nested quantifiers

- **Bound variable names don't matter**
  \[ \forall x \exists y P(x, y) = \forall a \exists b P(a, b) \]

- **Positions of quantifiers can sometimes change**
  \[ \forall x (Q(x) \land \exists y P(x, y)) = \forall x \exists y (Q(x) \land P(x, y)) \]

- **But: order is important...**

predicate with two variables

expression when true when false
---
\( \forall x \forall y P(x, y) \) when true when false
\( \exists x \forall y P(x, y) \) when true when false
\( \forall x \exists y P(x, y) \) when true when false
\( \exists x \forall y P(x, y) \) when true when false
\( \forall x \exists y P(x, y) \) when true when false

---

If the tortoise walks at a rate of one node per step, and the hare walks at a rate of two nodes per step, then the distance between them increases by one node per step.

If the tortoise is on node \( x \), and the hare is on node \( 2x \), then the distance between them increases by one node per step.
Quantification with two variables

<table>
<thead>
<tr>
<th>expression</th>
<th>when true</th>
<th>when false</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall x \forall y P(x,y)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\exists x \exists y P(x,y)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\forall x \exists y P(x,y)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\exists x \forall y P(x,y)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Logical inference

- So far we’ve considered:
  - How to understand and express things using propositional and predicate logic
  - How to compute using Boolean (propositional) logic
  - How to show that different ways of expressing or computing them are equivalent to each other

- Logic also has methods that let us infer implied properties from ones that we know
  - Equivalence is only a small part of this

Applications of logical inference

- Software Engineering
  - Express desired properties of program as set of logical constraints
  - Use inference rules to show that program implies that those constraints are satisfied

- Artificial Intelligence
  - Automated reasoning

- Algorithm design and analysis
  - e.g., Correctness, Loop invariants.

- Logic Programming, e.g. Prolog
  - Express desired outcome as set of constraints
  - Automatically apply logic inference to derive solution
Proofs

- Start with hypotheses and facts
- Use rules of inference to extend set of facts
- Result is proved when it is included in the set

An inference rule: Modus Ponens

- If $p$ and $p \rightarrow q$ are both true then $q$ must be true
- Write this rule as:
  \[
  \frac{p, p \rightarrow q}{\therefore q}
  \]
- Given:
  - If it is Monday then you have a 311 class today.
  - It is Monday.
- Therefore, by modus ponens:
  - You have a 311 class today.

Proofs can use equivalences too

Show that $\neg p$ follows from $p \rightarrow q$ and $\neg q$

1. $p \rightarrow q$ given
2. $\neg q$ given
3. $\neg q \rightarrow \neg p$ contrapositive of 1
4. $\neg p$ modus ponens from 2 and 3

Inference rules

- Each inference rule is written as:
  \[
  \frac{A, B}{\therefore C, D}
  \]
  ...which means that if both $A$ and $B$ are true then you can infer $C$ and you can infer $D$.
  - For rule to be correct $(A \land B) \rightarrow C$ and $(A \land B) \rightarrow D$ must be tautologies
- Sometimes rules don’t need anything to start with. These rules are called axioms:
  - e.g. Excluded Middle Axiom
    \[
    \frac{}{\therefore p \lor \neg p}
    \]

Simple propositional inference rules

Excluded middle plus two inference rules per binary connective, one to eliminate it and one to introduce it:

\[
\begin{align*}
\frac{p \land q}{\therefore p, q} & \quad \frac{p, q}{\therefore p \land q} \\
\frac{\therefore p}{} & \quad \frac{\therefore p \lor q, q \lor p}{\therefore p \lor \neg p} \\
\frac{p \lor q, \neg p}{\therefore q} & \quad \frac{\therefore p \lor q}{\therefore p \rightarrow q}
\end{align*}
\]

Direct Proof Rule
Not like other rules
important: applications of inference rules

• You can use equivalences to make substitutions of any sub-formula.

• Inference rules only can be applied to whole formulas (not correct otherwise)

  e.g. 1. \( p \rightarrow q \) given
  2. \((p \lor r) \rightarrow q\) intro \(\lor\) from 1.

  Does not follow! e.g. \( p=F, q=F, r=T \)

direct proof of an implication

• \( p \Rightarrow q \) denotes a proof of \( q \) given \( p \) as an assumption

• The direct proof rule:
  If you have such a proof then you can conclude that \( p \rightarrow q \) is true

Example:

1. \( p \) assumption
2. \( p \lor q \) intro for \(\lor\) from 1
3. \( p \rightarrow (p \lor q) \) direct proof rule