09
SOUNDNESS AND COMPLETENESS

Paal Antonsen
antonsp@tcd.ie
https://sites.google.com/site/paalantonsen/teaching/logic

*Formal Logic*
We began the course with an intuitive conception of validity.

**Validity: generic**
An argument $\langle X, A \rangle$ is valid iff in every case, if all the premises $X$ are true then the conclusion $A$ is also true.

We have introduced the language of propositional logic. What we have learned so far give us the resources to check the validity of arguments when their validity depends on their propositional argument form.

This was captured in a more precise definition of validity:

**Validity: propositional logic**
An argument $\langle X, A \rangle$ is valid in propositional logic ($X \models A$) iff on every evaluation, if all the premises $X$ are true then $A$ is also true.
Checking for validity: two tests

- Once we had a definition of validity (in propositional logic), we asked the question: how can we check whether an argument is valid?

  (A) Truth table test

  (B) Tree method test

  Both (A) and (B) are mechanical procedures. The only thing we need to do is applying the rules correctly. At the same time, (A) and (B) are also decidable: they always give a verdict about the validity of an argument.

  A test is decidable iff for any argument form, the test will either classify it as valid or as invalid. (It always gives a verdict).

  In the case of (B) this amounts to:

  For any \( \langle X, A \rangle \), correctly applying the rules will result in a completed tree for \( \langle X, \neg A \rangle \); either it will be closed (\( X \vdash A \)) or it will be open (\( X \nvdash A \)).
Once we had a definition of validity (in propositional logic), we asked the question: how can we check whether an argument is valid?

We have introduced two ways for checking validity:

(A) Truth table test
(B) Tree method test
Once we had a definition of validity (in propositional logic), we asked the question: how can we check whether an argument is valid?

We have introduced two ways for checking validity:

(A) Truth table test
(B) Tree method test

Both (A) and (B) are mechanical procedures. The only thing we need to do is applying the rules correctly. At the same time, (A) and (B) are also **decidable**: they always give a verdict about the validity of an argument.

A test is **decidable** iff for any argument form, the test will either classify it as valid or as invalid. (It always gives a verdict).

In the case of (B) this amounts to:

For any \( \langle X, A \rangle \), correctly applying the rules will result in a completed tree for \( \langle X, \sim A \rangle \); either it will be closed \((X \vdash A)\) or it will be open \((X \not\vdash A)\).
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 1

?  \[ p \supset (q \& r), \sim r \vdash \sim p \]

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(r)</th>
<th>(p \supset (q &amp; r))</th>
<th>(\sim r)</th>
<th>(\sim p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Q  How did you check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 1

\[ p \supset (q \& r), \sim r \vdash \sim p \]

<table>
<thead>
<tr>
<th></th>
<th>q</th>
<th>r</th>
<th></th>
<th>p \supset (q &amp; r)</th>
<th>\sim r</th>
<th>\sim p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Checking validity with method (A): truth tables
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 1
?

\( p \supset (q \& r), \sim r \vdash \sim p \)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 1

\[ p \supset (q \& r), \sim r \vdash \sim p \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \supset (q &amp; r)</th>
<th>\sim r</th>
<th>\sim p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

Q How did check for validity with truth tables?
A Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 1

\[ p \supset (q \& r), \sim r \vdash \sim p \]

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( r )</th>
<th>( p \supset (q &amp; r) )</th>
<th>( \sim r )</th>
<th>( \sim p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 1: classified as valid
✓  \( p \supset (q \& r), \sim r \vdash \sim p \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( r )</th>
<th>( p \supset (q &amp; r) )</th>
<th>(~ r )</th>
<th>(~ p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
How did check for validity with truth tables?

Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 2

\[ p \equiv (q \lor r), \sim r \models \sim p \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \equiv (q \lor r)</th>
<th>\sim r</th>
<th>\sim p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

**Q** How did check for validity with truth tables?

**A** Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

### Example 2

? \( p \equiv (q \lor r) \), \( \sim r \models \sim p \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( r )</th>
<th>( p \equiv (q \lor r) )</th>
<th>( \sim r )</th>
<th>( \sim p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 2

\[ p \equiv (q \lor r), \sim r \models \sim p \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \equiv (q \lor r)</th>
<th>\sim r</th>
<th>\sim p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Q  How did check for validity with truth tables?

A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 2

?  \( p \equiv (q \lor r), \sim r \models \sim p \)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p \equiv (q \lor r) )</td>
<td>( \sim r )</td>
<td>( \sim p )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>q</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?
A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 2

\[ p \equiv (q \lor r), \sim r \models \sim p \]

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \equiv (q \lor r)</th>
<th>\sim r</th>
<th>\sim p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 \equiv 0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0 \equiv 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0 \equiv 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0 \equiv 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0 \equiv 0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1 \equiv 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1 \equiv 1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1 \equiv 1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Checking validity with method (A): truth tables

Q  How did check for validity with truth tables?

A  Using the definitions of the connectives, fill out the truth table for each formula. If the conclusion has the value 1 in every row where all the premises have the value 1, then the argument is valid.

Example 2: classified as invalid

\[ p \equiv (q \lor r), \sim r \not\equiv \sim p \]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \equiv (q \lor r)</th>
<th>\sim r</th>
<th>\sim p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
 Checking validity with method (B): trees

Q  How did check for validity with trees?
A  Where \( \langle X, A \rangle \), we do so by constructing a tree for \( \langle X, \sim A \rangle \). Using the rules for the connectives, we completed a tree by resolving formulas and closing branches. If every branch of the tree close, then the argument is classified as valid.

**Closure**

<table>
<thead>
<tr>
<th>A</th>
<th>\sim A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>\times</td>
<td></td>
</tr>
</tbody>
</table>

**Conjunction**

<table>
<thead>
<tr>
<th>A &amp; B</th>
<th>\sim(A &amp; B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Disjunction**

<table>
<thead>
<tr>
<th>A \lor B</th>
<th>\sim(A \lor B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>\sim A \sim B</td>
</tr>
</tbody>
</table>

**Negation**

<table>
<thead>
<tr>
<th>\sim \sim A</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sim A</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A</td>
</tr>
</tbody>
</table>

**Conditional**

<table>
<thead>
<tr>
<th>A \supset B</th>
<th>\sim(A \supset B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sim A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>\sim B</td>
</tr>
</tbody>
</table>

**Biconditional**

<table>
<thead>
<tr>
<th>A \equiv B</th>
<th>\sim(A \equiv B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>\sim A \sim B</td>
</tr>
<tr>
<td>B</td>
<td>\sim B \sim A</td>
</tr>
</tbody>
</table>
Example 3

\[ p \supset (q \land r), \sim r \vdash \sim p \]
Example 3

? \( p \supset (q \land r) \), \( \sim r \vdash \sim p \)

\[
p \supset (q \land r) \\
\sim r \\
\sim \sim p
\]
Checking validity with method (B): trees

Example 3

\[ p \supset (q \land r), \sim r \vdash \sim p \]
Checking validity with method (B): trees

Example 3

\[ p \supset (q \& r), \sim r \vdash \sim p \]
Example 3

\[ p \supset (q \& r), \sim r \vdash \sim p \]
Checking validity with method (B): trees

Example 3

? \( p \supset (q \& r), \sim r \vdash \sim p \)
Example 3

\[ p \supset (q \& r), \sim r \vdash \sim p \]
Checking validity with method (B): trees

Example 3: classified as valid

\[ \checkmark \ p \supset (q \land r), \ \neg r \vdash \neg p \]
Example 4

\( p \land \sim q, r \supset q, (p \land r) \supset \sim s \vdash p \equiv \sim s \)
Example 4

\(? p \& \sim q, r \supset q, (p \& r) \supset \sim s \vdash p \equiv \sim s\)
Example 4

\[ p \land \sim q, \ r \supset q, \ (p \land r) \supset \sim s \Downarrow p \equiv \sim s \]
Example 4

\[ p \land \sim q, \quad r \supset q, \quad (p \land r) \supset \sim s \vdash p \equiv \sim s \]
Example 4

\[ p \land \sim q, r \supset q, (p \land r) \supset \sim s \vdash p \equiv \sim s \]
Example 4

\[ \begin{align*}
? & \quad p \land \neg q, \ r \supset q, \ (p \land r) \supset \neg s \vdash p \equiv \neg s \\
\end{align*} \]
Example 4

$p \land \sim q, r \supset q, (p \land r) \supset \sim s \vdash p \equiv \sim s$
Example 4

? \( p \& \sim q, \ r \supset q, \ (p \& r) \supset \sim s \vdash p \equiv \sim s \)
Example 4

\(? p \& \sim q, r \supset q, (p \& r) \supset \sim s \vdash p \equiv \sim s\)
Example 4

$p \& \sim q, r \supset q, (p \& r) \supset \sim s \vdash p \equiv \sim s$
Checking validity with method (B): trees

Example 4

? \(p \land \sim q, r \supset q, (p \land r) \supset \sim s \vdash p \equiv \sim s\)
Checking validity with method (B): trees

Example 4

? \( p \& \sim q, \ r \supset q, \ (p \& r) \supset \sim s \mid p \equiv \sim s \)
Example 4

\( p \& \sim q, \ r \supseteq q, \ (p \& r) \supset \sim s \vdash p \equiv \sim s \)
Example 4

\[ p \land \neg q, r \supset q, (p \land r) \supset \neg s \vdash p \equiv \neg s \]
Checking validity with method (B): trees

Example 4

? \( p \land \sim q, r \supset q, (p \land r) \supset \sim s \vdash p \equiv \sim s \)
Checking validity with method (B): trees

Example 4

\[ p \land \neg q, \ r \supset q, \ (p \land r) \supset \neg s \vdash p \equiv \neg s \]
Example 4

\[ \neg p \land \neg q, \, r \supset q, \, (p \land r) \supset \neg s \vdash p \equiv \neg s \]
Example 4

? \( p \& \sim q, r \supset q, (p \& r) \supset \sim s \vdash p \equiv \sim s \)
Checking validity with method (B): trees

Example 4: classified as invalid

\[ X \quad p \land \sim q, \quad r \supset q, \quad (p \land r) \supset \sim s \not\models p \equiv \sim s \]
Soundness and completeness

- The tree method is decidable, so every argument form will be classified as valid or invalid when we generate a tree. In other words, for every argument form \( \langle X, A \rangle \), either \( X \vdash A \) or \( X \not\vdash A \).

- But can we be sure that three method is reliable?
The tree method is decidable, so every argument form will be classified as valid or invalid when we generate a tree. In other words, for every argument form $\langle X, A \rangle$, either $X \vdash A$ or $X \not\vdash A$.

But can we be sure that the method is reliable?

In order for it to be reliable, we need to know that if an argument form is valid then all the completed trees generated from the initial list will close, and vice versa.
Soundness and completeness

- The tree method is decidable, so every argument form will be classified as valid or invalid when we generate a tree. In other words, for every argument form \( \langle X, A \rangle \), either \( X \vdash A \) or \( X \nvdash A \).

- But can we be sure that the method is reliable?

  In order for it to be reliable, we need to know that if an argument form is valid then all the completed trees generated from the initial list will close, and vice versa.

A logical system is **sound** iff if \( X \vdash A \) then \( X \models A \).

A logical system is **complete** iff if \( X \models A \) then \( X \vdash A \).

- When we want in showing that a logical system is sound and complete, we are talking properties of that system. We call this **metalogic**.
Soundness and completeness

**Soundness**

If $X \vdash A$ then $X \models A$

iff If $X \not\vdash A$ then $X \not\models A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \sim A \rangle$. 
Soundness and completeness

**Soundness**
If $X \vdash A$ then $X \models A$

iff If $X \not\vdash A$ then $X \not\models A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \neg A \rangle$.

**Assumption**: The argument $\langle X, A \rangle$ is invalid.
Soundness and completeness

**Soundness**
If $X \vdash A$ then $X \models A$

iff If $X \not\vdash A$ then $X \not\models A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \sim A \rangle$.

**Assumption**: The argument $\langle X, A \rangle$ is invalid.

There is then an evaluation $v$, such that $X_1, \ldots, X_n$ are true on $v$ and $A$ is false on $v$. In other words, $\sim A$ is true on $v$. 
Soundness and completeness

**Soundness**
If $X \vdash A$ then $X \models A$

iff If $X \not\models A$ then $X \not\vdash A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \sim A \rangle$.

**Assumption**: The argument $\langle X, A \rangle$ is invalid.

There is then an evaluation $v$, such that $X_1, \ldots, X_n$ are true on $v$ and $A$ is false on $v$. In other words, $\sim A$ is true on $v$.

The root for $\langle X_1, \ldots, X_n, \sim A \rangle$ therefore doesn’t close.
Soundness and completeness

Soundness

If \( X \vdash A \) then \( X \models A \)

iff If \( X \nvdash A \) then \( X \nmodels A \)

If \( \langle X, A \rangle \) is invalid then there is an open tree for \( \langle X, \sim A \rangle \).

Assumption: The argument \( \langle X, A \rangle \) is invalid.

There is then an evaluation \( v \), such that \( X_1, \ldots X_n \) are true on \( v \) and \( A \) is false on \( v \). In other words, \( \sim A \) is true on \( v \).

The root for \( \langle X_1, \ldots X_n, \sim A \rangle \) therefore doesn’t close.

To show Soundness, we can show that whenever we apply our rules on the root for \( \langle X_1, \ldots X_n, \sim A \rangle \) the tree will remain open.

To show this we must show that every rule of the tree method is truth preserving downwards. That is, whenever we apply rule it leads to there being at least one open branch where all the formulas are true on \( v \).
Soundness and completeness

**Soundness**

If $X \vdash A$ then $X \models A$

iff If $X \nvDash A$ then $X \nvDash A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \neg A \rangle$.

\[\neg\neg A\]

\[\begin{array}{c}
\neg A \\
\frac{}{A}
\end{array}\]

$v(\neg A) = \begin{cases} 1 & \text{if } v(A) = 0 \\ 0 & \text{otherwise} \end{cases}$

If $\neg\neg A$ is true on $v$ then $A$ is true $v$.

The rule for **negation** is truth preserving downwards.
Soundness and completeness

**Soundness**

If \( X \vdash A \) then \( X \models A \)

iff If \( X \not\vdash A \) then \( X \not\models A \)

If \( \langle X, A \rangle \) is invalid then there is an open tree for \( \langle X, \sim A \rangle \).

\[
\begin{align*}
(A \& B) & \quad \sim(A \& B) \\
A & \quad \sim A \\
B & \quad \sim B
\end{align*}
\]

\[
v(A \& B) = \begin{cases} 
1 & \text{if } v(A) = 1 \text{ and } v(B) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

If \( (A \& B) \) is true on \( v \) then \( A \) is true \( v \) and \( B \) is true on \( v \)

If \( \sim(A \& B) \) is true on \( v \) then \( \sim A \) is true \( v \) or \( \sim B \) is true on \( v \)

The rules for **conjunction** are truth preserving downwards.
Soundness and completeness

**Soundness**
If \( X \vdash A \) then \( X \models A \)
iff If \( X \not\vdash A \) then \( X \not\models A \)

If \( \langle X, A \rangle \) is invalid then there is an open tree for \( \langle X, \sim A \rangle \).

\[
(A \lor B) \quad \sim(A \lor B) \\
\text{ } \quad \text{ } \quad \quad \sim A \\
\text{ } \quad \text{ } \quad \quad \sim B
\]

\[
v(A \lor B) = \begin{cases} 
0 & \text{if } v(A) = 0 \text{ and } v(B) = 0 \\
1 & \text{otherwise}
\end{cases}
\]

If \( (A \lor B) \) is true on \( v \) then \( A \) is true on \( v \) or \( B \) is true on \( v \)

If \( \sim(A \lor B) \) is true on \( v \) then \( \sim A \) is true on \( v \) and \( \sim B \) is true on \( v \)

The rules for **disjunction** are truth preserving downwards.
Soundness and completeness

**Soundness**

If $X \vdash A$ then $X \models A$

If $X \not\models A$ then $X \not\models A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \sim A \rangle$.

If $(A \supset B)$ is true on $v$ then $\sim A$ is true $v$ or $B$ is true on $v$

If $\sim(A \supset B)$ is true on $v$ then $A$ is true $v$ and $\sim B$ is true on $v$

The rules for **conditional** are truth preserving downwards.
Soundness and completeness

**Soundness**
If $X ⊢ A$ then $X \models A$

iff If $X \not\models A$ then $X \not\models A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \sim A \rangle$.

\[(A \equiv B) \quad \sim(A \equiv B)\]
\[\begin{array}{ccc}
& A & \sim A \\
A & & \sim B \\
B & \sim B & \\
& B & \sim B
\end{array}\]

\[
v(A \equiv B) = \begin{cases} 
1 & \text{if } v(A) = 1 \text{ and } v(B) = 1 \\
1 & \text{if } v(A) = 0 \text{ and } v(B) = 0 \\
0 & \text{otherwise}
\end{cases}
\]

If $(A \equiv B)$ is true on $v$ then $A$ is true $v$ and $B$ is true on $v$ or $\sim A$ is true $v$ and $\sim B$ is true on $v$

If $\sim(A \equiv B)$ is true on $v$ then $\sim A$ is true $v$ and $B$ is true on $v$ or $A$ is true $v$ and $\sim B$ is true on $v$

The rules for **biconditional** are truth preserving downwards.
Soundness and completeness

### Soundness
If $X \vdash A$ then $X \models A$

iff If $X \not\vDash A$ then $X \not\models A$

If $\langle X, A \rangle$ is invalid then there is an open tree for $\langle X, \sim A \rangle$.

**Assumption**: The argument $\langle X, A \rangle$ is invalid.

There is then an evaluation $v$, such that $X_1, \ldots X_n$ is true on $v$ and $A$ is false on $v$. In other words, $\sim A$ is true on $v$.

The root for $\langle X_1, \ldots X_n, \sim A \rangle$ therefore doesn’t close.

We have seen that given **Assumption**, when the tree is completed, there is at least one branch where all formulas are true on $v$.

A branch closes iff both a formula $B$ and its negation $\sim B$ occur. But there is no evaluation such that both $B$ and $\sim B$ is true. Therefore, there is at least one open branch. That is, $X \not\vDash A$. We have shown **Soundness**.
Soundness and completeness

**Completeness**
If $X \models A$ then $X \vdash A$

iff If $X \not\models A$ then $X \not\vdash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.
Soundness and completeness

**Completeness**

If $X \models A$ then $X \vdash A$

iff If $X \not\models A$ then $X \not\vdash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

**Assumption:** There is an open tree for $\langle X, \sim A \rangle$. 
Soundness and completeness

**Completeness**

If $X \models A$ then $X \vdash A$

iff If $X \nvDash A$ then $X \nvDash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

**Assumption:** There is an open tree for $\langle X, \sim A \rangle$.

There is then at least one open branch $b$ where all formulas are resolved and $X_1, \ldots X_n, \sim A$ occurs in $b$. 
Soundness and completeness

**Completeness**
If $X \models A$ then $X \vdash A$

iff If $X \not\models A$ then $X \not\vdash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

**Assumption:** There is an open tree for $\langle X, \sim A \rangle$.

There is then at least one open branch $b$ where all formulas are resolved and $X_1, \ldots X_n, \sim A$ occurs in $b$.

There is then no formula $B$ and $\sim B$ occurring in $b$. There is then an evaluation satisfying all the atomic and negated atomic formulas in $b$. 
### Soundness and completeness

#### Completeness

If $X \vDash A$ then $X \vdash A$  

iff If $X \nvdash A$ then $X \nvdash A$  

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

#### Assumption: There is an open tree for $\langle X, \sim A \rangle$.

There is then at least one open branch $b$ where all formulas are resolved and $X_1, \ldots X_n, \sim A$ occurs in $b$.

There is then no formula $B$ and $\sim B$ occuring in $b$. There is then an evaluation satisfying all the atomic and negated atomic formulas in $b$.

To show **Completeness**, we can show that every rule is truth preserving upwards: whenever we have applied a rule, there is an evaluation $v$, such that if at least one of the resulting formulas are true on $v$ so is the resolved formula.
Soundness and completeness

Completeness
If \( X \models A \) then \( X \vdash A \)
iff If \( X \not\models A \) then \( X \not\vdash A \)
If there is an open tree for \( \langle X, \sim A \rangle \) then \( \langle X, A \rangle \) is invalid.

\[
\sim \sim A \quad \quad \\
\downarrow \quad \quad \quad \quad \\
A
\]

\[
\nu(\sim A) = \begin{cases} 
1 & \text{if } \nu(A) = 0 \\
0 & \text{otherwise}
\end{cases}
\]

If \( A \) is true on \( \nu \) then \( \sim \sim A \) is true \( \nu \)
The rule for \textbf{negation} is truth preserving upwards.
Soundness and completeness

**Completeness**

If $X \models A$ then $X \vdash A$

iff If $X \not\models A$ then $X \not\vdash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

\[
\begin{array}{c|cc}
(A \& B) & \sim(A \& B) \\
\hline
A & \sim A & \sim B \\
B
\end{array}
\]

\[
v(A \& B) = \begin{cases} 
1 & \text{if } v(A) = 1 \text{ and } v(B) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

If $A$ is true $v$ and $B$ is true on $v$ then $(A \& B)$ is true on $v$

If $\sim A$ is true $v$ or $\sim B$ is true on $v$ then $\sim(A \& B)$ is true on $v$

The rules for **conjunction** are truth preserving upwards.
Soundness and completeness

**Completeness**

If \( X \models A \) then \( X \vdash A \)

iff If \( X \not\models A \) then \( X \not\vdash A \)

If there is an open tree for \( \langle X, \sim A \rangle \) then \( \langle X, A \rangle \) is invalid.

\[
\begin{align*}
(A \lor B) & \quad \sim(A \lor B) \\
A & \quad B \\
\sim A & \\
\sim B
\end{align*}
\]

\[
v(A \lor B) = \begin{cases} 0 & \text{if } v(A) = 0 \text{ and } v(B) = 0 \\ 1 & \text{otherwise} \end{cases}
\]

If \( A \) is true \( v \) or \( B \) is true on \( v \) then \( (A \lor B) \) is true on \( v \)

If \( \sim A \) is true \( v \) and \( \sim B \) is true on \( v \) then \( \sim(A \lor B) \) is true on \( v \)

The rules for **disjunction** are truth preserving upwards.
Completeness
If $X \models A$ then $X \vdash A$

iff If $X \nvdash A$ then $X \nvdash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

$$v(A \supset B) = \begin{cases} 
0 & \text{if } v(A) = 1 \text{ and } v(B) = 0 \\
1 & \text{otherwise}
\end{cases}$$

If $\sim A$ is true $v$ or $B$ is true on $v$ then $(A \supset B)$ is true on $v$

If $A$ is true $v$ and $\sim B$ is true on $v$ then $\sim (A \supset B)$ is true on $v$

The rules for **conditional** are truth preserving upwards.
Soundness and completeness

**Completeness**

If $X \models A$ then $X \vdash A$

iff If $X \not\models A$ then $X \not\vdash A$

If there is an open tree for $\langle X, \sim A \rangle$ then $\langle X, A \rangle$ is invalid.

$(A \equiv B) \quad \sim(A \equiv B)$

\[
\begin{array}{c}
A & \sim A \\
\hline
\sim A & A \\
B & \sim B \\
\end{array}
\]

$v(A \equiv B) = \begin{cases} 
1 & \text{if } v(A) = 1 \text{ and } v(B) = 1 \\
1 & \text{if } v(A) = 0 \text{ and } v(B) = 0 \\
0 & \text{otherwise}
\end{cases}$

If $A$ is true $\nu$ and $B$ is true on $\nu$ or $\sim A$ is true $\nu$ and $\sim B$ is true on $\nu$ then $(A \equiv B)$ is true on $\nu$

If $\sim A$ is true $\nu$ and $B$ is true on $\nu$ or $A$ is true $\nu$ and $\sim B$ is true on $\nu$ then $\sim(A \equiv B)$ is true on $\nu$

The rules for **biconditional** are truth preserving upwards.
Soundness and completeness

Completeness
If \( X \models A \) then \( X \vdash A \)

iff If \( X \not\models A \) then \( X \not\models A \)

If there is an open tree for \( \langle X, \sim A \rangle \) then \( \langle X, A \rangle \) is invalid.

Assumption: There is an open tree for \( \langle X, \sim A \rangle \).

There is then at least one open branch \( b \) where all formulas are resolved and \( X_1, \ldots, X_n, \sim A \) occurs in \( b \).

There are rules for resolving all formulas that aren’t atomic and negated atomic, so since \( b \) is completed there is an evaluation \( v \), such that the atomic and negated atomic in \( b \) are true on \( v \).

We have seen that given Assumption there is an evaluation \( v \) such that \( X_1, \ldots, X_n, \sim A \) are all true on \( v \). Because, if the atomic and negated atomic in \( b \) are true on \( v \) then all the formulas of \( b \) are true on \( v \). Therefore, there is an evaluation where \( X_1, \ldots, X_n \) are true and \( A \) is false. That is, \( X \not\models A \). We have shown Completeness.