Lectures on Universal Logic

Lecture 11 – A Universal Natural Deduction System

This Lecture presents the universal logic **UL4**.

I use mostly standard **notation**: ∀ ∃ ¬ ∧ ∨ ∩ ∪ ⊃ ⊂ ⊆ ∈ ∉ ≠ ≡ ...

"→" is used for a conditional stronger than "⊃". "⇒" is used at the meta-level (e.g., in stating rules).

"A", "B"... are meta-language expressions for object language formulas.

"P( )", "P_2( )" … are meta-language expressions for object language general terms: "F( )" …

"á", "é" are meta-language expressions for object language singular terms: "a"…

Capital Greek letters mostly denote sets (of sentences). "Σ" is used for sums. "Δ" is used as truth operator for "true only", "∇" for "false only", "°A" for "A is consistent", "•A" for "A is inconsistent".

If the difference between syntax and semantics is important "proof system/theory" as usual means the systematisation of the relation of derivability, ⊢ , in contrast to the semantic consequence relation, ⊨. "⊢" and "⊨" are used with subscripts if a distinction between logics is needed in the context.

By *proven equivalents* A and B it is meant: ⊢A ≡ B.

By *provably equivalent* in contrast it is meant that A⊢B and B⊢A. Only replacement of provably equivalent formulas holds in **UL4**.

Proofs are often given in a **natural deduction** format without indenting or two dimensional rules.

Example: "< >" notes the assumptions a line depends on. We note the result of a (vertical) derivation in a (horizontal) formula by putting the assumptions mentioned in the dependency set of the last derived line on the left of "⊢".

1.<1>	A	PREM
2.<2>	A ⊃ B	PREM
3.<1.2>	B	(⊃E) 1,2

that is A, A⊃B ⊢ B.

The most simple case shows the identity/idempotency property of "$\vdash$" and "$\vDash$" respectively:

1.<1> A PREM

yields by just stopping here $A \vdash A$, since the first line is at the same time the last line.

The logic used here is an **Adaptive Logic** [cf. Lecture 10]. Retracting in the process of reasoning from a premise set cannot be completely avoided, since there is no general algorithmic procedure (for any logic) to test whether Γ or $\Gamma \cup \{A\}$ is consistent. So we often extend our premise set Γ by a new assumption on the supposition that this extension is consistent, although it sometimes turns out not to be.

Especially if $\Gamma \vdash A$ *depends* on $\Gamma \nvdash C$ and no negative test is available for $\Gamma \nvdash C$, then we have even no positive test for $\Gamma \vdash A$.

(The derivability of A may depend on the absence of C if say C states some exceptional condition on employing some rule to derive A.)

Retraction is of most interest with respect to internal dynamics, since given one and the same premise set the sentence A may be derivable at some stage and retracted later. A then might not belong to the final consequence set, but it appeared to during some stages of the reasoning process. An adaptive logic is characterized by: The *Upper Limit Logic* (ULL) allows for the unrestricted application of logical rules to derive the most consequences possible. Typically ULL is (standard) FOL. The *Lower Limit Logic* (LLL) is chosen to model some type of restricted reasoning. In our case it is a paraconsistent logic, i.e. a logic that blocks the application of some rules of standard logic. The *adaptive strategy* is the way to handle the management of restrictions and the corresponding retractions. An adaptive logic generates a set of consequences of an inconsistent Π that can lay between $Con_{LLL}(\Pi)$ and $Con_{ULL}(\Pi)$. The idea of adaptation is therefore: Think of some critical rules as applicable and make exceptions *only* if one of the premises is known to be inconsistent (or in some other ways to be explained below problematic). Since we do not know beforehand which premises *are* consistent, we may employ these rules incorrectly. That is why there is retraction. The application of that rule is retracted then. All consequences of that application are retracted as well. Given a premise set Γ one likes to know which of them may be abnormalities. *Abnormalities* here are, e.g., formulas of the form $A \wedge \neg A$. Some premise sets might be such that we know:

(1) $(p \wedge \neg p) \vee (q \wedge \neg q)$

whereas neither disjunct is a consequence (so far). So maybe each of them or either "p" or "q" behaves abnormally. "Dab(Π)" abbreviates the disjunction of $(A \wedge \neg A)$ for all $A \in \Delta$. "Dab(Π)" then expresses that at least one of the premises in Δ is abnormal. "Dab" means "disjunction of abnormalities". We are looking for *minimal* Dab-formulas (since the less disjuncts a Dab-formula has the more premises we have excluded as suspects). *Besides* the formulas occurring in a Dab-formula there might be formulas which are *already* known as being abnormal.

In general:

$$\Gamma \vdash_{LLL}(A \vee Dab(\Pi)) \text{ iff } \Gamma \vdash_{ULL} A$$

Here Π contains the formulas on the consistency of which the application of some rules used in deriving A depends. One can follow the *Minimal Abnormality* strategy, which with respect to (1), for example, assumes that once we consider the one abnormal we can take the other as normal (i.e., we can derive more consequences, since less exceptions are now operative). If at some later stage in a proof one can derive one of the disjuncts in (1) [in general in: Dab(Π)], then (1) [or Dab(Π)] is no longer *minimal*. So this Dab-formula is replaced by one stating that derived inconsistency. Retractions based on the supposed inconsistency of one of the other disjuncts are taken back then (by *marking/unmarking* lines in the proof, see below).

Proofs look like Natural Deduction Proofs with a further column:

n.<k,...> A Rule, m, l {B}

We number the lines and include in "< >" the premises a line depends on, then follows the formula, then a column naming the rule applied to get this line and the lines used in that application. The fifth column contains the set of formulas (possibly empty) on the consistency of which the derivability of the formula depends. These sets are called "conditions" (or "presuppositions", see below).

Conditions obey the following abstract rules:

(RU) If $A_1 ... A_n \vdash_{LLL} B$, then from $A_1 ... A_n$ on the conditions $\Pi_1 ... \Pi_n$ derive B on the condition $\Pi_1 \cup ... \cup \Pi_n$.

The rule (RU) concerns rules of Natural Deduction which do not require in LLL the consistency of the ingredient formulas. B just inherits the conditional dependencies.

Rules requiring such consistency operate on

(RC) If $A_1 ... A_n \vdash_{LLL}(B \vee Dab(\Pi_m))$, then from $A_1 ... A_n$ on the conditions $\Pi_1 ... \Pi_n$ derive B on the condition $\Pi_m \cup \Pi_1 \cup ... \cup \Pi_n$.

In this case consistency assumptions for the formulas in Π_m are added. The last line of a proof is the stage that the proof has arrived at. Now, if one of the formulas in the condition gets to be known as non fulfilling the essential criterion (here: consistency) the line is *marked*. The marking rule of the Minimal Abnormality strategy says roughly: If for A $\in \Pi_i$, A occurs in some Dab-formula, then line *i* is *not* marked because of that Dab-formula if there is another disjunct of that Dab-formula which is taken as unreliable. Lines that depend on a marked line have inherited the condition by either (RU) or (RC) and are, therefore, marked as well. Depending on the strategy – or the premise set – a line can get *unmarked* later, even in case of the reliabilist strategy.

Proof **Example**

Instead of simply writing "(RU)", "(RC)" the detailed rules are given.

1.<1>	$\neg p \wedge r$	PREM	$\emptyset$	
2.<2>	$q \supset p$	PREM	$\emptyset$	
3.<3>	$s \vee \neg r$	PREM	$\emptyset$	
4.<4>	$r \supset p$	PREM	$\emptyset$	
5.<5>	$p \vee \neg r$	PREM	$\emptyset$	
6.<1>	$\neg p$	$\wedge$E, 1	$\emptyset$	(RU)
7.<1>	r	$\wedge$E, 1	$\emptyset$	(RU)
8.<1,2>	$\neg q$	Contrap., 6, 2	{p}	(RC) marked at 10
9.<1,3>	s	$\vee$E, 3, 7	{r}	(RC) if marked at 10 unmarked at 11
10.<1,5>	$(\neg p \wedge p) \vee (\neg r \wedge r)$	Dilemma, $\wedge$I, 5, 6, 7	$\emptyset$	(RU)
11.<1,4>	$\neg p \wedge p$	$\supset$E, 4, 7	{r}	(RC)

In line 10 we get to know that at least one of "r" and "p" is inconsistent, so lines depending on them get marked. Given a Minimal Abnormality strategy *or* seeing in line 11 that "p" is inconsistent we can blame "p" for line 10 and unmark the lines depending on the consistency of "r". The Dab-formula in 10 is no longer *minimal* after 11. Given the dynamic character of the proofs one has to distinguish: derivability at some stage and final derivability. A is *finally derived* at line *i* of a proof at a stage *s* iff line *i* is unmarked at *s*, and whenever line *i* is marked in an extension of the proof, then there is a further extension in which line *i* is not marked. This property is (in most cases) not recursive. Even if final derivability is not recursive this resembles our actual reasoning where we (mostly) lack similar assurance against revision. There is *nothing* dynamic about *final* derivability. The *relative derivability statements*, however, (i.e. those statements like

$$\vdash((\neg p \wedge r)\wedge(q \supset p)\wedge(s \vee \neg r)\wedge(p \vee \neg r)) \supset s \qquad \textit{given } \{°r\}$$

expressing that something is derivable from a (empty) set of premises on the given set of presuppositions) *are* recursive enumerable. So one should not exaggerate the failure of enumerability of theorems!

In **Universal Natural Deduction** , working with the logic **UL4**, we take the outlined format of natural deduction, especially we use the fifth column to give set of formulas *presupposed* to be *true* (to apply the restricted rules, see below), and if needed the marked individual constants for the quantifier rules. We built this system by using the following ingredients:

- an adaptive version of standard **PC**;
- semantic and consistency operators (like in LFI-systems, "°" expresses consistency);
- quantifier rules in the vain of *Minimal Free Description Theory* (allowing for the use of descriptions, including those that are non-referring);
- identity rules that restrict substitution to consistent objects;
- rules for a stronger conditional, whether we really need this or not;
- basic rules for modalities.

The system thus is dynamic and maybe incomplete; but it is a truly universal logic. We do not follow the strict format of natural deduction in giving *only* the meaning of all connectives in introduction and elimination rules. The benefit of using **LP** as our fall-back paraconsistent logic is that all standard tautologies are **LP** valid. So we can simply include all **PC** tautologies. **UL4** thus keeps theorems and consequences apart: We have a lot of the standard theorems, but that does not mean that one can detach (such) conditionals in general. *Reasoning* thus is paraconsistent and to some extent Relevant, whereas *logical truth* need not be. A benefit of this approach is that we keep besides the introduction and elimination rules of the connectives a "real" semantics with truth conditions for complex formula. A universally quantified conditional as used in science thus still can be understood as asserting something about reality, and not as a hidden statement on assertability. The semantics thus kept preserves the intuitive understanding of the connectives and quantifiers.

Rules for derivations:

A line that reads

n.< > A

contains a *theorem*, since the sentence A does not depend on any assumption (the dependency set noted within "< >" is empty). Theorems can be introduced into derivations at any time. [Letters "n", "m" etc. are used to refer to unspecified line numbers. Remember: "A" is a schematic letter, the object language having sentences like "p", "F(a)" etc.]
To include **PC**-tautologies, which we know already, we have the rule:

(PC) n.< > A PC ∅

where A is *any* **PC**-theorem. The column with markings is empty.
For any other theorems (i.e. already proven **UL4**-theorems) we have:

(TH) n.< > A TH Γ

where A is any **UL4**-theorem. Γ contains the presuppositions. There cannot be a list of marked individual constants in theorems.

To introduce *assumptions* into a derivation we have the rule:

(AE) n.<n> A AE {sat(A)}

where we define satisfiability presuppositions by the schema

(sat) $\mathrm{sat}(A) \equiv \neg((\Delta A \wedge FA) \vee (\nabla A \wedge TA) \vee (\bullet A \wedge {}^{\circ}A))$

In case the presupposition later turns out to be violated lines depending on the assumption in question have to be retracted (as always). The satisfiability presupposition has to be made because the definition of **UL4**-consequences excludes the cases in which the premise set is unsatisfiable [see below].
(Typically assumption in arguments need not be considered true, but satisfiable at least.)

Conjunction Introduction has the form:

n.<m>	A	...	Γ
o.<k>	B	...	Λ
p.<m,k>	A ∧ B	(∧I) n, o	Γ ∪ Λ

Conjunction Elimination has the two forms:

n.<m> A ∧ B	...	Γ
o.<m> A	(∧E) n	Γ
n.<m> A ∧ B	...	Γ
o.<m> B	(∧E) n	Γ

[Note: Here and in the following rules "<m>" refers to an unspecific (number) of assumptions that the line depends on. Γ can, of course, be empty; if there are marked individual constants they are marked only in the line where the quantificational rule is employed, see below.]

Disjunction Introduction has the two forms:

n.<m> A	...	Γ
o.<m> A ∨ B	(∨I) n	Γ
n.<m> A	...	Γ
o.<m> B ∨ A	(∨I) n	Γ

Disjunction Elimination has the form:

n.<m>	A ∨ B	...	Γ
o.<k>	¬A	...	Λ
p.<m,k>	B	(∨E) n,o	Γ ∪ Λ ∪ {°A}

This is the restricted form of Disjunctive Syllogism.

Negation Introduction has the form:

n.<n>	A	AE	$\varnothing$
o.<m,n>	$\neg$A	...	Γ
p.<m>	$\neg$A	($\neg$I) n,o	Γ

If some assumption allows to derive its own negation, then this sentence can be stated negated *simpliciter* (i.e. the status as assumption is discharged, as indicated by the underlining in the line using ($\neg$I)). The usual form of Negation Introduction leads to trivialization in inconsistent contexts.

Negation Elimination has the form:

n.<m>	$\neg\neg$A	...	Γ
o.<m>	A	($\neg$E) n	Γ

Conditional Introduction (*Conditionalization*) has the form:

n.<n>	A	AE	$\varnothing$
o.<m,n>	B	...	Γ
p.<m>	A $\supset$ B	($\supset$I) n,o	Γ

This rule mirrors the Deduction Theorem. If the conditionalization is the last step of a derivation the restrictions on not having marked individual constants in it have to be kept.

Conditional Elimination (*Modus Ponens*) has the form:

n.<m>	A $\supset$ B ...		Γ
o.<k>	A	...	Λ
p.<m,k>	B	($\supset$E) n,o	$\Gamma \cup \Lambda \cup \{°A\}$

This is the restricted form of *Modus Ponens*.

Truth Introduction/Elimination follow the disquotational (T)-schema.

Strict Truth/Falsity will be defined notions. The Inconsistency operator is treated by rules as well.

Truth Introduction has the form:

n.<m>	A	...	Γ
o.<m>	TA	(TI) n	Γ

Truth Elimination is the converse:

n.<m>	TA	...	Γ
o.<m>	A	(TE) n	Γ

Falsity Introduction has the form:

n.<m>	$\neg A$	...	Γ
o.<m>	FA	(FI) n	Γ

Falsity Elimination is the converse:

n.<m>	FA	...	Γ
o.<m>	$\neg A$	(FE) n	Γ

Inconsistency Introduction has the form:

n.<m>	$A \wedge \neg A$	...	Γ
o.<m>	$^{\bullet}A$	($^{\bullet}$I) n	Γ

Inconsistency Elimination is the converse:

n.<m>	$^{\bullet}A$	...	Γ
o.<m>	$A \wedge \neg A \wedge \neg {^{\circ}A}$	($^{\bullet}$E) n	Γ

For simple truth we introduce its version of Convention (T):

n.<>	$\Delta A \equiv A$	(Δ)	$\varnothing$

Necessity Introduction (*Necessitation*) has the form:

n.< >	A	...	Γ
o.< >	$\Box A$	($\Box$I) n	Γ

A *theorem* (but not any sentence depending on further assumptions) can be necessitated.
Necessity Elimination has the form:

n.<m>	$\Box A$	...	Γ
o.<m>	A	...	Γ

Since necessity is taken here to be semantic necessity (not natural necessity or some more restricted version of necessity) it has to governed in the way of a normal modal logic of the strength of modal system **S5**. Therefore we need two further rules:
The rule corresponding to the K-Axiom of modal logic has the form:

n.<m>	$\Box(A \supset B)$	...	Γ
o.<k >	$\Box A \supset \Box B$	(K) n,o	Γ

The rule corresponding to the **S5-**Axiom has the form:

n.<m>	$\Diamond A$	...	Γ
o.<m>	$\Box\Diamond A$	(S5) n	Γ

Taking entailment to be *semantic entailment* in the sense that:

$$A \rightarrow B \Leftrightarrow_{UL4} \Box(A \supset B)$$

gives us *derived* introduction and elimination rules for "$\rightarrow$".
Entailment Introduction is a strict form of Conditionalization:

n.<n>	A	AE	$\varnothing$
o.<o>	B	AE	$\varnothing$
…			
r.<n,o>	C	...	Γ
s.< >	$A \wedge B \to C$	$(\to I)$ n,o,r	Γ

In strict conditionalization *all* assumptions have to be conditionalized (thus we get a *theorem* to be necessitated to yield the entailment).

Entailment Elimination is a version of *Modus Ponens*:

n.<m>	$A \to C$	...	Γ
o.<k>	A	...	Λ
p.<m,k>	C	$(\to E)$ n,o	$\Gamma \cup \Lambda \cup \{°A\}$

We introduce some further connectives by definitions. There are derivable introduction and elimination rules then. Within a derivation we use the definitions by referring to their name:

$(D\equiv)$	$A \equiv B \stackrel{\text{def}}{\equiv} (A \supset B) \wedge (B \supset A)$
$(D\leftrightarrow)$	$A \leftrightarrow B \stackrel{\text{def}}{\equiv} (A \to B) \wedge (B \to A)$
$(D\Diamond)$	$\Diamond A \stackrel{\text{def}}{\equiv} \neg\Box\neg A$
$(D\nabla)$	$\nabla A \stackrel{\text{def}}{\equiv} \Delta\neg A$
$(D°)$	$°A \stackrel{\text{def}}{\equiv} \Delta A \vee \nabla A$

We have to give the usual requirements on marking individual terms in case of applying Universal Generalization or Existential Specialization within a derivation. These are:

> Terms generalized in Universal Generalization and specialized to in Existential Specialization are *marked* at the *right* of such a line;
>
> - the marking also notes the dependencies on other individual terms in that line (in the form "a(e)" "a" being marked depended on "e");
> - markings may not be circular (i.e. we do not have "a(e)" *and* "e(a)");
> - no term may be marked twice;
> - marked terms may neither occur in the premises nor in the conclusion of a supposed valid derivation.
> - In applications of the quantifier rules one also has to meet the requirement that by generalising one constant to a variable "x", "x" will not be bound by already otherwise existing quantifiers. (∀I) and (∃I) require further on that "x" and the individual term occur at *exactly* the same places in a given sentence.

E!(á) says that the object denoted by á exists, ("E!( )" being the existence predicate is a logical constant.) Quantifiers refer to existing objects only. We assume that there *is* something:

n.< >	(∃x)E!(x)	(E!)	∅

[Note: Semantically speaking the extension of "E!( )" is the domain *at* a world index.]

Identity Introduction is valid for any object, existing or not:

n.< >	a = a	(=I)	∅

Identity Elimination (i.e. substitution of identicals) is more critical. It has to be restricted to avoid trivialization in a paraconsistent logic with as much expressive power as **UL4**. We have to presuppose that some object is not an inconsistent object to apply (=E) to it. We define a consistency predicate "K( )" for objects (as a logical constant, of course) to do this:

(DK) $K(á) \stackrel{def}{=} \neg(\exists P)(P(á) \wedge \neg P(á))$

UL4 so far is no 2nd order system, but we may employ (DK) in that we note $\neg K(á)$ in some line if for the object named á we could have a line with an instance of the scheme: $P(á) \wedge \neg P(á)$.

Identity Elimination then takes the form:

n.<m>	$P(á)$	...	Γ
o.<k>	$á = é$	...	Λ
p.<m,k>	$P(é)$	(=E) n,o	$\Gamma \cup \Lambda \cup \{K(é)\}$

Since we want to use description and modal operators we have to provide (=E) with a *provisio* in case descriptions are involved. In modal logic **S5** all modalities can be reduced to modalities of degree 1. We require as a *provisio* for Identity Elimination:

In case we have á = é, then:

(i) if á is a description and é an individual constant, é cannot be substituted into a modal context of "$\Diamond$",

(ii) if á is an individual constant and é a description, é cannot be substituted into a modal context of "$\Box$".

The following quantifier rules require following the rules of marking the constant generalized/specialized in ($\forall$I) and ($\exists$E), and the renaming of variables mentioned before.

$\forall$-Introduction (Universal Generalization) has the form:

n.<m>	$R(á,é)$	...	Γ	
o.<m>	$(\forall x)R(x,é)$	($\forall$I),n	$\Gamma \cup \{E!(á)\}$	á(é)

Thus the application of ($\forall$I) requires an existence assumption concerning á, since we conclude to a generalization about all *existing* objects. á is marked, here as depending on é.

∀-Elimination (Universal Instantiation) has the form:

n.<m>	(∀x)P(x)	...	Γ
o.<m>	P(é)	(∀E),n	Γ ∪ {E!(é)}

Since the generalization is (maybe) true of existing objects only the application of (∀E) presupposes that the constant specialized to names an existing object.

∃-Introduction (Existential Generalization) has the form:

n.<m>	P(á)	...	Γ
o.<m>	(∃x)P(x)	(∃I),n	Γ ∪ {E!(á)}

Thus the application of (∃I) requires an existence assumption concerning á, since we conclude to a generalization about some *existing* objects.

∃-Elimination (Existential Instantiation) has the form:

n.<m>	(∃x)R(x,á)	...	Γ
o.<m>	R(é,á)	(∃E),n	Γ ∪ {E!(é)} é(á)

Since the generalization is (maybe) true of existing objects only the application of (∃E) presupposes that the constant specialized to names an existing object. The name of the object is marked in its dependencies in the formula in question.
In case that existence assumptions are *explicitly* made the existence *presupposition* can be cancelled:

n.<m>	P(á)	...	Γ ∪ {E!(á)}
o.<o>	E!(á)	AE	
q.<m,o>	P(á)	(E!C),n,o	Γ

If the existence claim follows from the other assumptions the presupposition can be cancelled as well:

n.<m>	$P(\acute{a})$	...	$\Gamma \cup \{E!(\acute{a})\}$
o.<m>	$E!(\acute{a})$	…	$\Gamma \cup \{E!(\acute{a})\}$
q.<m>	$P(\acute{a})$	(E!C),n,o	Γ

Minimal Free Description Theory requires the uniqueness of a description with respect to the existing objects only. Otherwise it looks like the standard Russellian account of descriptions. We use the usual "ι"-notation, so that "ιxF(x)" means "*the* (unique) F".

The (MFD)-rule can be stated as the following two ways of interchangability:

n.<m>	$\iota x P(x) = \acute{a}$	...	Γ
o.<m>	$(\forall y)(\acute{a}=y \equiv P(y) \wedge (\forall z)(P(z) \supset z=y)$	(MFD),n	Γ
n.<m>	$(\forall y)(\acute{a}=y \equiv P(y) \wedge (\forall z)(P(z) \supset z=y)$	...	Γ
o.<m>	$\iota x P(x) = \acute{a}$	(MFD),n	Γ

The first conjunct in the equivalence states satisfaction of the defining property, the second expresses uniqueness.

In the context of quantificational rules we can now make clear the reference to a set of *presuppositions* above. Adaptive Logics speak of Dab-formula and corresponding sets of consistency assumptions. **UL4** notes these consistency assumptions as presuppositions to employ some restricted rules. Actually the consistency presupposition is "°A". In *Minimal Free Description Theory* usually a conjunct "E!(a)" is needed (e.g. as derivable line or *assumption*) to employ one of the quantifier rules. Since **UL4** is a dynamic logic already we need not work with "E!(a)" as a line in a derivation, but can note this also as a presupposition in the presupposition set Γ noted on the right. In case of Identity Elimination the presupposition is that we have a consistent object. We note this as the presupposition "K(a)" for an object a in question. Each of the sentences in the presupposition set has a negation. Once the negation of such a presupposition can be derived, all lines are retracted which depend on that presupposition (like in the original adaptive dynamics). The retraction thus

does not only concern the disappointment of consistency assumptions (either for a sentence or an object), but also the disappointment of existence presuppositions. If the last line A of a derivation has a non-empty presupposition set Γ, this means that the sentence in that line is derivable from the assumptions noted within "< >" given these further presuppositions.
Let Φ be the (possibly empty) set of assumptions and Γ the (possibly empty) set of presuppositions.
We have:

$$\neg(\exists B \in \Gamma)\ \Phi \vdash_{\mathbf{UL4}} \neg B \Rightarrow \Phi \vdash_{\mathbf{UL4}} A$$

To save labour and have derivation looking more closely like standard derivations we adopt the convention to drop noting Γ if Γ is empty.[1]
Relative derivability is recursive enumerable.
The *relative derivability statements*, i.e. those statements like

$$\vdash_{UL4} G(\iota x F(x)) \supset (a = \iota x F(x) \supset G(a)) \qquad \textit{given}\ \{K(a)\}$$

expressing that something is derivable from a (empty) set of premises on the given set of presuppositions, *are* recursive enumerable.

Consequence in **UL4** may be defined:

($\vDash_1$) $\Gamma \vDash_{UL4} A$ iff

in case that *all* $B \in \Gamma$ are true at least, then A is *true at least*.

Nothing needs to be said concerning the case that any $B \in \Gamma$ is false *only*. One has not to hold that then a consequence relationship holds. To do so would be Irrelevant.
To do so may come close to reintroducing Explosion, as well. ΔA and ∇A are incompatible, so both can never be true at the same time, so allowing for Irrelevant consequences would yield, for example:

[1] It may also be convenient to leave most presuppositions, especially satisfiability presuppositions, as being understood and return to them only in case of cancellation of lines. For the sake of getting to see all the presuppositions or to accustom to the adaptive procedures it may be useful to write them down for a while.

(**) $\nabla A, TA \vDash_{UL4} C$

for any C.

To insist that the “in case” has to be read as material implication as in **PC** just begs the questions against a Relevant meta-theory!

An improved Relevant definition of consequence in **UL4** might be:

($\vDash_2$) $\Gamma \vDash_{UL4} A$ iff *there are* models such that all $B \in \Gamma$ are true at least, and in case that all $B \in \Gamma$ are true at least in a model, then A is true at least in that model.

The existence condition rules out the Irrelevant cases and (**).

A consequence relation obtains if and only if all of the non-empty set of models that make the premises at least true make the consequence at least true.

As always $\vDash$ concerns the inheritance of truth. The second version, ($\vDash_2$), requires some reworking of the proof theory.

Changing the definition of consequence this way requires a further book keeping of presuppositions, in this case with respect to assumptions.

In as much as **UL4** has to be *correct* the basic rules must not support consequence claims that go against the definition above. Making an unsatisfiable assumption, however, would allow claims like

(***) $\nabla A \wedge TA \vDash_{UL4} \nabla A \wedge TA$

(****) $\nabla A \wedge TA \vDash_{UL4} \nabla A$

where the premise (set) is unsatisfiable and thus the claims are *supposedly* incorrect. If we consider these claims as *incorrect* – and not just Irrelevant – then the first definition of consequence is in trouble, since conjunction elimination would allow to derive (****). The proof theory allows to derive something that is not – strictly speaking – a violation of the definition of consequence given thus, but only because we deem it Irrelevant (the case of the assumption on the left being at least true just does not arise, thus it cannot violate the condition). This line of reasoning, however, leads to accepting (**) as *not incorrect*! And this may be too much, even if (**) is not accepted as *valid*. Still the first definition may be an option given a clear understanding of Relevance. Being silent on (**) as neither correct not incorrect, however, violates the otherwise assumed *tertium non datur*, and the meta-theory should not work with another logic than the logic, since a truly universal logic can be used as its own meta-logic.

Clearly, however, the solution for this first option's trouble is straight forward, given the second definition: assumptions (i.e. claims to be considered for further consequences) are presupposed not to be true, but to be *satisfiable*.

When applying the **assumption rule** (**AE**) we have to use the form

n.<n> A AE sat({A})

where we define the satisfiability presuppositions by the schema

(sat) $sat(\Gamma) \equiv \Gamma$ has a **UL4**-model where all $B \in \Gamma$ are true at least

Γ being a set of assumptions.

The set of assumptions Γ has to be *jointly satisfiable*.

With sat({A}) we note only the satisfiability of an *individual* assumption. If a line depends on several assumptions, the further assumptions entering into its derivation also have to enter the set the satisfiability of which is presupposed.

The presupposition of satisfiability is cancelled when Γ contains or entails for some A either

(i) $TA \wedge \nabla A$ or

(ii) $\nabla A \wedge \Delta A$ or

(iii) $\Delta A \wedge FA$ or

(iv) $(\bullet A \wedge {}^{\circ}A)$

In case the presupposition later turns out to be violated lines depending on the assumption in question have to be retracted (as always).

Since we *generally* have to presuppose the satisfiability of the set of assumptions which a line depends on, we may use the convention of not especially noting this in ordinary cases, but proceed according to a revision rule that all lines depending on an assumption that turned out to be unsatisfiable have to be taken back.

The relative derivability statements with respect to logical consequence (i.e. derivability from a set of assumptions) now carry the presupposition that the assumptions/premises are satisfiable (in the defined sense above):

$$p \wedge q \vdash_{UL4} p \text{ given sat}\{(p \wedge q)\}$$

expressing that something is derivable from a satisfiable set of premises.

Examples of **UL4**-derivations:

1.$\langle 1\rangle$	$G(\iota xF(x))$	AE	$\{sat(G(\iota xF(x)))\}$
2.$\langle 2\rangle$	$a = \iota xF(x)$	AE	$\{sat(a = \iota xF(x))\}$
3.$\langle 1,2\rangle$	$G(a)$	(=E)	$\{K(a),sat(1),sat(2)$[2]$\}$
4.$\langle 1\rangle$	$a = \iota xF(x) \supset G(a)$	$(\supset I)\underline{2},3$	$\{K(a),sat(1)\}$
5.$\langle\rangle$	$G(\iota xF(x)) \supset (a = \iota xF(x) \supset G(a))$	$(\supset I)\underline{1},4$	$\{K(a)\}$

1.$\langle 1\rangle$	$a = \iota xF(x)$	AE	$\{sat(1)\}$
2.$\langle 1\rangle$	$(\forall y)(a=y \equiv F(y) \wedge (\forall z)(F(z) \supset z=y)$	MFD,1	$\{sat(1)\}$
3.$\langle 1\rangle$	$a=a \equiv F(a) \wedge (\forall z)(F(z) \supset z=a)$	$(\forall E)2$	$\{E!(a),sat(1)\}$
4.$\langle\rangle$	$a=a$	(=I)	
5.$\langle 1\rangle$	$F(a) \wedge (\forall z)(F(z) \supset z=a)$	$(\supset E)3,4$	$\{E!(a),°(a=a),sat(1)\}$
6.$\langle 1\rangle$	$F(a)$	$(\wedge E)5$	$\{E!(a),°(a=a),sat(1)\}$
7.$\langle 1\rangle$	$(\exists x)F(x)$	$(\exists E)6$	$\{E!(a),°(a=a),sat(1)\}$
8.$\langle\rangle$	$a = \iota xF(x) \supset (\exists x)F(x)$	$(\supset E)\underline{1},7$	$\{E!(a),°(a=a)\}$

[2] “sat(n)“ being an obvious abbreviation for the claim that line *n* is satisfiable.

1.<1>	p	AE	{sat(1)}
2.<>	p ⊃ ¬ ¬ p	PC	
3.<1>	¬ ¬ p	(⊃E)1,2	{°p,sat(1)}
4.<4>	¬p ∨ q	AE	{sat(4)}
5.<1,4>	q	(∨E)3,4	{°p,° ¬ ¬p,sat(1),sat(4)}
6.<4>	p ⊃ q	(⊃I)1,5	{°p,° ¬ ¬p,sat(4)}
7.<>	(¬p ∨ q) ⊃ (p ⊃ q)	(⊃I)2,6	{°p,° ¬ ¬p}

1.<>	◇¬p ⊃ □◇¬p	(S5)(⊃I)	
2.<>	¬□◇¬p ⊃ ¬◇¬p	(PC)(⊃E)1	{°(◇¬p ⊃ □◇¬p)}
3.<>	◇□p ⊃ □p	(D◇)2	{°(◇¬p ⊃ □◇¬p)}
4.<>	□p ⊃ ◇□p	(TH)	
5.<>	□p ⊃ □□p	(⊃E)(⊃I)3,4	{°□p,°(◇¬p⊃□◇¬p)}
6.<6>	□p	AE	{sat(6)}
7.<6>	p	(□E)6	{sat(6)}
8.<6>	p ∨ q	(∨I)7	{sat(6)}
9.< >	□p ⊃ p ∨ q	(⊃I)6,8	
10.< >	□ (□p ⊃ p ∨ q)	(□I)9	
11.<>	□□p ⊃ □(p ∨ q)	(K)10	
12.<>	□p ⊃ □(p ∨ q)	(⊃E)(⊃I) 5,11	{°□p,°(◇¬p⊃□◇¬p)}

Additional rules may concern further connectives or incorporating central concepts of logic, formal ontology or mathematics. We here use abstract entities defined within **set theory**.
We add to the constants of **UL4** the expression "Set( )" with its obvious intended meaning, as well as the usual set theoretical symbols like curly brackets, "∈", "⊆" etc. and "|" to express set abstracts like {x | P(x)}. We have to add the usual definitions like:

(D⊆)	a ⊆ b	$\stackrel{def}{=}$	Set(a) ∧ Set(b) ∧ (∀x)(x∈a ⊃ x∈b)
(D<>)	<a,b>	$\stackrel{def}{=}$	{a,{a,b}}

Thus "∈" is taken as primitive.
U**L4** deals not only with sets, so we need a set predicate "Set( )" to sort out the empty cases with non-sets, i.e. since in the intended interpretation x ∈ y will be false if y is not a set, we have to avoid counterintuitive consequences of Irrelevant conditionals (e.g., (∀x,y)(x∈a ≡ y∈b) is true for any individuals *a* and *b* without these being identical).
We use an introduction rule (**Set**) for some of these cases:

n.<o>	á = {x | P(x)}	…	Γ
m.<o>	Set(á)	(Set),n	Γ

We define the *universal set* V

(DV) V $\stackrel{def}{=}$ {x | x = x}

Because of (=I) this is truly the universal set. Since we cannot exclude inconsistent objects like *a* with *a* ≠ *a* it is no option to define ∅ as {x | x ≠ x}!
A better idea is:

(D∅) ∅ $\stackrel{def}{=}$ {x | x ∉ V}

Extensionality of sets can be added as a rule to introduce identity of sets.

n.<k>	$Set(a) \wedge Set(b) \wedge (\forall x,y)(x \in a \equiv y \in b)$	…	Γ
m.<k>	$a = b$	(Ext),n	Γ

(=E) doing the rest for consistent sets, since both sides of "∈" are open for substitution. Reasoning with inconsistent sets may be impaired by (=E), but it is far from clear whether this is a disadvantage **Naïve Comprehension** is added as rule/axiom schema using the material conditional:

n.< >	$(\exists y)(Set(y) \wedge (\forall x)(x \in y \equiv P(x)))$	(NC)	∅

with no further restrictions. (NC) immediately gives us the *existence* of ∅ and V. Given the restrictions on detachment, **UL4**, however, may forbid or retract the application of detachment here. Consider, for example, the inconsistent object *a* with $a \neq a$. We have

1.< >	$a \in V \equiv a = a$	(NC), (DV), (∀E)	∅
2.< >	$a \in V$	(≡E), (=I), 1	{°a=a}
3.<3>	$a \neq a$	AE	{sat(a≠a)}
4.<3>	$a \neq a \wedge a = a$	(∧E), (=I), 3	{sat(a≠a)}

Now, given (4) "a = a" obviously isn't consistent, so (2) has to be retracted! That, of course, does not mean that $a \notin V$. Singular terms naming sets – including set abstracts! – need not refer by just being singular terms. In principle there would therefore be the option for *non-existent sets* being around. On the other hand (NC) just declares that any set whatsoever (i.e. any set defined by a set abstract) exists.

UL4 so far exhibits the *internal* dynamics of a premise set. One can add the *external* dynamics of **default reasoning** by introducing default-rules, adding their presuppositions to the presupposition set. A non-default connection between two predicates can be expressed by a universally quantified conditionals, like

$$(\forall x)(F(x) \supset G(x))$$

$$\Box(\forall x)(F(x) \supset G(x))$$

which can be added as either assumptions or further axioms/rules.
A default-rule can be given as a rule in which the singular term is schematic and which adds some presupposition to the presupposition set:

n.<m>	Bird(á)	…	Γ
o.<m>	Fly(á)	(dR_i),n	$\Gamma \cup$ { Fly(á) }

or

n.<m>	Bird(á)	…	Γ
o.<m>	Fly(á)	(dR_i),n	$\Gamma \cup$ {¬Penguin(á) }

In the first case the default-rule No. *i* is a *normal* default-rule stating that missing information to the contrary birds fly. In the second case default-rule No. *i* contains specific information on an exception by assuming that we have not the information that the object/animal under consideration is a penguin. As with other retractions we have to retract a consequence drawn under a presupposition in case the negation of the presupposition can be derived (e.g. in the second case in case we can later derive "Penguin(á)").