Logic: **Propositional Tree Rules** (Handout)

Here A and B can be any formula you like (e.g., A might be the formula (P&Q) → R).

Double Negation (¬¬)

```
¬¬A
 |
 A
```

Conjunction (&)

```
  A & B
    |
    A
    B
```

Negated Conjunction (¬&)

```
¬(A & B)
 /    \
¬A    ¬B
```

Disjunction (∨)

```
 A ∨ B
 /   \
A     B
```

Negated Disjunction (¬∨)

```
¬(A ∨ B)
    |
   ¬A
   ¬B
```

Conditional (→)

```
  A → B
  /    \
¬A      B
```

Negated Conditional (¬→)

```
¬(A → B)
    |
    A
   ¬B
```

Biconditional (↔)

```
  A ↔ B
  /    \
A      ¬A
B      ¬B
```

Negated Biconditional (¬↔)

```
¬(A ↔ B)
  /    \
 A     ¬A
¬B      B
```

Using these nine simple “algorithmic” rules you can:

(i) **prove** any **valid** sequent;

(ii) construct a **counter-example** for any **invalid** sequent.

Flow Diagram for Constructing Truth Trees

(Step 1) List the elements of the initial list, adding a line number for each.

⇓

⇒ (Step 2) Pick a formula in an open branch and apply one of the tree rules and write down the result, adding **lines numbers** and **annotations**.

⇑ ⇓

(Step 3) Put a tick (✓) next to the formula used.

⇑ ⇓

(Step 4) Examine any branch. If it contains a formula A and its negation ¬A, then place **-X-** after the final formula in the branch. That branch is **closed**.

⇑ ⇓

(Step 5) If every branch is closed, then the tree is **CLOSED**. **STOP**

If some branch is not closed, …

⇑ ⇓

(Step 6) if every formula in every branch has been used and some branch is still open, then the tree is **FINISHED** and **OPEN**. Then **STOP**.

If some formula has not been used, ….

⇑ ⇓

⇐ (Step 7) go back to (Step 2)