Predicate Logic

Robert Keller
February 2011

Predicate Calculus Language

- E is the start symbol
- E ::= A | // Atom (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: ∀ ∃ ¬ ∧ ∨ → ↔
- Atom (A) now requires a more complex production

Atomic Formula

- Informally, evaluates to a truth value {T or F}, once argument individuals are substituted.

Atomic Formulas

- 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

- ∀ is a "wholesale" version of ∧

- ∃ is a "wholesale" version of ∨

- $\forall x\ P(x)$ 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 $P(x_0) \wedge P(x_1) \wedge P(x_2) \wedge \ldots$

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

“Terms”: 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 symbol
• f(b, y)	function applications
• g(h(b), c, h(y))	
• g(a, b, g(a, b, c))	

Examples of Atomic Formulas

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

Examples of “Literals”

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

Examples of Formulas

- Any Quantifier-Free Formula
- $\exists x\ p(x)$
- $\forall y\ (p(y) \wedge q(y))$
- $\forall y\ \exists x\ (p(f(x, y)) \rightarrow q(y))$
- $\forall x\ (p(f(x, y)) \vee q(x))$
- $\forall y\ (q(y) \rightarrow \exists x\ p(f(x, y)))$

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.

- The particular structures are called “Interpretations”.

Example:

Interpretation for the natural numbers

- The intended domain is {0, 1, 2, 3, …}.
- 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

- 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(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".

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

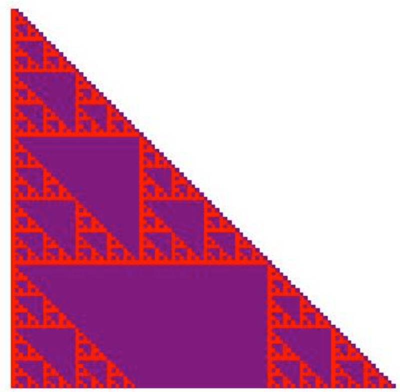
- Trivial group: {0} $e = 0$, $f(0, 0) = 0$
- 2-element group: {0, 1} $e = 0$, $f(x, y) = x + y \pmod 2$
- Z_p: {0, 1, …, p-1} for any prime p,
  $e = 0$, $f(x, y) = x + y \pmod p$
- Tire rotations
- Particle spins (physics)
- Rubik's cube twists
- Many others

Examples of Groups

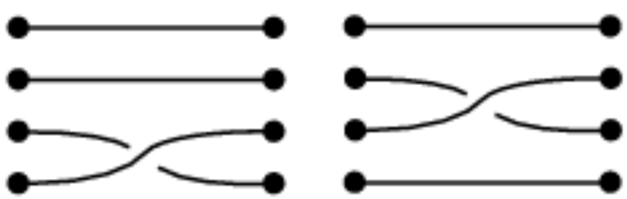
$$|\psi\rangle \rightarrow e^{-\frac{i}{\hbar}(2\pi)S_z}|\psi\rangle = -|\psi\rangle$$

spins

4 elements



cellular automaton
based on group



braids

twists: 43252003274489856000 elements

Joke involving groups… By Tommaso Dorigo

- At PhyStat 2011, … Kyle Cranmer was showing results of very CPU-intensive calculations of **renormalization-group equations** used to derive measurable parameters of Supersymmetry from the value of basic parameters at a high-energy scale.

- He was mentioning that the original calculation used to take 720 CPU-days, but that they had found a series of shortcuts using neural networks, and the result was a huge improvement in speed: this was now a 1-minute calculation!

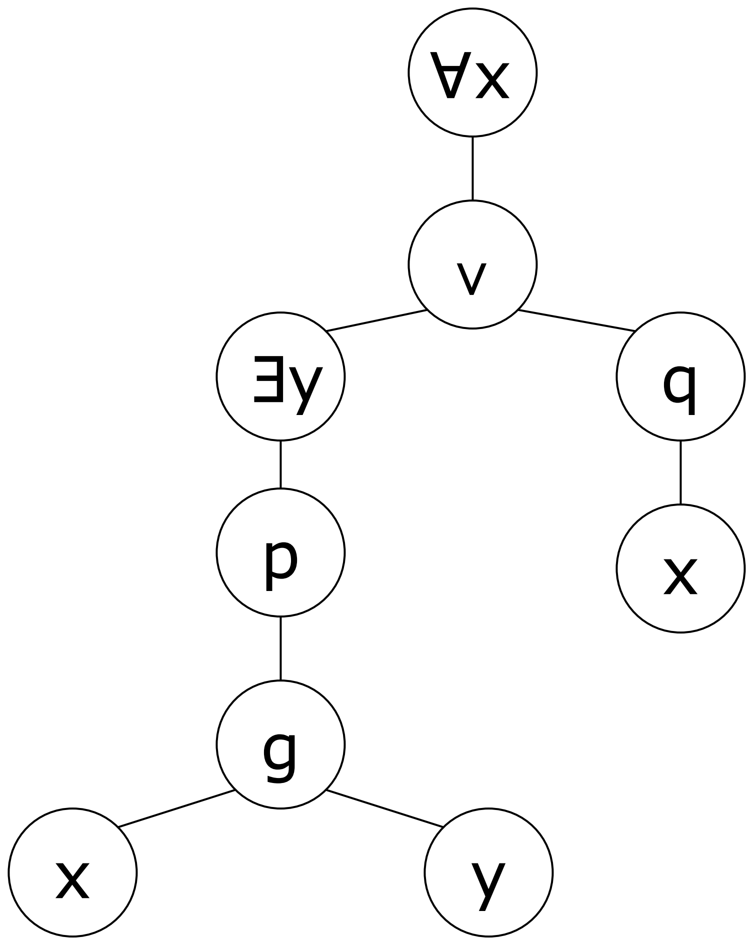
- Sitting in second row next to me was John Conway … As Kyle was describing the new program with its speedy calculation of SUSY model parameters and likelihoods, he went audibly: "Is there an iPhone app for that ?".

Inside Joke

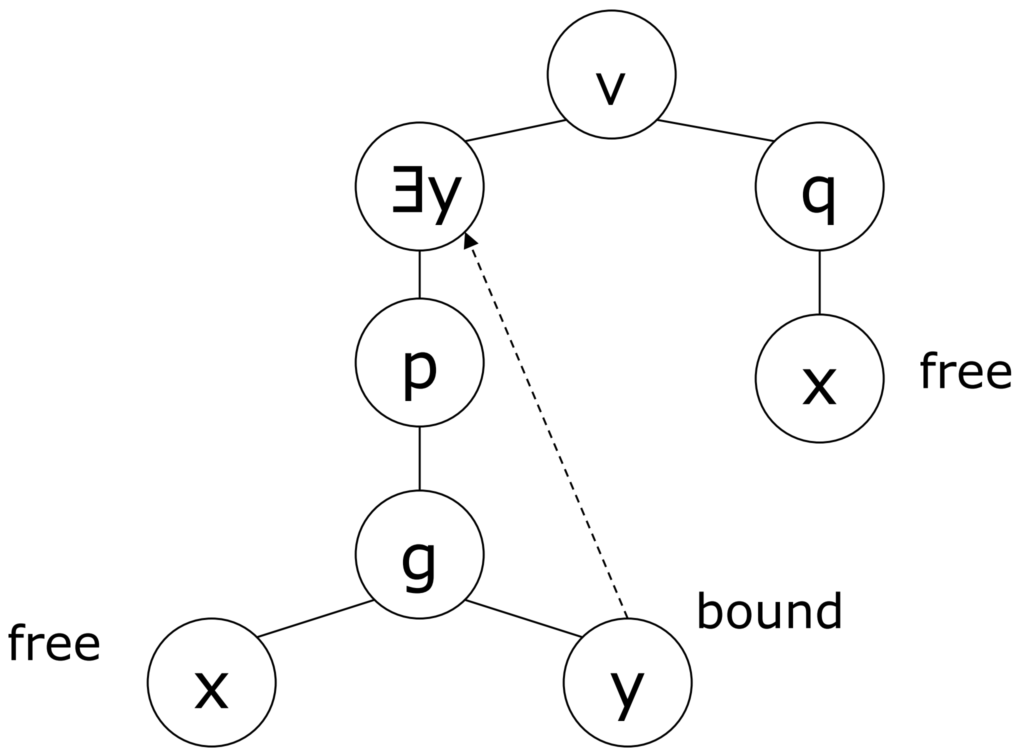
- Q: Why didn't Newton discover group theory?
- A: Because he wasn't Abel.

Syntax Trees (or "Parse" Trees)

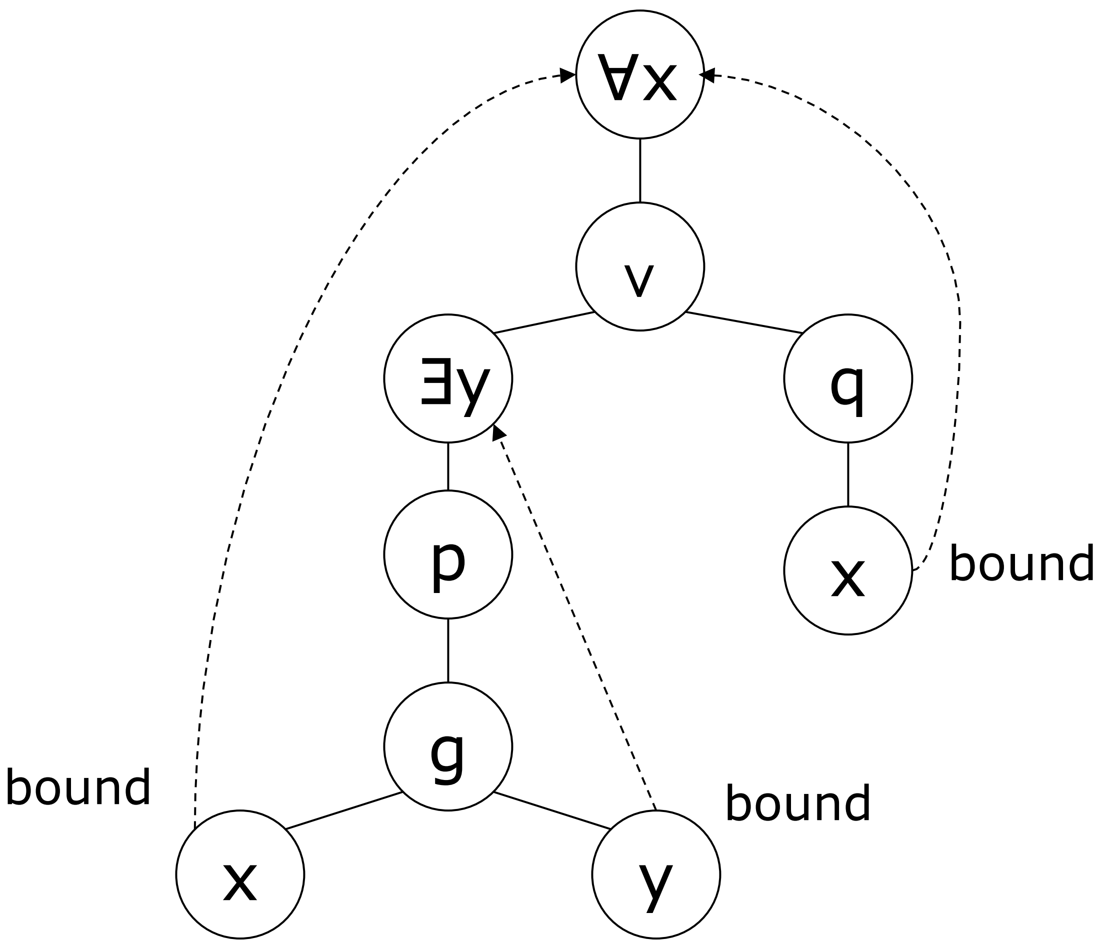
- We are assuming familiarity with syntax trees from CS 60.
- ∀x, ∃x are treated as if **1-ary operators**.
- Example: ∀x ((∃y p(g(x, y))) ∨ q(x))



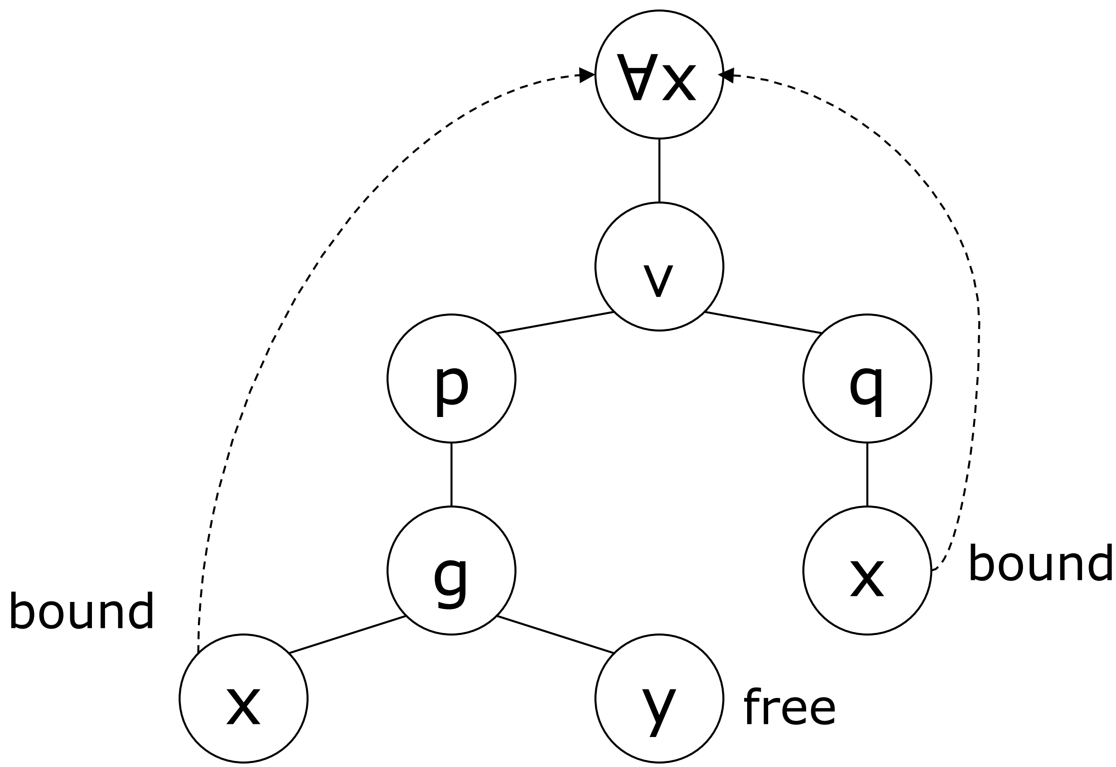
Free and Bound Variable Instances



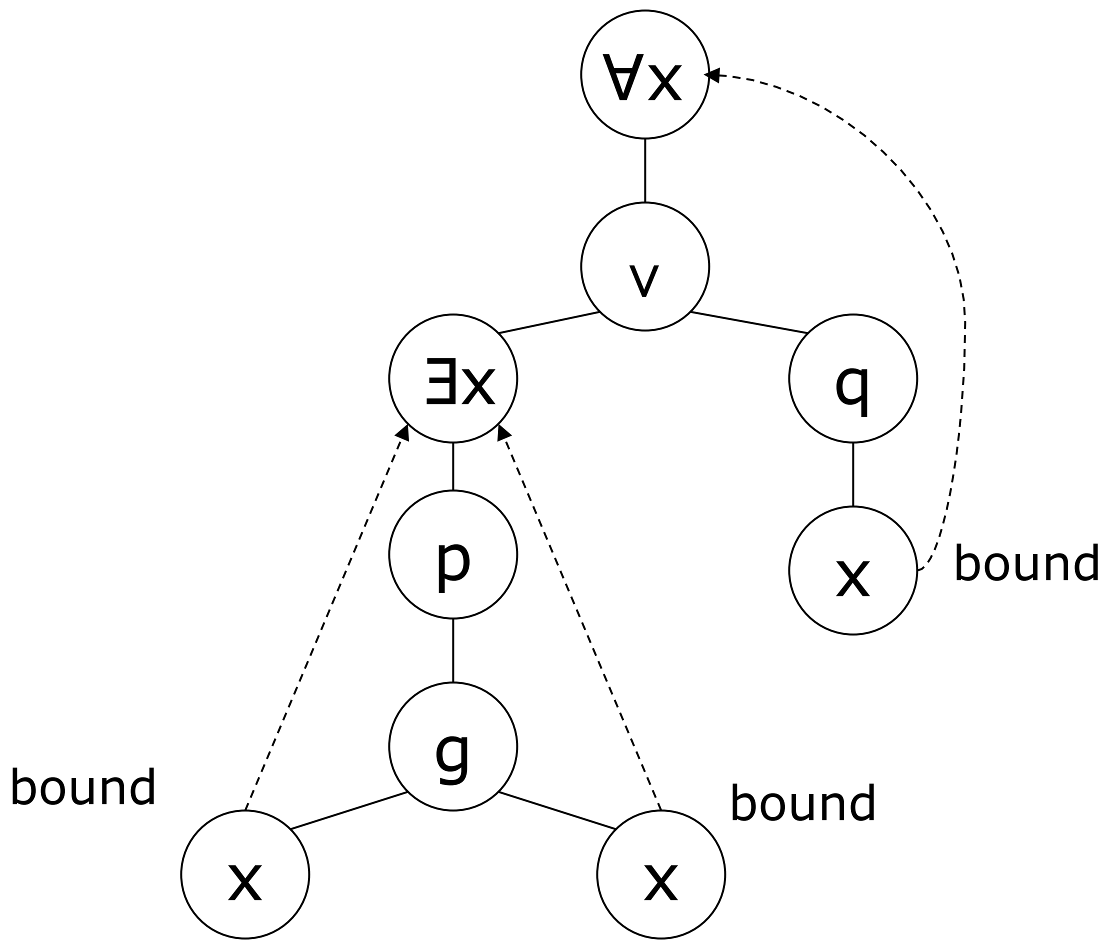
Free and Bound Variable Instances



Free and Bound Variable Instances



Free and Bound Variable Instances



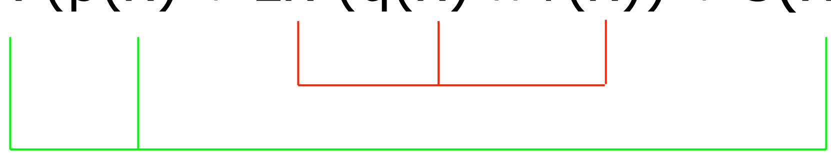
Scope of Variables

- 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 ∀x E, or ∃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 x:

$$\forall x\ (p(x) \vee \exists x\ (q(x) \wedge r(x)) \vee s(x, y))$$



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.

- Example: One of the x's renamed to u:

$$\forall x\ (p(x) \vee \exists u\ (q(u) \wedge r(u)) \vee s(x, y))$$

Definition of Free and Bound Instances

- In a **term or atomic formula**, every instance of a variable is free.

- If φ is a formula, then any **free instances** of a variable x **become bound** in $\forall x\ \varphi$ and $\exists x\ \varphi$.

- The free instances of variables in φ and ψ remain free in $(\neg\varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, and $(\varphi \rightarrow \psi)$.

- The bound instances of variables in φ and ψ remain bound in $(\neg\varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, and $(\varphi \rightarrow \psi)$.

Substitutability Restriction

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

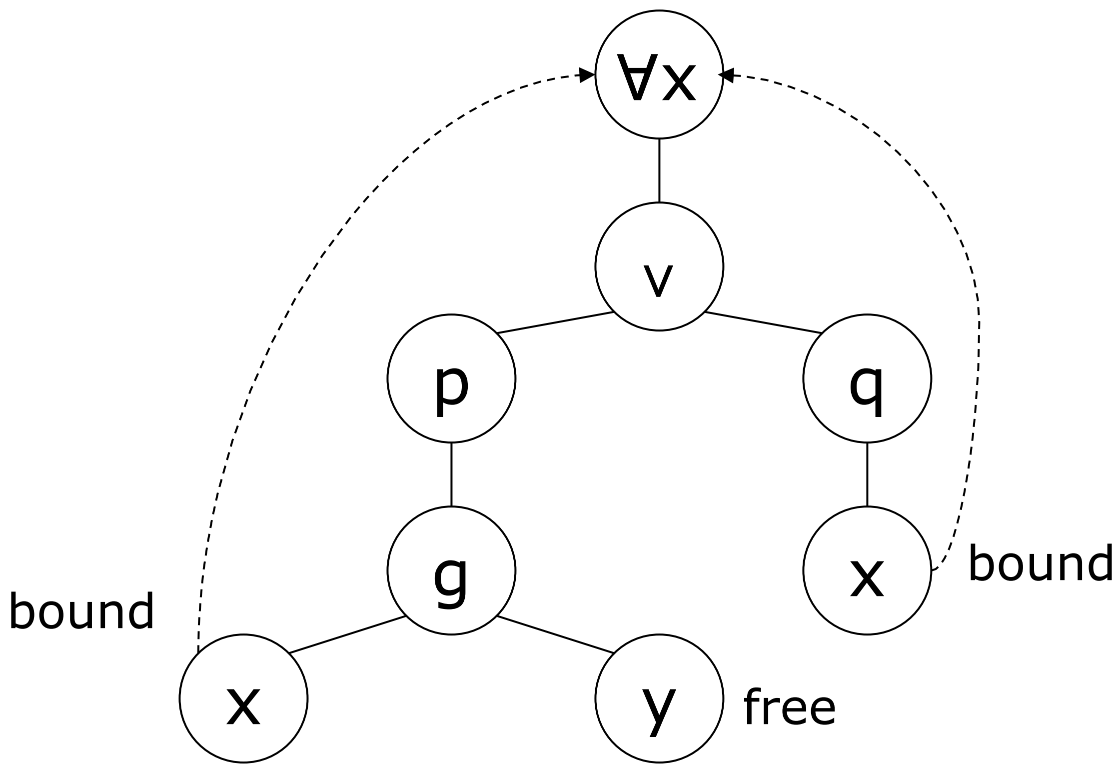
- While this is easy syntactically, there is a semantic restriction that must be observed:
  - In substituting a term for a variable within a formula, **no variables *within* the term can become bound** as a result of the substitution.

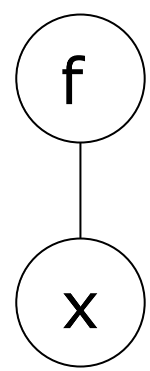
- If t is a term, v is a variable, and F is a formula, and the above restriction applies, we say that

"t is free to replace v in F"

(or more conventionally, **"t is free for v in F")**

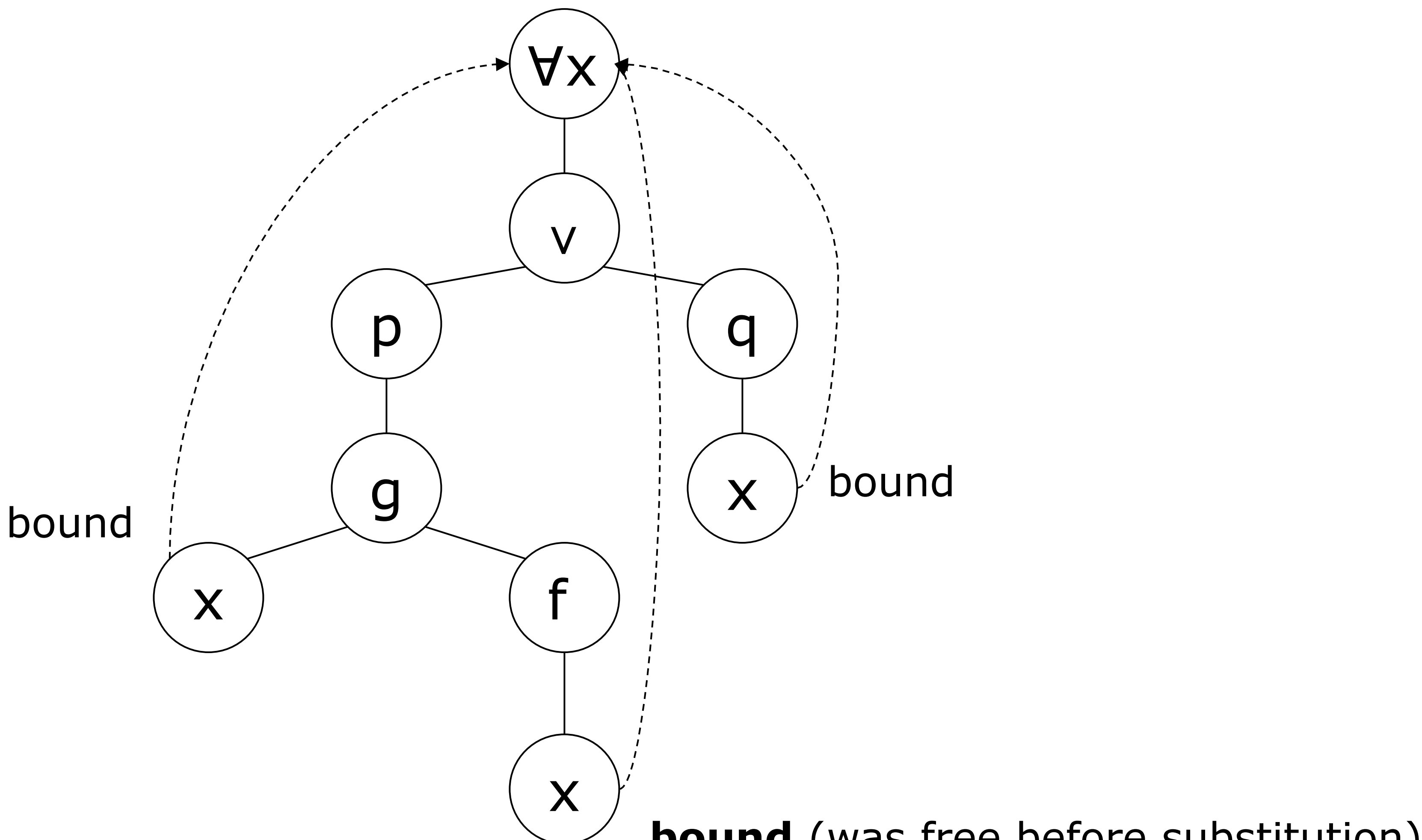
Non-Substitutability Example





f(x) is **not** free to replace y in
∀x (p(g(x, y)) v q(x))

Non-Substitutability Example



$\forall x\ (p(g(x,\ \mathbf{f(x)}))\ \vee\ q(x))$

Substitution Notation

- If t is a term, v is a variable, and F is a formula, and

  t is free to replace v in F

  then by

  F[t/v]

  we mean the result of substituting t for every **free** occurrence of v in F. (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 F 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 F.

F[f(z)/y] is $\forall x\ (p(g(x, f(z))) \lor q(x))$.

Substitution Notation Example

Let F 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 F (vacuously)
because there are no free instances of x.

F[f(x)/x] is the same as F;
there are no free instances of x in F.

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:
  - ∀
  - ∃
- These will be added to our propositional natural deduction rules.

∀-Elimination Rule ∀E

- $$\frac{\forall x\ \varphi}{\varphi[t/x]} \quad \forall E$$

  where t is any term that is free to replace x in φ.

- **What the rule says:**

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

Why the Substitution Qualification is Necessary

- $$\frac{\forall x\ \varphi}{\varphi[t/x]} \quad \forall E$$

  where t is any term that is free to replace x in φ.

- Correct example: z is free to replace x in ∃y p(y, x)
  1. ∀x ∃y p(y, x) Premise
  2. ∃y p(y, z) ∀E 1 (substituting **z** for x)

- Incorrect example: y is **not** free to replace x in ∃y p(y, x)
  1. ∀x ∃y p(y, x) Premise
  2. ∃y p(y, y) ∀E 1 (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.

Fresh x_0

.

.

.

$\varphi[x_0/x]$

$$\frac{\boxed{\begin{array}{c}\text{Fresh } x_0 \\ \vdots \\ \varphi[x_0/x]\end{array}}}{\forall x\ \varphi}\ \forall I$$

- x_0 is a "fresh" variable otherwise unused in the proof.
- x_0 must naturally be free to replace x in φ, but since x_0 is "fresh", this should never be an issue; It can't become bound.

∀-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.

Work backward to use ∀I

- Unless your goal is proving something of the form ∀x … you won’t know to open a box with a fresh variable.

∀E ∀I Example

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

$\forall x\ p(x)$	Premise
$\forall y\ \ p(y)$	Desired conclusion

Work backward

∀E ∀I Example

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

1.		$\forall x\ p(x)$	Premise
2.	y_0		Fresh var
3.		$p(y_0)$	
4.	$\forall y\ p(y)$		∀I

∀E ∀I Example

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

1.	$\forall x\ p(x)$	Premise
2.	y_0	Fresh var
3.	$p(y_0)$	1, ∀E
4.	$\forall y\ \ p(y)$	2-3, ∀I

∀E ∀I Example

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

1.	$\forall x\,(p(x) \to q(x))$	Premise
2.	$\forall x\ \ p(x)$	Premise
3.	x_0	Fresh var
4.		
5.	$q(x_0)$	
6.	$\forall x\, q(x)$	∀I

∀E ∀I Example

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

1.	$\forall x\,(p(x) \rightarrow q(x))$	Premise
2.	$\forall x\;p(x)$	Premise
3.	x_0	Fresh var
4.	$p(x_0) \rightarrow q(x_0)$	1, ∀E
5.	$p(x_0)$	2, ∀E
6.	$q(x_0)$	4, 5 →E
7.	$\forall x\,q(x)$	3-6, ∀I

∀E ∀I English Equivalent

- Derive $\forall x\, (p(x) \rightarrow q(x)), \forall x\, p(x) \vdash \forall x\, q(x)$:
- "Assume $\forall x\, (p(x) \rightarrow q(x))$ and $\forall x\, p(x)$.

  Let x_0 be an arbitrary element. [open a box]

  From the the first assumption $p(x_0) \rightarrow q(x_0)$, and from the second $p(x_0)$, hence also $q(x_0)$ by *modus ponens*.

  Since x_0 was chosen arbitrarily, $q(x_0)$ gives us $\forall x\, q(x)$."
  [close box]

∀E ∀I Example

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

  $\forall x\ \forall y\ p(x, y)$ Premise

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

Where ∀I is to be used, work backward.

∀E ∀I Example

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

1.		$\forall x\ \forall y\ p(x, y)$	Premise
2.	y_0		Fresh
3.	x_0		Fresh
4.		$\forall y\ p(x_0, y)$	1, ∀E
5.		$p(x_0, y_0)$	4, ∀E
6.		$\forall x\ p(x, y_0)$	3-5, ∀I
7.		$\forall y\ \forall x\ p(x, y)$	2-6, ∀I

∃-Introduction Rule (∃I)

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

  where t is any term that is free to replace x in φ.

- **What the rule says:**

  If we have exhibited a formula φ 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)

- $\dfrac{\varphi[t/x]}{\exists x\ \varphi}$ ∃E_

  where t is any term that is free to replace x in φ.

- 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 ∨-Introduction.

Why lose information?

- For one thing, the specific term t derived might not be “exportable”;

  it could depend on some fresh variable introduced inside the box.

∀E ∃I Example

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

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

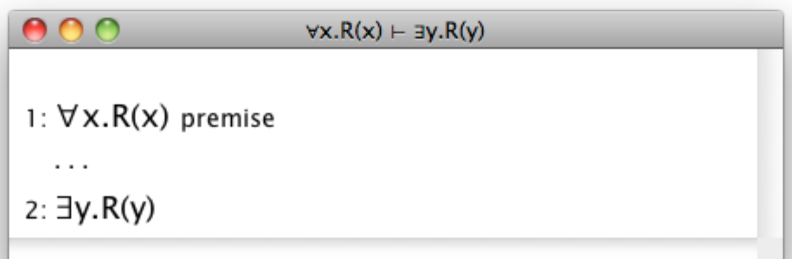
Note: x is free to replace x in p(x),
since nothing is bound in p(x).

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**.

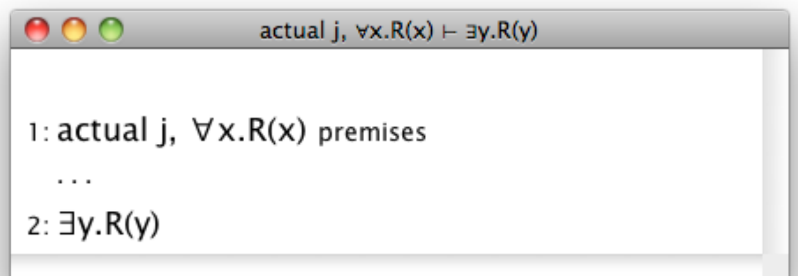
Most treatments assume this, but not all.
For example, Richard Bornat, the author of JAPE does not.
(Allowing empty domains is analogous to allowing 0 as a number.)

JAPE Examples

Not JAPE-Provable (no assumption that something exists)

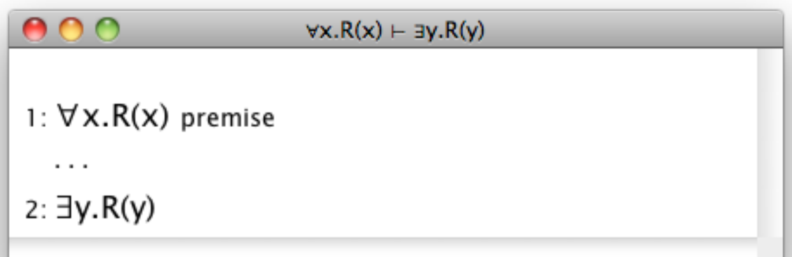


JAPE-Provable (actual a means something exists)

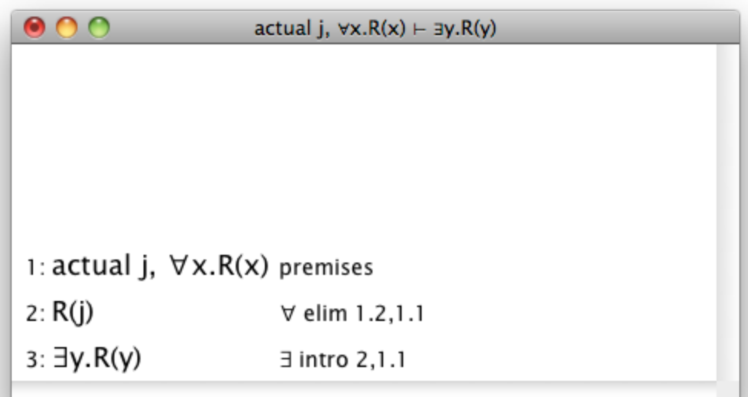


JAPE Examples

This is listed in Invalid Conjectures.



JAPE Proof



∃-Elimination Rule (∃E)

	x_0	$\varphi[x_0/x]$
		.
		.
		.
$\exists x \varphi$		χ

$$\frac{\exists x\varphi \qquad \boxed{x_0 \quad \varphi[x_0/x] \; \ldots \; \chi}}{\chi} \quad (\exists E)$$

- Here x_0 is a "fresh" variable otherwise unused in the proof.
- x_0 must be free to replace x in φ,
  but since x_0 is "fresh", this should never be an issue.
- This rule is analogous to ∨ Elimination.

∃-Elimination Rule (∃x E)

$$\dfrac{\exists x\ \varphi \qquad \boxed{\begin{array}{ll} x_0 & \varphi[x_0/x] \\ & \vdots \\ & \chi \end{array}}}{\chi}\ \exists E$$

- **What this rule says:**
- Assume that we have derived $\exists x\ \varphi$. One use we can make of this fact is to let x_0 be ***an*** x such that $\varphi[x_0/x]$. There can be no other constraints on x_0. If we then derive χ from the assumption about φ, then we can conclude χ in general.

∃I ∃E Example

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

1.		∀x (p(x) → q(x))	Premise
2.		∃x p(x)	Premise
3.	x_0	$p(x_0)$	Fresh var, Assumption
4.		$p(x_0) \rightarrow q(x_0)$	1, ∀E
5.		$q(x_0)$	3, 4, →e
6.		(∃x) q(x)	5, ∃I
7.		(∃x) q(x)	2, 3-6, ∃E

same formula (lines 6 and 7)

- In the ∃E rule, φ is identified with p(x), while χ is identified with ∃x q(x).

- Try not to be confused by the fact that ∃ is in the conclusion. The ∃ in 2 is what was eliminated.

∃I ∃E Example in English

- Derive $\forall x\ (p(x) \rightarrow q(x)),\ \exists x\ p(x)\ \vdash \exists x\ q(x)$:
- "Assume $\forall x\ (p(x) \rightarrow q(x))$ and $\exists x\ p(x)$.

  Let x_0 be such that $p(x_0)$.

  By the first assumption, $p(x_0) \rightarrow q(x_0)$.
  Hence $q(x_0)$ by modus ponens.

  As we've exhibited an x such that $q(x)$,
  conclude $\exists x\ q(x)$."

∃I ∃E **Incorrect** Proof Example

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

1.		$\forall x\,(p(x) \rightarrow q(x))$	Premise
2.		$\exists x\, p(x)$	Premise
3.	x_0	$p(x_0)$	∃E
4.		$p(x_0) \rightarrow q(x_0)$	1, ∀E
5.		$q(x_0)$	3, 4, →E
6.		$q(x_0)$	3-5, ∃E
7.		$(\exists x)\, q(x)$	6, ∃I

- Why incorrect?
- Formulas containing x_0 cannot be carried outside the box.
- The box for ∃E has two purposes:
  - Restricting the scope of the introduced variable.
  - Restricting the scope of the assumption.

Caution: ∃E

- Normally, ∃E can only be used to introduce a variable once. You cannot use it to introduce a second distinct variable.

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

- In contrast, you can use ∃I as many times as you want (not that it will always help).

Quantifier rule summary

	Introduction	**Elimination**
$\forall$	Fresh x_0 . . . $\varphi[x_0/x]$ ——— $\forall x\ \varphi$ $\quad \forall I$	$\forall x\ \varphi$ ——— $\varphi[t/x]$ $\quad \forall E$ (t is free to replace x)
$\exists$	$\varphi[t/x]$ ——— $\exists x \varphi$ $\quad \exists I$ (t is free to replace x)	x_0 $\quad \varphi[x_0/x]$ . . . χ $\exists x\ \varphi$ ——— χ $\quad \exists E$

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 that are of interest. In natural deduction, these formulas will appear on the left-hand side of a sequent.

- It can then be proved separately that the syntactic rules are in agreement with the semantics of the intended **interpretation**.

Interpretation I = (Δ, μ)

- An **interpretation** for a set of terms and formulas consists of:
  - A (usually non-empty) **domain** Δ: that 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 μ are the values ***assigned*** by the interpretation.
- The domain, Δ, may also be called the “universe” or “domain of discourse”.
- Non-empty domain is required if there are function symbols.

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 the same as 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 = (Δ, μ)

- If a formula has **free variables**, add a ∀ quantifier for each such variable in front of the formula. (Free variables are understood to mean ∀-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**.
- This method will break down the formula recursively.

Truth of a Formula
Relative to an Interpretation I = (Δ, μ)

- For any closed formula φ, we will define the truth value I[φ] ∈ {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 $\forall x$** φ, then

  $I[\forall x\ \varphi] = T$ iff $I[\varphi[d/x]] = T$ for every $d \in \Delta$.

- Recalling that we are using d as **constant** symbol in the case of substitution.

Truth of a Formula
Relative to an Interpretation I = (Δ, μ)

- If the formula is of the **form ∃x** φ, then

  I[∃x φ] = T iff

  I[φ[d/x]] = T for *some* d ∈ Δ.

Truth of a Formula
Relative to an Interpretation I = (Δ, μ)

- I[φ ∧ ψ] = T iff I[φ] = T **and** I[ψ] = T.

- (Recall there are no free variables.)

- I[φ ∨ ψ] = T iff I[φ] = T **or** I[ψ] = T.

Truth of a Formula Relative to an Interpretation $I = (\Delta, \mu)$

- $I[\varphi \rightarrow \psi] = T$ iff either $I[\varphi] = F$ or $I[\psi] = T$.
- $I[\varphi \leftrightarrow \psi] = T$ iff $I[\varphi] = I[\psi]$.
- $I[\neg\varphi] = T$ iff $I[\varphi] = F$.
- If the formula φ 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 a value $I[t] \in \Delta$ of each **term** t recursively:
  - If t is a **constant symbol** c, then $I[t] = \mu(c)$, the assigned value in Δ.
  - 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 a value $I[E] \in \{T, F\}$ of each atomic formula E **recursively**:
  - If E is an **atomic formula** $p(t_1, t_2, \ldots, t_n)$, where p is an n-ary predicate symbol, and the t_i are terms, 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 of terms** definition 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
  - Δ = {0, 1, 2} domain
  - μ(c) = 0 constant
  - μ(f) = {0 $\rightarrow$ 1, 1 $\rightarrow$ 2, 2 $\rightarrow$ 0} function
  - μ(q) = {(2, 0), (1, 2), (0, 1))} predicate
    the set of pairs for which μ(q) is T
- Thus:

  - I[f(f(c)] = μ(f)(I[f(c)]) = μ(f)(μ(f)(I[c])) = μ(f)(μ(f)(μ(c))) = 2

  - I(q(f(f(c)), c)) = μ(q)(I[f(f(c)], μ(c)) = μ(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.

- *Suppose* interpretation I assigns
  - Δ = {0, 1, 2} domain
  - μ(c) = 0 constant
  - μ(f) = {0 → 1, 1 → 2, 2 → 0} function
  - μ(q) = {(2, 0), (1, 2), (0, 1))} predicate
    the set of pairs for which μ(q) is T
- Thus:

  - I[f(f(c)] = μ(f)(I[f(c)]) = μ(f)(μ(f)(I[c])) = μ(f)(μ(f)(μ(c))) = 2

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

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,

  $$\forall x\ P(x) \models \exists x\ P(x)$$

  The premise is *vacuously* true for an empty-domain, but the conclusion cannot be true.

A Prolog Programming for Evaluting I[ ] **when the domain is finite**

- See truth.pro on the website.

Satisfaction and Models

- An interpretation I ***satisfies*** a formula E iff I [E] = T.
- We also say that I ***is a model for*** E when I [E] = T.
- Caution: Some authors, such as Huth&Ryan, use “model” to mean “interpetation”.
- 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,

  $$\varphi_1, \ldots, \varphi_n \models \psi$$

  means:

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

  also satisfies ψ.

- $\Gamma \models \psi$, where Γ is a ***set*** of formulas, can be restated, by extending **model** to mean an interpretation satisfies the entire set Γ, as:

  Every model for Γ is also a model for ψ.

Validity

- When the left-hand side is empty:

  $\models \psi$

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

- In this case, every interpretation for ψ 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.

Soundness and Completeness

- As with propositional logic, we define:
- **Soundness** of a set of derivation rules:

  For any set of formulas Γ and any formula ψ:

  $\Gamma \vdash \psi$ implies $\Gamma \models \psi$

- **Completeness** of a set of derivation rules:

  For any set of formulas Γ and any formula ψ:

  $\Gamma \models \psi$ 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 Valid and 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:
  - $\Delta = \{1, 2\}$
  - $\mu(A) = \{2\}$
  - $\mu(B) = \{2\}$

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.

ND Equality Rules

- Natural Deduction typically introduces rules for equality (from which the axioms can be derived).
  - $$\frac{}{t = t} \quad =\text{I where t is any term}$$
  - $$\frac{s = t \qquad \varphi[s/x]}{\varphi[t/x]} \quad =\text{E}$$

where s and t are any terms
and x is any variable, provided
s and t are free to replace x in φ.

Equality Formulas ("Axioms" in some systems) (Derivable from ND Rules)

- Four types of formulas characterize equality:
  - $\forall x\ (x = x)$ — reflexive
  - $\forall x\ \forall y\ (x = y \rightarrow y = x)$ — symmetric
  - $\forall x\ \forall y\ \forall z\ ((x = y \wedge y = z) \rightarrow (x = z))$ — transitive
  - $\forall x_1 \ldots \forall x_n\ \forall y_1 \ldots \forall y_n$ $(x_1 = y_1 \wedge \ldots \wedge x_n = y_n) \rightarrow (f(x_1, \ldots x_n) = f(y_1, \ldots y_n))$ — substitution
    where f is any n-ary function symbol
  - $\forall x_1 \ldots \forall x_n\ \forall y_1 \ldots \forall y_n$ $(x_1 = y_1 \wedge \ldots \wedge x_n = y_n) \rightarrow (p(x_1, \ldots x_n) \rightarrow p(y_1, \ldots y_n))$ — substitution
    where p is any n-ary predicate symbol

Example

$$\frac{}{t = t} \; =\text{I}$$

$$\frac{s = t \qquad \varphi[s/x]}{\varphi[t/x]} \; =\text{E}$$

- Prove the symmetry rule:

  u = v |— v = u

  where u and v are any terms, from the two ND equality rules.
  The "trick" here is finding the right φ.
  We use x = u for φ, so that φ[u/x] is u=u, an instance of =I. This gives φ[v/x] as v=u, the desired conclusion.

  1. u = v Premise
  2. u = u =I (identified as φ[u/x])
  3. v = u 1, 2, =E (identified as φ[v/x])

Example

$$\frac{}{t = t} \; {=}I$$

$$\frac{s = t \qquad \varphi[s/x]}{\varphi[t/x]} \; {=}E$$

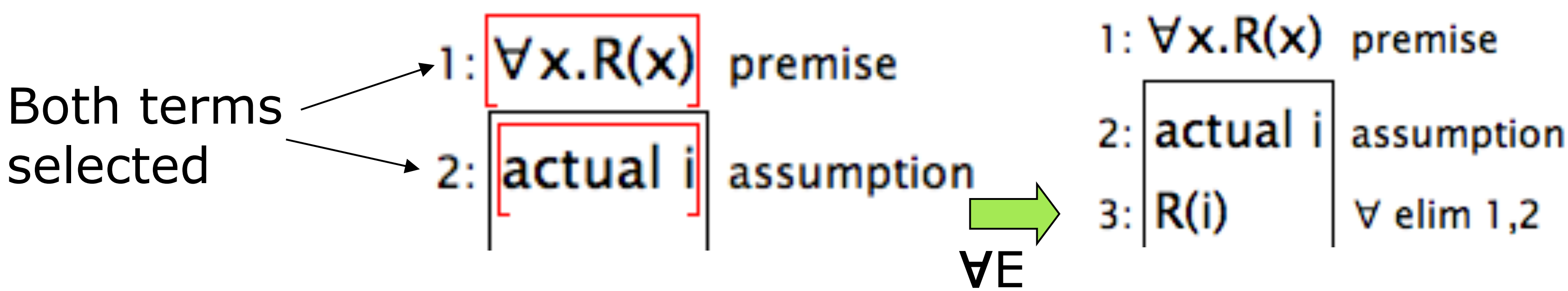
- Prove the transitivity rule:
  u = v, v = w |— u = w
  where u, v, and w are terms, from the two ND equality rules.
  Here we let φ be u = x, to use =E.
  We identify s = t in the rule with v = w.

1. v = w Premise
2. u = v Premise (identified as φ[v/x])
3. u = w 1, 2, =E (identified as φ[w/x])

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):



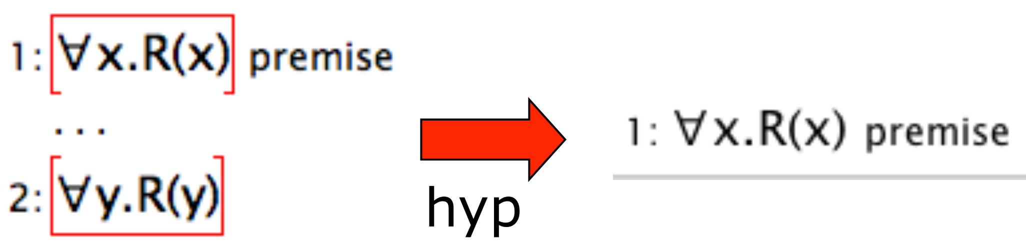
Note: If the red bracket opens **downward**, the item is usable as a hypothesis. If **upward**, a conclusion. In some cases both apply, and you need to click above or below to indicate which.

JAPE Examples

- ∀ Introduction (working backward), followed by ∀ Elimination (working forward)

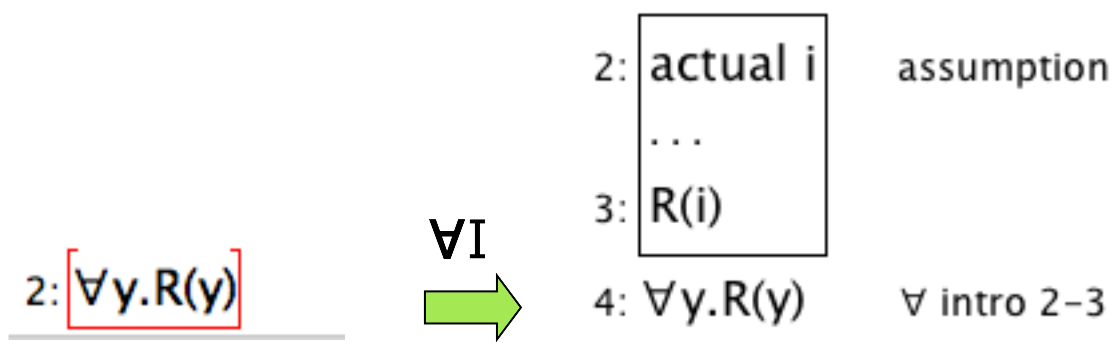
1:	∀x.R(x)	premise
2:	actual i	assumption
3:	R(i)	∀ elim 1,2
4:	∀y.R(y)	∀ intro 2-3

- 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.



JAPE Examples

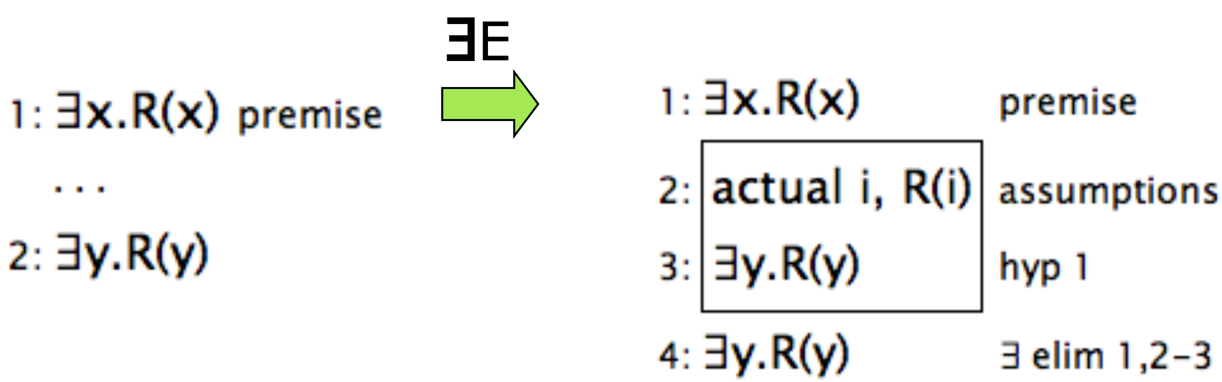
- **∀ Introduction** (working ***backward***) introduces a fresh variable. Variables are often helpful in completing a proof. Of course, the variable **can't be taken outside the box**.



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.



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)

no variable

- 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)

2: actual i, ∃y.R(i,y) assumptions

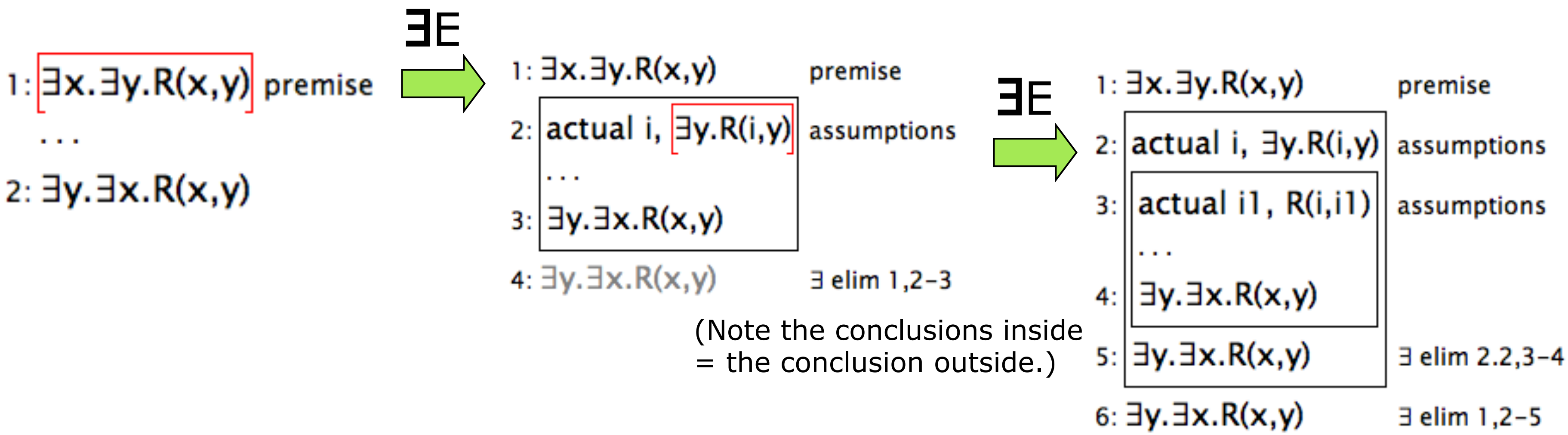
. . .

3: ∃x.R(x,i)

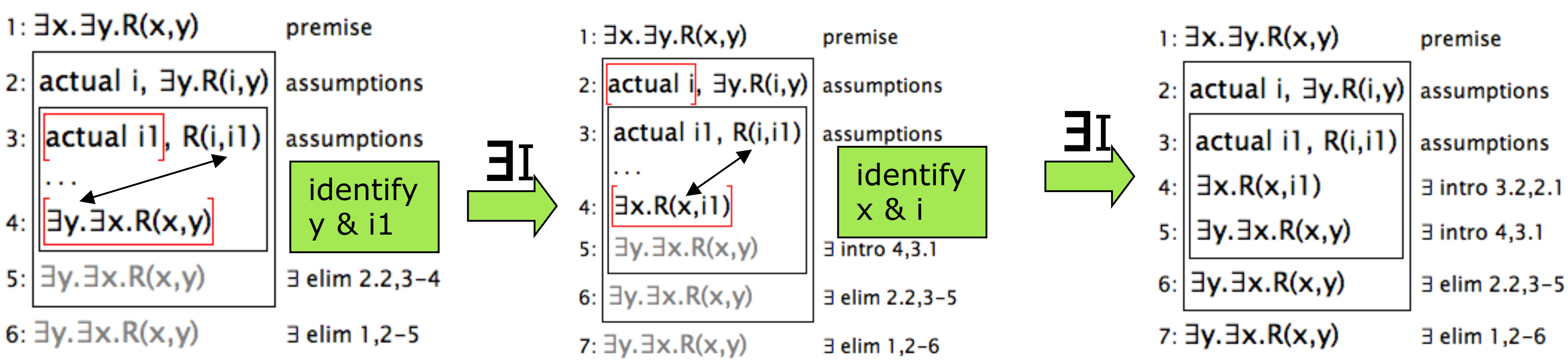
4: ∃y.∃x.R(x,y) ∃ intro 3,2.1

Proof of a sequent using ∃E and ∃I

Work forward to introduce variables, by **eliminating** ∃'s (opening boxes):



then introduce ∃'s working backward **in the right order**:



Example using all four quantifier rules

1:	∃x.∀y.R(x,y)	premise
	. . .	
2:	∀y.∃x.R(x,y)	

∀I →

1:	∃x.∀y.R(x,y)	premise
2:	actual i	assumption
	. . .	
3:	∃x.R(x,i)	
4:	∀y.∃x.R(x,y)	∀ intro 2–3

∃E →

1:	∃x.∀y.R(x,y)	premise
2:	actual i	assumption
3:	actual i1, ∀y.R(i1,y)	assumptions
	. . .	
4:	∃x.R(x,i)	
5:	∃x.R(x,i)	∃ elim 1,3–4
6:	∀y.∃x.R(x,y)	∀ intro 2–5

∃I →

1:	∃x.∀y.R(x,y)	premise
2:	actual i	assumption
3:	actual i1, ∀y.R(i1,y)	assumptions
	. . .	
4:	R(i1,i)	
5:	∃x.R(x,i)	∃ intro 4,3.1
6:	∃x.R(x,i)	∃ elim 1,3–5
7:	∀y.∃x.R(x,y)	∀ intro 2–6

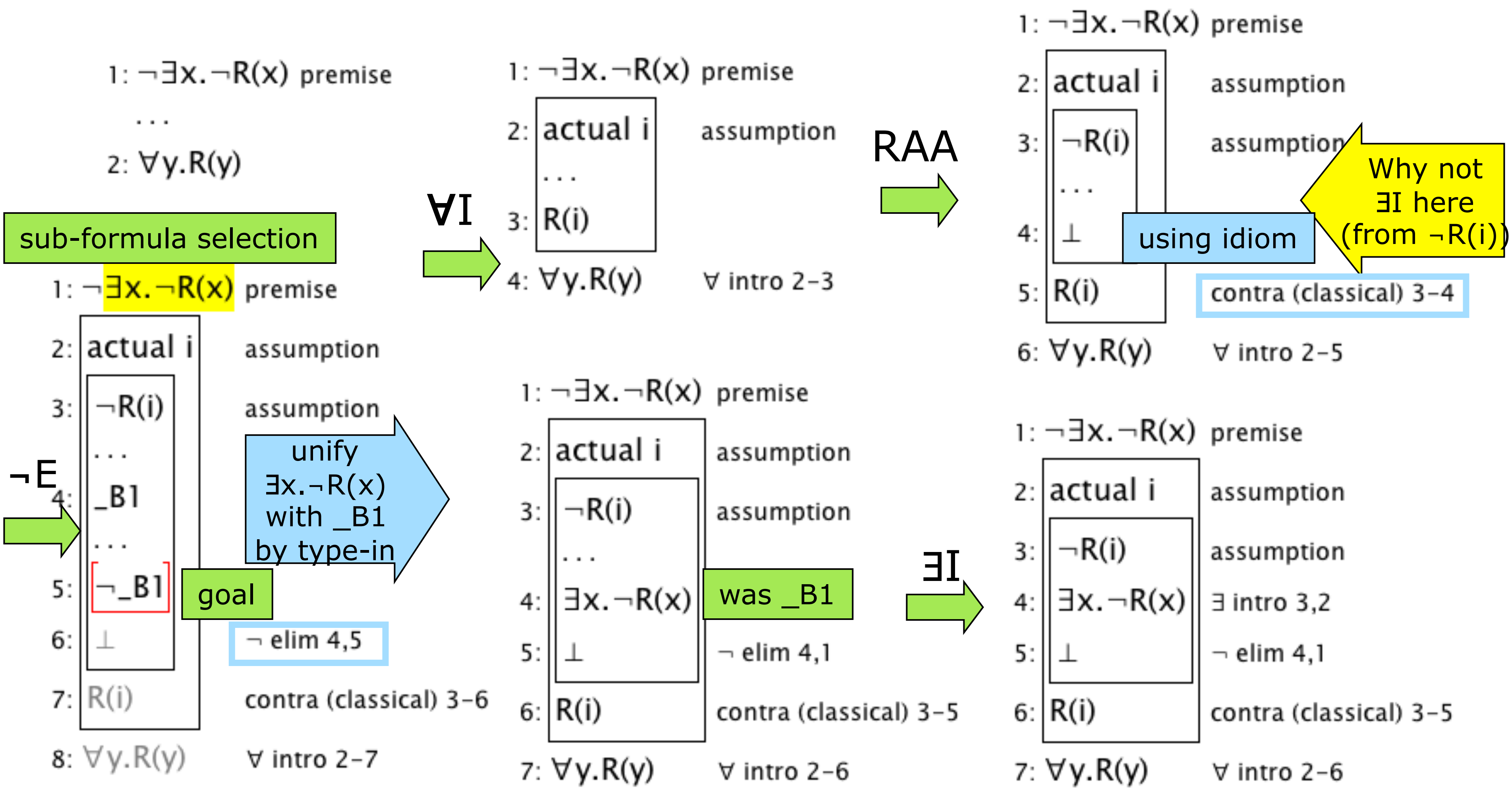
∀E →

1:	∃x.∀y.R(x,y)	premise
2:	actual i	assumption
3:	actual i1, ∀y.R(i1,y)	assumptions
4:	R(i1,i)	∀ elim 3.2,2
5:	∃x.R(x,i)	∃ intro 4,3.1
6:	∃x.R(x,i)	∃ elim 1,3–5
7:	∀y.∃x.R(x,y)	∀ intro 2–6

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
  - ¬ 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. ∃I won’t work forward (needs variable and body).
This example uses the previous idiom.



JAPE

- **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 ∃I needs a term.
- Can't go forward, because ∀E needs a variable.

∀ intro (introduces variable)
∃ intro (needs variable)

∀ elim (needs variable)
∃ elim (assumption & variable)

JAPE

- 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.T' as a premise. (one place where T is useful, but others could be used).

1:	actual i, $\forall x.R(x)$	premises
2:	$R(i)$	$\forall$ elim 1.2,1.1
3:	$\exists x.R(x)$	$\exists$ intro 2,1.1

1:	$\exists x.\top$, $\forall x.R(x)$	premises
2:	actual i, $\top$	assumptions
3:	$R(i)$	$\forall$ elim 1.2,2.1
4:	$\exists y.R(y)$	$\exists$ intro 3,2.1
5:	$\exists y.R(y)$	$\exists$ elim 1.1,2-4

- 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: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

A Tricky One

1: actual j, actual k — premises

. . .

2: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

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 — premises

2: $\_E \vee \neg \_E$ — Theorem $E \vee \neg E$

. . .

3: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

LEM to the rescue (used as a lemma)

unify R(j) with _E

1: actual j, actual k — premises

2: $R(j) \vee \neg R(j)$ — Theorem $E \vee \neg E$

. . .

3: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

$\vee$E

1: actual j, actual k — premises

2: $R(j) \vee \neg R(j)$ — Theorem $E \vee \neg E$

3: $R(j)$ — assumption

. . .

4: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

5: $\neg R(j)$ — assumption

. . .

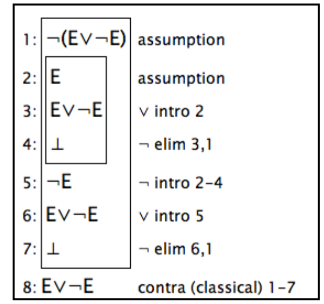
6: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

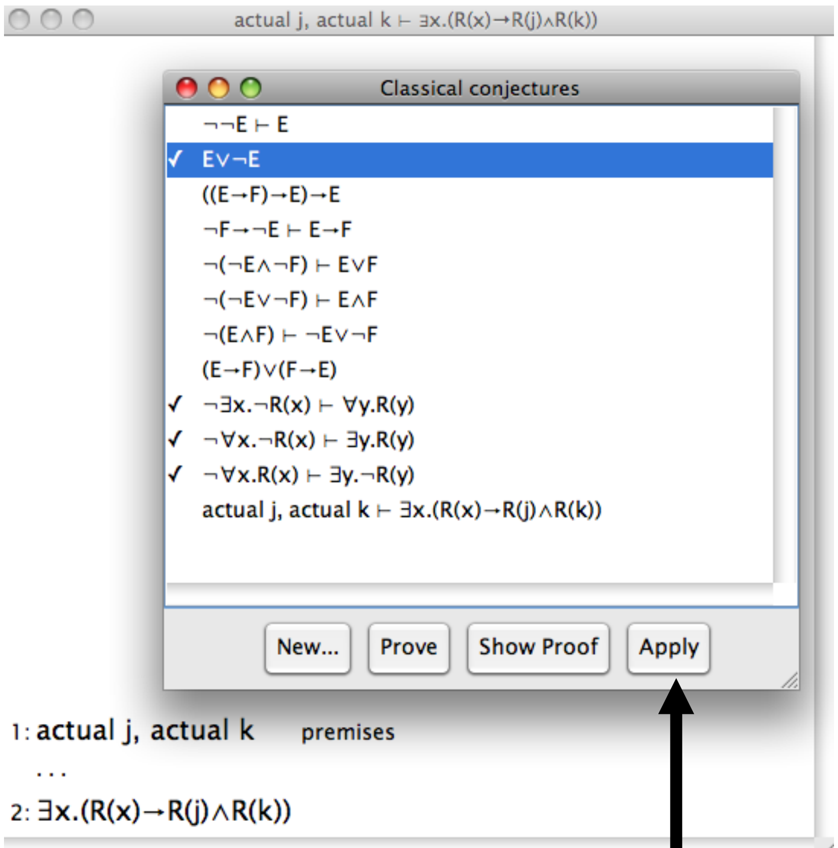
7: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$ — $\vee$ elim 2,3–4,5–6

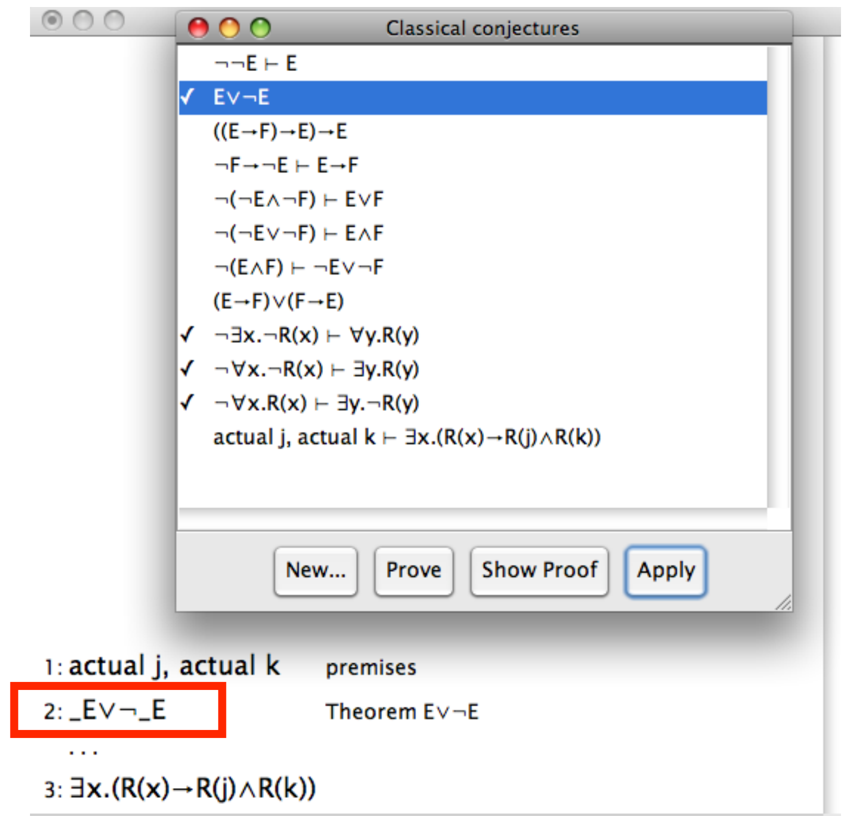
What x would make this work?

What x would make this work?

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 ∃I

3:	$R(j)$	assumption
	. . .	
4:	$\exists x.(R(x) \to R(j) \wedge R(k))$	

3:	$R(j)$	assumption
4:	$R(k)$	assumption
5:	$R(j) \wedge R(k)$	$\wedge$ intro 3,4
6:	$R(k) \to R(j) \wedge R(k)$	$\to$ intro 4-5
7:	$\exists x.(R(x) \to R(j) \wedge R(k))$	$\exists$ intro 6,1.2

Continuing the tricky proof …

For the Bottom Box

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

5:	$\neg R(j)$	assumption
	…	
6:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	

8:	$\neg R(j)$	assumption
9:	$R(j)$	assumption
10:	$\bot$	¬ elim 9,8
11:	$R(j) \wedge R(k)$	contra (constructive) 10
12:	$R(j) \rightarrow R(j) \wedge R(k)$	→ intro 9–11
13:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	∃ intro 12,1.1

Completed Proof

1:	actual j, actual k	premises
2:	$R(j) \vee \neg R(j)$	Theorem $E \vee \neg E$
3:	$R(j)$	assumption
4:	$R(k)$	assumption
5:	$R(j) \wedge R(k)$	$\wedge$ intro 3,4
6:	$R(k) \rightarrow R(j) \wedge R(k)$	$\rightarrow$ intro 4-5
7:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	$\exists$ intro 6,1.2
8:	$\neg R(j)$	assumption
9:	$R(j)$	assumption
10:	$\bot$	$\neg$ elim 9,8
11:	$R(j) \wedge R(k)$	contra (constructive) 10
12:	$R(j) \rightarrow R(j) \wedge R(k)$	$\rightarrow$ intro 9-11
13:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	$\exists$ intro 12,1.1
14:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	$\vee$ elim 2,3-7,8-13

An Analogous Sequent

1: actual i premise

. . .

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 ∨ ¬E
- But what is E?

1: actual i premise

...

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

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

- Use unification to assign formulat to E

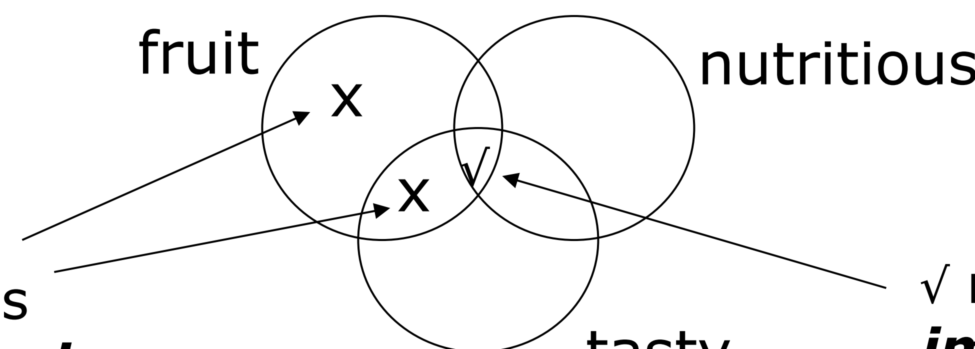
Syllogisms (WP)

- 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: Being a syllogism does not require validity.

Proving a Syllogism Using Venn Diagram

- All fruit is nutritious.
- Some fruit is tasty.
- Therefore, some tasty things are nutritious.



x means
excluded
possibilities
(fruit that is not nutritious)

√ means
implied possibility
non-empty category
(tasty that is nutritious)

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

Statement	**Translation**
All humans are mortal.	$\forall x\ (H(x) \rightarrow M(x))$
Socrates is a human.	$H(s)$
Socrates is mortal.	$M(s)$

This syllogism happens to be valid.

Syllogistic Forms

Statement Form	**Translation**
All S is/are/has… P.	$\forall x\ (S(x) \rightarrow P(x))$
Some S is P.	$\exists x\ (S(x) \land P(x))$
No S is P.	$\neg\exists x\ (S(x) \land P(x))$
Some S is not P.	$\exists x\ (S(x) \land \neg P(x))$
No S is not P.	$\neg\exists x\ (S(x) \land \neg P(x))$
All S is not P.	$\forall x\ (S(x) \rightarrow \neg P(x))$

Are any forms equivalent to one another?

Example: Translate this syllogism, then try to prove it.

- All fruit is nutritious.
- Some fruit is tasty.
- Some tasty things are nutritious.

Example: Translate this syllogism, then try to prove it.

- No humans are perfect.
- All perfect creatures are mythical.
- Some mythical creatures are not human.

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.

Constructive $\rightarrow$ vs. Classical $\leftarrow$

1:	$E \wedge F$	premise
2:	$\neg E \vee \neg F$	assumption
3:	$\neg E$	assumption
4:	E	$\wedge$ elim 1
5:	$\bot$	$\neg$ elim 4,3
6:	$\neg F$	assumption
7:	F	$\wedge$ elim 1
8:	$\bot$	$\neg$ elim 7,6
9:	$\bot$	$\vee$ elim 2,3–5,6–8
10:	$\neg(\neg E \vee \neg F)$	$\neg$ intro 2–9

1:	$\neg(\neg E \vee \neg F)$	premise
2:	$\neg E$	assumption
3:	$\neg E \vee \neg F$	$\vee$ intro 2
4:	$\bot$	$\neg$ elim 3,1
5:	E	contra (classical) 2–4
6:	$\neg F$	assumption
7:	$\neg E \vee \neg F$	$\vee$ intro 6
8:	$\bot$	$\neg$ elim 7,1
9:	F	contra (classical) 6–8
10:	$E \wedge F$	$\wedge$ intro 5,9

Constructive → vs. Classical ←

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

1:	$\neg\exists x.\neg R(x)$	premise
2:	actual i	assumption
3:	$\neg R(i)$	assumption
4:	$\exists x.\neg R(x)$	$\exists$ intro 3,2
5:	$\bot$	$\neg$ elim 4,1
6:	$R(i)$	contra (classical) 3-5
7:	$\forall y.R(y)$	$\forall$ intro 2-6

Note Rule Parallels

1:	$E \wedge F$	premise
2:	$\neg E \vee \neg F$	assumption
3:	$\neg E$	assumption
4:	E	$\wedge$ elim 1
5:	$\bot$	$\neg$ elim 4,3
6:	$\neg F$	assumption
7:	F	$\wedge$ elim 1
8:	$\bot$	$\neg$ elim 7,6
9:	$\bot$	$\vee$ elim 2,3-5,6-8
10:	$\neg(\neg E \vee \neg F)$	$\neg$ intro 2-9

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

Note Rule Parallels

1:	$\neg(\neg E \vee \neg F)$	premise
2:	$\neg E$	assumption
3:	$\neg E \vee \neg F$	$\vee$ intro 2
4:	$\perp$	$\neg$ elim 3,1
5:	E	contra (classical) 2–4
6:	$\neg F$	assumption
7:	$\neg E \vee \neg F$	$\vee$ intro 6
8:	$\perp$	$\neg$ elim 7,1
9:	F	contra (classical) 6–8
10:	$E \wedge F$	$\wedge$ intro 5,9

1:	$\neg\exists x.\neg R(x)$	premise
2:	actual i	assumption
3:	$\neg R(i)$	assumption
4:	$\exists x.\neg R(x)$	$\exists$ intro 3,2
5:	$\perp$	$\neg$ elim 4,1
6:	$R(i)$	contra (classical) 3–5
7:	$\forall y.R(y)$	$\forall$ intro 2–6

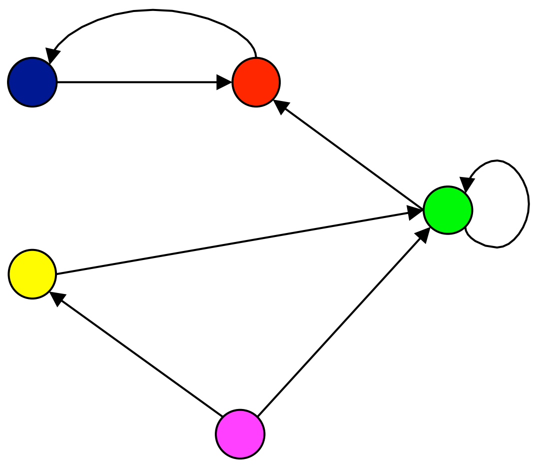
Fun with Relations

- A 2-ary predicate represents a binary relation, i.e. a set of pairs of domain elements.

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

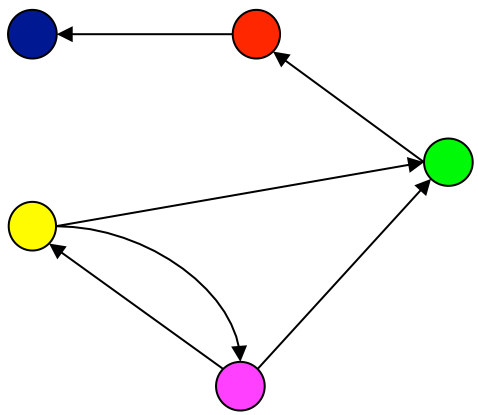
- In the following, what formula characterizes each relation represented by predicate L (sometimes using “loves” for analogy), and possibly the predicate =.

Everybody loves somebody.



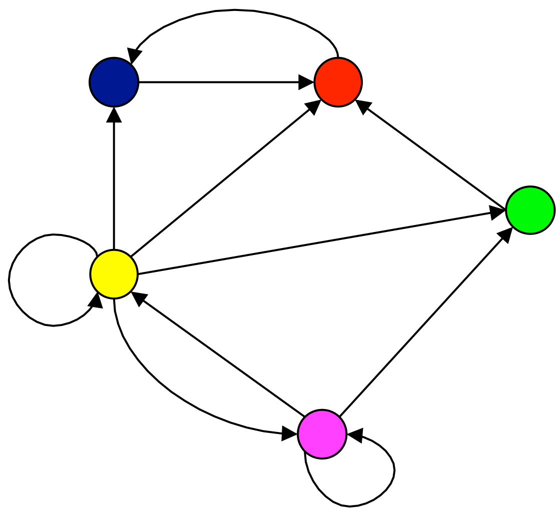
$\forall x \, \exists y \, L(x, y)$

Everybody is loved by somebody.

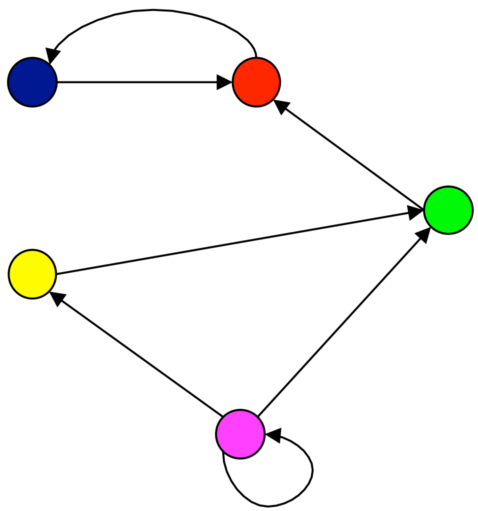


Somebody loves everybody.

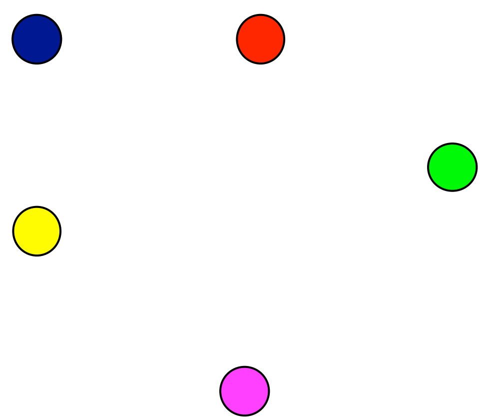
"Pollyanna"



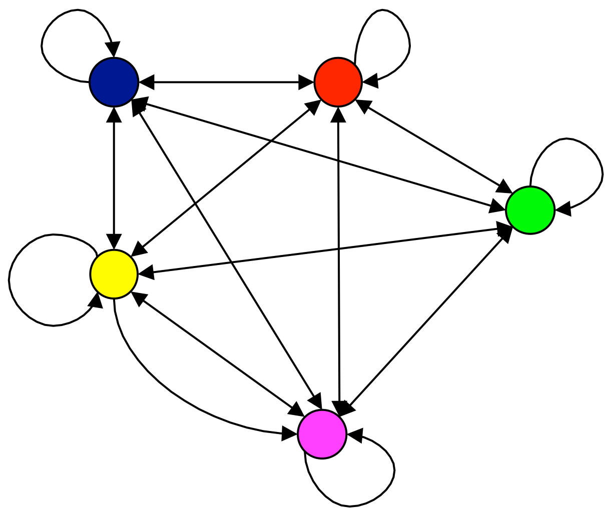
Nobody loves everybody.



Nobody loves somebody.

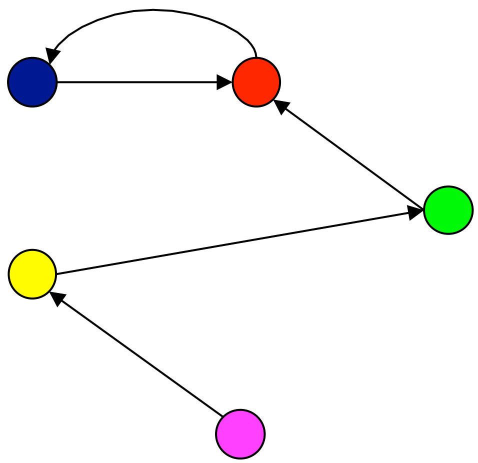


Everybody loves everybody.

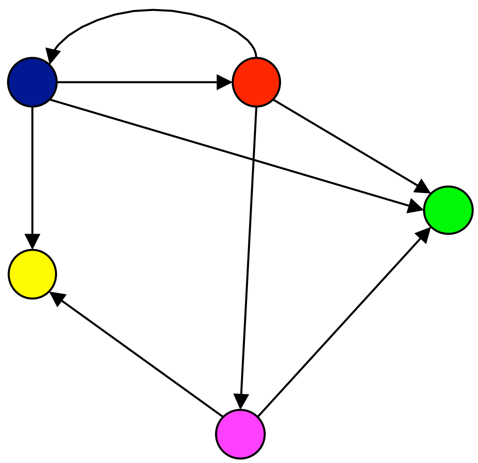


“Commune”

Everybody loves exactly one.

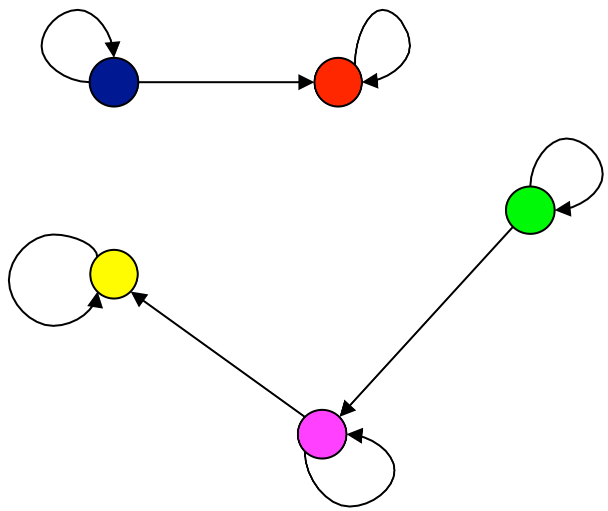


Nobody loves exactly one.

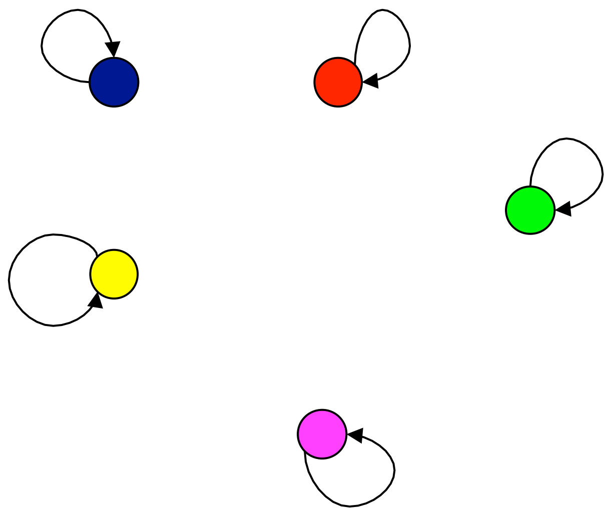


Everybody loves him/herself.

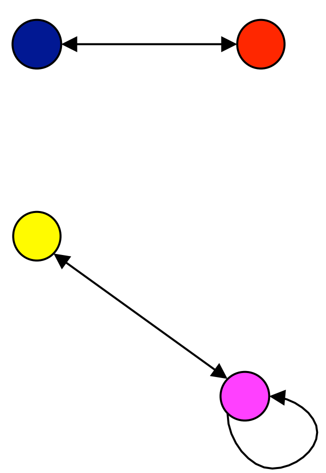
"Reflexive"



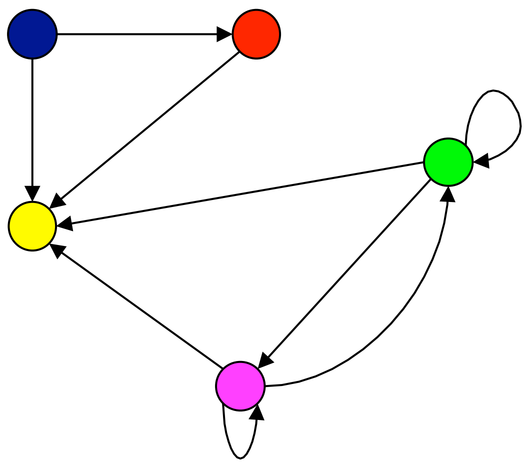
Everybody loves him/herself and only him/herself.



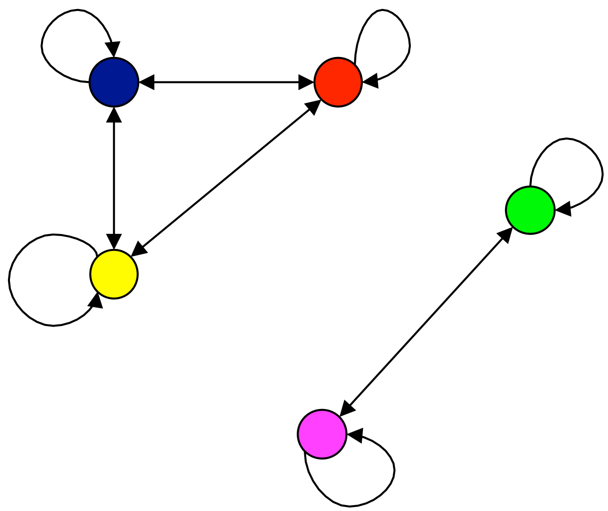
L is symmetric.
If x loves y then y loves x.



L is transitive.

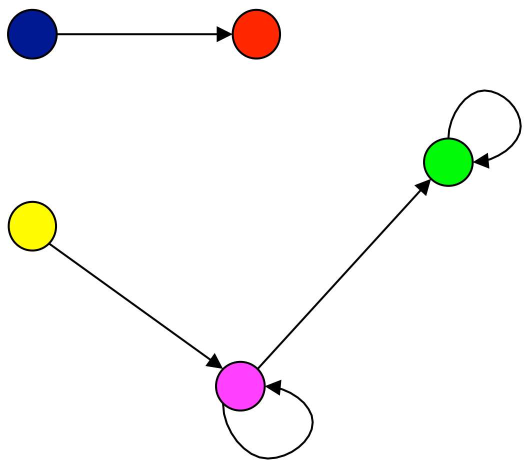


L is an equivalence relation

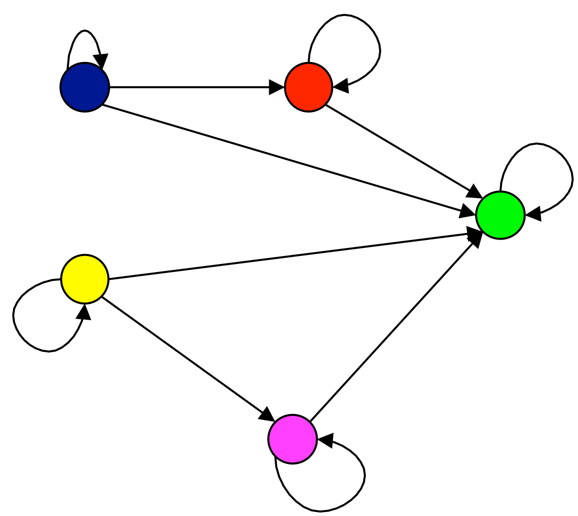


Reflexive,
symmetric,
transitive

L is antisymmetric.

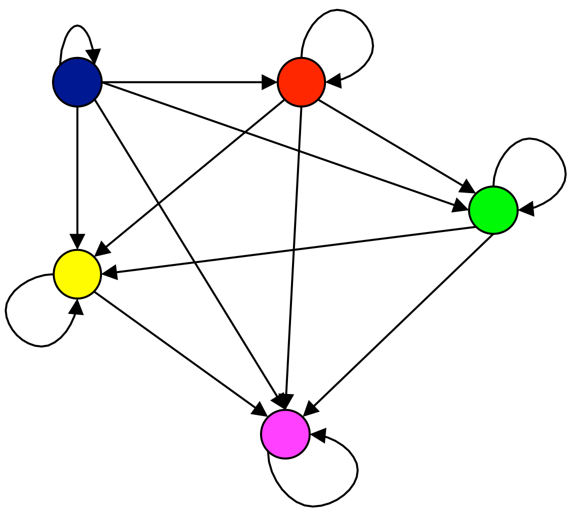


L is a partial order ("poset").



Reflexive,
Antisymmetric,
Transitive

L is a linear (or total) order.



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:	$\forall x.\forall y.\forall z.((R(x,y)\wedge R(y,z))\rightarrow R(x,z))$	premise
3:	actual i	assumption
4:	$\forall y.\forall z.((R(i,y)\wedge R(y,z))\rightarrow R(i,z))$	$\forall$ elim 2,3
5:	$\forall y.(R(i,y)\rightarrow R(y,i))$	$\forall$ elim 1.2,3
6:	$\exists y.R(i,y)$	$\forall$ elim 1.1,3
7:	actual i1	assumption
8:	$R(i,i1)$	assumption
9:	$\forall z.((R(i,i1)\wedge R(i1,z))\rightarrow R(i,z))$	$\forall$ elim 4,7
10:	$(R(i,i1)\wedge R(i1,i))\rightarrow R(i,i)$	$\forall$ elim 9,3
11:	$R(i,i1)\rightarrow R(i1,i)$	$\forall$ elim 5,7
12:	$R(i1,i)$	$\rightarrow$ elim 11,8
13:	$R(i,i1)\wedge R(i1,i)$	$\wedge$ intro 8,12
14:	$R(i,i)$	$\rightarrow$ elim 10,13
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:
  - $\forall x \forall y \exists z F(x, y, z)$
  - $\forall x \forall y \forall z \forall z' (F(x, y, z) \land F(x, y, z') \rightarrow z = z')$
- We'd still need axioms for equality.

Example: Group theory without function symbols (c is unit)

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