sie trägt in natürlicher Weise eine ***Ordnung***, deren strikte Form mit $<_I$ und deren nicht-strikte Form mit $\leq_I$ bezeichnet sei und so definiert ist: Sind (A, B) und (A', B') formale Begriffe in $\underline{\mathfrak{B}}$(**I**), so definiert man

(4.5) (A, B) $\leq_I$ (A', B') :gdw. A $\subseteq$ A' , und das ist gleichwertig zu B' $\subseteq$ B.

Gilt (A,B) $<_I$ (A',B'), so heißt (A,B) ein formaler **Unterbegriff** von (A‘,B‘) bzw. (A‘,B‘) ein formaler **Oberbegriff** von. Die Ordnung $\leq_I$ besagt mit (4.5): ***je größer der Umfang, desto kleiner der Inhalt***; und je kleiner der Umfang, desto größer der Inhalt – ganz im Sinne der intuitiven Vorstellung von der "Extension" und der "Intension" eines "Begriffs".

Die geordnete Menge ($\underline{\mathbb{B}}$(**I**), $\leq_I$) heißt der **Begriffsverband** zum Kontext K(**I**); sie hat eine ***sehr schöne*** FBA-Struktur: nämlich die eines sog. **vollständigen Verbandes**[1].

Fassen wir alle Umfänge von $\underline{\mathbb{B}}$(**I**) in einer Menge $\mathbb{U}$(**I**) und alle Inhalte in einer Menge $\mathbb{J}$(**I**) zusammen, so ist sowohl $\mathbb{U}$(**I**) als auch $\mathbb{J}$(**I**) durch die Teilmengeninklusion $\subseteq$ geordnet, und die geordneten Mengen ($\mathbb{U}$(**I**), $\subseteq$) , ($\mathbb{J}$(**I**), $\subseteq$) sind sogar zuein-

---

[1] DEF.: Ein **„vollständiger Verband“** ist eine geordnete Menge (**B**, ≤), in der für ***jede*** Teilmenge X⊆**B** das „Supremum“ supX := Min{ y∈**B**| x≤y für alle x∈X} = „kleinste obere Schranke von X“ und das „Infimum“ infX := Max{y∈**B**| y≤x für alle x∈X} = „größte untere Schranke von X“ existieren und Elemente der Menge **B** sind (supX∈**B**, infX∈**B**) (Das ist allgemein *nicht für jede* Ordnung der Fall!). Insbesondere gilt: inf**B** = sup∅ = *kleinstes* Element von **B**, sup**B** = inf∅ = *größtes* Element von **B**. (Beachte, dass, nach mathematischer Konvention, die leere Menge ∅ ja Teilmenge einer beliebigen Menge ist!). Hiermit kann man auf dem Verband (**B**, ≤) zwei binäre Verknüpfungen ∧, ∨ einführen durch x∧y := inf{x,y}, x∨y := sup{x,y} für beliebige x, y ∈**B**. Sie haben für alle x, y, z∈**B** folgende Eigenschaften:

(K)	Kommutativität:	$x\vee y = y\vee x$,	$x\wedge y = y\wedge x$
(As)	Assoziativität:	$x\vee(y\vee z) = (x\vee y)\vee z$,	$x\wedge(y\wedge z) = (x\wedge y)\wedge z$
(Ap)	Absorption:	$x\vee(x\wedge y) = x\wedge(x\vee y) = x$	
(Id)	Idempotenz:	$x\vee x = x\wedge x = x$	
(Vt)	≤ -Verträglichkeit:	$x\wedge y \leq x \leq x\vee y$,	$x\wedge y \leq y \leq x\vee y$.

ander duale **vollständige Verbände**, geben also beide *zusammen* die Struktur des Begriffsverbandes ($\underline{\mathfrak{B}}$(**I**), $\leq_I$) wieder.

Zurück zur Informatiker-Sprache. Setzen wir im Konzept „Ontology“ die Regeln **(2.1), (2.2), (3.1), (3.2)** und die **Anmerkung-#** in Kap.3.1 voraus, so ersieht man:

- Jeder Umfang A $\in \mathbb{U}$(**I**) ist gerade die Ausprägungsmenge inst(c) der Instanzen einer Klasse c $\in \mathbb{C}$.
- Jeder Inhalt B $\in \mathbb{J}$(**I**) ist gerade die Menge att(c) der Attribute einer Klasse c.
- Damit ist das taxonomisch geordnete Klassensystem ($\mathbb{C}$, $\leq_C$) dargestellt durch den ***vollständigen Verband*** ($\mathbb{U}$(**I**), $\subseteq$).
- Das geordnete System ([$\mathbb{A}$], $\leq_A$) der Attributemengen wird durch den ***vollständigen Verband*** ($\mathbb{J}$(I), $\subseteq$) dargestellt. Das System [$\mathbb{A}$] trägt also eine ***Ordnung*** $\leq_A$, die ebenfalls durch die Mengeninklusion ausgedrückt wird. Die Ordnung $\leq_A$ ist wegen (4.5) „dual“ zur Klassentaxonomie $\leq_C$. Man könnte $\leq_A$ als die „**Attribute-Taxonomie**“ bezeichnen.

Bemerkenswert ist also, dass in einem O-Schema, das den Regeln (2.1), (2.1), (3.1), (3.2) und der **Anmerkung-#** in Kap.3.1 gehorcht, das taxonomisch geordnete Klassensystem ($\mathbb{C}$, $\leq_C$) dual-isomorph ist zum geordneten Attribute*mengen-*

system ([$\mathfrak{A}$], $\leq_A$), und dass das Stück $\mathbf{O}_{Tax}$:=($\mathbb{C}$, $\leq_C$, $\mathfrak{A}$, VEB) des O-Schemas durch den formalen Kontext

K(**I**) = (**G**, $\mathfrak{A}$, $\underline{\mathbf{I}}$) bzw. durch den formalen Begriffsverband

($\underline{\mathfrak{B}}$(**I**), $\leq_I$) bzw. durch die beiden zueinander "dualen" vollständigen Verbände

($\mathbb{U}$(**I**), $\subseteq$) ~ ($\mathbb{C}$, $\leq_C$), ($\mathbb{J}$(**I**), $\subseteq$) ~ ([$\mathfrak{A}$], $\leq_A$)

dargestellt werden kann. Wir haben also für das Stück $\mathbf{O}_{Tax}$ des O-Schemas folgenden Zusammenhang zwischen Informatiker- und Mathematiker-Sprache:

- Jede im O-Schema auftretende Klasse c∈$\mathbb{C}$ wird in der FBA dargestellt durch den **Umfang** A eines formalen Begriffs (A, B)∈$\underline{\mathfrak{B}}$(**I**).
- Jede zu einer Klasse c∈$\mathbb{C}$ gehörige Attributemenge att(c)⊆$\mathfrak{A}$ wird in der FBA dargestellt durch den **Inhalt** B eines formalen Begriffs (A, B)∈$\underline{\mathfrak{B}}$(**I**).
- Jedes zusammengehörige Paar (c, att(c)), bestehend aus einer Klasse c∈$\mathbb{C}$ und ihrer Attributemenge att(c) ⊆ $\mathfrak{A}$, wird in der FBA dargestellt durch einen **formalen Begriff** (A, B) ∈ $\underline{\mathfrak{B}}$(**I**) zum formalen Kontext K(**I**)=(**G**, $\mathfrak{A}$, $\underline{\mathbf{I}}$).
- Die **Vererbung von Attributen** („Merkmalen“) drückt sich in der FBA aus als Vererbung ***von oben nach unten*** im Sinne der Begriffsverbandsordnung $\leq_I$. „Dual“ dazu entspricht dem eine ***Vererbung der Instanzen*** („Gegenstände“) ***von unten nach oben*** im Sinne der Begriffsverbandsordnung $\leq_I$. Das ist eine sehr schöne

„Symmetrie“, zwischen den Instanzen („Gegenständen“) und den Attributen („Merkmalen“), die in der üblichen Ontology-Definition nicht so klar zum Ausdruck kommt wie in der FBA.

- Das gesamte durch die Klassentaxonomie geordnete System ($\mathbb{C}$, $\leq_C$) der Klassen wird in der FBA dargestellt durch den **Umfangsverband** ($\mathbb{U}$(**I**), $\subseteq$).
- „Dual“ dazu: Das gesamte – ebenfalls geordnete – Attribute*mengen*system ([$\mathfrak{A}$], $\leq_A$) wird in der FBA dargestellt durch den **Inhaltsverband** ($\mathbb{J}$(**I**), $\subseteq$).

Der größte formale Begriff ist (**G**, $\mathbf{G}^{\uparrow I}$), der kleinste ist ($\mathfrak{A}^{\downarrow I}$, $\mathfrak{A}$) (wobei in Anwendungen oft $\mathbf{G}^{\uparrow I}=\varnothing$ bzw. oft $\mathfrak{A}^{\downarrow I}=\varnothing$ ausfällt). Daher: Die größte *Klasse* (oft als „**TopClass**“ bezeichnet und oft weggelassen, sofern $\mathbf{G}^{\uparrow I}=\varnothing$ ist) wird durch die Menge **G** *aller* in der Ontology ermittelbaren Instanzen dargestellt; es gibt aber auch die kleinste *Klasse* (oft als „**BottomClass**“ bezeichnet und oft weggelassen, sofern $\mathfrak{A}^{\downarrow I}=\varnothing$ ist); sie wird durch die Menge $\mathfrak{A}^{\downarrow I}$ dargestellt, das ist die Menge der Instanzen denen *alle* Attribute von $\mathfrak{A}$ zukommen. Die größte *Attributemenge* ist die Menge $\mathfrak{A}$; es gibt aber auch die kleinste Attributemenge; sie wird durch die Menge $\mathbf{G}^{\uparrow I}$ dargestellt, das ist die Menge der Attribute, die *allen* Instanzen zukommt.

Es gibt also eine durchgehende „Symmetrie“ zwischen Klassen und ihren Attributemengen. In der FBA nennt man das die „*Dualität*“

(zwischen „Gegenständen“ und „Merkmalen“ im Kontext oder auch die „*Dualität*“ zwischen Umfängen und Inhalten im Begriffsverband).

4.2 Eine philosophisch-historische Anmerkung

Es wäre nach meiner Ansicht also auch in einer „Ontology“ besser, „Begriffe“ nicht durch die Klassen c sondern durch die ***Paare*** (c, att(c)) darzustellen! Leider hat sich aber bei Informatikern einseitig und asymmetrisch die „Klasse“ als Begriffs-darstellung eingebürgert, wogegen die Attribute „nur so nebenher vererbt“ werden; und das gibt unnötige Streitereien.
Diese asymmetrische Sichtweise ist uralt: Solange es „philosophische Ontologie“ gibt, hat man in erster Linie auf die „Objekte“/ „Dinge“ gestarrt und ihre „Eigenschaften“ erst in zweiter Linie ernstgenommen. Später hat man sogar angefangen, über die „Dinge an sich“ (also „ohne Eigenschaften“) zu spekulieren. (Erst *David Hume* und *Immanuel Kant* haben mit diesem Unsinn aufgeräumt). Ich nenne das scherzhaft die ***„platonisch-mittelalterliche Backsteinmentalität“***. Diese hat natürlich mit der *noch viel älteren* menschlichen Erfahrung zu tun, dass man einen Backstein in die Hand nehmen und sehen bzw. spüren kann, dass er rot / schwer / fest / kalt / usw… ist. Aber seine „Röte“ / sein „Gewicht“ / seine „Festigkeit“ / seine „Kälte“ / etc… kann man nicht in die Hand nehmen, nicht sehen, nicht spüren, denn es sind ja nur nützliche ***Abstraktionen***, die erst im menschlichen Gehirn erzeugt

werden und uns Menschen recht gut unterstützen, ***planend*** mit unserer Umwelt zurechtzukommen. Beeinflusst aber durch jene „*Backsteinmentalität*“ kamen alte Philosophen – wie etwa der antike ***Platon***, und später besonders einige mittelalterlich-christlichen Scholastiker – auf die verrückte Idee, auch „die Röte“ / „das Gewicht“ / „die Festigkeit“ / „die Kälte“ / etc. … ebenfalls wie „höhere“, evtl. sogar „ewige“ Backsteine zu behandeln. Daraus entstand u.a. die absurde mittelalterliche Streitfrage, ob die sogenannten ***Universalia*** (Allgemeinbegriffe wie z.B. „die Röte“, „das Gewicht“, etc. …) unabhängig von der Wahrnehmung ihrer Ausprägungen „existieren“ oder nicht „existieren“. (Das ***Absurde*** dabei waren eigentlich weniger jene Universalia, sondern der ***völlig unreflektierte*** Gebrauch des Wortes „existieren“! Daran hat ja seit eh und je die sog. „philosophische Ontologie“ gekrankt.)

Man sieht: **Das derzeitige informatische, asymmetrische Konzept „Ontology“ ist durch eine lange, nicht besonders originelle Vorgeschichte belastet.**

4.3 Berücksichtigung der Attribut*__werte__* durch Bildung eines mehrwertigen Kontextes

Die Inzidenzaussage „g **I** a“ im formalen Kontext K(**I**) = (**G**, $\mathfrak{A}$, $\underline{\mathbf{I}}$) besagt nur, dass der Instanz g∈**G** ein Attribut a∈$\mathfrak{A}$ zugeordnet ist, so dass a(g) ein definierter Attribut*wert* ist. Der Attributwert a(g) selbst ist damit noch nicht ermittelt. K(**I**) ist sozusagen ein „einfacher“ formaler Kontext; seine Inzidenzrelation ist ***binär***. Will man im Stück $\mathbf{O}_{Tax}$ = ($\mathbb{C}$, $\leq_C$, $\mathfrak{A}$, VEB) bei Instanziierung auch die Attribut*werte* a(g) selbst berücksichtigen (soweit sie definiert sind), so bildet man gemäß dem Vorschlag der FBA [1] einen sog. **mehrwertigen formalen Kontext**

K(**I***) **= (G,** $\mathfrak{A}$, W, $\underline{\mathbf{I}}$*),

wobei W die Menge aller möglichen *Werte* von Attributen aus $\mathfrak{A}$ ist; und **I*** keine binäre, sondern eine aus der binären Relation **I** abgeleitete ***3-stellige*** Relation $\underline{\mathbf{I}}$*⊆**G**×$\mathfrak{A}$×W ist, für die gilt:

(g, a, w)∈$\underline{\mathbf{I}}$* :⇔ (g, a)∈$\underline{\mathbf{I}}$ und a(g)=w (für g∈**G**, a∈$\mathfrak{A}$, w∈W).

Nach FBA kann man K(**I***) in einen „größeren“ aber „einfachen“ Kontext K(**I****) = (**G**, W, **I****) mit ***binärer*** Inzidenzrelation $\underline{\mathbf{I}}$** umwandeln, indem man **G** als „Gegenstandsmenge“ beibehält, aber statt $\mathfrak{A}$ die Wertemenge W als „Merkmalsmenge“ nimmt, und $\underline{\mathbf{I}}$** definiert durch (g, w) ∈ $\underline{\mathbf{I}}^{**}$:⇔ es gibt ein Attribut a∈ $\mathfrak{A}$, so dass a(g) definiert ist und den Wert a(g) = w hat.

Da nun K(**I****) wieder ein „einfacher“ formaler Kontext ist, gehört zu ihm ein formaler Begriffsverband **<u>B</u>**(**<u>I</u>****). Bei den formalen Begriffen von **<u>B</u>**(**<u>I</u>****) bestehen die Umfänge nun aus Mengen von Instanzen, die Inhalte aus Mengen von Attribut-***werten***.

4.4 Formale Kontexte der Relationship-Types

Nun betrachten wir das volle O-Schema $\mathbf{O}_{full}$:= ($\mathbb{C}$, $\leq_C$, $\mathfrak{A}$, **RELT**, VEB), beziehen also die Menge **RELT** der Relationship-Types mit ein.

Nehmen wir im O-Schema einmal zwei Klassen c, d ∈ $\mathbb{C}$ her, die durch einen Relationship-Type r ∈ **RELT** verbundenen sind, relt(r) = (c, d). Es bestehe also die relationale Teilkonfiguration **c r d.**

Beispiel für eine Teilkonfiguration **c r d** (vgl. Abb.1): ^Künstler <>erzeugt ^Kunstwerk.

Ich verwende im Folgenden die etwas kürzere Notation „<u>c</u>“ statt „inst(c)“ für die Menge aller Instanzen der Klasse c.

Sind <u>c</u> := inst(c), <u>d</u> := inst(d) die Instanzenmengen der Klassen c, d und ist °r die zum Relationship-Type r gehörige Aussageform, und bilden wir daraus die mathematische Relation

$\underline{r}$:={(g, m) | $g \in \underline{c}$, $m \in \underline{d}$, g ♈ m} $\subseteq$ $\underline{c} \times \underline{d}$,

so bekommen wir gemäß FBA den **formalen Kontext**

K($\underline{r}$) := ($\underline{c}$, $\underline{d}$, $\underline{r}$).

K($\underline{r}$) stellt die Teilkonfiguration **c r d** des O-Schemas dar. Die „**Gegenstände**“ von K($\underline{r}$) sind die Instanzen der Klasse c, die „**Merkmale**“ sind die Instanzen der Klasse d. Die Inzidenzrelation $\underline{r}$ induziert wieder die beiden Abbildungen

$\uparrow$r: Pot$\underline{c}$$\to$Pot$\underline{d}$, $\downarrow$r: Pot$\underline{d}$$\to$Pot$\underline{c}$, („Galois-Verbindung“), definiert für beliebige Teilmengen $X \subseteq \underline{c}$, $Y \subseteq \underline{d}$ durch:

$X^{\uparrow r}$:={$y \in \underline{d}$| für alle $x \in X$: x ♈ y}, $Y^{\downarrow r}$:={$x \in \underline{c}$| für alle $y \in Y$: x ♈ y}.

Ein **formaler Begriff** des Kontextes K($\underline{r}$) ist gemäß FBA-Sprache ein Mengenpaar (A, B) mit $A \subseteq \underline{c}$, $B \subseteq \underline{d}$, das die Bedingungen

$A^{\uparrow r}$ =B, $B^{\downarrow r}$ =A,

erfüllt. A heißt der **Umfang**, B der **Inhalt** des formalen Begriffs (A, B).

In Worten: **Der „Inhalt“ B besteht aus denjenigen Instanzen von d, die mit *allen* Instanzen des „Umfangs“ A in der Beziehung ♈ stehen ; der „Umfang“ A besteht aus denjenigen Instanzen von c, die mit *allen* Instanzen des „Inhalts“ B in der Beziehung ♈ stehen.**

Das klingt kompliziert, weil es ganz allgemein formuliert ist. Im konkreten Fall einer „kleinen“ Ontology ergibt sich oft etwas ganz Einfaches.

Beispiel (vgl. Abb.1): Klassen c:= ^Maler, d:= ^Bild, Relationship-Type r:= <>malt. c = ^Maler ist dann die Menge aller `Malerinstanzen`, d = ^Bild ist die Menge aller `Bildnstanzen`, die in der betreffenden Ontologie ermittelbar sind, und r = <>malt = {(x, y) | >x °<>malt > y} ist die zum Relationship-Type <>malt gehörige **Relation.** Eine Ausprägung davon wäre zum Beispiel: `>Pablo_Picasso` °<> `malt` `>Junge_mit_der_Pfeife`. Nehmen wir nun bei der gewählten Instanziierung an, Picasso habe alle seine Bilder selbst gemalt (ohne Beteiligung von Schülern), dann bildet das Mengenpaar (A, B) mit

A:= { `>Pablo_Picasso`} = $B^{\downarrow \underline{r}}$ (eine nur 1-elementige Menge!) ,

B:= Menge aller Bilder von Pablo Picasso = $A^{\uparrow \underline{r}}$,

einen **Formalen Begriff** zum formalen Kontext (^Maler, ^Bild, <>malt).

Der zum formalen Kontext K(r) gehörige **Begriffsverband** sei mit (**B**(r), $\leq_r$) bezeichnet. **B**(r) trägt in natürlicher Weise eine **Ordnung** „$\leq_r$", die definiert ist durch

$(X, Y) \leq_r (X', Y') :\Leftrightarrow X \subseteq X', Y' \subseteq Y$ für formale Begriffe (X,Y), (X',Y') ∊ **B**(r).

Im Kontext (c, d, r) wird die Klasse c durch den Umfang c des *größten* Begriffs (c, $c^{\uparrow r}$), die Klasse d durch den Inhalt d des *kleinsten* formalen Begriffs ($d^{\downarrow r}$, d) dargestellt. (Die Umfänge / Inhalte der formalen Begriffe „zwischen" dem kleinsten und dem größten stellen sozusagen „Verfeinerungen" der Klassen c, d dar, gehören aber selbst nicht mehr zum ursprünglichen Klassensystem $\mathbb{C}$.)

So kann man ***jede*** relationale Teilkonfiguration **c r d** des O-Schemas [c, d$\in\mathbb{C}$, r$\in$**RELT**, relt(r)=(c,d)] durch einen ***formalen Kontext*** K($\underline{r}$) = ($\underline{c}$, $\underline{d}$, $\underline{r}$), und damit durch einen ***Begriffsverband*** ($\underline{\mathbf{B}}$($\underline{r}$),$\leq$r), darstellen.

4.5 „Verfeinerung" einer Ontology im Sinne der FBA

In der üblichen Definition einer Ontology begnügt man sich mit nur ***zwei*** Ebenen: Der oberen Ebene des **O-Schemas** und der unteren Ebene der **Instanziierung**. Setzt man, wie eben beschrieben, FBA-Methoden ein, so bekommt man eine „***Verfeinerung***" in ***drei*** Ebenen im folgenden Sinne:

- Die **Oberste Ebene** bildet das Stück $\mathbf{O}_{Tax}$:=($\mathbb{C}$, $\leq_C$, $\mathfrak{A}$, VEB) des **O-Schemas.** $\mathbf{O}_{Tax}$ wird durch den formalen Kontext K(**I**) = (**G**, $\mathfrak{A}$, $\underline{\mathbf{I}}$) bzw. dessen Begriffsverband ($\underline{\mathbb{B}}$(**I**), $\leq_I$) dargestellt, wobei die formalen Begriffe gerade die Paare (c, att(c)) aus Klassen c$\in\mathbb{C}$ und ihren Attributemengen att(c)$\subseteq\mathfrak{A}$ sind. Dabei erfolgt die **Vererbung** der **Attribute** „von oben nach unten" entlang der Klassentaxonomie $<_C$ und, symmetrisch („dual") dazu, erfolgt die „**Vererbung**" der **Instanzen** „von unten nach oben" entlang $<_C$.
- Eine **mittlere Ebene** berücksichtigt alle **Relationship-Types** der Menge **RELT**. Diese Ebene wird durch alle formalen Kontexte K($\underline{r}$) = ($\underline{c}$, $\underline{d}$, $\underline{r}$) bzw. die

formalen Begriffsverbände (**B**(r), $\leq_r$) dargestellt, wobei r∈ **RELT** je ein Relationship-Type mit relt(c, d) ist, der gewisse Paare c, d ∈ $\mathbb{C}$ zu einer relationalen Teilkonfiguration **c r d** verbindet. Der *Umfang* des jeweils $\leq_r$-*größten* formalen Begriffs von (**B**(r), $\leq_r$) repräsentiert die Klasse c, der *Inhalt* des jeweils $\leq_r$-*kleinsten* formalen Begriffs von (**B**(r), $\leq_r$) repräsentiert die Klasse d. Die *Inzidenzrelation* r = {(g,h)∈ c×d| g °r h} repräsentiert den *Relationship-Type* r.

Anmerkungen zur mittleren Ebene:

(i) Diese „mittlere Ebene“ sehe ich als eine „Verfeinerung“ an, weil hier als „Mitteldinge“ zwischen dem Bereich der Klassen/Attribute und dem Bereich der Instanzen/Attributwerte die (aus Instanzen bestehenden) Umfänge und Inhalte bezüglich der ***Relationship-Types*** untersucht werden können.

(ii) Wie schon in **Kap.3.4** erwähnt, sind in dieser mittleren Ebene hauptsächlich die formalen Kontexte der „***maximalen***“ Relationship-Types relevant. Die Kontexte der andern Unter-Relationship-Types werden daraus als „Teilkontexte“ durch **Vererbung** abgeleitet.

(iii) Diese mittlere Ebene kann man sogar durch ***einen einzigen*** formalen Kontext K(**i**) := ($\mathbf{F}_{(RELT)}$, **RELT**, **i**) darstellen, wobei $\mathbf{F}_{(RELT)}$ die ***Vereinigung*** aller Begriffsverbände **B**(r) (r∈ RELT) ist, und die Inzidenzrelation **i** einfach definiert ist durch (A,B) **i** r :⇔ (A,B)∈ **B**(**r**) für (A,B)∈ $\mathbf{F}_{(RELT)}$, r∈ **RELT**. K(**i**) nenne ich einen *„F-Compound Context“;* die formalen Begriffe dazu nenne ich *„F-Compounds“*.

Vgl. *Lübbert* [18]. Sowohl das System der ($\mathbf{F}_{(RELT)}$**,** $\leq_F$) der „Gegenstande“ als auch das System (RELT, $\leq_{RELT}$) der „Merkmale“ sind in natürlicher Weise geordnet. Die Ordnung $\leq_F$ ist durch die Ordnungen $\leq_r$ (r$\in$ **RELT**) der Begriffsverbände ($\underline{\mathbf{B}}(\underline{r})$, $\leq$r) induziert; die Ordnung $\leq_{RELT}$ ist die oben beschriebene **Relationship-Taxonomie**. Der F-Compound Context und sein F-Compound-Begriffsverband dienen hauptsächlich dazu, sich klar zu machen, dass nicht nur einzelne Instanzenpaare (x, y) sondern auch formale Begriffe (A, B)$\in$ $\mathbf{F}_{(RELT)}$ zu *verschiedenen* Relationship-Types r, s $\in$ **RELT** in Beziehung stehen können.

- Die **unterste Ebene** bildet, wie gehabt, die **Instanziierung**. Diese wiederum wird gemäß FBA dargestellt durch den ***mehrwertigen Kontext***

  K(**I***) **= (G, *A*,** W, $\underline{\mathbf{I}}$*) oder auch durch den daraus gebildeten „einfachen“ formalen Kontext K(**I****) = (**G**, W, **I****) und dessen Begriffsverband ($\underline{\mathbf{B}}(\underline{\mathbf{I}}^{**})$, $\leq_{I^{**}}$).

5 Schlussbemerkung

Ich habe hier zu schildern versucht, dass die ***Struktur*** des (mir bekannten) informatischen Konzepts „Ontology“ symmetrischer und mathematisch präziser mit Hilfe der **FBA-Sprache** formuliert werden kann, wodurch die Implementierungspraxis der Informatiker eine wohldefinierte mathematische Basis erhält. Mit Kenntnis dieser Basis darf der Implementierer dann im Bedarfsfalle von der mathematischen Grundstruktur abweichen, weiß aber dann genau um die Probleme, die eventuell durch diese Abweichung entstehen können.

Ich meine, Informatiker könnten sich besser über die verschiedenen Varianten ihres Konzepts „Ontology“ einigen – und wären darüber hinaus „auf der sicheren Seite“ in der Implementierungspraxis – wenn sie sich etwas ernsthafter die dazugehörigen ***mathematischen Grundlagen*** zu Gemüte führen und außerdem genauer zwischen „***Struktur***“ und „***Deutung***“ (Anwendungsmöglichkeiten) ***unterscheiden*** würden, als wenn sie sich nur an alten philosophischen und „metaphysik-geschwängerten“ Anschauungen aus Antike und Mittelalter orientieren.

Über das in dieser Notiz Angedeutete hinaus gibt es aber noch mehrere andere Möglichkeiten, wie man die konventionelle Definition eines Konzepts „Ontology“ mit **FBA**-Methoden „verfeinern“ oder „erweitern“ könnte. Insbesondere bietet sich mit FBA-Methoden der Entwurf einer „Ontology“ an, wo die Relationship-Types nicht nur binär sondern auch ***mehr als 2-stellig*** sein dürfen (also mehr als nur je zwei Klassen betreffen) – vgl. etwa [6], [7], [28]. Darauf kann ich aber der Kürze halber hier nicht weiter eingehen.

6 Literatur

[1] **B. Ganter / R. Wille: „Formale Begriffsanalyse“ (FBA);** Springer 1996.
[Die am meisten benutzte Referenz]

[2] G. Pickert: „Einführung in Ontologien“, Humbold-Universität Berlin, Feb. 2011. http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0203/SemWeb/artikel/2/Pickert_Ontologien_final.pdf
(diente zusammen mit [12] als Motivation zur Abfassung dieser Note)

[3] W. Bartussek, B. Humm, A. Reibold, T. Zeh (Ontologie-Arbeitskreis der Hochschule Darmstadt): „Zur Definition von ‚Ontologie’ in den Informationswissenschaften“, Oktober 2010 / Mai 2011, Hochschule Darmstadt (h_da).

[4] Autorenkollektiv der Ontologie-Arbeitsgruppe der Hochschule Darmstadt (h_da): „Was ist eigentlich Ontologie“, Jan.2012.

[6] T. Zeh: „Darstellung von Texten in den drei Formalismen Power Context Families (PCF), Relational Data System (RDS) und Entity Relational System (ERS) anhand des Fallbeispiels „Biographie von Albert Einstein“, Hochschule Darmstadt, 13.10.2010.

[7] K. E. Wolff: „Relational Semantic Systems, Power Context Families, and Concept Graphs, Hochschule Darmstadt, 2009,
http://www.fbmn.fh-darmstadt.de/home/wolff

[8] K. E. Wolff: „Relational Scaling in Relational Semantic Systems", Hochschule Darmstadt, 2009, http://www.fbmn.fh-darmstadt.de/home/wolff

[9] Wikipedia: „Vererbung (Programmierung)", Version 20.11.2011

[10] Wikipedia: „Ontologie (Informatik)", Version Nov. 2013

[12] A. Maedche & V. Zacharias: „Clustering Ontology-based Metadata in the Semantic Web", FZI Research Center for Information Technologies at the University of Karlsruhe, Research Group WIM, 2002/2003 (?)
(diente zusammen mit [2] als Motivation zur Abfassung dieser Note)

[13] Nicola Guarino: „Formal Ontology, Conceptual Analysis and Knowledge Representation", 1995 (?)

[14] John F. Sowa: "KR Ontology", http://www.jfsowa.com/ontology/ , *last modified*: Nov.2010

[16] M. Kreuzer / S. Kühling: „Logik für Informatiker". Vlg. Pearson Studium, 2006.

[17] C. Lübbert: „Mehrwertige Logiken", V11, Dez.2011, www.cl-diesunddas.de

[18] C. Lübbert: „Ontologie-Definition auf Basis von FBA", V3.14, Darmstadt, Aug. 2012

[19] H. Herre et.al.: „General Formal Ontology (GFO)“, Part I, Basic Principles. Leipzig, July 2006. www.onto-med.de

[20] Barry Smith: „Against Fantology“, (in Experience and Analysis), Vienna, 2005

[25] Stanford Encyclopedia of Philosophy: „Mereology“, 2003 / 2009. http://plato.stanford.edu/entries/mereology/

[26] Stanford Encyclopedia of Philosophy: “States of Affaires”, März 2012 http://plato.stanford.edu/entries/states-of-affairs/

[28] R. Wille: „Conceptual Structures of Multicontexts“, FB Mathematik, TU Darmstadt, 1996

Mit freundlichem Gruß

Christoph Lübbert