COMP0009 Logic slides 3. Propositional Tableau.

October 12, 2023

Proof by Contradiction

- To prove a formula ϕ,

Proof by Contradiction

- To prove a formula ϕ, first
- Assume $\neg\phi$,

Proof by Contradiction

- To prove a formula ϕ, first
- Assume $\neg\phi$, then
- Deduce a contradiction $q \wedge \neg q$.

Proof by Contradiction

- To prove a formula ϕ, first
- Assume $\neg\phi$, then
- Deduce a contradiction $q \wedge \neg q$. Hence,
- Conclude ϕ

Propositional Tableaux

- Given the formula ϕ, a tableau can tell us whether it is **satisfiable or not**

Propositional Tableaux

- Given the formula ϕ, a tableau can tell us whether it is **satisfiable or not**
- It does so by **decomposing** the formulae according to certain rules and growing a labelled tree to the point that only literals are left

Propositional Tableaux

- Given the formula ϕ, a tableau can tell us whether it is **satisfiable or not**
- It does so by **decomposing** the formulae according to certain rules and growing a labelled tree to the point that only literals are left
- A branch is closed if it has one node labelled by a proposition and another node labelled by its negation.

Propositional Tableaux

- Given the formula ϕ, a tableau can tell us whether it is **satisfiable or not**
- It does so by **decomposing** the formulae according to certain rules and growing a labelled tree to the point that only literals are left
- A branch is closed if it has one node labelled by a proposition and another node labelled by its negation.
- A tableau is closed if every branch is closed.

Propositional Tableaux

- Given the formula ϕ, a tableau can tell us whether it is **satisfiable or not**
- It does so by **decomposing** the formulae according to certain rules and growing a labelled tree to the point that only literals are left
- A branch is closed if it has one node labelled by a proposition and another node labelled by its negation.
- A tableau is closed if every branch is closed.
- If the tableau closes then ϕ is **unsatisfiable**

Propositional Tableaux

- Given the formula ϕ, a tableau can tell us whether it is **satisfiable or not**
- It does so by **decomposing** the formulae according to certain rules and growing a labelled tree to the point that only literals are left
- A branch is closed if it has one node labelled by a proposition and another node labelled by its negation.
- A tableau is closed if every branch is closed.
- If the tableau closes then ϕ is **unsatisfiable**
- If the tableau never closes then ϕ is **satisfiable**

The idea

To test whether $((\phi \wedge \neg\phi) \vee (\psi \wedge \neg\psi))$ is satisfiable

$$((\phi \wedge \neg\phi) \vee (\psi \wedge \neg\psi)) \quad \checkmark$$

$$\begin{array}{cc}
(\phi \wedge \neg\phi) \quad \checkmark & (\psi \wedge \neg\psi) \quad \checkmark \\
\phi & \psi \\
\neg\phi & \neg\psi \\
\oplus & \oplus
\end{array}$$

The idea

To test whether $((\phi \land \neg\phi) \lor (\psi \land \neg\psi))$ is satisfiable

$$
\begin{array}{ccc}
 & ((\phi \land \neg\phi) \lor (\psi \land \neg\psi)) \;\; \checkmark & \\
\swarrow & & \searrow \\
(\phi \land \neg\phi) \;\; \checkmark & & (\psi \land \neg\psi) \;\; \checkmark \\
| & & | \\
\phi & & \psi \\
| & & | \\
\neg\phi & & \neg\psi \\
\oplus & & \oplus
\end{array}
$$

Both branches close $\Rightarrow$ not satisfiable.

Tableaux

- Formally, a tableau T is a type of binary tree where every node is labelled by a formula.

Tableaux

- Formally, a tableau T is a type of binary tree where every node is labelled by a formula.
- At first, T has only one node, containing the formula ϕ whose satisfiability we are investigating.

Tableaux

- Formally, a tableau T is a type of binary tree where every node is labelled by a formula.
- At first, T has only one node, containing the formula ϕ whose satisfiability we are investigating.
- Every formula in the tableau, except literals, gets "expanded" and ticked.
- If a branch of T contains p and $\neg p$ then it is **closed**. Otherwise it is **open**.
- If every branch of T is closed then T is **closed**. Otherwise it is **open**.

Tableau Expansion Rules

α formulas

$$(\phi \wedge \psi)$$

Observe that $(\phi \wedge \psi)$ is true if and only if ϕ is true and ψ is true.

Tableau Expansion Rules

α formulas

$$
\begin{array}{c}
(\phi \wedge \psi) \quad \checkmark \\
| \\
\phi \\
| \\
\psi
\end{array}
$$

Observe that $(\phi \wedge \psi)$ is true if and only if ϕ is true and ψ is true.

Means that nodes corresponding to ϕ and ψ are added at every leaf of T below the current node $(\phi \wedge \psi)$.

Tableau Expansion Rules

α formulas

Three other kinds of α formulas.

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

Tableau Expansion Rules

α formulas

Three other kinds of α formulas.

$$\begin{array}{c}\neg\neg\phi \\ | \\ \phi\end{array} \qquad\qquad \begin{array}{c}\neg(\phi \vee \psi) \quad \checkmark \\ | \\ \neg\phi \\ | \\ \neg\psi\end{array}$$

Observe that $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$

Tableau Expansion Rules

α formulas

Three other kinds of α formulas.

$$\begin{array}{c} \neg\neg\phi \\ | \\ \phi \end{array} \qquad \begin{array}{c} \neg(\phi \vee \psi) \quad \checkmark \\ | \\ \neg\phi \\ | \\ \neg\psi \end{array} \qquad \begin{array}{c} \neg(\phi \to \psi) \quad \checkmark \\ | \\ \phi \\ | \\ \neg\psi \end{array}$$

Observe that $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$

Observe that $\neg(\phi \to \psi) \equiv \neg(\neg\phi \vee \psi) \equiv \phi \wedge \neg\psi$

Tableau Expansion Rules

β formulas

$$(\phi \vee \psi)$$

For this case, at each leaf of the tableau below the current node, we create two new leaves, one labelled ϕ the other labelled ψ.

Tableau Expansion Rules

β formulas

$$\begin{tikzcd}[column sep=small] & (\phi \vee \psi) \quad \checkmark \arrow[dl, no head] \arrow[dr, no head] & \\ \phi & & \psi \end{tikzcd}$$

For this case, at each leaf of the tableau below the current node, we create two new leaves, one labelled ϕ the other labelled ψ.
Observe that $(\phi \vee \psi)$ is true if and only if ϕ is true or ψ is true.

Tableau Expansion Rules

β formulas

There are two other kinds of β formulas.

$$
\begin{array}{ccc}
 & \neg(\phi \wedge \psi) \quad \checkmark & \\
 / & & \backslash \\
\neg\phi & & \neg\psi
\end{array}
$$

Observe that $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$

Tableau Expansion Rules

β formulas

There are two other kinds of β formulas.

$$\begin{array}{ccc} & \neg(\phi \wedge \psi) \quad \checkmark & \\ / & & \backslash \\ \neg\phi & & \neg\psi \end{array} \qquad\qquad \begin{array}{ccc} & (\phi \rightarrow \psi) \quad \checkmark & \\ / & & \backslash \\ \neg\phi & & \psi \end{array}$$

Observe that $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$

Observe that $(\phi \rightarrow \psi) \equiv \neg\phi \vee \psi$

Tableaux

- A tableau is **complete** if every node is either ticked (already expanded) or a literal.
- If ϕ is at the root of a **complete open** tableau, then ϕ is **satisfiable**.

Propositional Tableaux

Remember: In a proof system we are interested in validity, not just satisfiability. Fortunately, we know that:

$$\begin{aligned} \phi \text{ is satisfiable} \quad &\Longleftrightarrow \quad \neg\phi \text{ is not valid} \\ \phi \text{ is valid} \quad &\Longleftrightarrow \quad \neg\phi \text{ is not satisfiable} \end{aligned}$$

Propositional Tableaux

Remember: In a proof system we are interested in validity, not just satisfiability. Fortunately, we know that:

$$\begin{aligned} \phi \text{ is satisfiable} \quad &\iff \quad \neg\phi \text{ is not valid} \\ \phi \text{ is valid} \quad &\iff \quad \neg\phi \text{ is not satisfiable} \end{aligned}$$

Solution: In order to test whether a formula ϕ is valid we see whether the tableau for $\neg\phi$ closes

Example 1: Is formula $(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ valid?

$$(1)\, \neg(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$$

Example 1: Is formula $(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ valid?

$$\begin{array}{c} (1)\, \neg(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r))) \quad \checkmark \\ \Big| \, {\scriptstyle \alpha(1)} \\ (2)\, ((p \wedge q) \to r) \\ \Big| \\ (3)\, \neg((p \to r) \vee (q \to r)) \end{array}$$

Example 1: Is formula $(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$ valid?

$$(1)\ \neg(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r))) \quad \checkmark$$

$$\alpha(1)$$

$$(2)\ ((p \wedge q) \rightarrow r) \quad \checkmark$$

$$(3)\ \neg((p \rightarrow r) \vee (q \rightarrow r))$$

$$\beta(2)$$

$$(4)\ \neg(p \wedge q) \qquad\qquad (5)\ r$$

Example 1: Is formula $(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$ valid?

(1) $\neg(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$ ✓

$\alpha(1)$

(2) $((p \wedge q) \rightarrow r)$ ✓

(3) $\neg((p \rightarrow r) \vee (q \rightarrow r))$ ✓

$\beta(2)$

(4) $\neg(p \wedge q)$	(5) r
$\alpha(3)$	$\alpha(3)$
(6) $\neg(p \rightarrow r)$	(8) $\neg(p \rightarrow r)$
(7) $\neg(q \rightarrow r)$	(9) $\neg(q \rightarrow r)$

Example 1: Is formula $(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$ valid?

$(1)\ \neg(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))\quad \checkmark$

$\alpha(1)$

$(2)\ ((p \wedge q) \rightarrow r)\quad \checkmark$

$(3)\ \neg((p \rightarrow r) \vee (q \rightarrow r))\quad \checkmark$

$(4)\ \neg(p \wedge q)\quad \checkmark$ $\qquad \beta(2) \qquad$ $(5)\ r$

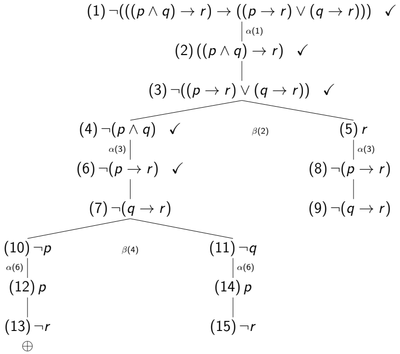
$\alpha(3)$ $\qquad\qquad$ $\alpha(3)$

$(6)\ \neg(p \rightarrow r)$ $\qquad\qquad$ $(8)\ \neg(p \rightarrow r)$

$(7)\ \neg(q \rightarrow r)$ $\qquad\qquad$ $(9)\ \neg(q \rightarrow r)$

$(10)\ \neg p$ $\qquad \beta(4) \qquad$ $(11)\ \neg q$

Example 1: Is formula $(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$ valid?



Example 1: Is formula $(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$ valid?

$(1)\ \neg(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r))) \quad \checkmark$

$\alpha(1)$

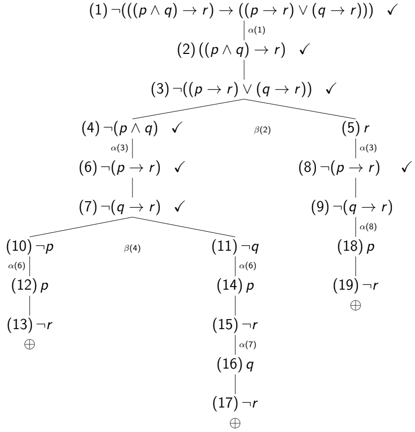
$(2)\ ((p \wedge q) \rightarrow r) \quad \checkmark$

$(3)\ \neg((p \rightarrow r) \vee (q \rightarrow r)) \quad \checkmark$

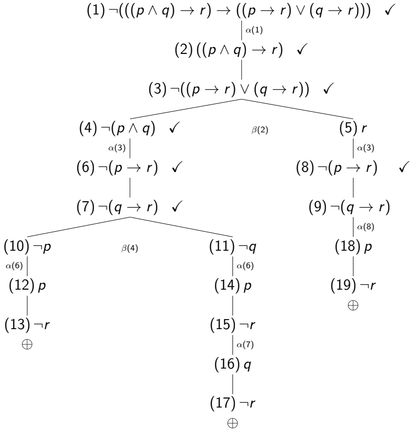
$\beta(2)$

- $(4)\ \neg(p \wedge q) \quad \checkmark$
  - $\alpha(3)$
  - $(6)\ \neg(p \rightarrow r) \quad \checkmark$
  - $(7)\ \neg(q \rightarrow r) \quad \checkmark$
  - $\beta(4)$
    - $(10)\ \neg p$
      - $\alpha(6)$
      - $(12)\ p$
      - $(13)\ \neg r$
      - $\oplus$
    - $(11)\ \neg q$
      - $\alpha(6)$
      - $(14)\ p$
      - $(15)\ \neg r$
      - $\alpha(7)$
      - $(16)\ q$
      - $(17)\ \neg r$
      - $\oplus$
- $(5)\ r$
  - $\alpha(3)$
  - $(8)\ \neg(p \rightarrow r)$
  - $(9)\ \neg(q \rightarrow r)$

Example 1: Is formula $(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ valid?



Example 1: Is formula $(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ valid?



Closed tableau — **root formula is unsatisfiable, original formula is valid.**

Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?

$(1)\, \neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$

Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?

$$
\begin{array}{c}
(1)\ \neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r)) \quad \checkmark \\
\Big| \ \alpha(1) \\
(2)\ ((p \to r) \vee (q \to r)) \\
\Big| \\
(3)\ \neg((p \vee q) \to r)
\end{array}
$$

Example 2: Is formula $(((p \to r) \lor (q \to r)) \to ((p \lor q) \to r))$ valid?

$(1)\ \neg(((p \to r) \lor (q \to r)) \to ((p \lor q) \to r))$ ✓

$\alpha(1)$

$(2)\ ((p \to r) \lor (q \to r))$ ✓

$(3)\ \neg((p \lor q) \to r)$

$\beta(2)$

$(4)\ (p \to r)$ | $(5)\ (q \to r)$

Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?

$$(1)\ \neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r)) \quad \checkmark$$

$\alpha(1)$

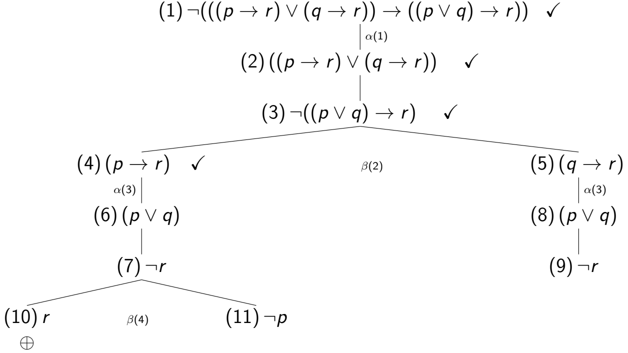
$$(2)\ ((p \to r) \vee (q \to r)) \quad \checkmark$$

$$(3)\ \neg((p \vee q) \to r) \quad \checkmark$$

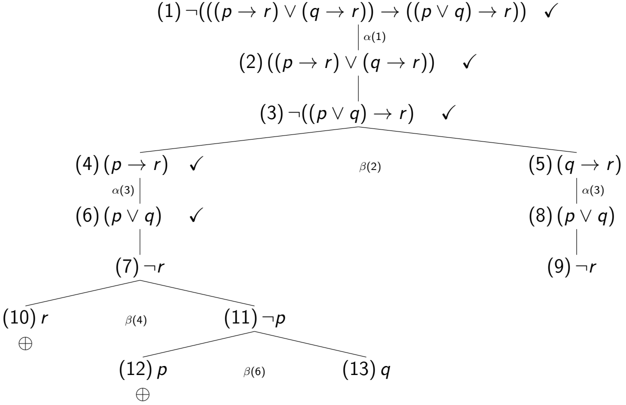
$\beta(2)$

$(4)\ (p \to r)$	$(5)\ (q \to r)$
$\alpha(3)$	$\alpha(3)$
$(6)\ (p \vee q)$	$(8)\ (p \vee q)$
$(7)\ \neg r$	$(9)\ \neg r$

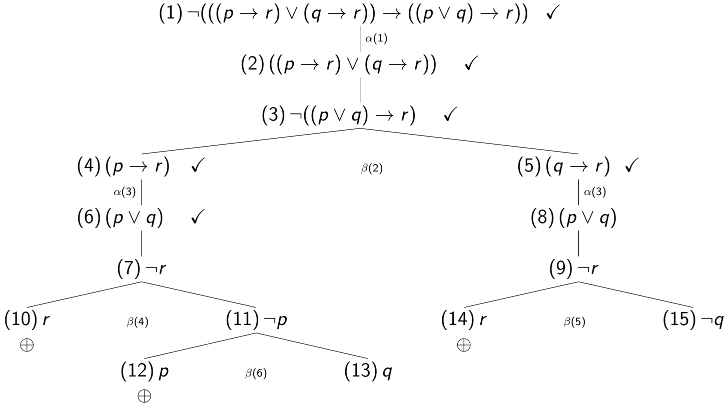
Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?



Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?



Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?



Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?

(1) $\neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ ✓

$\alpha(1)$

(2) $((p \to r) \vee (q \to r))$ ✓

(3) $\neg((p \vee q) \to r)$ ✓

$\beta(2)$

- (4) $(p \to r)$ ✓
  - $\alpha(3)$
  - (6) $(p \vee q)$ ✓
  - (7) $\neg r$
  - $\beta(4)$
    - (10) r $\oplus$
    - (11) $\neg p$
      - $\beta(6)$
        - (12) p $\oplus$
        - (13) q
- (5) $(q \to r)$ ✓
  - $\alpha(3)$
  - (8) $(p \vee q)$ ✓
  - (9) $\neg r$
  - $\beta(5)$
    - (14) r $\oplus$
    - (15) $\neg q$
      - $\beta(8)$
        - (16) p
        - (17) q $\oplus$

Example 2: Is formula $(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ valid?

(1) $\neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$ ✓

$\alpha(1)$

(2) $((p \to r) \vee (q \to r))$ ✓

(3) $\neg((p \vee q) \to r)$ ✓

$\beta(2)$

- (4) $(p \to r)$ ✓
  - $\alpha(3)$
  - (6) $(p \vee q)$ ✓
  - (7) $\neg r$
  - $\beta(4)$
    - (10) r $\oplus$
    - (11) $\neg p$
      - $\beta(6)$
        - (12) p $\oplus$
        - (13) q
- (5) $(q \to r)$ ✓
  - $\alpha(3)$
  - (8) $(p \vee q)$ ✓
  - (9) $\neg r$
  - $\beta(5)$
    - (14) r $\oplus$
    - (15) $\neg q$
      - $\beta(8)$
        - (16) p
        - (17) q $\oplus$

Two open branches — **root formula is satisfiable, original formula is not valid.**

Disjunctive Normal Form (DNF)

A **DNF** formula is a disjunctions of clauses where each clause is a conjunction of literals. E.g.

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

Also

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

Every propositional formula has an equivalent formula in DNF. A **CNF** formula is a conjunction of clauses where each clause is disjunction of literals. E.g.

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

Converting a formula to DNF

1. By tableau
2. By truth table
3. By logical equivalences (De Morgan's laws, distribution laws etc., later)

Using tableau to find DNF equivalent

- Place ϕ at root of new tableau.
- Expand until tableau T is completed (no nodes left to expand).
- For each open branch Θ of T let

$$C_\Theta = \bigwedge\{\text{literals in } \Theta\}$$

- Then

$$\phi \equiv \bigvee_{\text{open branches } \Theta} C_\Theta$$

Example 2: Is formula $(((p \to r) \lor (q \to r)) \to ((p \lor q) \to r))$ valid?

(1) $\neg(((p \to r) \lor (q \to r)) \to ((p \lor q) \to r))$ ✓

$\alpha(1)$

(2) $((p \to r) \lor (q \to r))$ ✓

(3) $\neg((p \lor q) \to r)$ ✓

$\beta(2)$

(4) $(p \to r)$ ✓ | (5) $(q \to r)$ ✓

$\alpha(3)$ | $\alpha(3)$

(6) $(p \lor q)$ ✓ | (8) $(p \lor q)$ ✓

(7) $\neg r$ | (9) $\neg r$

$\beta(4)$: (10) r $\oplus$ | (11) $\neg p$

$\beta(6)$: (12) p $\oplus$ | (13) q

$\beta(5)$: (14) r $\oplus$ | (15) $\neg q$

$\beta(8)$: (16) p | (17) q $\oplus$

The **DNF** for $\neg(((p \to r) \lor (q \to r)) \to ((p \lor q) \to r))$ is $(\neg r \land \neg p \land q) \lor (\neg r \land \neg q \land p)$

DNF by truth table

$\neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$

p	q	r	$\neg(((p \to r) \vee (q \to r)) \to ((p \vee q) \to r))$	
0	0	0	0	
0	0	1	0	
0	1	0	1	$(\neg p \wedge q \wedge \neg r)$
0	1	1	0	$\vee$
1	0	0	1	$(p \wedge \neg q \wedge \neg r)$
1	0	1	0	
1	1	1	0	

Problem

- How do you convert a propositional formula into an equivalent CNF?
- E.g.

$$((p \land \neg q) \lor (q \land \neg r)) \equiv ??$$