Lecture 17

1.11 Truth trees

(1) Truth tables are laborious to construct.

(2) Truth tables don't generalise to FOL.

Truth trees are usually more efficient than truth tables and do generalize to FOL.

Truth trees ≠ parse trees

- We use a data structure, a tree, to represent the truth value options of a wff.

- What is not true is false – we only represent what is true.

- The truth tree algorithm is a divide and conquer algorithm.

- The truth-tree algorithm is about <u>satisfiability</u>.

There are two kinds of rules: <u>α-rules</u> and <u>β-rules</u>

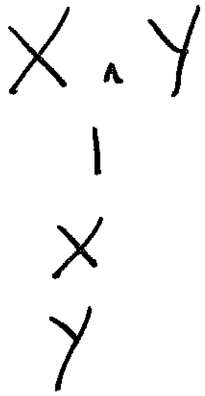
α-rule	β-rule
α $\mid$ α_1 α_2	α $\alpha_1 \qquad \alpha_2$

<u>α-rule</u>: α is T precisely when both α_1 and α_2 are T.

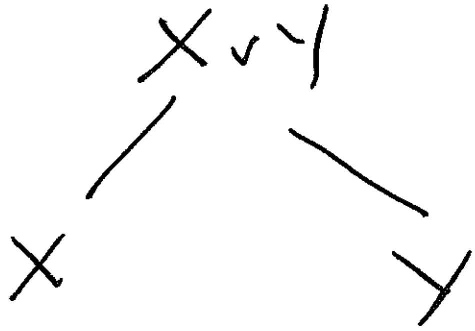
<u>β-rule</u>: α is T precisely when at least one of α_1 and α_2 is T.

Examples

(1)



(2)



Both of these trees contain exactly the same information as the truth tables

---

We shall exclude $\oplus$ from what follows

since $X \oplus Y \equiv \neg(X \leftrightarrow Y)$

Possible shapes of wff

1. $X \wedge Y$
2. $\neg(X \vee Y)$
3. $\neg(X \rightarrow Y)$
4. $\neg\neg X$
5. $\neg(X \wedge Y)$
6. $X \vee Y$
7. $X \rightarrow Y$
8. $X \leftrightarrow Y$
9. $\neg(X \leftrightarrow Y)$

We now represent the truth tables for these wff in terms of α-rules or β-rules.

---

① $X \wedge Y$:

$$\begin{array}{c} X \wedge Y \\ | \\ X \\ Y \end{array}$$

---

② $\neg(X \vee Y)$

$(\equiv \neg X \wedge \neg Y)$

$$\begin{array}{c} \neg(X \vee Y) \\ | \\ \neg X \\ \neg Y \end{array}$$

---

③ $\neg(X \to Y)$

$\equiv \neg(\neg X \vee Y)$

$\equiv \neg\neg X \wedge \neg Y$

$\equiv X \wedge \neg Y$

$$\begin{array}{c} \neg(X \to Y) \\ | \\ X \\ \neg Y \end{array}$$

---

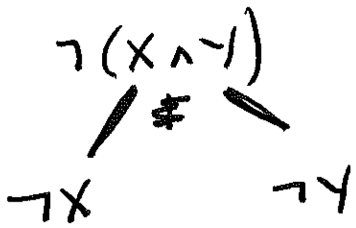
④

$$\begin{array}{c} \neg\neg X \\ | \\ X \end{array}$$

double negation

---

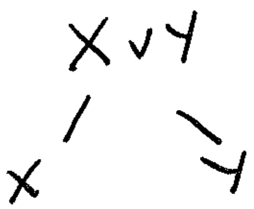
⑤



sie $\neg(X \wedge Y) \equiv \neg X \vee \neg Y$

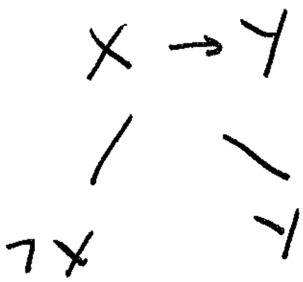
---

⑥



---

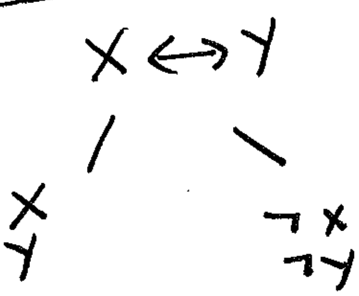
⑦



sie $X \to Y \equiv \neg X \vee Y$

---

⑧



$$X \leftrightarrow Y \equiv (X \to Y) \wedge (Y \to X)$$

$$\equiv (\neg X \vee Y) \wedge (\neg Y \vee X)$$

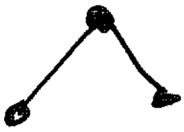
$$\equiv ((\neg X \vee Y) \wedge \neg Y) \vee ((\neg X \vee Y) \wedge X)$$

$$\equiv (\neg X \wedge \neg Y) \vee \overbrace{(Y \wedge \neg Y)}^{f} \vee \underbrace{(\neg X \wedge X)}_{f} \vee (Y \wedge X)$$

$$\equiv (\neg X \wedge \neg Y) \vee (X \wedge Y)$$

(9) $\neg(X \leftrightarrow Y)$

X, $\neg Y$ | $\neg X$, Y

For each of our 9 shapes we have drawn a tree  or  which completely describes the truth table of the conn (i.e. tree - tree)

Example

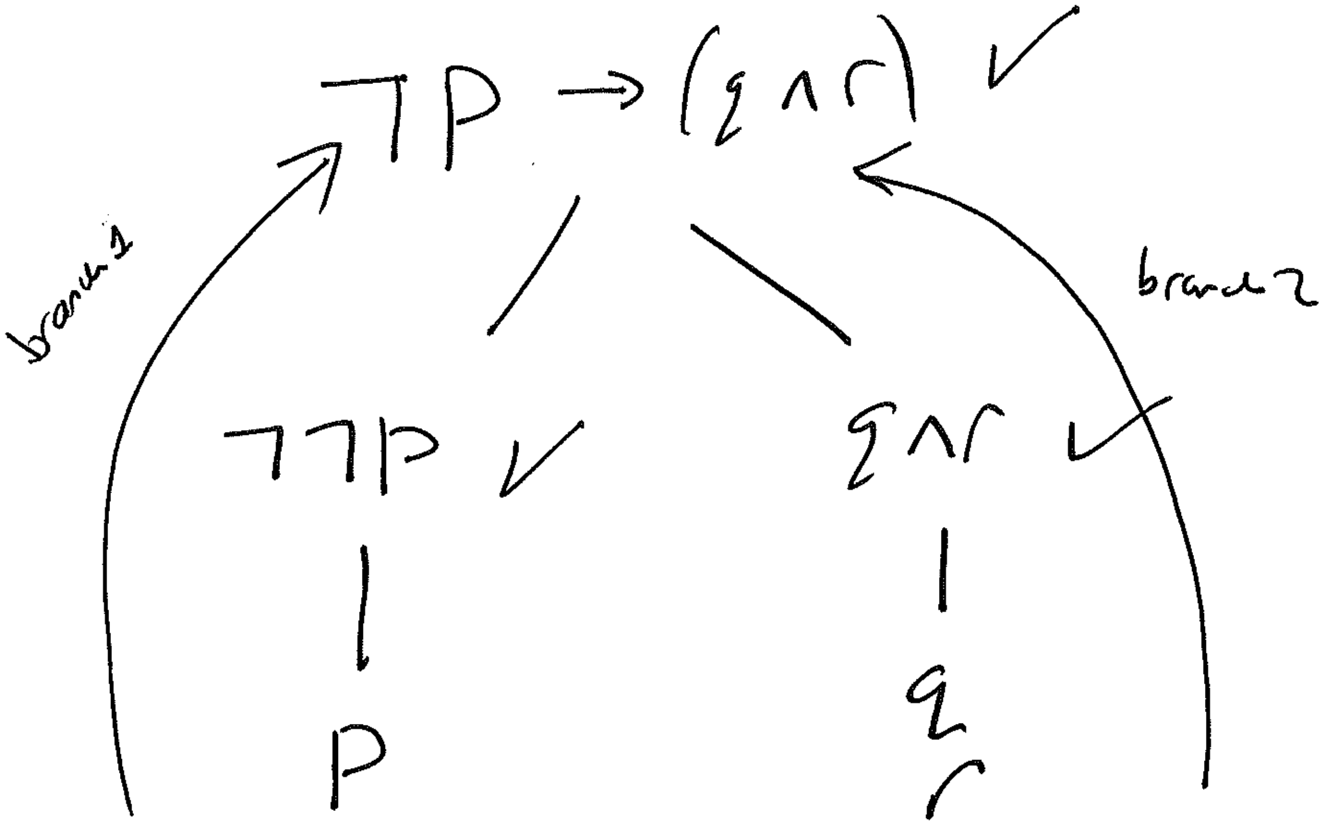
$X \leftrightarrow Y$

X, Y | $\neg X$, $\neg Y$

X	Y	X ⟷ Y
T	T	T
T	F	F (default)
F	T	F (default)
F	F	T

Example Construct a truth tree for $\neg p \rightarrow (q \wedge r)$

(**NB** Not a parse tree!)



$\neg p \rightarrow (q \wedge r)$ is true when

p is true or $q \wedge r$ is true

$\therefore \neg p \rightarrow (q \wedge r) \equiv p \vee (q \wedge r)$

We can use this truth tree to write down the truth table of $\neg p \to (q \wedge r)$ (though we would not usually do this)

p	q	r	$\neg p \to (q \wedge r)$	
T	T	T	T	since p true
T	T	F	T	
T	F	T	T	
T	F	F	T	
F	T	T	T	q & r true
F	T	F	F	default.
F	F	T	F	
F	F	F	F	