Predicate Logic

Robert Keller
February 2009

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
  ∀V E | // Universally-quantified formula
  ∃V E // Existentially-quantified formula

- Precedence, tightest first: ∀ ∃ ¬ ∧ ∨ → ↔
- Atom (A) now requires a more complex production

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.

Examples of Terms

- b constant
- y variable
- 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 one.
  - p(b)
  - ¬q(y)
  - ¬p(f(b, y))
  - r(a, g(h(b), c, h(y)))
- Literals become important in resolution theorem proving.

Examples of Quantifier-Free Formulas

- $p(b) \vee p(c)$
- $p(y) \wedge q(y)$
- $p(f(b, y)) \rightarrow q(y)$
- $\neg r(a, g(h(b), c, h(y)))$

Examples of Formulas

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

- (Quantifier-free formulas are also formulas.)

Preview of Semantics

- We will give details of semantics later on. However, a preview is helpful to understand certain syntactic considerations.
- Predicate logic describes characteristics of particular kinds of structures, such as sets with certain algebraic properties.

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).

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 one-to-one.]

Example:
Interpretations for “Groups”

- The domain is non-empty (as always).
- The domain can be finite or infinite.
- There is a constant symbol u (unit).
- There is a 2-ary function f (group multiplication).
- There is a 2-ary predicate = (equals).

Some formulas for groups

- $\forall x\ f(u, x) = x$

  [u is an identity]

- $\forall x\ \forall y\ \forall z\ f(x, f(y, z)) = f(f(x, y), z)$

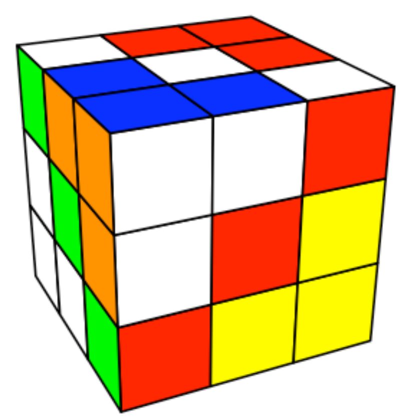
  [f is associative]

- $\forall x\ \exists y\ f(x, y) = u$

  [existence of inverse]

Examples of Groups

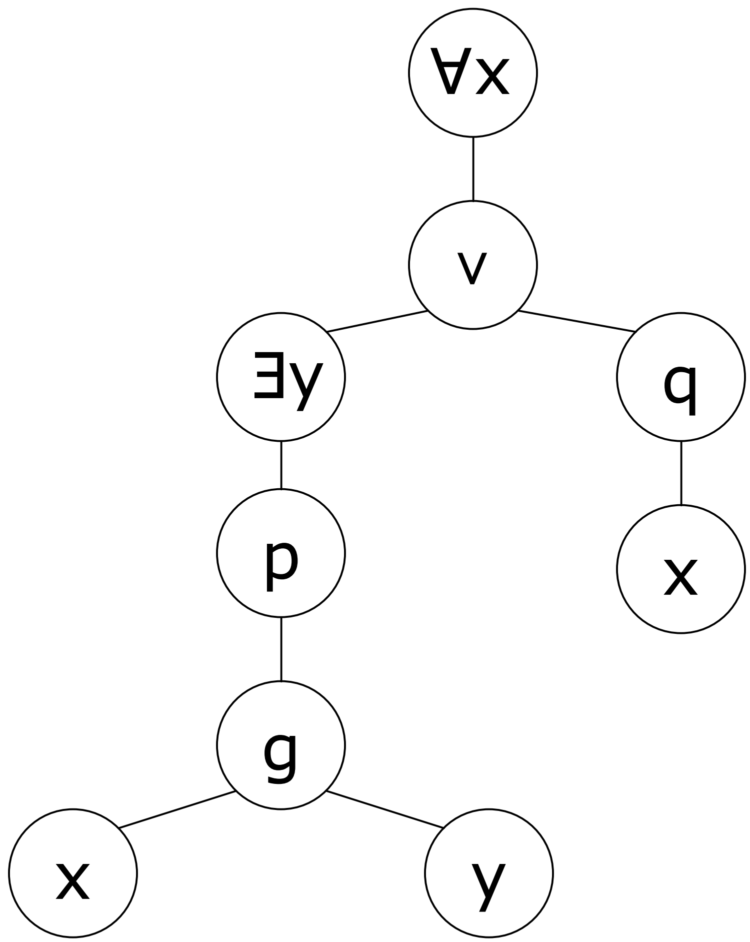
- Trivial group: {0} u = 0, f(0, 0) = 0
- 2-element group: {0, 1} u = 0, f(x, y) = x + y (mod 2)
- p-element group: {0, 1, …, p-1} for any prime p,
  u = 0, f(x, y) = x + y (mod p)
- Rubik's cube twists
- Particle spins (physics)
- Tire rotations
- Many others



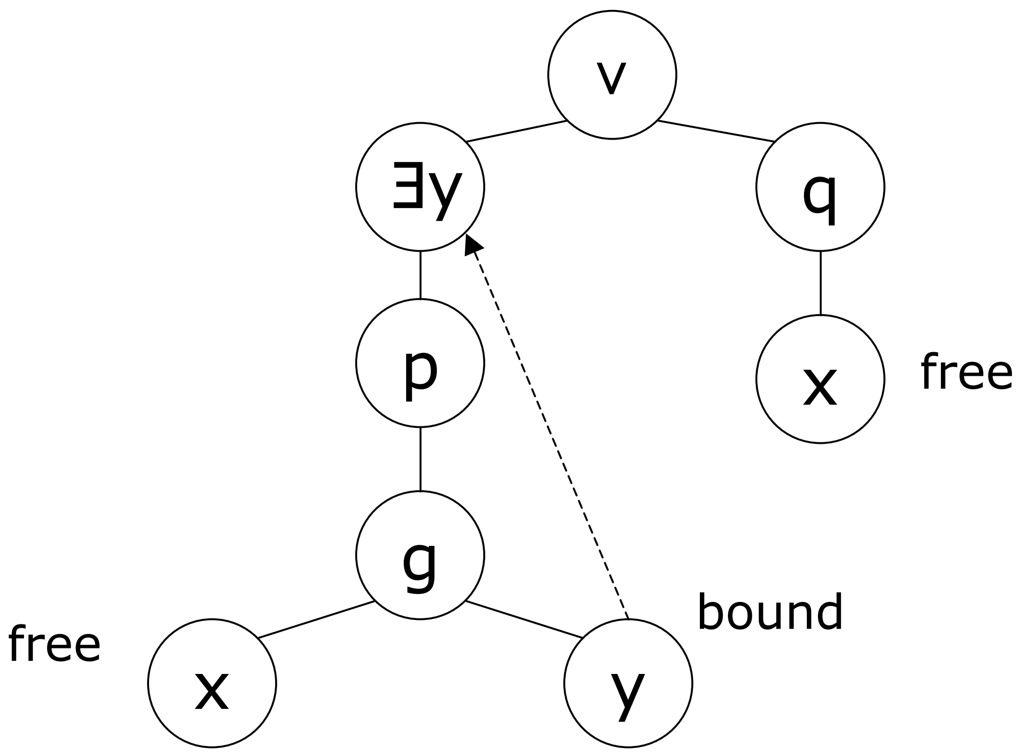
43252003274489856000 positions

Syntax Trees (or "Parse" Trees)

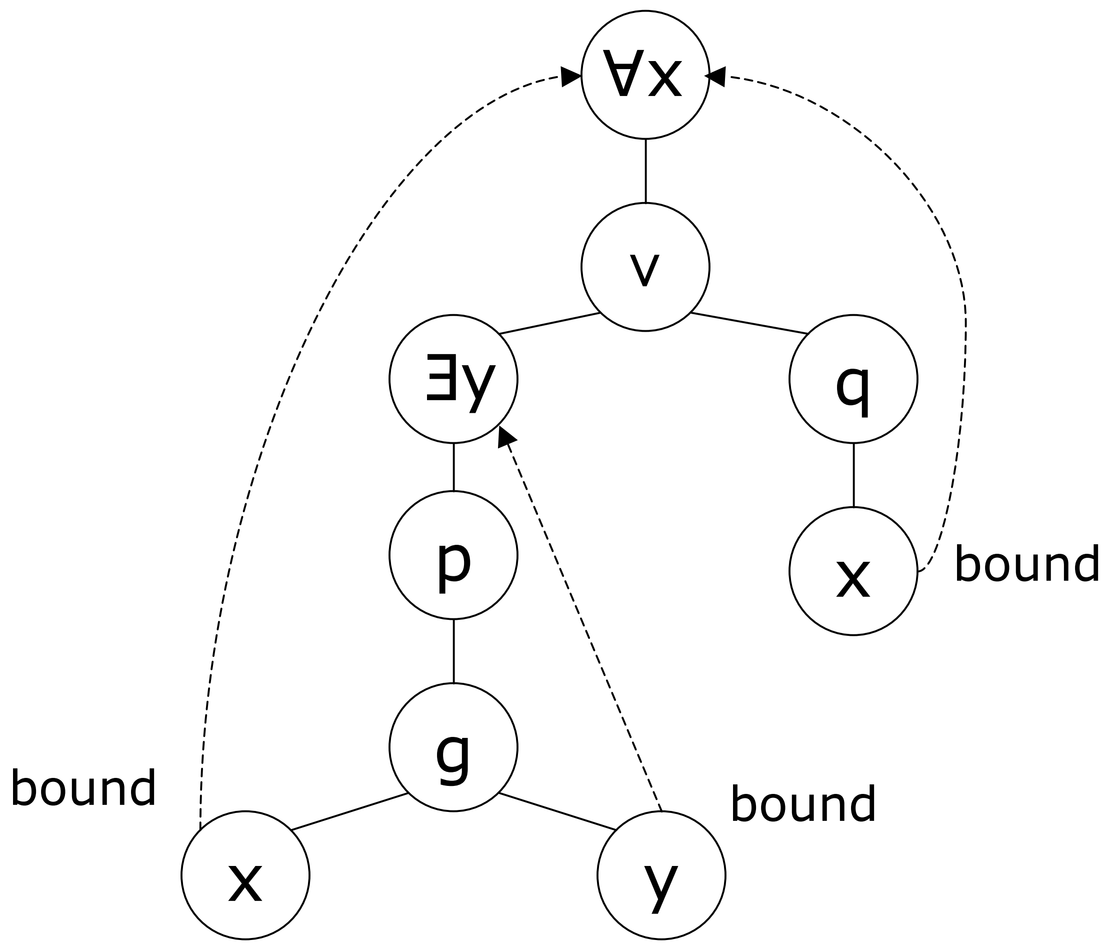
- We are assuming familiarity with syntax trees from CS 60.
- ∀x, ∃x are treated as if **1-ary operators**.
- Example: ∀x ((∃y p(g(x, y))) ∨ q(x))



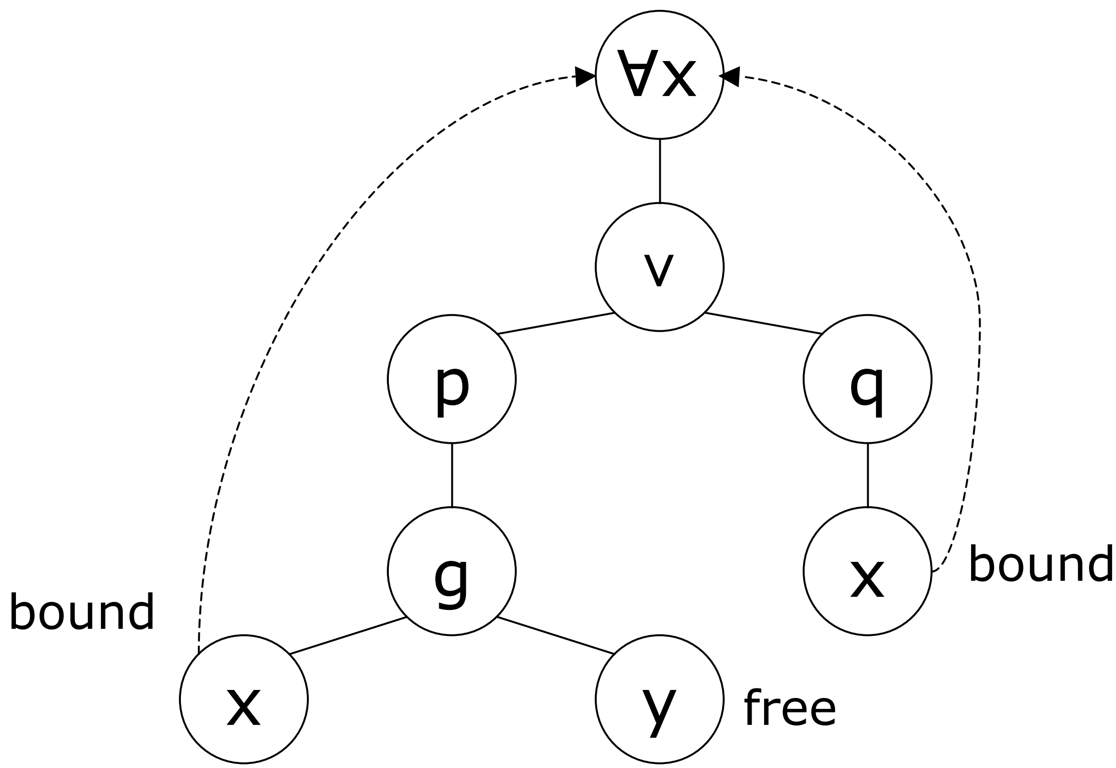
Free and Bound Variable Instances



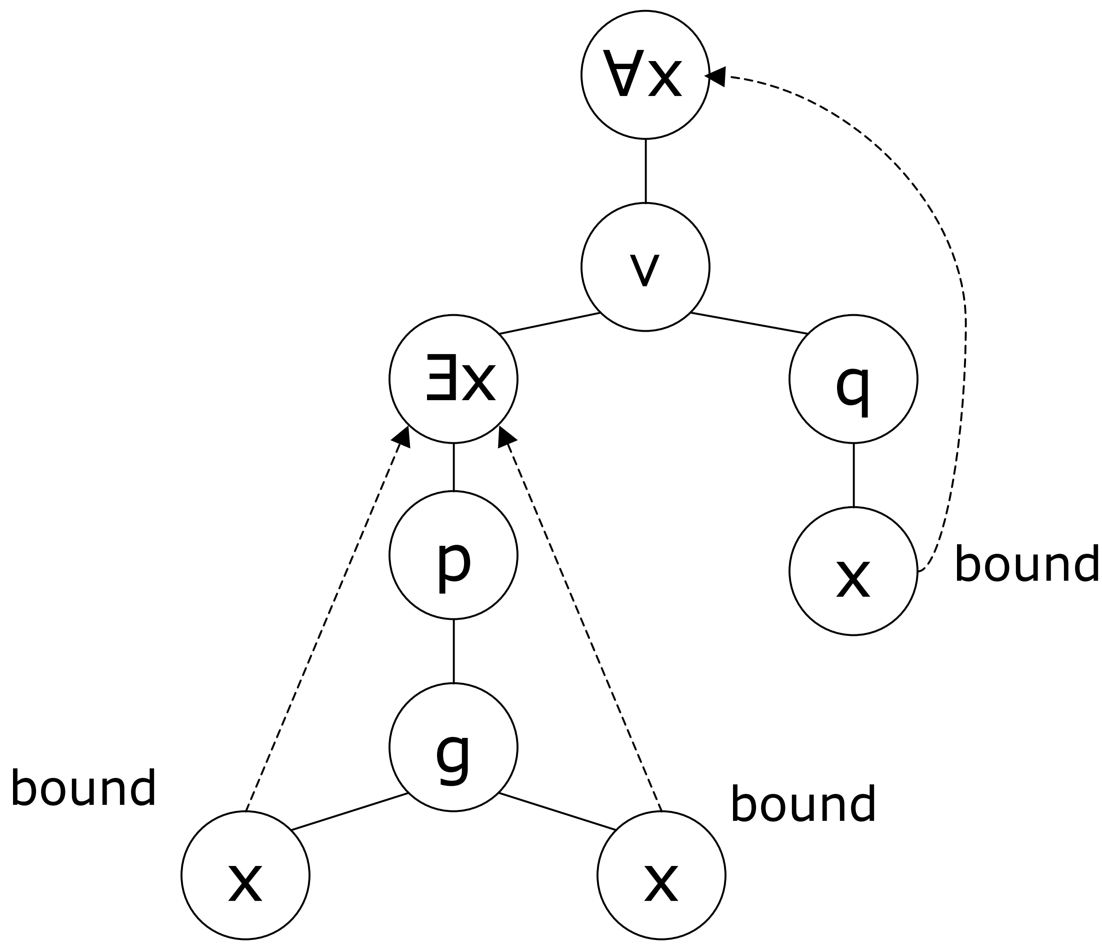
Free and Bound Variable Instances



Free and Bound Variable Instances



Free and Bound Variable Instances



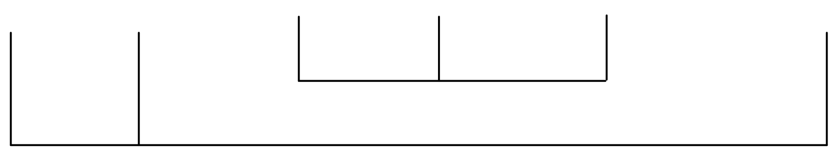
Scope of Variables

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

Scope Defined Inductively

- For a quantifier-free formula, the scope of each variable is the entire formula.
- For ∀x E, or ∃x E, the scope of x is inside E, but not 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

- 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, 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:

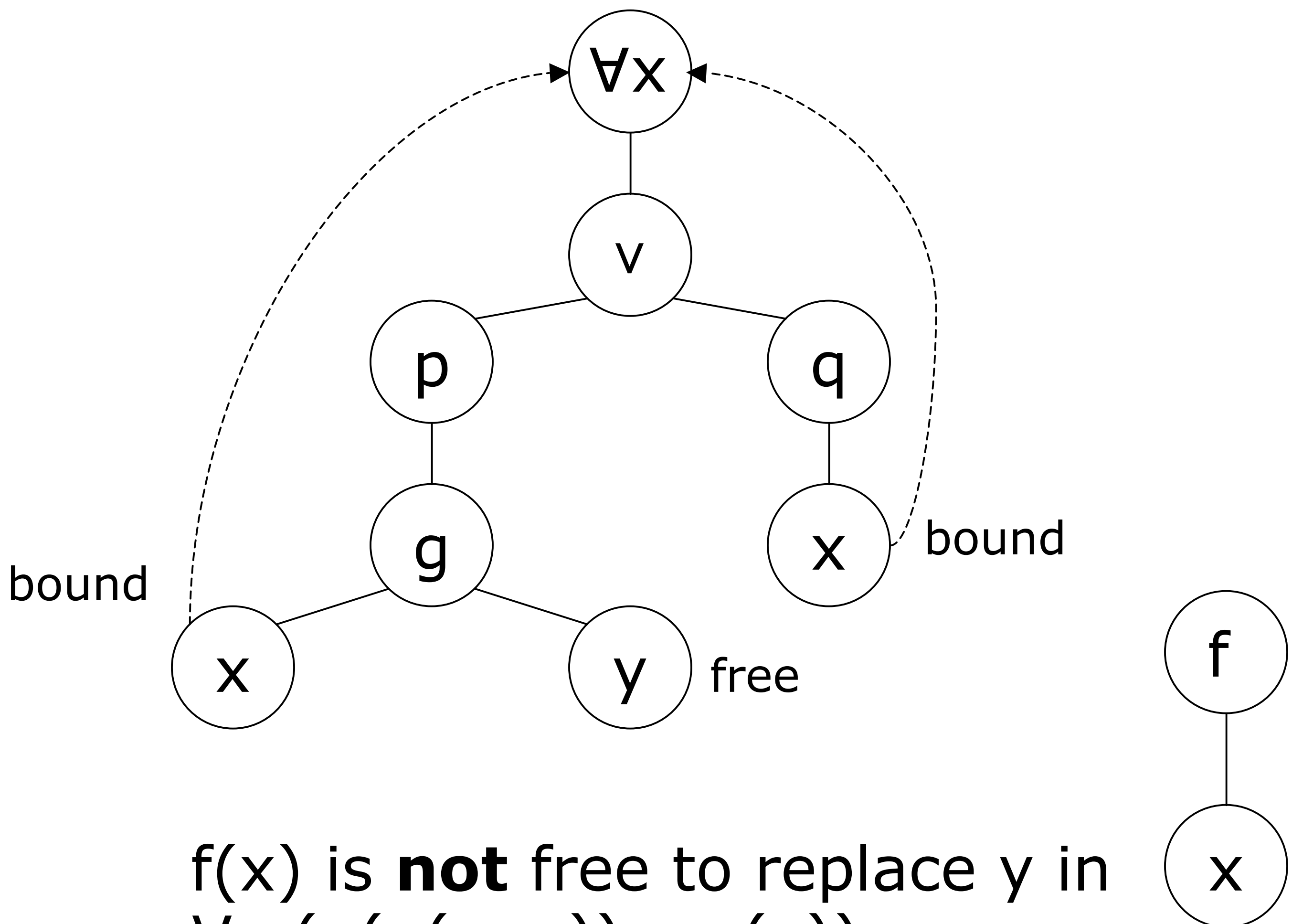
  - In substituting a term for a variable within a formula, **no variables *within* the term can become bound** as a result of the substitution.

- 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"

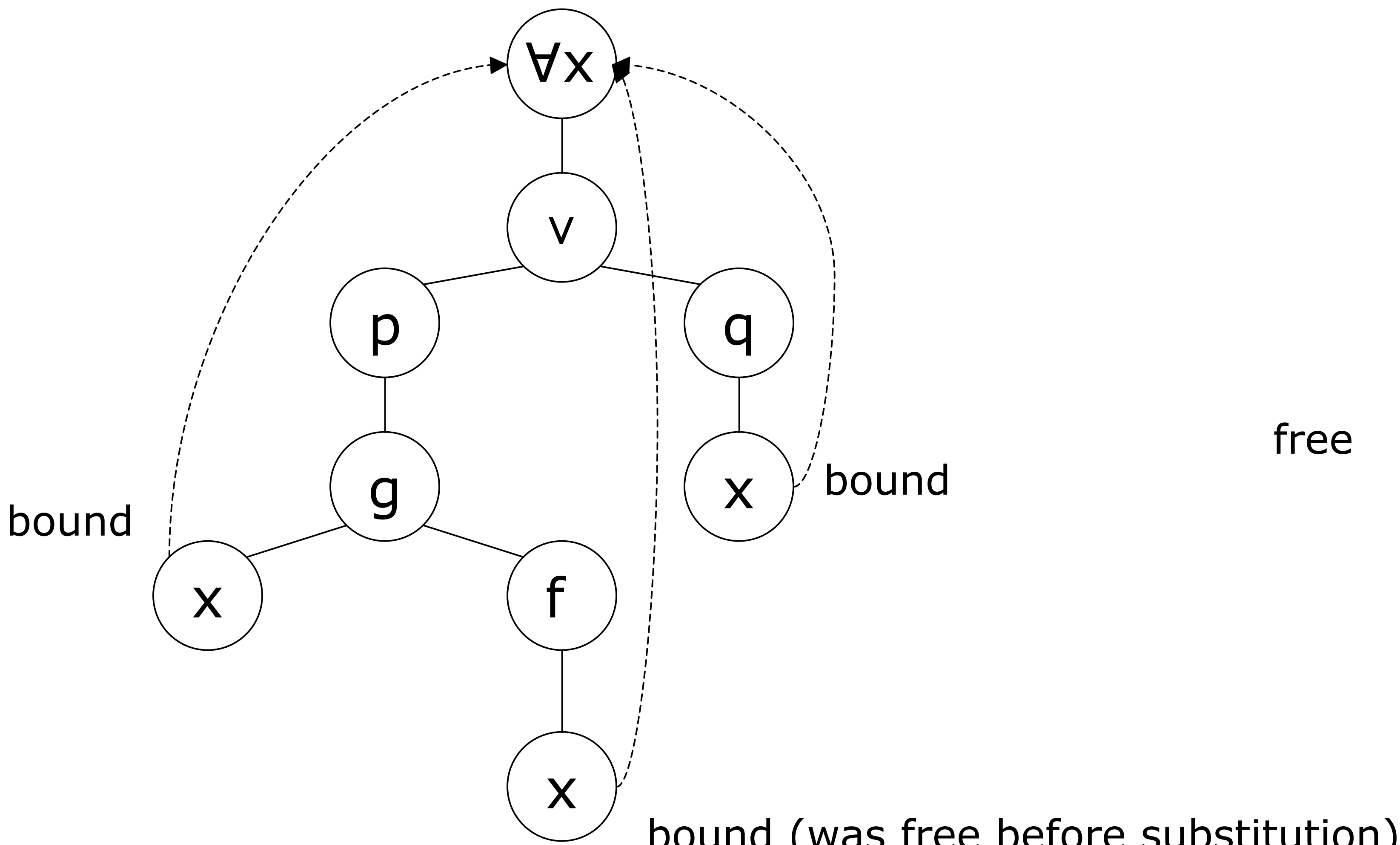
(or more conventionally, **"t is free for v in F")**

Non-Substitutability Example



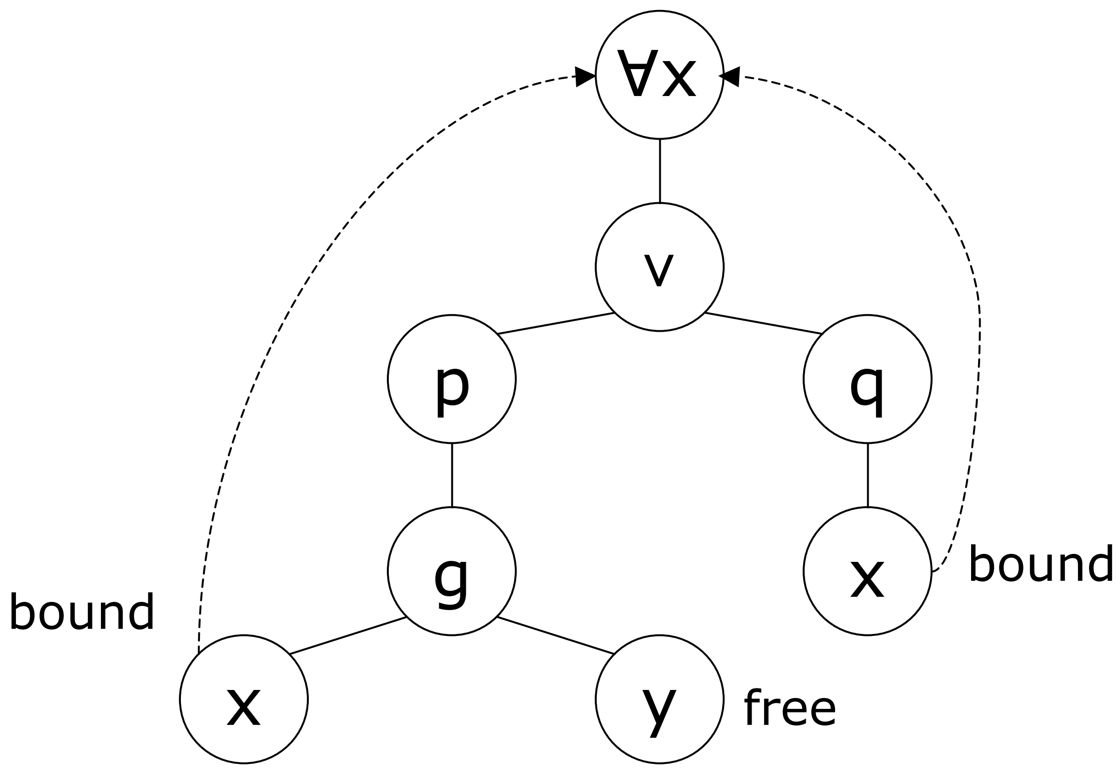
f(x) is **not** free to replace y in
∀x (p(g(x, y)) ∨ q(x))

Non-Substitutability Example

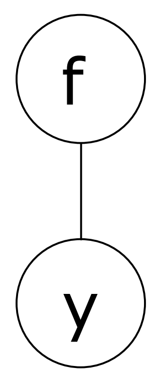


$\forall x\ (p(g(x, \mathbf{f(x)})) \lor q(x))$

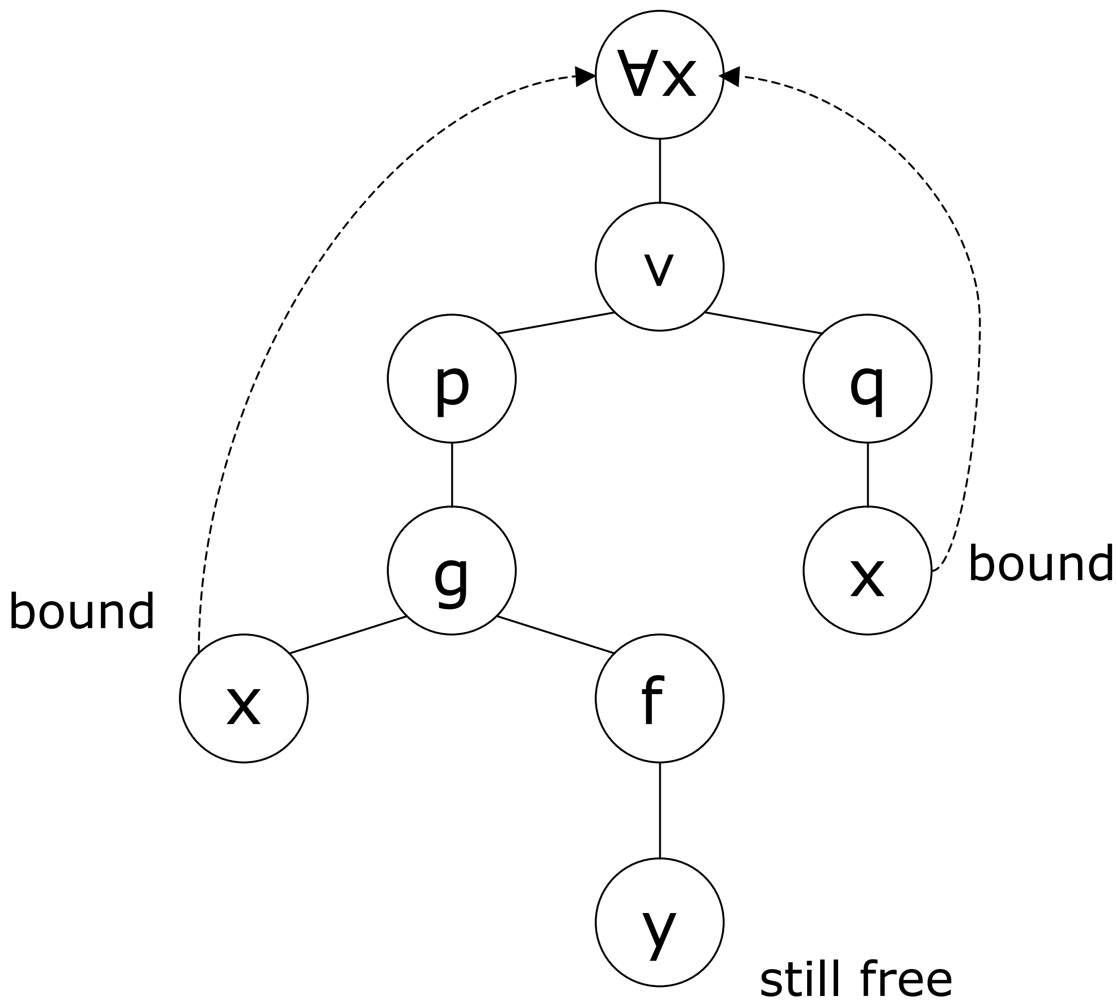
Substitutability Example



f(y) **is** free to replace y in
$\forall x\ (p(g(x, y)) \vee q(x))$



Substitutability Example



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

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.

  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

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

Let v be the variable y.

Let t be the term f(y).

f(y) **is** free to replace y in F.

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

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(y) **is** free to replace x in F;
there are no free instances of x.

F[f(y)/y] is the same as F;
there are no instances of y 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**.

Purpose of Separating Syntax and Semantics

- Reasoning about semantics is often very complex.
- Logical syntax allows reasoning without revisiting semantic details at every step.

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 (non-empty) **domain** Δ: that contains all individuals of interest.
  - For each **constant symbol** c, an element $\mu(c) \in \Delta$.
  - For each **function symbol** f, a function $\mu(f)$: $\Delta^n \rightarrow \Delta$.
  - For each **predicate symbol** p, a function $\mu(p)$: $\Delta^n \rightarrow \{T, F\}$.
  - For each **variable symbol** x, an element $\mu(x) \in \Delta$.
- Δ may also be called the "universe".

The value of **terms** under an interpretation

- An interpretation $I = (\Delta, \mu)$ determines a value $I[t] \in \Delta$ of each term recursively:
  - If t is a constant symbol c, then $I[t] = \mu(c)$.
  - If t is a variable symbol v, then $I[t] = \mu(v)$.
  - If t is $f(t_1, t_2, \ldots, t_n)$ where the t_i are terms, then

    $I[t] = \mu(f)(I[t_1], I[t_2], \ldots I[t_n])$.
- Notation: Stuff inside [...] is *syntactic*, not an expression in the meta-language. So it is not pre-evaluated as if a function.

Notation: I[d/x]

- If $I = (\Delta, \mu)$ is an interpretation, and x is a variable symbol, then by $I_{[d/x]}$, we mean the interpretation (Δ, μ') that is

  identical to I, *except* that

  $$\mu'(x) = d.$$

The value of **formulas** under an interpretation

- An interpretation $I = (\Delta, \mu)$ determines a value $I[t] \in \{T, F\}$ of each formula E recursively:
  - If E is an atomic formula $p(t_1, t_2, \ldots, t_n)$, where p is a predicate symbol, and the t_i are terms, then

    $I[E] = \mu(p)(I[t_1], I[t_2], \ldots I[t_n])$ [using the interpretation of terms]

  - If E is $E_1 \circ E_2$, then $I[E] = I[E_1] \circ I[E_2]$ where o is one of $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
  - If E is $\neg E_1$, then $I[E] = \neg I[E_1]$
  - If E is $\forall x\, E_1$, then I[E] is T if $I_{[d/x]}[E_1] = T$ for *every* d in Δ.
  - If E is $\exists x\, E_1$, then I[E] is T if $I_{[d/x]}[E_1] = T$ for *some* d in Δ.

Example

- Formula E is: q(f(f(x)), c)
- An interpretation I is
  - $\Delta = \{0, 1, 2\}$
  - $\mu(c) = 0$
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$
    the set of pairs for which $\mu(q)$ is T
  - $\mu(x) = 2$
- Thus:
  - $I[f(f(x)] = \mu(f)(I[f(x)]) = \mu(f)(\mu(f)(I[x])) = \mu(f)(\mu(f)(\mu(x))) = 1$
  - $I(q(f(f(x)), c)) = \mu(q)(I[f(f(x)], \mu(c)) = \mu(q)(1, 0) = T$

Example

- Formula E is: q(f(f(x)), c)
- An interpretation I is
  - Δ = {0, 1, 2}
  - μ(c) = 0
  - μ(f) = {0 $\rightarrow$ 1, 1 $\rightarrow$ 2, 2 $\rightarrow$ 0}
  - μ(q) = {(0, 2), (1, 0), (2, 1)}
    the set of pairs for which μ(q) is T
  - μ(x) = 0 <<< changed
- Thus:
  - I[f(f(x)] = μ(f)(I[f(x)]) = μ(f)(μ(f)(I[x])) = μ(f)(μ(f)(μ(x))) = 2
  - I(q(f(f(x)), c)) = μ(q)(I[f(f(x)], μ(c)) = μ(q)(2, 0) = F <<< note

Example

- Formula E is: $\exists x\ q(f(f(x)), c)$
- An interpretation I is
  - $\Delta = \{0, 1, 2\}$
  - $\mu(c) = 0$
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$
    the set of pairs for which $\mu(q)$ is T
- Thus:
  - $I[\exists x\ q(f(f(x)), c)] = T$ iff for *some* d in Δ, $I_{[d/x]}[q(f(f(x)), c)]$
  - We saw two slides ago that there is such a d, namely d = 2.
  - So $I[\exists x\ q(f(f(x)), c)] = T$.

Example

- Formula E is: $\forall x\ q(f(f(x)), c)$
- An interpretation I is
  - $\Delta = \{0, 1, 2\}$
  - $\mu(c) = 0$
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$
    the set of pairs for which $\mu(q)$ is T
- Thus:
  - $I[\forall x\ q(f(f(x)), c)]$ = T iff for *all* d in Δ, $I_{[d/x]}[q(f(f(x)), c)]$ = T.
  - We saw two slides ago that there is a d, namely 0, such that $I_{[d/x]}[q(f(f(x)), c)]$ = F.
  - So $I[\forall x\ q(f(f(x)), c)]$ = F.

New Example

- Formula is: $\forall x\ (q(x, c) \rightarrow q(f(x), c))$
- An interpretation I is
  - $\Delta = \{0, 1, 2\}$
  - $\mu(c) = 0$
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$
    the set of pairs for which $\mu(q)$ is T
- Thus:

  $I[\forall x\ (q(x, c) \rightarrow q(f(x), c))] = T$ iff for *all* d in Δ,
  $I_{[d/x]}[q(x, c) \rightarrow q(f(x), c)] = T$.

  - We can check for each d whether or not this is true:
    - d = 0: $I_{[0/x]}[q(x, c) \rightarrow q(f(x), c)] = T$
    - d = 1: $I_{[1/x]}[q(x, c) \rightarrow q(f(x), c)] = F$
    - d = 2: $I_{[2/x]}[q(x, c) \rightarrow q(f(x), c)] = T$
- So $I[\forall x\ (q(x, c) \rightarrow q(f(x), c))] = F$
  (but $I[\exists x\ (q(x, c) \rightarrow q(f(x), c))] = T$)

Satisfaction

- An interpretation I ***satisfies*** a formula E iff I[E] = T. We also say that E ***is valid under*** I. We also say that I is a ***model*** for E.

- 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, 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 relevant interpretation is a model.

$\models$ in predicate calculus vs. propositional

- The predicate version of $\models \psi$ is a **very broad** statement:
  - The set of applicable structures is generally infinite.
  - If a given domain is infinite, so is the set of assignments.
- Intuitively there is much less likely to be an algorithm to check whether $\models \psi$ for predicate calculus in the way there is for the propositional calculus.

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.**

Equality Axioms

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

ND Equality Rules

- Natural Deduction typically introduces rules for equality (from which the axioms can be derived).
  - $$\frac{}{t = t}$$ where t is any term =I
  - $$\frac{s = t \qquad \varphi[s/x]}{\varphi[t/x]}$$ where s and t are any terms =E
    (s and t must be free to replace x in φ)

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 $\Gamma \vdash \psi$

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

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

  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

- $$\frac{\forall x\ \varphi}{\varphi[t/x]} \qquad (\forall E)$$

  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.

$$\frac{\boxed{\begin{array}{l} x_0 \\ \quad\vdots \\ \varphi[x_0/x] \end{array}}}{\forall x\ \varphi} \quad (\forall I)$$

- 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; 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 is an **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.

$\forall$E $\forall$I Example

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

1.		$\forall x\ p(x)$	Premise
2.	x_0		Fresh var
3.		$p(x_0)$	$\forall$E 1
4.	$\forall y\ p(y)$		$\forall$I 2-3

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

∀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.

  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 $\forall x\, q(x)$.

∀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.

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

∃-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** then we can conclude that there is **an** x for which the formula is true.

∃-Introduction Rule (∃I)

- $\underline{\varphi[t/x]}$ ∃E
  $\exists x\ \varphi$

  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

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

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

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

Note: Here is one place we rely on the semantic domain being **non-empty**.

∃-Elimination Rule (∃E)

$$
\begin{array}{lll}
 & \boxed{\begin{array}{ll} x_0 & \varphi[x_0/x] \\ & \vdots \\ & \chi \end{array}} & \\
\exists x \varphi & & (\exists E) \\
\hline
 & \chi &
\end{array}
$$

- 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.

∃-Elimination Rule (∃x E)

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

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

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

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

- 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.

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

- 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 well help).

Quantifier rule summary

	Introduction	**Elimination**
$\forall$	$\dfrac{\boxed{x_0 \; \ldots \; \varphi[x_0/x]}}{\forall x \; \varphi}$ $\forall I$	$\dfrac{\forall x \;\; \varphi}{\varphi[t/x]}$ (t is free to replace x) $\forall E$
$\exists$	$\dfrac{\varphi[t/x]}{\exists x \; \varphi}$ (t is free to replace x) $\exists I$	$\dfrac{\exists x \; \varphi \quad \boxed{x_0 \;\; \varphi[x_0/x] \ldots \; \chi}}{\chi}$ $\exists E$

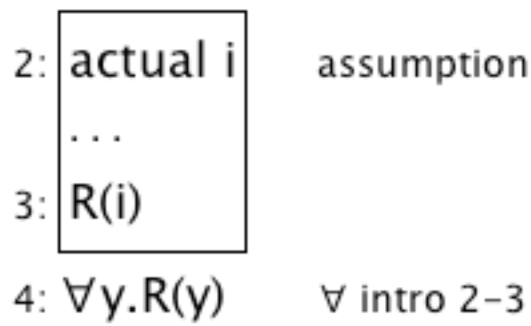
JAPE Examples

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

1:	$\forall x.R(x)$	premise
2:	actual i	assumption
3:	$R(i)$	∀ elim 1,2

JAPE Examples

- **∀ Introduction** (working backward) introduces a fresh variable.
  The variable can't be taken outside the box.



JAPE Examples

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

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

- Note: JAPE will *unify* the above premise and conclusion, so a *shorter* proof, using the 'hyp' rule is:

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

. . .

2: $\forall y.R(y)$

→

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

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.

1:	$\exists y.R(y),\ \forall x.(R(x) \rightarrow S(x))$	premises
2:	actual i, R(i)	assumptions
	. . .	
3:	$\exists y.S(y)$	
4:	$\exists y.S(y)$	∃ elim 1.1,2-3

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 identified by the user.

1:	∃y.R(y), ∀x.(R(x)→S(x))	premises
2:	actual i, R(i)	assumptions
3:	R(i)→S(i)	∀ elim 1.2,2.1
4:	S(i)	→ elim 3,2.2
	...	
5:	∃y.S(y)	
6:	∃y.S(y)	∃ elim 1.1,2-5

→

1:	∃y.R(y), ∀x.(R(x)→S(x))	premises
2:	actual i, R(i)	assumptions
3:	R(i)→S(i)	∀ elim 1.2,2.1
4:	S(i)	→ elim 3,2.2
5:	∃y.S(y)	∃ intro 4,2.1
6:	∃y.S(y)	∃ elim 1.1,2-5

Sometimes the steps have to be taken in an unexpected order, e.g. ∃E won't work forward

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

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

1:	¬∃x.¬R(x)	premise
2:	actual i	assumption
3:	¬R(i)	assumption
	. . .	
4:	⊥	
5:	R(i)	contra (classical) 3-4
6:	∀y.R(y)	∀ intro 2-5

Why not ∃I here?

1:	¬∃x.¬R(x)	premise
2:	actual i	assumption
3:	¬R(i)	assumption
	. . .	
4:	_B1	
	. . .	
5:	¬_B1	
6:	⊥	¬ elim 4,5
7:	R(i)	contra (classical) 3-6
8:	∀y.R(y)	∀ intro 2-7

unify step

1:	¬∃x.¬R(x)	premise
2:	actual i	assumption
3:	¬R(i)	assumption
	. . .	
4:	∃x.¬R(x)	
5:	⊥	¬ elim 4,1
6:	R(i)	contra (classical) 3-5
7:	∀y.R(y)	∀ intro 2-6

1:	¬∃x.¬R(x)	premise
2:	actual i	assumption
3:	¬R(i)	assumption
4:	∃x.¬R(x)	∃ intro 3,2
5:	⊥	¬ elim 4,1
6:	R(i)	contra (classical) 3-5
7:	∀y.R(y)	∀ intro 2-6

JAPE

- The non-empty universe is not assumed in JAPE!!
- If you need this, you must introduce a premise that there is at least one element, by including 'actual i' as a premise.
- Valid in textbooks, but not provable in JAPE:

1: ∀x.R(x) premise

...

2: ∃x.R(x)

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

∀ intro (introduces variable) ∀ elim (needs variable)

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

JAPE

- The non-empty universe is not assumed in JAPE.
- If you need this, you must introduce a premise that there is at least one element, by including ‘actual i’ as a premise.

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

- 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: $\exists x.(R(x) \to R(j) \land R(k))$

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: $\_E \lor \neg \_E$ Theorem $E \lor \neg E$

. . .

3: $\exists x.(R(x) \to R(j) \land R(k))$

1: actual j, actual k   premises

2: $R(j) \lor \neg R(j)$ Theorem $E \lor \neg E$

. . .

3: $\exists x.(R(x) \to R(j) \land R(k))$

1: actual j, actual k   premises

2: $R(j) \lor \neg R(j)$ Theorem $E \lor \neg E$

3:	$R(j)$	assumption
	. . .	
4:	$\exists x.(R(x) \to R(j) \land R(k))$	
5:	$\neg R(j)$	assumption
	. . .	
6:	$\exists x.(R(x) \to R(j) \land R(k))$	

7: $\exists x.(R(x) \to R(j) \land R(k))$ $\lor$ elim 2,3-4,5-6

What x would make this work?

What x would make this work?

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	premises
2:	$R(j) \vee \neg R(j)$	Theorem $E \vee \neg E$
3:	$R(j)$	assumption
	. . .	
4:	$\exists x.(R(x) \to R(j) \wedge R(k))$	
5:	$\neg R(j)$	assumption
	. . .	
6:	$\exists x.(R(x) \to R(j) \wedge R(k))$	
7:	$\exists x.(R(x) \to R(j) \wedge R(k))$	$\vee$ elim 2,3-4,5-6

. . .

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:	$R(k) \to R(j) \wedge R(k)$	$\to$ intro 4-5
7:	$\exists x.(R(x) \to R(j) \wedge R(k))$	$\exists$ intro 6,1.2
8:	$\neg R(j)$	assumption
9:	$R(j) \to R(j) \wedge R(k)$	Theorem $\neg E \vdash E \to F$ 8
0:	$\exists x.(R(x) \to R(j) \wedge R(k))$	$\exists$ intro 9,1.1
1:	$\exists x.(R(x) \to R(j) \wedge R(k))$	$\vee$ elim 2,3-7,8-10

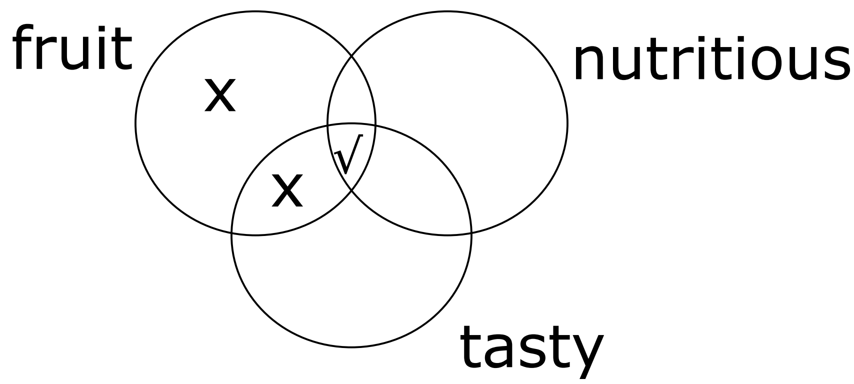
Syllogisms (WP)

- 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.
- Some tasty things are 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.	$\exists x\ (S(x) \wedge P(x))$
No S is P.	$\neg\exists x\ (S(x) \wedge P(x))$
Some S is not P.	$\exists x\ (S(x) \wedge \neg P(x))$
No S is not P.	$\neg\exists x\ (S(x) \wedge \neg P(x))$
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
  - $(p \wedge q) \leftrightarrow \neg(\neg p \vee \neg q)$ $\quad$ $(\neg p \vee \neg q) \leftrightarrow \neg(p \wedge q)$
  - $(p \vee q) \leftrightarrow \neg(\neg p \wedge \neg q)$ $\quad$ $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$
- For quantifiers, we have analogous rules
  - $\forall x\ P(x) \leftrightarrow \neg(\exists x\ \neg P(x))$ $\quad$ $\exists x\ \neg P(x) \leftrightarrow \neg \forall x\ P(x)$
  - $\exists x\ P(x) \leftrightarrow \neg(\forall x\ \neg P(x))$ $\quad$ $\neg \exists x\ P(x) \leftrightarrow \forall x\ \neg P(x)$
- 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

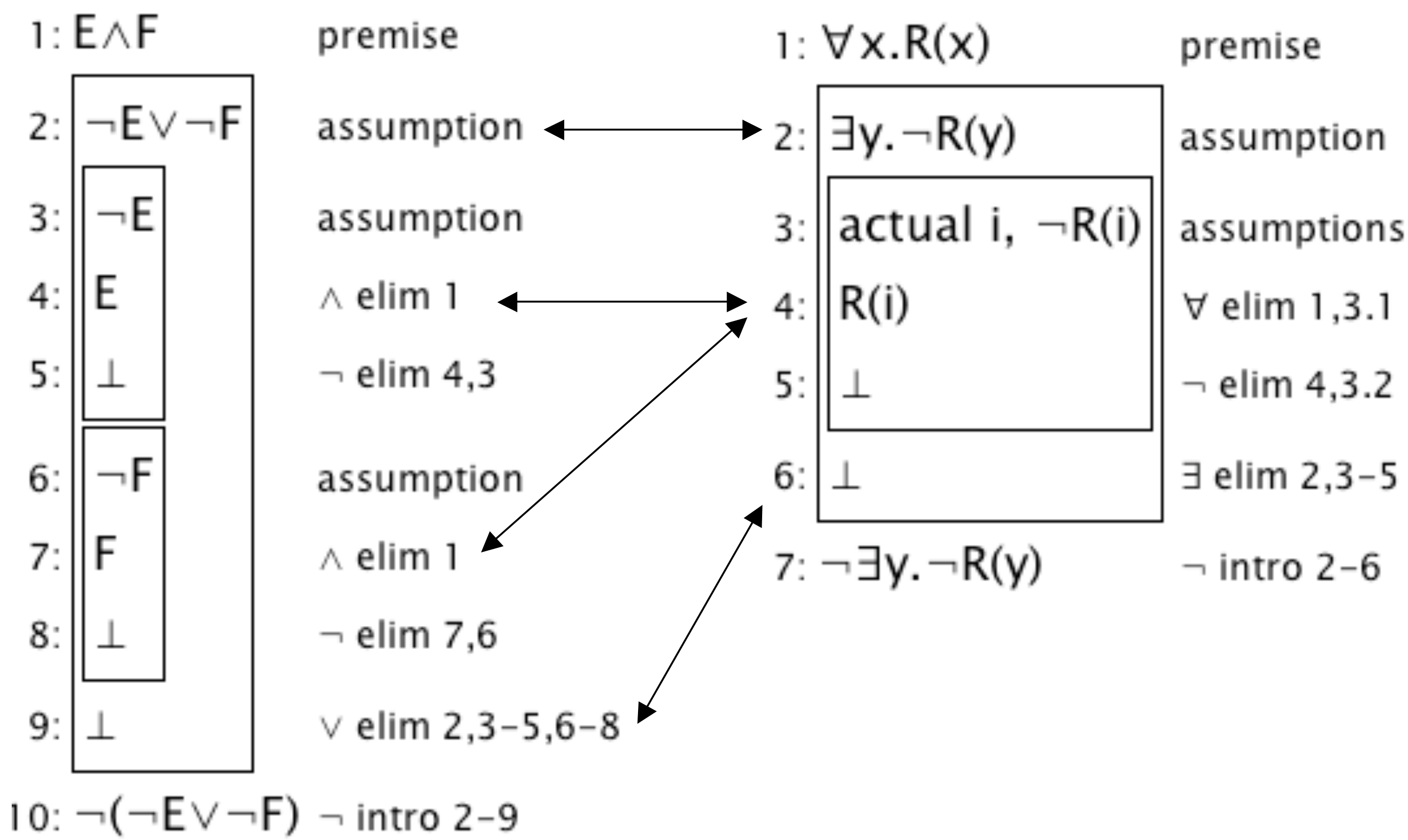
1:	$\neg(\neg E \vee \neg F)$	premise
2:	$\neg E$	assumption
3:	$\neg E \vee \neg F$	$\vee$ intro 2
4:	$\bot$	$\neg$ elim 3,1
5:	E	contra (classical) 2–4
6:	$\neg F$	assumption
7:	$\neg E \vee \neg F$	$\vee$ intro 6
8:	$\bot$	$\neg$ elim 7,1
9:	F	contra (classical) 6–8
10:	$E \wedge F$	$\wedge$ intro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

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

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



Note Rule Parallels

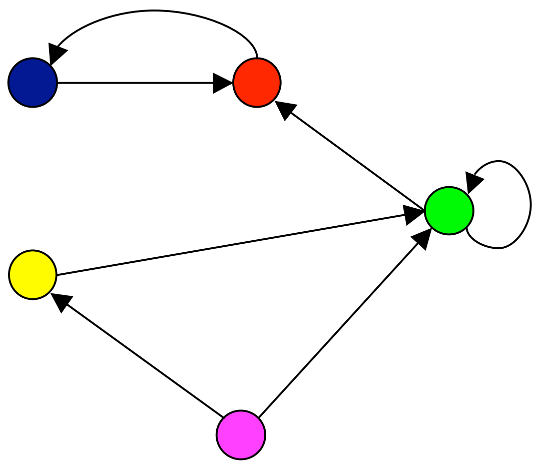
1:	$\neg(\neg E \vee \neg F)$	premise
2:	$\neg E$	assumption
3:	$\neg E \vee \neg F$	$\vee$ intro 2
4:	$\bot$	$\neg$ elim 3,1
5:	E	contra (classical) 2–4
6:	$\neg F$	assumption
7:	$\neg E \vee \neg F$	$\vee$ intro 6
8:	$\bot$	$\neg$ elim 7,1
9:	F	contra (classical) 6–8
10:	$E \wedge F$	$\wedge$ intro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

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

Fun with Relations

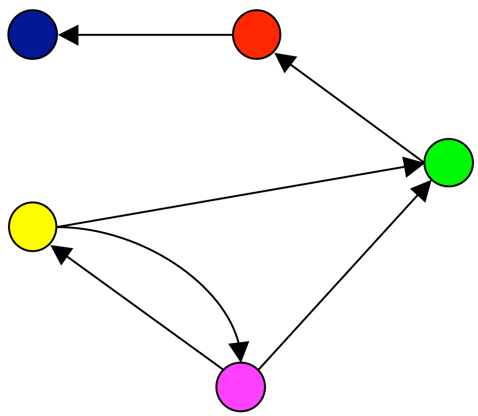
- A 2-ary predicate represents a binary relation, i.e. a set of pairs of domain elements.
- Various properties of relations can be expressed using predicate logic formulas.
- In the following, what formula characterizes each relation represented by predicate L (sometimes using “loves” for analogy), and possibly the predicate =.

Everybody loves somebody.

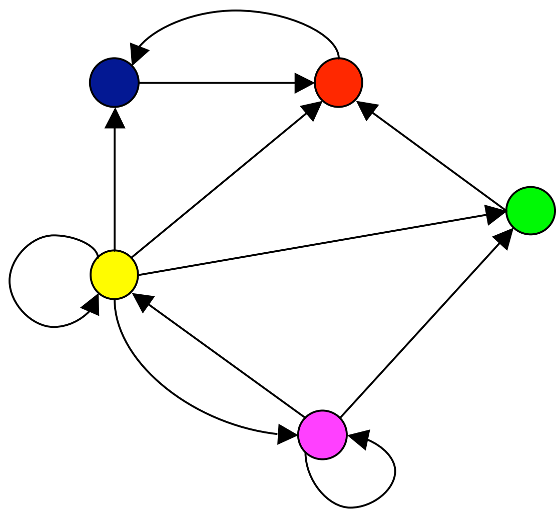


$\forall x\, \exists y\ L(x, y)$

Everybody is loved by somebody.

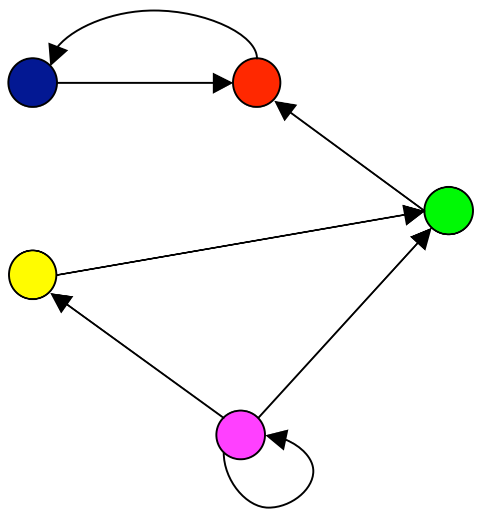


Somebody loves everybody.

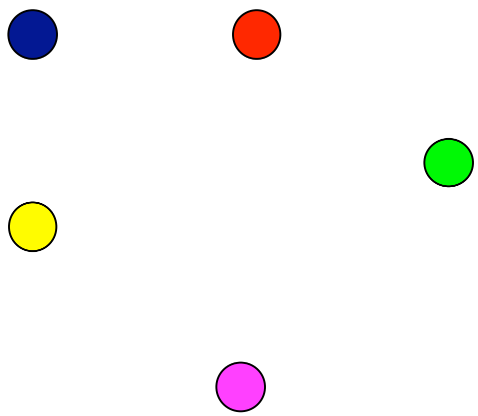


"Pollyanna"

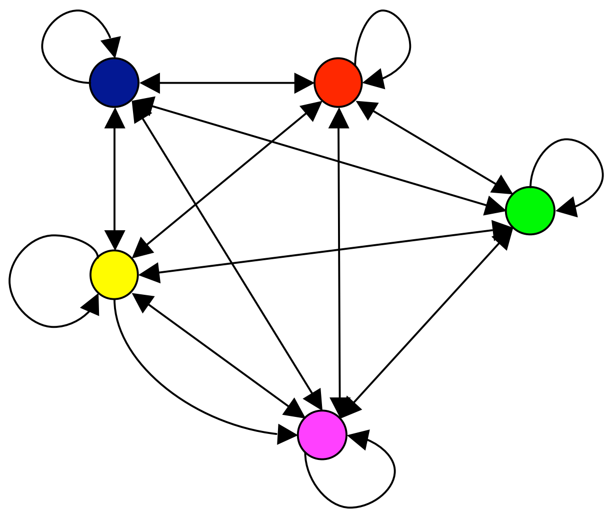
Nobody loves everybody.



Nobody loves somebody.

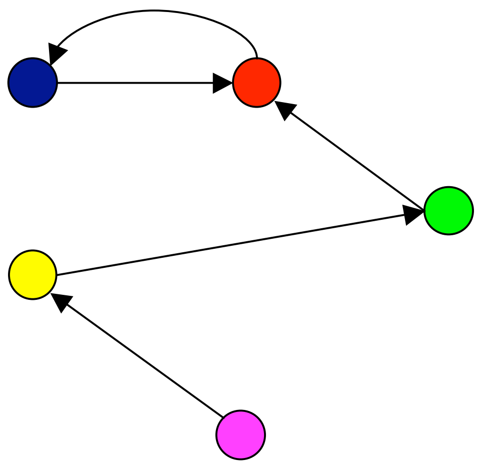


Everybody loves everybody.

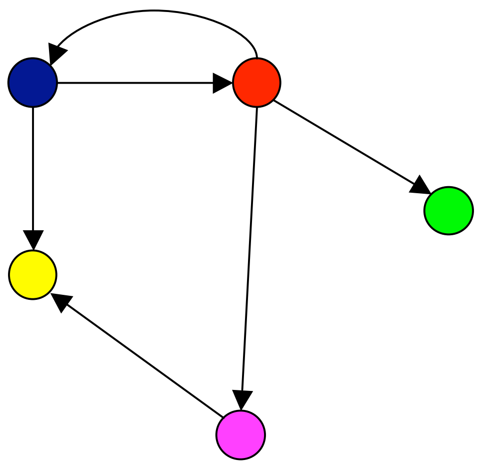


"Commune"

Everybody loves exactly one.

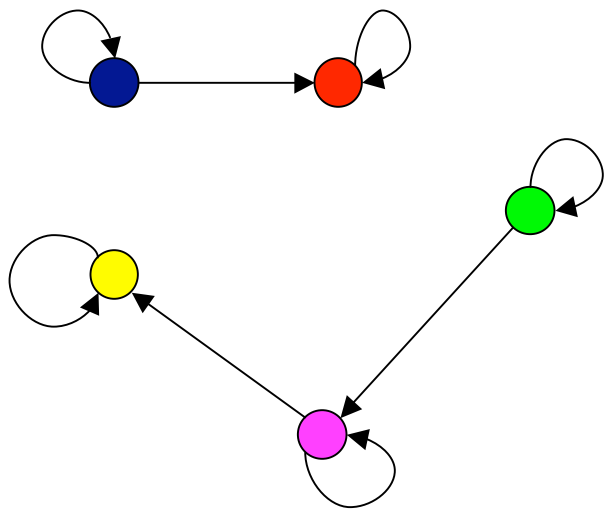


Nobody loves exactly one.

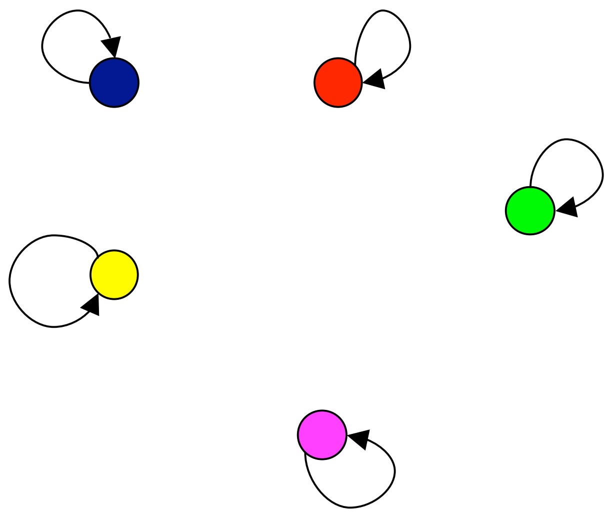


Everybody loves him/herself.

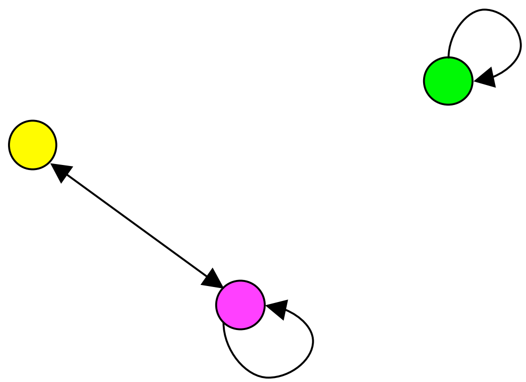
"Reflexive"



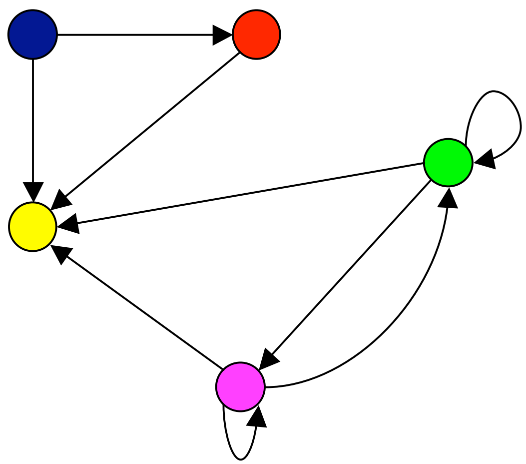
Everybody loves him/herself and only him/herself.



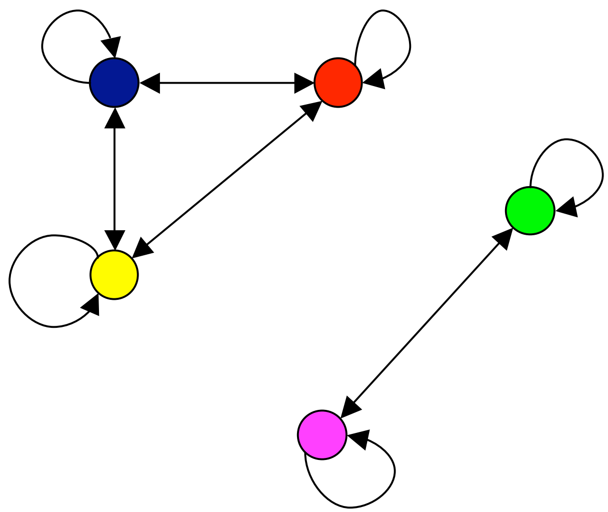
L is symmetric.



L is transitive.

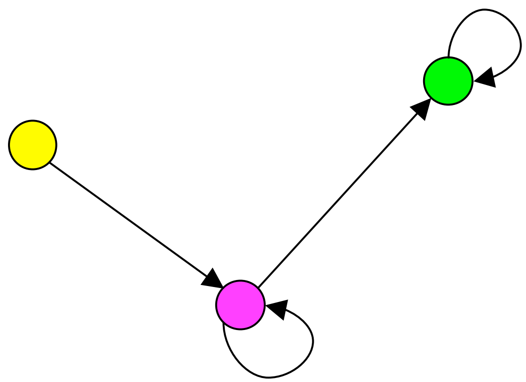


L is an equivalence relation

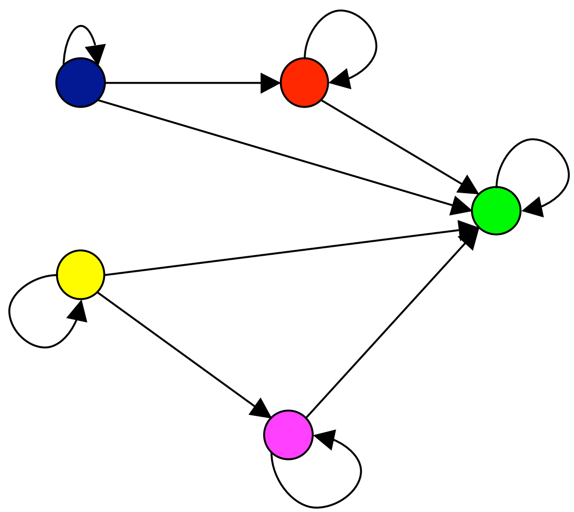


Reflexive, symmetric, transitive

L is antisymmetric.

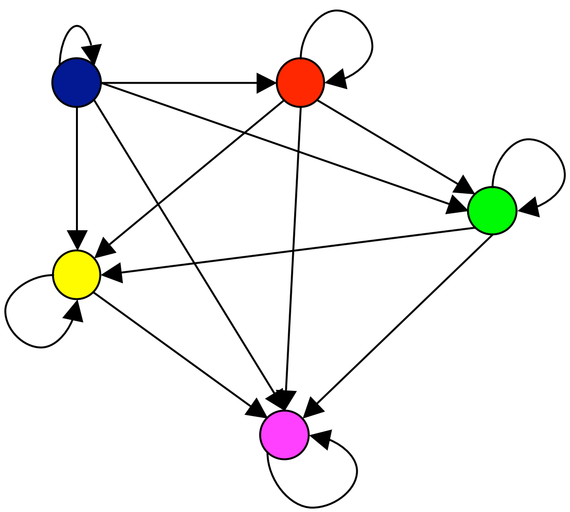


L is a partial order ("poset").



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

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:
  - $\forall x\ \forall y\ \exists z\ F(x, y, z)$
  - $\forall x\ \forall y\ \forall z\ \forall z'\ (F(x, y, z) \wedge F(x, y, z') \rightarrow z = z')$
- We'd still need axioms for equality.

Example: Group theory (e is unit)

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

- It is a pain to do without function symbols!