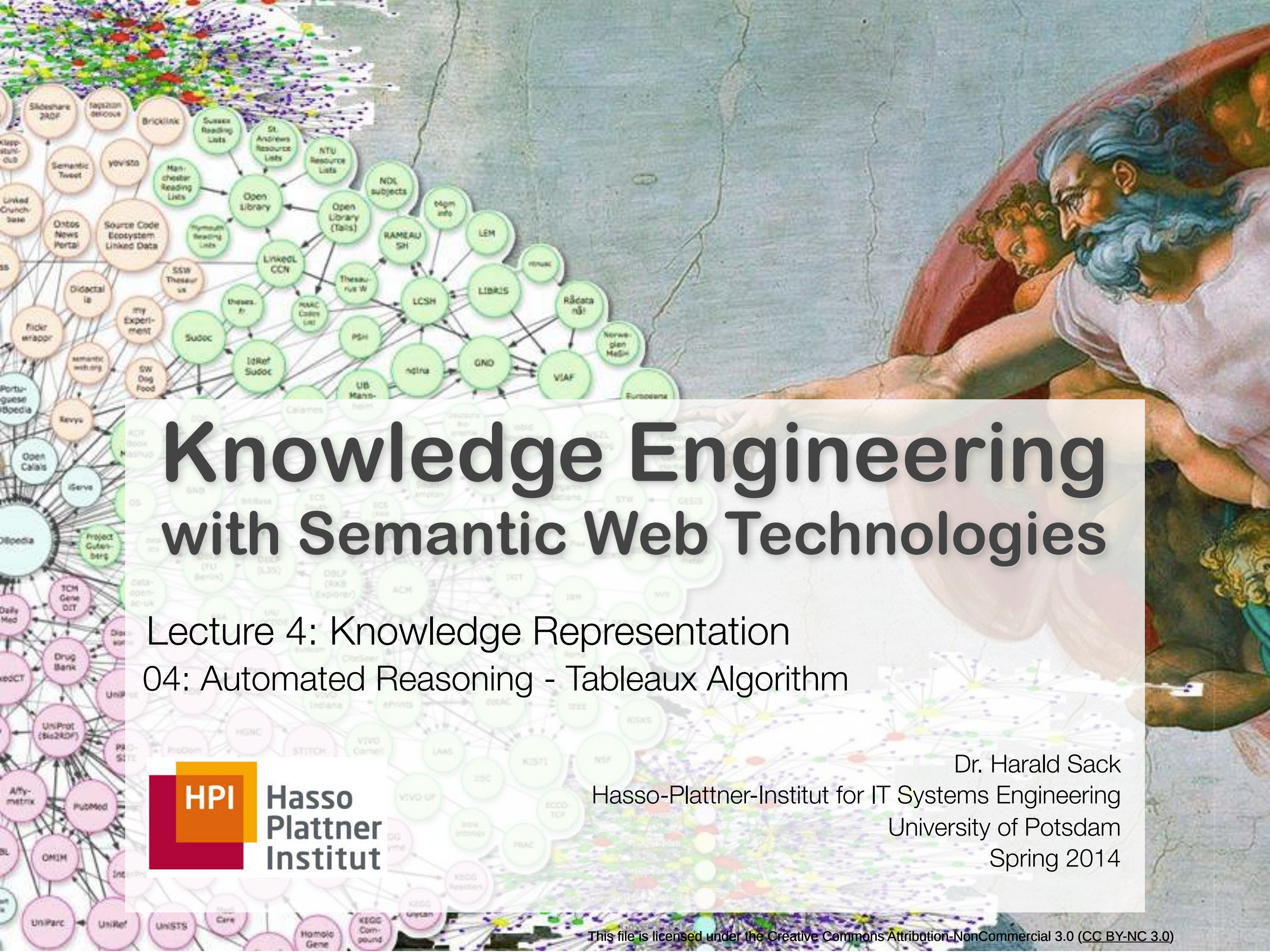


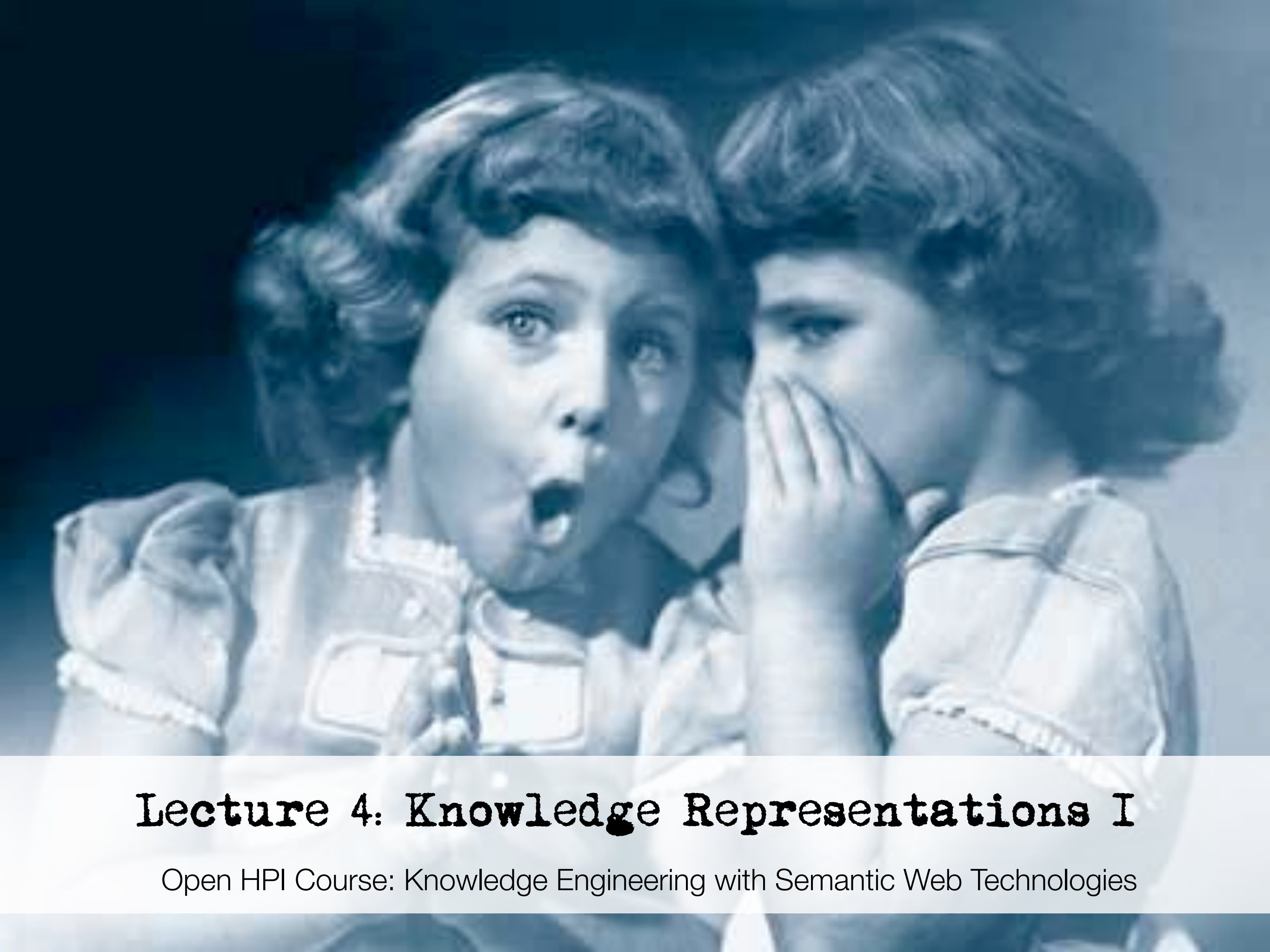


Knowledge Engineering with Semantic Web Technologies

Lecture 4: Knowledge Representation
04: Automated Reasoning - Tableaux Algorithm



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



Lecture 4: Knowledge Representations I

Open HPI Course: Knowledge Engineering with Semantic Web Technologies

3
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

04 Automated Reasoning - Tableau Algorithm

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 04: Knowledge Representations Pt.1

Logical Equivalences



Augustus De Morgan
(1806-1871)

$$F \wedge G \equiv G \wedge F$$

$$F \vee G \equiv G \vee F$$

$$F \rightarrow G \equiv \neg F \vee G$$

$$F \leftrightarrow G \equiv (F \rightarrow G) \wedge (G \rightarrow F)$$

$$\neg(F \wedge G) \equiv \neg F \vee \neg G$$

$$\neg(F \vee G) \equiv \neg F \wedge \neg G$$

DeMorgans' Laws

$$\neg\neg F \equiv F$$

$$F \wedge t \equiv F$$

$$F \wedge f \equiv f$$

$$F \vee t \equiv t$$

$$F \vee f \equiv F$$

$$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$$

$$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$$

$$\neg(\forall X) F \equiv (\exists X) \neg F$$

$$\neg(\exists X) F \equiv (\forall X) \neg F$$

$$(\forall X)(\forall Y) F \equiv (\forall Y)(\forall X) F$$

$$(\exists X)(\exists Y) F \equiv (\exists Y)(\exists X) F$$

$$(\forall X) (F \wedge G) \equiv (\forall X) F \wedge (\forall X) G$$

$$(\exists X) (F \vee G) \equiv (\exists X) F \vee (\exists X) G$$

$$F \wedge \neg F = f$$

$$F \vee \neg F = t$$

Commutativity of Quantifiers

- Quantifiers (of the same sort) are commutative

$$\begin{aligned}(\forall X)(\forall Y) F &\equiv (\forall Y)(\forall X) F \\ (\exists X)(\exists Y) F &\equiv (\exists Y)(\exists X) F\end{aligned}$$

- But

$$(\exists X)(\forall Y) F \not\equiv (\forall Y)(\exists X) F$$

- Example:

- $(\exists x)(\forall y): \text{loves}(x, y)$

„There exists somebody, who loves everybody.“

- $(\forall y)(\exists x): \text{loves}(x, y)$

„Everybody is loved by somebody.“

Tableaux Algorithm

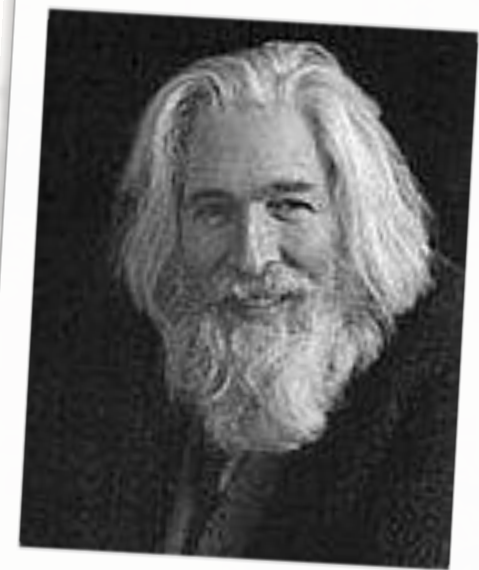


Evert Willem Beth
(1908-1964)

- For **automated reasoning**, we need syntactic algorithms to check the consistency of logical assertions
- The **method of analytical tableaux** is based on disjunctive normal form of logical expressions
 - invented by Dutch logician Evert Willem Beth in 1955 and simplified by Raymond Smullyan.

- **Basic Idea of Tableaux Algorithm:**

- Proof algorithm (decision procedure) to check the consistency of a logical formula by inferring that its negation is a contradiction (***proof by refutation***).



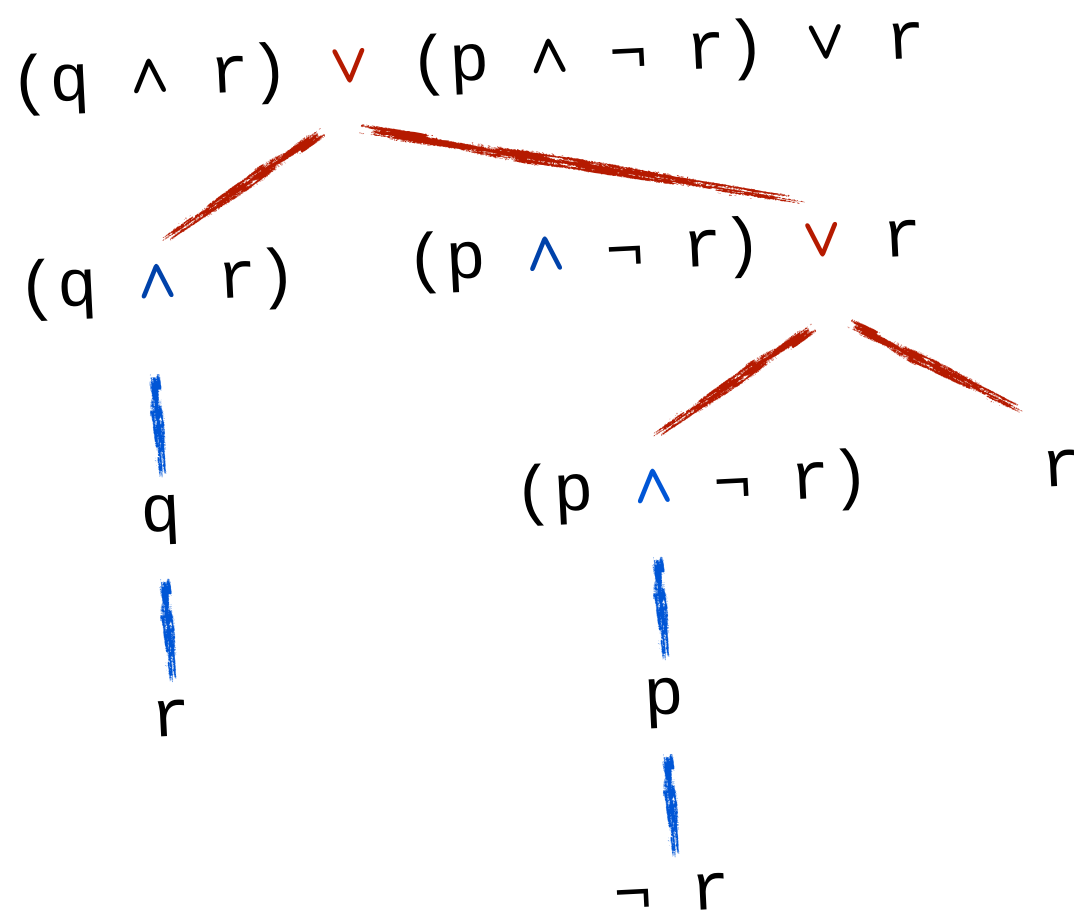
Raymond Merrill Smullyan

Beth, Evert W., 1955. "Semantic entailment and formal derivability", *Mededlingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde*, N.R. Vol 18, no 13, 1955, pp 309–42.

Tableaux Algorithm for PL

(1) Construct **Decision Tree**, where each node is marked with a logical formula.

- A **path from the root to a leaf** is the **conjunction** of all formulas represented within the nodes of the path;
- a **branch of the path** represents a **disjunction**.



Tableaux Algorithm for PL

- ⁸
- (1) Construct **Decision Tree**, where each node is marked with a logical formula.
 - A **path from the root to a leaf** is the **conjunction** of all formulas represented within the nodes of the path;
 - a **branch of the path** represents a **disjunction**.
 - (2) The tree is created by successive application of the **Tableaux Extension Rules** (which are selected non deterministically).
 - (3) A **path** in the Tableaux is **closed**,
 - if along the path as well X as $\neg X$ for a formula X occurs (X doesn't have to be atomic) or
 - if **false** occurs .
 - (4) A tableaux is **fully expanded**, if no more extension rules can be applied.
 - (5) A tableaux is called **closed**, if all its paths are closed.
 - (6) A **Tableaux Proof** for a formula X is a closed tableaux for $\neg X$.

Tableaux Extension Rules for PL

- for PL:

$\frac{\neg\neg X}{X}$	$\frac{\neg T}{F}$	$\frac{\neg F}{T}$
------------------------	--------------------	--------------------

- for conjunctive Formula (α -Rules):

α	$X \wedge Y$	$\neg(X \vee Y)$	$\neg(X \Rightarrow Y)$
α_1	X	$\neg X$	X
α_2	Y	$\neg Y$	$\neg Y$

- for disjunctive formula (β -Rules):

β	$X \vee Y$	$\neg(X \wedge Y)$	$(X \Rightarrow Y)$
$\beta_1 \mid \beta_2$	X Y	$\neg X \mid \neg Y$	$\neg X \mid Y$

Tableaux Algorithm (PL) -Example1

10

proof: $((q \wedge r) \Rightarrow (\neg q \vee r))$

α -Rule
 $\neg(X \Rightarrow Y)$
 X
 $\neg Y$

(1) $\neg((q \wedge r) \Rightarrow (\neg q \vee r))$

(2) $\alpha, 1: (q \wedge r)$

(3) $\alpha, 1: \neg(\neg q \vee r) = q \wedge \neg r$

Tableaux Algorithm (PL) -Example1

11

proof: $((q \wedge r) \Rightarrow (\neg q \vee r))$

(1) $\neg((q \wedge r) \Rightarrow (\neg q \vee r))$

(2) $\alpha, 1: (q \wedge r)$

(3) $\alpha, 1: \neg(\neg q \vee r) = q \wedge \neg r$

α -Rule

X	$\wedge$	Y
X		
Y		

(4) $\alpha, 2: q$

(5) $\alpha, 2: r$

(6) $\alpha, 3: q$

(7) $\alpha, 3: \neg r$

- path is closed
- tableaux is closed

Tableaux Algorithm (PL) -Example2

proof: $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$

α -Rule

$\neg(X \Rightarrow Y)$

X

$\neg Y$

(1) $\neg((p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)))$

(2| α from 1) $(p \Rightarrow (q \Rightarrow r))$

(3| α from 1) $\neg((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$

(4| α from 3) $(p \Rightarrow q)$

(5| α from 3) $\neg(p \Rightarrow r)$

(6| α from 5) p

(7| α from 5) $\neg r$

Tableaux Algorithm (PL) -Example2

proof: $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$

(1) $\neg((p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)))$

(2| α from 1) $(p \Rightarrow (q \Rightarrow r))$

(3| α from 1) $\neg((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$

(4| α from 3) $(p \Rightarrow q)$

(5| α from 3) $\neg(p \Rightarrow r)$

(6| α from 5) p

(7| α from 5) $\neg r$

(8| β from 2) $\neg p$ | (9| β from 2) $(q \Rightarrow r)$

(10| β from 9) $\neg q$ | (11| β from 9) r

(12| β from 4) $\neg p$ | (13| β from 4) q

β -Rule

$(X \Rightarrow Y)$

$\neg X$ | Y

Tableaux Algorithm Extension (FOL)

14

- as for propositional logic - X and Y stand for arbitrary (FOL) formulas
- Additional Rules for quantified formulas :

$$\frac{\gamma}{\gamma[t]}$$

$$\frac{\delta}{\delta[c]}$$

- γ for universally quantified formulas, δ existentially quantified formulas, with:

γ	$\gamma[t]$	δ	$\delta[c]$
$\forall x.\Phi$	$\Phi[x \leftarrow t]$	$\exists x.\Phi$	$\Phi[x \leftarrow c]$
$\neg \exists x.\Phi$	$\neg \Phi[x \leftarrow t]$	$\neg \forall x.\Phi$	$\neg \Phi[x \leftarrow c]$

- t is an arbitrary ground term
(i.e. doesn't contain variables that are bound in Φ),
- c is a „new“ constant

Tableaux Algorithm (FOL) -Example3

γ	$\gamma[t]$
$\forall x.\Phi$	$\Phi[x \leftarrow t]$
$\neg \exists x.\Phi$	$\neg \Phi[x \leftarrow t]$
δ	$\delta[c]$
$\exists x.\Phi$	$\Phi[x \leftarrow c]$
$\neg \forall x.\Phi$	$\neg \Phi[x \leftarrow c]$

Proof: $(\forall x . P(x) \vee Q(x)) \Rightarrow (\exists x . P(x)) \vee (\forall x . Q(x))$

(1) $\neg((\forall x . P(x) \vee Q(x)) \Rightarrow (\exists x . P(x)) \vee (\forall x . Q(x)))$

(2| α from 1) $(\forall x . P(x) \vee Q(x))$

(3| α from 1) $\neg((\exists x . P(x)) \vee (\forall x . Q(x)))$

(4| α from 3) $\neg(\exists x . P(x))$

(5| α from 3) $\neg(\forall x . Q(x))$

(6| δ from 5) $\neg Q(c)$

(7| γ from 4) $\neg P(c)$

(8| γ from 2) $P(c) \vee Q(c)$

(9| β from 8) $P(c)$ | (10| β from 8) $Q(c)$



05 Description Logics - ALC

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 04: Knowledge Representations Pt.1