Predicate Logic

Robert Keller
February 2013
Examples of Predicate Logic Formulas

- \( p(x) \)
- \( q(a) \)
- \( r(x, f(a)) \)
- \( p(x) \rightarrow r(x, f(a)) \)
- \( q(a) \rightarrow \exists x \ r(x, f(a)) \)
- \( \forall y (q(y) \rightarrow \exists x \ r(x, f(y)) \)
- \( \exists x \ q(z) \land \forall y (q(y) \rightarrow \exists x \ r(x, f(y)) \)
Predicate Calculus Abstract Syntax

- E is the start symbol
- E ::= A // Atom or atomic formula
  - ¬E // Negation (not)
  - E ∧ E // Conjunction (and)
  - E ∨ E // Disjunction (or)
  - E → E // Implication (implies)
  - E ↔ E // If-and-only-if
  - ⊥ // Bottom
  - T // Top
  - ∀V E // Universally-quantified formula
  - ∃V E // Existentially-quantified formula

- V means variable symbol (see next page)
- Precedence, tightest first: ∀ ∃ ¬ ∧ ∨ → ↔
- Atomic formula (A) requires a more complex production
Atomic Formulas

• Informally, an atomic formula is the smallest unit that evaluates to a truth value \{T or F\}, once individuals are substituted for arguments.

• Atomic formulas don’t contain connectives or quantifiers.

• They are analogous to proposition symbols in proposition logic.
Examples of Predicate Logic Formulas

- $p(x)$
- $q(a)$
- $r(x, f(a))$
- $p(x) \rightarrow r(x, f(a))$
- $q(a) \rightarrow \exists x \ r(x, f(a))$
- $\forall y \ (q(y) \rightarrow \exists x \ r(x, f(y)))$
- $\exists x \ q(z) \land \forall y \ (q(y) \rightarrow \exists x \ r(x, f(y)))$
Atomic Formula Syntax

A ::= P(L)  // Predicate applied to list of terms
L ::= T | T ‘,’ L  // List of terms
T ::= V | C | F(L)  // Term

V ::= ‘x’ | ‘y’ | ‘z’ | ...  // Variable symbols
P ::= ‘p’ | ‘q’ | ‘r’ | ...  // Predicate symbols
C ::= ‘a’ | ‘q’ | ‘c’ | ...  // Constant symbols
F ::= ‘f’ | ‘g’ | ‘h’ | ...  // Function symbols

Some predicates and functions may be abbreviated in infix form, e.g.
  = < < ... will be infix predicate symbols
  + * / ... will be infix function symbols
We will not bother with a special grammar for these, although it can
be done.
Arities

- In addition, predicate and function symbols have an “arity” (number of arguments) which we don’t show explicitly.

- Most of the time, we will not overload the symbols, but rather assume a fixed arity for a given symbol.

- So we will not typically use both $f(a, b)$ (2-ary) and $f(a)$ (1-ary), for example, in the same discussion.
Quantifiers

• $\forall$ is a “wholesale” version of $\land$

• $\exists$ is a “wholesale” version of $\lor$

• $\forall x \ P(x)$ is like $P(x_0) \land P(x_1) \land P(x_2) \land \ldots$

  except that we don’t know how many elements there are.
Quantifiers

- $\forall x \ P(x)$ is like $P(x_0) \land P(x_1) \land P(x_2) \land \ldots$

  just as $\sum_x f(x)$ is like $f(x_0) + f(x_1) + f(x_2) + \ldots$
First-Order Logic

- Our focus here is *first-order logic* (FOL) or *first-order predicate calculus* (FOPC).
  

- Second-, and higher-, order logic would include quantification over predicates and functions. It is not within the scope of this course, but may get brief mention.
  
“Term”: a term to remember

- A **term** designates an individual in a **domain** (to be introduced later).

- A term can be:
  - A **constant symbol**, naming the individual
  - A **variable symbol**, naming a generic individual
  - A **function** applied to some terms as arguments, the result of which is **the individual the function produces**.
Examples of Terms

- b  constant symbol
- y  variable
- f(b, y)  function applications
- g(h(b), c, h(y))
- g(a, b, g(a, b, c))

- Atomic formulas are not terms, although they look similar.
- Atomic formulas can contain terms as arguments.
Examples of Atomic Formulas

- $p(b)$
- $q(y)$
- $p(f(b, y))$
- $r(a, g(h(b), c, h(y)))$

The arguments must be terms.
Examples of “Literals”

- A **literal** is an atomic formula, or the **negation** of an atomic formula.
  - \( p(b) \)
  - \( \neg q(y) \)
  - \( \neg p(f(b, y)) \)
  - \( r(a, g(h(b), c, h(y))) \)

- Literals become important in resolution theorem proving, discussed later.
Examples of **Quantifier-Free** Formulas

- Any atomic formula
- \( p(b) \lor p(c) \)
- \( p(y) \land q(y) \)
- \( p(f(b, y)) \to q(y) \)
- \( \neg r(a, g(h(b), c, h(y))) \)
Examples of Formulas

- Any Quantifier-Free Formula
- \( \exists x \, p(x) \)
- \( \forall y \, (p(y) \land q(y)) \)
- \( \forall y \, \exists x \, (p(f(x, y)) \rightarrow q(y)) \)
- \( \forall x \, (p(f(x, y)) \lor q(x)) \)
- \( \forall y \, (q(y) \rightarrow \exists x \, p(f(x, y))) \)
Structural Summary

- **Formula**
  - has (*
  - contains

- **Quantifier-Free Formula**
  - is-a

- **Atomic Formula**
  - contains

- **Predicate symbol**

- **Term**
  - has
  - > 1
  - ≤ 1
  - *
  - *

- **Function symbol**

- **Constant symbol**

- **Variable**

* means “zero or more”
“Well-Formed Formulas”

• WFFs (sometimes pronounced “woofs”)

• We don’t deal with formulas that are *not* well-formed. In the second part of the course, we discuss grammars and parsing.

• See
  http://en.wikipedia.org/wiki/Well-formed_formula
Preview of Semantics

• We will give details of semantics later on. However, a preview is helpful to understand certain syntactic considerations.

• Predicate logic can be used to describe characteristics of particular kinds of structures, such as sets with certain algebraic properties or real-world objects.

• The particular structures are called “Interpretations”. Interpretations which make a set of formulas true are called “models”.
Example:

Interpretation for the natural numbers

- The intended domain is \{0, 1, 2, 3, \ldots\}.
- There is a constant symbol 0.
- There is a 1-ary function s (successor).
  Informally, \( s(n) = n+1 \).
- There is a 2-ary predicate \( = \) (equals).
Individuals in the Domain

- Any individual natural number can be described by a term

- 0 describes the number ‘0’
- s(t) where t is a term, describes 1+ the number described by t.

- s(0) describes ‘1’
- s(s(s(s(s(s(s(s(s(0)))))))))) describes ____?
Some formulas for this interpretation

- \( \forall n \neg (s(n) = 0) \)
  
  “0 is not the successor of anything”.

- \( \forall m (\neg (m = 0) \rightarrow (\exists n) (m = s(n))) \)
  
  “Anything other than 0 is the successor of something”.

- \( \forall m \forall n ((s(m) = s(n)) \rightarrow m = n) \)
  
  “Successor is a one-to-one function”.

22
Example:

Interpretations for “Groups”

- The domain is non-empty.

- The domain can be finite or infinite.

- There is a constant symbol $e$ (identity element of the group).

- There is a 2-ary function $f$ (group “multiplication”).

- There is a 2-ary predicate $=$ (equals).
Some formulas for groups

- \( \forall x \ f(e, x) = x \)  
  [e is an identity]

- \( \forall x \ \forall y \ \forall z \ f(x, f(y, z)) = f(f(x, y), z) \)  
  [f is associative]

- \( \forall x \ \exists y \ f(x, y) = e \)  
  [existence of inverse]
Examples of Groups

• Trivial group: \( \{0\} \) \( e = 0, f(0, 0) = 0 \)

• 2-element group: \( \{0, 1\} \) \( e = 0, f(x, y) = x + y \pmod 2 \)

• \( \mathbb{Z}_p \): \( \{0, 1, \ldots, p-1\} \) for any prime \( p \),
  \( e = 0, f(x, y) = x + y \pmod p \)

• Tire rotations

• Particle spins (physics)

• Rubik’s cube states

• Many others
Examples of Groups

4 elements

|ψ⟩ \rightarrow e^{-\frac{i}{\hbar}(2\pi)S_z} |ψ⟩ = -|ψ⟩

spins

cellular automaton based on a group

braids

states: 43252003274489856000 elements
Syntax Trees (or “Parse” Trees)

- We are assuming familiarity with syntax trees from CS 60.
- ∀x, ∃x are treated as if 1-ary operators.
- Example: ∀x ((∃y p(g(x, y))) ∨ q(x))
Bound and Free Variables

- A variable $\nu$ is **bound** in a formula if it occurs in a sub-tree having $\exists \nu$ or $\forall \nu$ at the root.

- Otherwise it is **free** in the formula.
Free and Bound Variable Instances

\[ \exists y \, p \quad q \]

\[ g \quad x \quad y \]

\[ \text{free} \quad \text{bound} \]
Analogues in Calculus

\[ \sum_{k=1}^{10} f(k, n) \quad \text{k is bound, n is free} \]

\[ \lim_{h \to 0} \frac{f(x + h) - f(x)}{h} \quad \text{h is bound, x is free} \]
Free and Bound Variable Instances

∀x ∨ ∃y p q

bound

∀x

∃y

p

q

bound

bound

bound

x

y

bound
Free and Bound Variable Instances

\[ \forall x \lor p \land q \]
Free and Bound Variable Instances

∀x ∨ ∃x p q x

 bound

bound

bound

bound

∀x

∃x

p

q

g

x

x

bound
Scope of Variables

- The same variable may be used more than once in a formula, with different “meanings”.

- The idea of **scope** clarifies these separate meanings.

- For a formula $\forall x \ E$, or $\exists x \ E$, the scope of $x$ extends only inside $E$, and not beyond.

- Similar to scope in programming languages
Scope Defined Inductively

- For a quantifier-free formula, the scope of each variable is the entire formula.

- For $\forall x \ E$, or $\exists x \ E$, the scope of $x$ is inside $E$, but not including inside any quantification of the same variable inside $E$.

- Example: Two distinct scopes of variable $x$:

$$
\forall x \ (p(x) \lor \exists x \ (q(x) \land r(x)) \lor s(x, y))
$$
Renaming Variables

• Although not required, it is better to avoid using the same variable for more than one scope.

• Bound variables can be renamed to a fresh variable to accomplish this. All instances within the scope must be renamed.

• Example: One of the x’s renamed to u:

\[
\forall x (p(x) \lor \exists u (q(u) \land r(u)) \lor s(x, y))
\]
Improper renaming

- $\forall x \ (p(x) \lor \exists x \ (q(x) \land r(x)) \lor s(x, y))$

- Can’t rename just one of the inner $x$’s

- $\forall x \ (p(x) \lor \exists u \ (q(u) \land r(x)) \lor s(x, y))$

- The scope of the $x$ in $r(x)$ would change.
Definition of Free and Bound Instances

• In a term and in a quantifier-free formula, every instance of a variable is free.

• If \( \varphi \) is a formula, then any free instances of a variable \( x \) are bound in \( \forall x \varphi \) and \( \exists x \varphi \).

• The free instances of variables in \( \varphi \) and \( \psi \) remain free in \( \neg \varphi \), \( \varphi \lor \psi \), \( \varphi \land \psi \), and \( \varphi \rightarrow \psi \).

• The bound instances of variables in \( \varphi \) and \( \psi \) remain bound in \( \neg \varphi \), \( \varphi \lor \psi \), \( \varphi \land \psi \), and \( \varphi \rightarrow \psi \).
Substitutability Restriction

- We are going to need to be able to substitute terms for free variables in various formulas.

- While this is easy syntactically, there is a semantic restriction that must be observed:
  - In substituting a term for a variable within a formula, no variables within the term can become bound as a result of the substitution.
  - If \( t \) is a term, \( v \) is a variable, and \( F \) is a formula, and the above restriction applies, we say that
    
    \[ \text{“} t \text{ is free to replace } v \text{ in } F \text{”} \]
    
    (or more conventionally, \[ \text{“} t \text{ is free for } v \text{ in } F \text{”} \])
Non-Substitutability Example

\[ \forall x (p(g(x, y)) \lor q(x)) \]

\( f(x) \) is **not** free to replace \( y \) in \( \forall x (p(g(x, y)) \lor q(x)) \)
Non-Substitutability Example

\[ \forall x \ (p(g(x, f(x))) \lor q(x)) \]

\(\text{bound} \) (was free before substitution)
Example of Violation

• ∃x x<y
• We don’t allow substitution of x for y because the meaning would change:
• ∃x x<x
Substitution Notation

• If $t$ is a term, $v$ is a variable, and $E$ is a formula, and $t$ is free to replace $v$ in $E$

then by

$$E[t/v]$$

we mean the result of substituting $t$ for every free occurrence of $v$ in $E$. (We leave the bound occurrences of $v$ as they were.)

This notation and substitution itself are to be used only when the substitutability restriction applies.

Note: $[ / ]$ is meta-syntax; these symbols do not appear in the resulting formula.
Substitution Notation Example

Let $E$ be the formula

$$\forall x \ (p(g(x, y)) \lor q(x))$$

Let $v$ be the variable $y$.

Let $t$ be the term $f(z)$.

$f(z)$ is free to replace $y$ in $E$.

$E[f(z)/y]$ is $\forall x \ (p(g(x, f(z))) \lor q(x))$. 
Substitution Notation Example

Let $E$ be the formula

$$\forall x \ (p(g(x, y)) \lor q(x))$$

Let $v$ be the variable $x$.

Let $t$ be the term $f(y)$.

$f(x)$ is free to replace $x$ in $E$ (vacuously) because there are no free instances of $x$.

$E[f(x)/x]$ is the same as $E$; there are no free instances of $x$ in $E$. 
A simpler way of writing substitutions

If \( E \) is a formula, then \( E(x) \) identifies any \textit{free} occurrences of \( x \) in \( E \). (There might not be any.)

If \( t \) is a term free for \( x \) in \( E \), then \( E(t) \) is the result of substituting \( t \) for all free instances of \( x \).
Example

• Suppose $E$ is $\forall x \ (p(g(x, y)) \lor q(y))$

• Then $E(y)$ has $y$ identified with the two occurrences of $y$.

• $f(z)$ is free for $y$ in $E$.

• $E(f(z))$ is $\forall x \ (p(g(x, f(z))) \lor q(f(z)))$
Syntax vs. Semantics

- Predicate logic proofs, in a system such as natural deduction, focus on syntax: each formula in the derivation is mechanically-checkable to be derivable from earlier formulas using only the given rules.

- The semantics or meaning of a formula is determined by separate considerations. Each formula is making a statement about some kind of underlying structure.
Why Separate Syntax from Semantics?

• Reasoning about semantics is often very complex.

• Reasoning syntactically allows reasoning without revisiting semantic details at every step.
Natural Deduction Rules for Predicate Logic
Natural Deduction Rules

- We need introduction and elimination rules for both:
  - $\forall$
  - $\exists$
- These will be added to our propositional natural deduction rules.
∀-Elimination Rule ∀E

- \[ \frac{∀x \, \varphi}{∀E} \frac{∀E}{\varphi[t/x]} \]

where \( t \) is any term that is free to replace \( x \) in \( \varphi \).

- **What the rule says:**

  *If* we have derived a universally-quantified formula \( \varphi \),
  *then* the formula \( \varphi \) with any (appropriately-qualified) **specific instance** of \( x \) substituted for \( x \) is also derivable.
Two ways of writing

1. $\forall x \varphi \forall E \varphi[t/x]$

2. $\forall x \varphi(x) \forall E \varphi(t)$
Why the Substitution Qualification is Necessary

•  \( \forall x \, \varphi \) \hspace{1cm} \forall E \hspace{1cm} \varphi[t/x] \hspace{1cm} \\

where \( t \) is any term that is free to replace \( x \) in \( \varphi \).

• Correct example: \( z \) is free to replace \( x \) in \( \exists y \, p(y, x) \)
  1. \( \forall x \, \exists y \, p(y, x) \) \hspace{1cm} Premise
  2. \( \exists y \, p(y, z) \) \hspace{1cm} \forall E 1 \hspace{1cm} (substituting \( z \) for \( x \))

• Incorrect example: \( y \) is not free to replace \( x \) in \( \exists y \, p(y, x) \)
  1. \( \forall x \, \exists y \, p(y, x) \) \hspace{1cm} Premise
  2. \( \exists y \, p(y, y) \) \hspace{1cm} \forall E 1 \hspace{1cm} (substituting \( y \) for \( x \))

• For instance, \( p \) could be \( > \) in the domain of natural numbers.
∀-Introduction Rule (∀I)

- This rule uses a sub-derivation, with no formula assumed, but with a fresh variable introduced.

\[
\begin{array}{c}
\text{Fresh } x_0 \\
\vdots \\
\vdots \\
\vdots \\
\varphi[x_0/x] \\
\hline
\forall x \varphi
\end{array}
\]

- \( x_0 \) is a “fresh” variable otherwise unused in the proof.
- \( x_0 \) must be free to replace \( x \) in \( \varphi \), but since \( x_0 \) is “fresh”, this should never be an issue. It can’t become bound.
Another way of writing

\[ \forall x_0 \varphi(x_0) \]

\[
\begin{array}{c}
\text{Fresh } x_0 \\
\vdots \\
\varphi(x_0)
\end{array}
\]

\[ \forall x \varphi \]
∀-Introduction Rule

- **What this rule says:**
  
  If we have argued to derive a term \( \varphi[x_0/x] \) where \( x_0 \) represents a **totally arbitrary** value of \( x \), then we are justified in concluding \( \forall x \varphi \).

- The key is the word “arbitrary”; there can be **no constraints** attached to \( x_0 \).

- Note: Once the conclusion \( \forall x \varphi \) is drawn, \( x_0 \) is **discharged** and cannot be further used outside the box.
Use $\forall I$ working backward

Unless your **goal** is proving something of the form $\forall x \ldots$ you won’t know to open a box with a fresh variable.
∀E ∀I Example

Derive $\forall x \ p(x) \vdash \forall y \ p(y)$

$\forall x \ p(x)$  \hspace{1cm} \text{Premise}

$\forall y \ p(y)$  \hspace{1cm} \text{Desired conclusion}

Work backward
### ∀E ∀I Example

Derive \( \forall x \, p(x) \vdash \forall y \, p(y) \)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>( \forall x , p(x) )</td>
<td>Premise</td>
</tr>
<tr>
<td>2.</td>
<td>( y_0 )</td>
<td>Fresh var</td>
</tr>
<tr>
<td>3.</td>
<td>( p(y_0) )</td>
<td>1, ∀E</td>
</tr>
<tr>
<td>4.</td>
<td>( \forall y , p(y) )</td>
<td>2-3, ∀I</td>
</tr>
</tbody>
</table>
∀E ∀I Example

Derive $\forall x (p(x) \rightarrow q(x))$, $\forall x p(x) \mid \rightarrow \forall x q(x)$

1. $\forall x (p(x) \rightarrow q(x))$  Premise
2. $\forall x p(x)$  Premise
3. $x_0$  Fresh var
4.  
5. $q(x_0)$  
6. $\forall x q(x)$  3-6, ∀I
\[ \forall E \setminus \forall I \text{ Example} \]

Derive \( \forall x \ (p(x) \rightarrow q(x)), \forall x \ p(x) \vdash \forall x \ q(x) : \]

1. \( \forall x \ (p(x) \rightarrow q(x)) \) \quad \text{Premise}

2. \( \forall x \ p(x) \) \quad \text{Premise}

3. \( x_0 \) \quad \text{Fresh var}

4. \( p(x_0) \rightarrow q(x_0) \) \quad 1, \forall E

5. \( p(x_0) \) \quad 2, \forall E

6. \( q(x_0) \) \quad 4, 5 \rightarrow E

7. \( \forall x \ q(x) \) \quad 3-6, \forall I
∀E ∀I English Equivalent

- Derive ∀x (p(x) → q(x)), ∀x p(x) ⊢ ∀x q(x):

- “Assume ∀x (p(x) → q(x)) and ∀x p(x).

Let x₀ be an arbitrary element. [open box]

From the first assumption p(x₀) → q(x₀), and from the second p(x₀), hence also q(x₀) by modus ponens.

Since x₀ was chosen arbitrarily, q(x₀) gives us [close box] ∀x q(x).”
∀E ∀I Example

Derive ∀x ∀y p(x, y) ⊢ ∀y ∀x p(x, y)

∀x ∀y p(x, y) Premise

∀y ∀x p(x, y)

Where ∀I is to be used, work backward.
### ∀E ∀I Example

Derive $\forall x \forall y \ p(x, y) \vdash \forall y \forall x \ p(x, y)$

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\forall x \forall y \ p(x, y)$</td>
<td>Premise</td>
</tr>
<tr>
<td>2</td>
<td>$y_0$</td>
<td>Fresh</td>
</tr>
<tr>
<td>3</td>
<td>$x_0$</td>
<td>Fresh</td>
</tr>
<tr>
<td>4</td>
<td>$\forall y \ p(x_0, y)$</td>
<td>1, ∀E</td>
</tr>
<tr>
<td>5</td>
<td>$p(x_0, y_0)$</td>
<td>4, ∀E</td>
</tr>
<tr>
<td>6</td>
<td>$\forall x \ p(x, y_0)$</td>
<td>3-5, ∀I</td>
</tr>
<tr>
<td>7</td>
<td>$\forall y \forall x \ p(x, y)$</td>
<td>2-6, ∀I</td>
</tr>
</tbody>
</table>
∃-Introduction Rule (∃I)

- \[ \frac{\varphi[t/x]}{\exists x \varphi} \] (∃I)

where t is any term that is free to replace x in \( \varphi \).

- What the rule says:

If we have exhibited a formula \( \varphi \) in which variable x is replaced by a specific instance t (a term) then we can conclude that there is an \( x \) for which the formula is true.
∃-Introduction Rule (∃I)

- $\varphi[t/x] \quad \exists I$
  
  $\exists x \varphi$

  where $t$ is any term that is free to replace $x$ in $\varphi$.

- In essence, this rule loses information, by replacing knowledge of a specific $x$ for which is true with the statement that there is some such $x$.

- It is analogous to rule $\lor$-Introduction.
Another way of writing

• $\varphi(t) \quad \exists I$
  $\exists x \varphi(x)$
Why would you want to lose information?

- For one thing, the specific term derived might not be “exportable”; it could depend on some fresh variable introduced inside the box, which doesn’t make sense outside.
∀E ∃I Example

- Assume there is a constant symbol a.
- Derive $\forall x \ p(x) \vdash \exists x \ p(x)$ :

1. $\forall x \ p(x)$  
   Premise
2. $p(a)$  
   1, ∀E
3. $\exists x \ p(x)$  
   2, ∃I
The previous example is rare.

- As with ∨ Introduction,

∃ Introduction is almost never the last line of a proof when the premise and conclusion are equivalent.
Slight Controversy

What if there are no constant symbols? Use a variable instead.

Derive $\forall x \ p(x) \vdash \exists x \ p(x)$:

1. $\forall x \ p(x)$  \hspace{1cm} {Premise}
2. $p(x)$ \hspace{1cm} 1, $\forall$E
3. $\exists x \ p(x)$ \hspace{1cm} 2, $\exists$I

Note: $x$ is free to replace $x$ in $p(x)$, since nothing is bound in $p(x)$.

The legitimacy of step 2 is questionable. It amounts to assuming that there is at least one thing in the domain, i.e. a non-empty domain.

Most treatments assume this, but not all. For example, Richard Bornat, the author of JAPE does not. (Allowing empty domains is analogous to allowing 0 as a number.)
JAPE Examples

Not JAPE-Provable
(no assumption that something exists)

JAPE-Provable
(actual \( j \) means something exists, i.e. \( a \) is a constant)
JAPE Examples

This is listed in Invalid Conjectures.

JAPE Proof with actual j added
Summary

If using JAPE and you want to make the non-empty domain assumption, include “actual j” as a premise.

Alternatively, if $T$ is available, you could include $\exists x. T$, from which actual $j$ could be derived. Or you could include $\exists x. \varphi$ where $\varphi$ is any provable formula.
∃-Elimination Rule (∃E)

Here $x_0$ is a “fresh” variable otherwise unused in the proof.

$x_0$ must be free to replace $x$ in $\varphi$, but since $x_0$ is “fresh”, this should never be an issue.

This rule is analogous to $\lor$ Elimination.
∃-Elimination Rule (∃E)

What this rule says:

- Assume that we have derived ∃x ϕ (on left). We can make use of this fact by letting x₀ be an x such that ϕ[x₀/x]. There can be no other constraints on x₀. If we then derive χ from the assumption about ϕ, then we can conclude χ in general.
### Example

**Derive** $\forall x \ (p(x) \to q(x)), \exists x \ p(x) \vdash \exists x \ q(x)$:

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\forall x \ (p(x) \to q(x))$</td>
<td>Premise</td>
</tr>
<tr>
<td>2</td>
<td>$\exists x \ p(x)$</td>
<td>Premise</td>
</tr>
<tr>
<td>3</td>
<td>$p(x_0)$</td>
<td>Fresh var, Assumption</td>
</tr>
<tr>
<td>4</td>
<td>$p(x_0) \to q(x_0)$</td>
<td>1, $\forall$E</td>
</tr>
<tr>
<td>5</td>
<td>$q(x_0)$</td>
<td>3, 4, $\to$E</td>
</tr>
<tr>
<td>6</td>
<td>$\exists x \ q(x)$</td>
<td>5, $\exists$I</td>
</tr>
<tr>
<td>7</td>
<td>$\exists x \ q(x)$</td>
<td>2, 3-6, $\exists$E</td>
</tr>
</tbody>
</table>

- In the $\exists$E rule template, $\varphi$ is identified with $p(x)$, while $\chi$ is identified with $\exists x \ q(x)$.
- Try not to be confused by the fact that $\exists$ is in the conclusion. The $\exists$ in 2 is what was eliminated.
Example is tricky

This rule is the hardest for students to get right, so study the examples carefully.
∃I ⊢ ∃E Example in English

- Derive ∀x (p(x) → q(x)), ∃x p(x) ⊢ ∃x q(x):

- “Assume ∀x (p(x) → q(x)) and ∃x p(x).

Let x₀ be such that p(x₀), by the second assumption.

By the first assumption, p(x₀) → q(x₀).
Hence q(x₀) by modus ponens.

As we have exhibited an x (namely x₀) such that q(x), conclude ∃x q(x).”
**Incorrect** Proof Example

Derive $\forall x (p(x) \rightarrow q(x))$, $\exists x p(x) \vdash \exists x q(x)$:

1. $\forall x (p(x) \rightarrow q(x))$ \hspace{1cm} Premise
2. $\exists x p(x)$ \hspace{1cm} Premise
3. $x_0$ \hspace{1cm} $p(x_0)$ \hspace{1cm} $\exists E$
4. $p(x_0) \rightarrow q(x_0)$ \hspace{1cm} 1, $\forall E$
5. $q(x_0)$ \hspace{1cm} 3, 4, $\rightarrow E$
6. $q(x_0)$ \hspace{1cm} 3-5, $\exists E$
7. $(\exists x) q(x)$ \hspace{1cm} 6, $\exists I$

**Why incorrect?**
- Formulas containing $x_0$ cannot be carried outside the box.
- The box for $\exists E$ has two purposes:
  - Restricting the scope of the introduced variable $x_0$.
  - Restricting the scope of the assumption.
Caution: $\exists \mathcal{E}$

- Normally, $\exists \mathcal{E}$ can only be used to introduce a variable \textbf{once} inside a box. You \textbf{cannot} use it to introduce a \textbf{second} distinct variable in the same box.

- In other words, $\exists x \varphi$ says that \textbf{an} $x$ exists, but \textbf{not necessarily more than one} $x$.

- In contrast, you can use $\exists \mathcal{I}$ as many times as you want (not that it will always help).
## Quantifier rule summary

<table>
<thead>
<tr>
<th></th>
<th><strong>Introduction</strong></th>
<th><strong>Elimination</strong></th>
</tr>
</thead>
</table>
| \( \forall \) | \[ \begin{array}{c}
    \text{Fresh } x_0 \\
    \vdots \\
    \varphi[x_0/x] \\
\end{array} \] \[ \forall x \varphi \]
|       | \[ \forall x \varphi \] \[ \varphi[t/x] \] | \[ \forall x \varphi \] \[ \varphi[t/x] \] \[ \forall E \] 
|       | \[ \forall x \varphi \] \[ \varphi[t/x] \] \[ \forall E \] \[ (t \text{ is free to replace } x) \] |
| \( \exists \) | \[ \varphi[t/x] \] \[ \exists x \varphi \] \[ \exists E \] \[ (t \text{ is free to replace } x) \] | \[ \exists x \varphi \] \[ \varphi[x_0/x] \] \[ \vdots \] \[ x_0 \] \[ \vdots \] \[ x \] \[ \exists E \] |
Parallels

- \( \forall \) Elimination similar to \( \land \) Elimination

\[
\forall x \varphi \\
\varphi[t/x] \quad \forall E \\
(t \text{ is free to replace } x)
\]

\[
\frac{F_1 \land F_2 \quad \land E}{F_1} \quad \frac{F_1 \land F_2 \quad \land E}{F_2} \quad \ldots \quad \frac{F_1 \land F_2 \land F_3 \quad \land E}{F_3}
\]
Parallels

- $\exists$ Introduction similar to $\lor$ Introduction

Think of $\exists x \varphi(x)$ as $\varphi(x_1) \lor \varphi(x_2) \lor ...$

(t is free to replace x)
Parallels

• ∀ Introduction similar to ∧ Introduction

Think of ∀x φ(x) as φ(x₀) ∧ φ(x₁) ∧ φ(x₂) ∧ ...

\[
\begin{array}{c}
\text{Fresh } x_0 \\
. \\
. \\
. \\
\varphi[x_0/x] \\
\forall x \varphi
\end{array}
\]

\[
\text{∀I}
\]

\[
\begin{array}{c}
F_1 \\
F_2 \\
F_3
\end{array}
\]

\[
F_1 \land F_2 \land F_3 \land \text{I}
\]

86
Parallels

- $\exists$ Elimination similar to $\lor$ Elimination

Think of $\exists x \varphi(x)$ as $\varphi(x_0) \lor \varphi(x_1) \lor \varphi(x_2) \lor ...$

In $\lor E$, there is one box for each disjunct.
More Tips on JAPE Usage

• Introduction rules are both **backward** rules.

• Elimination rules are both **forward** rules.

• \( \exists I \) and \( \forall E \) don’t require special handling.

• \( \exists E \) and \( \forall I \) require a term to be present for unification:
  • Both require identifying the **term to be substituted** for the quantified variable.
JAPE Example

- Cannot do this, for a reason

1: ∀x.(R(x)→S(x)), ∃y.R(y) premises

\[ \exists z. S(z) \]

("variable" means "term")

To make an \( \exists \) intro step backwards, you have to use a conclusion of the form \( \exists x. A \), and select a pseudo-assumption of the form actual i. You selected the conclusion \( \exists z. S(z) \) (which is ok), but you didn't select a hypothesis.
JAPE Example

- Cannot do this

1. $\forall x.(R(x) \rightarrow S(x))$, $\exists y.R(y)$ premises
   ...
2. $\exists z.S(z)$

(“variable” means “term”)
JAPE Example

- Start here

\[ 1: \forall x. (R(x) \rightarrow S(x)), \exists y. R(y) \text{ premises} \]

\[ \ldots \]

2: \exists z. S(z)

- Giving this:

\[ 1: \forall x. (R(x) \rightarrow S(x)), \exists y. R(y) \text{ premises} \]

\[ \boxed{\text{actual i, } R(i) \text{ assumptions}} \]

\[ \ldots \]

3: \exists z. S(z)

4: \exists z. S(z) \quad \exists \text{ elim 1.2,2-3} \]
JAPE Example

- Now can use either $\exists I$ or $\forall E$ with variable $i$.

1: $\forall x. (R(x) \rightarrow S(x)), \ \exists y. R(y)$ premises

2: actual $i, R(i)$

... assumptions

3: $\exists z. S(z)$

4: $\exists z. S(z)$ $\exists$ elim 1.2,2–3

OR

1: $\forall x. (R(x) \rightarrow S(x)), \ \exists y. R(y)$ premises

2: actual $i, R(i)$

... assumptions

3: $\exists z. S(z)$

4: $\exists z. S(z)$ $\exists$ elim 1.2,2–3
Next Steps

Two steps to closure:

1: \( \forall x. (R(x) \rightarrow S(x)), \exists y. R(y) \) premises
2: actual i, R(i)
3: \( R(i) \rightarrow S(i) \)
4: \( S(i) \)
5: \( \exists z. S(z) \)
6: \( \exists z. S(z) \)

\[ \exists \text{ intro 1,2,2-5} \]

\[ \exists \text{ elim 1,2,2-4} \]

\[ \forall \text{ elim 1,1,2,1} \]

OR

1: \( \forall x. (R(x) \rightarrow S(x)), \exists y. R(y) \) premises
2: actual i, R(i)
3: \( R(i) \rightarrow S(i) \)
4: \( S(i) \)
5: \( \exists z. S(z) \)

\[ \exists \text{ intro 1,2,2-4} \]

\[ \exists \text{ elim 1,2,2-4} \]

\[ \forall \text{ elim 1,1,2,1} \]
Other Examples

• I’ve added more examples to:

Proofs of Interest (Google Presentation)
Syllogisms

- A syllogism consists of three parts: the major premise, the minor premise, and the conclusion. In Aristotle, each of the premises is in the form "Some/all A belong to B," where "Some/All A" is one term and "belong to B" is another, but more modern logicians allow some variation. Each of the premises has one term in common with the conclusion: in a major premise, this is the major term (i.e., the predicate) of the conclusion; in a minor premise, it is the minor term (the subject) of the conclusion. For example:

  - Major premise: All humans are mortal.
  - Minor premise: Socrates is a human.
  - Conclusion: Socrates is mortal.

- Each of the three distinct terms represents a category, in this example, "human," "mortal," and "Socrates." "Mortal" is the major term; "Socrates," the minor term. The premises also have one term in common with each other, which is known as the middle term — in this example, "human."

Note: Stating a syllogism does not require validity.
Codifying Syllogisms using Predicate Logic

• Use unary predicates.
  • $S(x)$: “$x$ is an $S$”, “$x$ has an $S$”, “$x$ belongs to $S$”, etc.

• Use quantifiers for some, all
  • $\forall$  $\exists$

• Use connectives
  • $\neg$  $\rightarrow$

• Use constant symbols for individuals
Translating a Syllogism

<table>
<thead>
<tr>
<th>Statement</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>All humans are mortal.</td>
<td>$\forall x \ (H(x) \rightarrow M(x))$</td>
</tr>
<tr>
<td>Socrates is a human.</td>
<td>$H(s)$</td>
</tr>
<tr>
<td>Therefore Socrates is mortal.</td>
<td>$M(s)$</td>
</tr>
</tbody>
</table>

This syllogism happens to be valid.
Syllogistic Forms

<table>
<thead>
<tr>
<th>Statement Form</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>All S is/are/has... P.</td>
<td>$\forall x \ (S(x) \rightarrow P(x))$</td>
</tr>
<tr>
<td>Some S is P.</td>
<td>$\exists x \ (S(x) \land P(x))$</td>
</tr>
<tr>
<td>No S is P.</td>
<td>$\neg \exists x \ (S(x) \land P(x))$</td>
</tr>
<tr>
<td>Some S is not P.</td>
<td>$\exists x \ (S(x) \land \neg P(x))$</td>
</tr>
<tr>
<td>No S is not P.</td>
<td>$\neg \exists x \ (S(x) \land \neg P(x))$</td>
</tr>
<tr>
<td>All S is not P.</td>
<td>$\forall x \ (S(x) \rightarrow \neg P(x))$</td>
</tr>
</tbody>
</table>

Are any forms equivalent to one another?
Example: Translate this syllogism, then try to prove it.

- All fruit is nutritious.
- Some fruit is tasty.
- Therefore some tasty things are nutritious.
DeMorgan’s Rules for Quantifiers

- Recall DeMorgan’s rules for propositions
  - \((p \land q) \leftrightarrow \neg(\neg p \lor \neg q)\)
  - \((\neg p \lor \neg q) \leftrightarrow \neg(p \land q)\)
  - \((p \lor q) \leftrightarrow \neg(\neg p \land \neg q)\)
  - \(\neg(p \lor q) \leftrightarrow (\neg p \land \neg q)\)

- For quantifiers, we have analogous rules
  - \(\forall x P(x) \leftrightarrow \neg(\exists x \neg P(x))\)
  - \(\exists x \neg P(x) \leftrightarrow \neg\forall x P(x)\)
  - \(\exists x P(x) \leftrightarrow \neg(\forall x \neg P(x))\)
  - \(\neg\exists x P(x) \leftrightarrow \forall x \neg P(x)\)

- Note that in some cases, only one direction of implication is constructive.
Semantics of Predicate Logic

What is truth?
Interpretations of Formulas

- The **structure(s)** of interest in specific derivations are generally **not totally specified** in the system of derivation itself.

- Instead, we rely on certain formulas ("axioms") to **characterize** the properties of these structures.

- In natural deduction, these formulas will appear on the left-hand side of a sequent, or understood as lemmas.

- It can then be proved separately that the syntactic rules are in agreement with the semantics of the intended **interpretation**.
Interpretation $I = (\Delta, \mu)$

- An **interpretation** for a set of terms and formulas consists of:
  - A **domain** $\Delta$ (usually non-empty): contains all **individuals** of interest.
  - For each **constant symbol** $c$ *in the language*, an element $\mu(c) \in \Delta$.
  - For each $n$-ary **function symbol** $f$, a function $\mu(f): \Delta^n \rightarrow \Delta$.
  - For each $n$-ary **predicate symbol** $p$, a function $\mu(p): \Delta^n \rightarrow \{T, F\}$.
  - The values of $\mu$ are the values **assigned** by the interpretation.
  - The domain, $\Delta$, may also be called the “universe” or “domain of discourse”.
Constant Symbols for a Domain

• In what follows, we will assume that there is a unique constant symbol for each domain element.

• The symbol will be **identified with** the element itself.

• Example: If $\Delta = \{1, 2, 3\}$, we will assume constant symbols 1, 2, 3.

• **This is only for sake of exposition. The symbols do not form a permanent part of the language.**
Truth of a Formula Relative to an Interpretation $I = (\Delta, \mu)$

- If a formula has free variables, add a $\forall$ quantifier for each such variable in front of the formula. (Free variables are understood to mean $\forall$-quantified by convention.)

- The result is called the “closure” of the formula.

- Now proceed assuming the formula has no free variables, i.e. the formula is closed.

- Our method will break down the formula recursively.
**Truth** of a Formula relative to an Interpretation $I = (\Delta, \mu)$

• For any closed formula $\varphi$, we will define the truth value $I[\varphi] \in \{T, F\}$.

• Brackets $[\ ]$ are used to emphasize that what is inside them is syntactic.
Truth of a Formula Relative to an Interpretation $I = (\Delta, \mu)$

- If the formula is of the form $\forall x \varphi$, then
  \[ I[\forall x \varphi] = T \text{ iff for every } d \in \Delta: I[\varphi[d/x]] = T \]
- Recalling that we are using $d$ as constant symbol in the case of substitution.
Truth of a Formula Relative to an Interpretation $I = (\Delta, \mu)$

- If the formula is of the form $\exists x \varphi$, then

$$I[\exists x \varphi] = T \iff \text{for some } d \in \Delta: I[\varphi[d/x]] = T.$$
Truth of a Formula Relative to an Interpretation $I = (\Delta, \mu)$


- (Recall there are no free variables.)

- $I[\varphi \lor \psi] = T$ iff $I[\varphi] = T$ or $I[\psi] = T$.

- Note that this is the same as the semantics for proposition logic.
Truth of a Formula Relative to an Interpretation $I = (\Delta, \mu)$


- $I[\varphi \leftrightarrow \psi] = T$ iff $I[\varphi] = I[\psi]$.

- $I[\neg \varphi] = T$ iff $I[\varphi] = F$.

- Finally, if the formula $\varphi$ is atomic, then $I[\varphi]$ is determined according to the following slides.
The value of terms under an interpretation

- An interpretation \( I = (\Delta, \mu) \) determines, for each term \( t \), a value \( I[t] \in \Delta \) of recursively:
  - If \( t \) is a constant symbol \( c \), then \( I[t] = \mu(c) \), the assigned value in \( \Delta \).
  - If \( t \) is a function symbol applied to terms, \( f(t_1, t_2, \ldots, t_n) \) where the \( t_i \) are terms, then
    \[
    I[t] = \mu(f)(I[t_1], I[t_2], \ldots I[t_n])
    \]
    recalling that \( \mu(f) \) is the function that interpretation \( I \) assigns the function symbol \( f \).
The value of **atomic formulas** under an interpretation

- An interpretation $I = (\Delta, \mu)$ determines for each atomic formula $E$ a value $I[E] \in \{T, F\}$ **recursively**:

  - If $E = p(t_1, t_2, \ldots, t_n)$, where $p$ is an $n$-ary predicate symbol, and the $t_i$ are its term arguments, then
    
    $$I[E] = \mu(p)(I[t_1], I[t_2], \ldots I[t_n]) \in \{T, F\}$$

    where $\mu(p)$ is the **predicate** $I$ assigns to $p$.

[using the **value** $I[t_i]$ of terms $t_i$ presented previously]
Example

- Atomic Formula is: $q(f(f(c)), c)$, where $q$ is a predicate symbol, $f$ is a function symbol, and $c$ is a constant.

- Suppose interpretation $I$ assigns
  - $\Delta = \{0, 1, 2\}$ domain
  - $\mu(c) = 0$ constant
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$ function
  - $\mu(q) = \{(2, 0), (1, 2), (0, 1)\}$ predicate
    the set of pairs for which $\mu(q)$ is $T$

- Thus:
  - $I[f(f(c))] = \mu(f)(I[f(c)]) = \mu(f)(\mu(f)(I[c])) = \mu(f)(\mu(f)(\mu(c))) = 2$
  - $I(q(f(f(c)), c)) = \mu(q)(I[f(f(c)], \mu(c)) = \mu(q)(2, 0) = T$
Example

• Atomic Formula is: \(q(f(f(c)), f(c))\), where \(q\) is a predicate symbol, \(f\) is a function symbol, and \(c\) is a constant.

• \textit{Suppose} interpretation \(I\) assigns
  • \(\Delta = \{0, 1, 2\}\) domain
  • \(\mu(c) = 0\) constant
  • \(\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}\) function
  • \(\mu(q) = \{(2, 0), (1, 2), (0, 1)\}\) predicate
    the set of pairs for which \(\mu(q)\) is \(T\)

• Thus:
  • \(I[f(f(c))] = \mu(f)(I[f(c)]) = \mu(f)(\mu(f)(I[c])) = \mu(f)(\mu(f)(\mu(c))) = 2\)
  • \(I(q(f(f(c)), f(c))) = \mu(q)(I[f(f(c)], I[f(c)]) = \mu(q)(2, 1) = F\)
Example

- Formula is: $\exists x \ q(f(f(c)), f(x))$, where $q$ is a predicate symbol, $f$ is a function symbol, and $c$ is a constant.

- Suppose interpretation $I$ assigns
  - $\Delta = \{0, 1, 2\}$ domain
  - $\mu(c) = 0$ constant
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$ function
  - $\mu(q) = \{(2, 0), (1, 2), (0, 1)\}$ predicate
    the set of pairs for which $\mu(q)$ is $T$

- According to our rules $I[\exists x \ q(f(f(c)), f(x))] = T$ iff at least one of these is true:
  - $I[q(f(f(c)), f(0))]$ which is the same as $I[q(2, 1)]$
  - $I[q(f(f(c)), f(1))]$ which is the same as $I[q(2, 2)]$
  - $I[q(f(f(c)), f(2))]$ which is the same as $I[q(2, 0)]$

- As $q(2, 0) \in \mu(q)$, $I[\exists x \ q(f(f(c)), f(x))] = T$. 

115
Example

- Formula is: $\forall x \ q(f(f(c)), f(x))$, where $q$ is a predicate symbol, $f$ is a function symbol, and $c$ is a constant.

- Suppose interpretation $I$ assigns
  - $\Delta = \{0, 1, 2\}$ domain
  - $\mu(c) = 0$ constant
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$ function
  - $\mu(q) = \{(2, 0), (1, 2), (0, 1)\}$ predicate
    the set of pairs for which $\mu(q)$ is T

- According to our rules $I[\exists x \ q(f(f(c)), f(x))] = T$ iff all of these are true:
  - $I[q(f(f(c)), f(0))]$ which is the same as $I[q(2, 1)]$
  - $I[q(f(f(c)), f(1))]$ which is the same as $I[q(2, 2)]$
  - $I[q(f(f(c)), f(2))]$ which is the same as $I[q(2, 0)]$

- As $q(2, 1)$ is not in $\mu(q)$, $I[\forall x \ q(f(f(c)), f(x))] = F.$
Fun with Interpretations

• A 2-ary predicate represents the binary relation in an interpretation, i.e. a set of pairs of domain elements.

• Various properties of relations can be expressed using predicate logic formulas.

• In the following, what formula characterizes each relation represented by predicate L (sometimes using “loves” for analogy), and possibly the predicate =.


Everybody loves somebody.

∀x ∃y L(x, y)
Everybody is loved by somebody.
Somebody loves everybody.

“Pollyanna”
Nobody loves everybody.
Nobody loves somebody.
Everybody loves everybody.

“Commune”
Everybody loves exactly one.
Everybody loves him/herself.

“Reflexive”
“Narcissists’ Convention”
L is transitive.
L is an equivalence relation

Reflexive, symmetric, transitive
L is antisymmetric.
L is a partial order ("poset").

Reflexive, Antisymmetric, Transitive
Empty Domains

- Customarily domains are required to be non-empty.

- Certain entailments that would be true under non-empty domains become false if the domain is empty.

- For example,
  \[ \forall x \ P(x) \implies \exists x \ P(x) \]
  The premise is *vacuously* true for an empty-domain, but the conclusion cannot be true.
We can construct a program for evaluating $I$ when the domain is finite

- For infinite domains, this is not possible.
Satisfaction and Models

• An interpretation I satisfies a formula E iff $I[E] = T$.

• We also say that I is a model for E in this case.

• Caution: Some authors, such as Huth&Ryan, use “model” to mean “interpretation”.

• A formula is satisfiable iff there is an interpretation that satisfies it, otherwise it is unsatisfiable.
Formalizing Semantic Entailment $|=\$

- When $\varphi_1, \ldots, \varphi_n, \psi$ are predicate calculus formulas,

$$\varphi_1, \ldots, \varphi_n |= \psi$$

means:

Every interpretation $I$ that satisfies each of the formulas $\varphi_1, \ldots, \varphi_n$ also satisfies $\psi$.

$\Gamma |= \psi$, where $\Gamma$ is a set of formulas:

- Extend “model for” $\Gamma$ to mean an interpretation satisfies the entire set $\Gamma$, as:

  Every model for $\Gamma$ is also a model for $\psi$. 
Validity

- When the left-hand side is empty:
  \(|= \psi\)

  we say that is **universally valid**, or just plain **valid**.

- In this case, every interpretation for \(\psi\) is a model.

- **Validity** in predicate calculus is analogous to **tautology** in propositional calculus.
|= in predicate calculus vs. propositional

- The predicate version of $|= \psi$ is a very broad statement:
  - The domain of an interpretation can be infinite.
  - The set of applicable interpretation is generally infinite.
- Intuitively there is much less likely to be an algorithm to check whether $|= \psi$ for predicate calculus in the way there is for the propositional calculus.
Predicate Calculus “with Equality”

- There is one exception to the “all interpretations” definitions of validity when the $=$ predicate symbol is being used:

  Equality is always interpreted as **identity** on the domain of the interpretation.
ND Equality Rules

• Natural Deduction typically introduces rules for equality (from which the axioms can be derived).
  
  • \[ t = t \] =I where \( t \) is any term

  • \[ \frac{s = t \quad \varphi[s/x]}{\varphi[t/x]} \] =E

where \( s \) and \( t \) are any terms and \( x \) is any variable, provided \( s \) and \( t \) are free to replace \( x \) in \( \varphi \).
Equality Formulas ("Axioms" in some systems) 
(Derivable from ND Rules)

- Four types of formulas characterize equality:
  
  \( \forall x \ (x = x) \) \hspace{2cm} \text{reflexive}

  \( \forall x \ \forall y \ (x = y \rightarrow y = x) \) \hspace{2cm} \text{symmetric}

  \( \forall x \ \forall y \ \forall z \ ((x = y \land y = z) \rightarrow (x = z)) \) \hspace{2cm} \text{transitive}

  \( \forall x_1 \ldots \forall x_n \ \forall y_1 \ldots \forall y_n \)
  \( \ (x_1 = y_1 \land \ldots \land x_n = y_n) \rightarrow (f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)) \) \hspace{2cm} \text{substitution}

  where \( f \) is any \( n \)-ary function symbol

  \( \forall x_1 \ldots \forall x_n \ \forall y_1 \ldots \forall y_n \)
  \( \ (x_1 = y_1 \land \ldots \land x_n = y_n) \rightarrow (p(x_1, \ldots, x_n) \rightarrow p(y_1, \ldots, y_n)) \) \hspace{2cm} \text{substitution}

  where \( p \) is any \( n \)-ary predicate symbol
Example

Prove the symmetry rule:
\[ u = v \quad |\quad v = u \]
where \( u \) and \( v \) are any terms, from the two ND equality rules.

The “trick” here is finding the right \( \varphi \).

**Use** \( x = u \) **for** \( \varphi \), so that \( \varphi[u/x] \) is \( u=u \), an instance of \( =I \).
This gives \( \varphi[v/x] \) as \( v=u \), the desired conclusion.

1. \( u = v \) \hspace{1cm} \text{Premise}
2. \( u = u \) \hspace{1cm} =I \quad \text{(identified as} \ \varphi[u/x])
3. \( v = u \) \hspace{1cm} 1, 2, =E \quad \text{(identified as} \ \varphi[v/x])
Example

Prove the transitivity rule:

\[ u = v, \ v = w \vdash u = w \]

where \( u, v, \) and \( w \) are terms, from the two ND equality rules.

Here we let \( \varphi \) be \( u = x \), to use \( =E \) rule. Identify \( s = t \) in the \( =E \) rule with \( v = w \).

1. \( v = w \) \hspace{1cm} \text{Premise}
2. \( u = v \) \hspace{1cm} \text{Premise (identified as} \varphi[v/x])
3. \( u = w \) \hspace{1cm} 1, 2, =E (identified as \varphi[w/x])
Shortcutting Equality

- Having to spell out every instance of an equality rule can detract from the flow of a proof.

- So we will adopt the practice of combining uses of equality with other axioms, rules, and lemmas, knowing that we could spell them out if need be. (Be careful tho’!)
Theories

• In logic, a **theory** is a **set of formulas**.

• Usually a theory is determined by a set of **axioms**. The formulas in the theory are those derivable from the axioms using the rules of inference.
Examples of Theories

• **Group Theory** is the set of formulas derived from the group theory axioms:

\[
\forall x \forall y \forall z \left[ x + (y + z) = (x + y) + z \right], \\
\forall x \left[ x + 0 = x \right], \\
\forall x \exists y \left[ x + y = 0 \right].
\]

Here 0 is a constant symbol, + is a 2-ary function symbol, = is a 2-ary predicate symbol.
An Alternate Version of Group Theory
(adds inverse function – and symmetry)

- $\forall x \forall y \forall z \ (x+((y+z)) = (((x+y)+z))$ A1
- $\forall x \ (x+0 = x) \land (0+x = x)$ A2
- $\forall x \ (x+(-x) = 0) \land ((-x)+x = 0)$ A3

(-x is the y that exists, on the previous slide)

Note that symmetry does not say:
$\forall x \forall y \ x + y = y + x$
That does not follow from the group axioms.
Examples of Formulas Derivable in Group Theory

- $\forall x \ (-(-x)) = x$
- $\forall x \forall y \ -((x+y)) = ((-y)+(-x))$
- $\forall a \forall b \forall x \ (a+x = b) \rightarrow x = ((-a)+b)$
Sample Proof in Group Theory
\( \forall x \ (-(-x)) = x \) (Combining steps)

1. \(-x_0 + (-(-x_0)) = 0\)  
   \(\text{A3, } \forall E, \land E\)
2. \(x_0 + ((-x_0) + (-(-x_0))) = x_0 + 0\)  
   \(1, =\)
3. \((x_0 + (-x_0)) + (-(-x_0)) = x_0 + 0\)  
   \(2, A1\)
4. \(0 + (-(-x_0)) = x_0 + 0\)  
   \(3, A2, =\)
5. \(0 + (-(-x_0)) = x_0\)  
   \(4, A2, =\)
6. \((-(-x_0)) = x_0\)  
   \(5, A2, =\)
7. \(\forall x \ (-(-x)) = x\)  
   \(1-6, \forall I\)
Examples of Theories

- **Abelian Groups** adds the commutative axiom to those of group theory:

  $$\forall x \forall y [x + y = y + x]$$

- The set of provable formulas changes as a result.
Examples of Theories

• Theory of **Commutative Rings** adds a new function symbol, a new constant symbol, and more axioms to the theory of Abelian Groups:

\[
\forall x \forall y \forall z \ [x \cdot y = y \cdot x], \\
\forall x \forall y \forall z \ [(x \cdot y) \cdot z = x \cdot (y \cdot z)], \\
\forall x \forall y \forall z \ [x \cdot (y + z) = (x \cdot y) + (x \cdot z)], \\
\forall x \ [x \cdot 1 = x], \\
0 \neq 1.
\]
Limitations of Theories

• Some theories *cannot* be expressed with only a **finite** set of axioms in **first-order** logic (but can be in second-order).

• Example: Torsion-Free Abelian Groups adds an infinite number of axioms (one for each \( n \), where \( nx \) means \( x+x+\ldots+x \) \( n \) times): \[ \forall x \left[ x \neq 0 \rightarrow nx \neq 0 \right] \]
Soundness and Completeness

• As with propositional logic, we define:

• **Soundness** of a set of derivation rules:
  
  For any set of formulas $\Gamma$ and any formula $\psi$:
  
  \[ \Gamma \vdash \psi \text{ implies } \Gamma \models \psi \]

• **Completeness** of a set of derivation rules:
  
  For any set of formulas $\Gamma$ and any formula $\psi$:
  
  \[ \Gamma \models \psi \text{ implies } \Gamma \vdash \psi \]

• It can be shown that our natural deduction framework has both of these properties [cf. van Dalen, *Logic and Structure*]
Examples of (Universally) Valid vs. Invalid Formulas
Invalid Formulas Valid Under Specific Interpretations
Showing a Formula Invalid

- **Find a counterexample**: an interpretation under which the formula is not valid.

**Example:**  \( \forall x \ (A(x) \rightarrow B(x)) \rightarrow (\exists x \ A(x) \rightarrow \forall x \ B(x)) \)

**Interpretation:**
- \( \Delta = \{1, 2\} \)
- \( \mu(A) = \{2\} \)
- \( \mu(B) = \{2\} \)
More JAPE Examples

- **∀ Elimination** (working ***forward***) instantiates a ∀-quantified variable with a term that already exists (in this case, \(i\)).

- **Both** the term and the ∀ formula must be selected (using shift-click to add one or the other):

  ![Diagram showing ∀ Elimination]

  Note: If the red bracket opens **downward**, the item is usable as a hypothesis. If **upward**, a conclusion. In some cases both apply, and you need to click above or below to indicate which.
JAPE Examples

- $\forall$ Introduction (working backward), followed by $\forall$ Elimination (working forward)

\[\begin{align*}
1: & \ \forall x. R(x) \text{ premise} \\
2: & \ \text{actual i} \quad \text{assumption} \\
3: & \ R(i) \quad \forall \text{ elim 1,2} \\
4: & \ \forall y. R(y) \quad \forall \text{ intro 2-3}
\end{align*}\]

- Note: JAPE will **unify** the above premise and conclusion, so a **shorter** proof, using the ‘hyp’ rule is, but this might be confusing because we end up with no $y$.

\[\begin{align*}
1: & \ \forall x. R(x) \text{ premise} \\
\cdots & \\
2: & \ \forall y. R(y) \quad \text{hyp} \\
\end{align*}\]
JAPE Examples

- $\forall$ Introduction (working \textit{backward}) introduces a fresh variable. Variables are often helpful in completing a proof. Of course, the variable \textit{can't be taken outside the box}.

\[\begin{array}{c}
2: \forall y. R(y) \\
\forall I \\
3: R(i) \\
4: \forall y. R(y) \quad \forall \text{ intro } 2-3
\end{array}\]

$i$ is meaningless out here
JAPE Examples

- **∃ Elimination** (working *forward*) introduces a fresh variable for a sub-proof.
- It *needs a goal, in order to introduce the goal for the sub-proof* (inside the box). You may need to identify the goal if not obvious.
- In this example, the goal is implicit, and the proof is completed in one step.

```
1: ∃x.R(x) premise
   ...
2: ∃y.R(y)
```

Note: As before, JAPE will also *unify* the above premise and conclusion in a single step, making a proof unnecessary.
JAPE Examples

• **∃ Introduction** (working *backward*) needs a term that it can use as an instantiation for the ∃ variable.
• The **JAPE ND theory doesn’t have functions yet, so all such terms will be variables**.
• The variable must be selected by the user.
• We can’t use ∃I here, because there is no variable available.

```
1: ∃x.R(x) premise
   ...
2: ∃y.R(y)
```

• Here is an example with a variable that *can* be used (but leads to a dead end):

```
2: actual i, ∃y.R(i,y) assumptions
   ...
3: ∃y.∃x.R(x,y)

∃I (with i for y)
4: ∃y.∃x.R(x,y)
```
Proof of a sequent using $\exists E$ and $\exists I$

Work forward to introduce variables, by eliminating $\exists$’s (opening boxes):

1: $\exists x. \exists y. R(x, y)$ premise

... 

2: $\forall y. \exists x. R(x, y)$

$\exists E$

1: $\exists x. \exists y. R(x, y)$ premise

2: actual $i$, $\exists y. R(i, y)$ assumptions

3: $\exists y. R(i, y)$

4: $\exists y. \exists x. R(x, y)$ $\exists$ elim 1,2–3

(Note the conclusions inside = the conclusion outside.)

3: actual $i$, $\exists y. R(i, y)$ assumptions

4: $\exists y. \exists x. R(x, y)$ $\exists$ elim 1,2–5

then introduce $\exists$’s working backward in the right order:

1: $\exists x. \exists y. R(x, y)$ premise

2: actual $i$, $\exists y. R(i, y)$ assumptions

3: actual $i_1$, $R(i_1)$ assumptions

4: $\exists y. R(x, i_1)$ $\exists$ elim 2.2,3–4

5: $\exists y. \exists x. R(x, y)$ $\exists$ elim 1,2–5

6: $\exists y. \exists x. R(x, y)$

7: $\exists y. \exists x. R(x, y)$

...
Example using all four quantifier rules

1: \( \exists x. \forall y. R(x, y) \) premise

\[
\begin{align*}
2: \forall y. \exists x. R(x, y) \\
\text{...}
\end{align*}
\]

\( \forall \) introduction

1: \( \exists x. \forall y. R(x, y) \) premise

2: actual i

\[
\begin{align*}
3: & \exists x. R(x, i) \\
\text{...}
\end{align*}
\]

\( \exists \) elimination

4: \( \forall y. \exists x. R(x, y) \) \( \forall \) introduction 2–3

1: \( \exists x. \forall y. R(x, y) \) premise

2: actual i

\[
\begin{align*}
3: & \exists x. R(x, i) \\
\text{...}
\end{align*}
\]

\( \exists \) elimination

4: \( \exists x. R(x, i) \)

5: \( \exists x. R(x, i) \)

6: \( \forall y. \exists x. R(x, y) \) \( \forall \) introduction 2–5

1: \( \exists x. \forall y. R(x, y) \) premise

2: actual i

\[
\begin{align*}
3: & \exists x. R(x, i) \\
\text{...}
\end{align*}
\]

\( \exists \) elimination

4: \( \exists x. R(x, i) \)

5: \( \exists x. R(x, i) \)

6: \( \forall y. \exists x. R(x, y) \) \( \forall \) introduction 2–6

7: \( \forall y. \exists x. R(x, y) \) \( \forall \) introduction 2–6

164
A Common Idiom in JAPE

- These steps are often done in sequence, but it is not always obvious when to use them:
  - contra (classical) = RAA
  - \( \neg \) elimination (introduces skeletal formulas)
  - unify one of the skeletal formulas with an existing sub-formula
Sometimes the steps have to be taken in a round-about order, e.g. \( \exists I \) won’t work forward (needs variable and body). **This example uses the previous idiom.**
The non-empty universe assumption is not assumed in JAPE!!

If you need this, you must introduce a premise that there is at least one element. How to do this is shown on the next slide.

Proved in textbooks, but not provable in JAPE:

1: $\forall x. R(x)$ premise

... 

2: $\exists x. R(x)$

Can’t go backward, because $\exists I$ needs a term.

Can’t go forward, because $\forall E$ needs a variable.
If you need the non-empty universe assumption, you must introduce a premise that **there is at least one element**, by including ‘actual i’, or ‘∃x.Τ’ as a premise. (one place where Τ is useful, but others could be used).

See Bornat’s book “Proof and Disproof ... “ for discussion on why this philosophy is better.
A Tricky One

1: actual j, actual k premises

... 

2: \( \exists x. (R(x) \rightarrow R(j) \land R(k)) \)
A Tricky One

Note: This does not say that \( j \) and \( k \) are distinct. They could be two names for the same individual.

1: actual \( j \), actual \( k \)  
2: \( \exists x.(R(x) \rightarrow R(j) \land R(k)) \)

What \( x \) would make this work?

1: actual \( j \), actual \( k \)  
2: \( \_E \lor \neg \_E \)  
3: \( \exists x.(R(x) \rightarrow R(j) \land R(k)) \)

LEM to the rescue (used as a lemma)

unify \( R(j) \) with \( \_E \)

What \( x \) would make this work?

\[ \lor \text{E} \]
How to introduce LEM (it must be proved first)

Click to apply as lemma

Voila!
Continuing the tricky proof ...

For the Top Box

\( x = k \) (an actual) will enable \( \exists I \)
Continuing the tricky proof ...

**For the Bottom Box**

\( x = j \) (an actual) will enable \( \exists I \) (using contra)

\[
\begin{align*}
5: & \quad \neg R(j) \\
6: & \quad \exists x. (R(x) \to R(j) \land R(k))
\end{align*}
\]

\[
\begin{align*}
8: & \quad \neg R(j) \\
9: & \quad R(j) \\
10: & \quad \bot \\
11: & \quad R(j) \land R(k) \\
12: & \quad R(j) \to R(j) \land R(k) \\
13: & \quad \exists x. (R(x) \to R(j) \land R(k))
\end{align*}
\]
Completed Proof

1: actual j, actual k  
2: \( R(j) \vee \neg R(j) \)  
3: \( R(j) \)  
   \[ \text{premises} \]  
   \[ \text{Theorem } E \vee \neg E \]  
4: \( R(k) \)  
   \[ \text{assumption} \]  
5: \( R(j) \land R(k) \)  
   \[ \land \text{ intro } 3,4 \]  
6: \( R(k) \rightarrow R(j) \land R(k) \)  
   \[ \rightarrow \text{ intro } 4-5 \]  
7: \( \exists x. (R(x) \rightarrow R(j) \land R(k)) \)  
   \[ \exists \text{ intro } 6,1,2 \]  
8: \( \neg R(j) \)  
   \[ \text{assumption} \]  
9: \( R(j) \)  
   \[ \text{assumption} \]  
10: \( \bot \)  
   \[ \neg \text{ elim } 9,8 \]  
11: \( R(j) \land R(k) \)  
   \[ \text{contra (constructive) } 10 \]  
12: \( R(j) \rightarrow R(j) \land R(k) \)  
   \[ \rightarrow \text{ intro } 9-11 \]  
13: \( \exists x. (R(x) \rightarrow R(j) \land R(k)) \)  
   \[ \exists \text{ intro } 12,1,1 \]  
14: \( \exists x. (R(x) \rightarrow R(j) \land R(k)) \)  
   \[ \lor \text{ elim } 2,3-7,8-13 \]
An Analogous Sequent

1: actual i

... 

2: \( \exists x. (R(x) \rightarrow \forall y. R(y)) \)

“If there is at least one person, then there is someone \((x)\) such that if \(x\) is happy then everyone is happy.”
Key

- How to use the LEM to create a dichotomy?
- $E \lor \neg E$
- But what is $E$?

- Possibilities for $E$:
  - $\exists x. R(x)$
  - $\forall y. R(y)$

- Use unification to assign formula to $E$
Constructive \( \rightarrow \) vs. Classical \( \leftarrow \)
Constructive $\rightarrow$ vs. Classical $\leftarrow$

Constructive:

1: $\forall x. R(x)$  \hspace{1cm} premise
2: $\exists y. \neg R(y)$  \hspace{1cm} assumption
3: actual $i, \neg R(i)$  \hspace{1cm} assumptions
4: $R(i)$  \hspace{1cm} $\forall$ elim 1, 3.1
5: $\bot$  \hspace{1cm} $\neg$ elim 4, 3.2
6: $\bot$  \hspace{1cm} $\exists$ elim 2, 3–5
7: $\neg \exists y. \neg R(y)$  \hspace{1cm} $\neg$ intro 2–6

Classical:

1: $\neg \exists x. \neg R(x)$  \hspace{1cm} premise
2: actual $i$  \hspace{1cm} assumption
3: $\neg R(i)$  \hspace{1cm} assumption
4: $\exists x. \neg R(x)$  \hspace{1cm} $\exists$ intro 3, 2
5: $\bot$  \hspace{1cm} $\neg$ elim 4, 1
6: $\bot$  \hspace{1cm} contra (classical) 3–5
7: $\forall y. R(y)$  \hspace{1cm} $\forall$ intro 2–6
Note Rule Parallels

1: \( E \land F \)
   premise
2: \( \neg E \lor \neg F \)
   assumption
3: \( \neg E \)
   assumption
4: \( E \)
   \( \land \) elim 1
5: \( \bot \)
6: \( \neg F \)
   assumption
7: \( F \)
   \( \land \) elim 1
8: \( \bot \)
9: \( \bot \)
10: \( \neg (\neg E \lor \neg F) \)
    \( \neg \) intro 2-9

1: \( \forall x. R(x) \)
   premise
2: \( \exists y. \neg R(y) \)
   assumption
3: \( \text{actual } i, \neg R(i) \)
   assumptions
4: \( R(i) \)
5: \( \bot \)
6: \( \bot \)
7: \( \neg \exists y. \neg R(y) \)
    \( \neg \) intro 2-6
Note Rule Parallels

1: \neg (\neg E \lor \neg F) \text{ premise}

2: \neg E \hspace{1cm} \text{assumption}

3: \neg E \lor \neg F \hspace{1cm} \lor \text{ intro 2}

4: \bot \hspace{1cm} \neg \text{ elim 3,1}

5: E \hspace{1cm} \text{contra (classical) 2–4}

6: \neg F \hspace{1cm} \text{assumption}

7: \neg E \lor \neg F \hspace{1cm} \lor \text{ intro 6}

8: \bot \hspace{1cm} \neg \text{ elim 7,1}

9: F \hspace{1cm} \text{contra (classical) 6–8}

10: E \land F \hspace{1cm} \land \text{ intro 5,9}

1: \neg \exists x. \neg R(x) \text{ premise}

2: \text{actual i} \hspace{1cm} \text{assumption}

3: \neg R(i) \hspace{1cm} \text{assumption}

4: \exists x. \neg R(x) \hspace{1cm} \exists \text{ intro 3,2}

5: \bot \hspace{1cm} \neg \text{ elim 4,1}

6: R(i) \hspace{1cm} \text{contra (classical) 3–5}

7: \forall y. R(y) \hspace{1cm} \forall \text{ intro 2–6}
A proof

- Suppose everyone loves somebody, and loves is symmetric and transitive.
- Then loves is reflexive.

```
1: \forall x. \exists y. R(x,y), \ \forall x. \forall y. (R(x,y) \rightarrow R(y,x)) premises
2: \forall x. \forall y. \forall z. ((R(x,y) \land R(y,z)) \rightarrow R(x,z)) premise
3: actual i
4: \forall y. \forall z. ((R(i,y) \land R(y,z)) \rightarrow R(i,z)) assumption
5: \forall y. (R(i,y) \rightarrow R(y,i)) \forall elim 1,2,3
6: \exists y. R(i,y) \forall elim 1,3
7: actual i1
8: R(i,i1)
9: \forall z. ((R(i,i1) \land R(i1,z)) \rightarrow R(i,z)) assumption
10: (R(i,i1) \land R(i1,i)) \rightarrow R(i,i) \forall elim 4,7
11: R(i,i1) \rightarrow R(i1,i) \forall elim 9,3
12: R(i1,i) \forall elim 5,7
13: R(i,i1) \land R(i1,i) \land intro 8,12
14: R(i,i) \neg elim 11,8
15: R(i,i) \exists elim 6,7-14
16: \forall x. R(x,x) \forall intro 3-15
```
Informal Proof

- **Assertion:** If everyone loves somebody, and loves is symmetric and transitive, then loves is reflexive.

- Let $x_0$ be an arbitrary element, to show $x_0$ loves $x_0$.

- Since everyone loves someone, let $y_0$ be someone $x_0$ loves.

- By symmetry, $y_0$ loves $x_0$ too.

- By transitivity, since $x_0$ loves and $y_0$ loves $x_0$, $x_0$ loves $x_0$. 
Caution

• Without the assumption:
  Everyone loves somebody
• the assertion that loves is reflexive does not hold.

• Show this by giving a counterexample.
How to do without function symbols

• Every n-ary function is an (n+1)-ary relation.
• For example, a binary function f can be represented by a 3-ary relation F.
• F(x, y, z) means f(x, y) = z.
• Functionality induces some additional axioms for F:
  • ∀x ∀y ∃z F(x, y, z)
  • ∀x ∀y ∀z ∀z’ (F(x, y, z) ∧ F(x, y, z’) → z = z’)
• We’d still need axioms for equality.
Example: Group theory without function symbols (e is identity)

- \( \forall x \forall y \exists z \ F(x, y, z) \)
- \( \forall x \forall y \forall z \forall z' \ (F(x, y, z) \land F(x, y, z') \rightarrow z = z') \)
- \( \forall x \forall y \forall z \exists v(F(x, y, v) \land F(v, z, w)) \rightarrow \exists u(F(y, z, u) \land F(x, u, w)) \)
- \( \forall x \ F(x, e, x) \)
- \( \forall x \ F(e, x, x) \)
- \( \forall x \exists y \ F(x, y, e) \)
- + Equality axioms