Natural Deduction for Predicate Calculus

Robert Keller
31 January 2005

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)
  ⊥ | // Bottom
  (∀V)E | // Universally-quantified formula
  (∃V)E // Existentially-quantified formula

- 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
g(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(g(b, y))
- r(a, g(h(b), c, h(y)))

Examples of “Literals”

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

- Note: There is no production for literals specifically in the grammar.

Examples of Quantifier-Free Formulas

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

Examples of Formulas

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

- Note: Some authors (e.g. Hein) don’t use parens around the quantified variables. It is ok not to, but I’ve gotten into the habit.

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.

Preview of Semantics

- Underlying the meaning of a formula will be a **structure** consisting of:
  - A non-empty domain of elements
  - Domain elements associated with each constant
  - Functions associated with each function symbol: These map tuples of domain elements to domain elements
  - Predicates (aka relations) associated with each predicate symbol: These map tuples of domain elements to truth values.

- Although we may have a ***particular*** structure in mind when we write formulas, the formulas are generally **not limited** to this structure and we have no way in general to require them to be so limited.

Example:

Structure N 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 e (equals).

Some formulas for the structure N

- $(\forall n)\ \neg e(s(n), 0)$

  [0 is not the successor of anything.]

- $(\forall m)\ (\neg e(m, 0) \rightarrow (\exists n)\ e(m, s(n)))$

  [Anything other than 0 is the successor of something.]

- $(\forall m)(\forall n)\ (e(s(m), s(n)) \rightarrow e(m, n))$

  [Successor is one-to-one.]

Example:
Structure G for "Groups"

- The domain is non-empty (as always).
- There is a constant symbol u (unit).
- There is a 2-ary function f (group multiplication).
- There is a 2-ary predicate e (equals).

Some formulas for structures G

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

  [u is an identity]

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

  [f is associative]

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

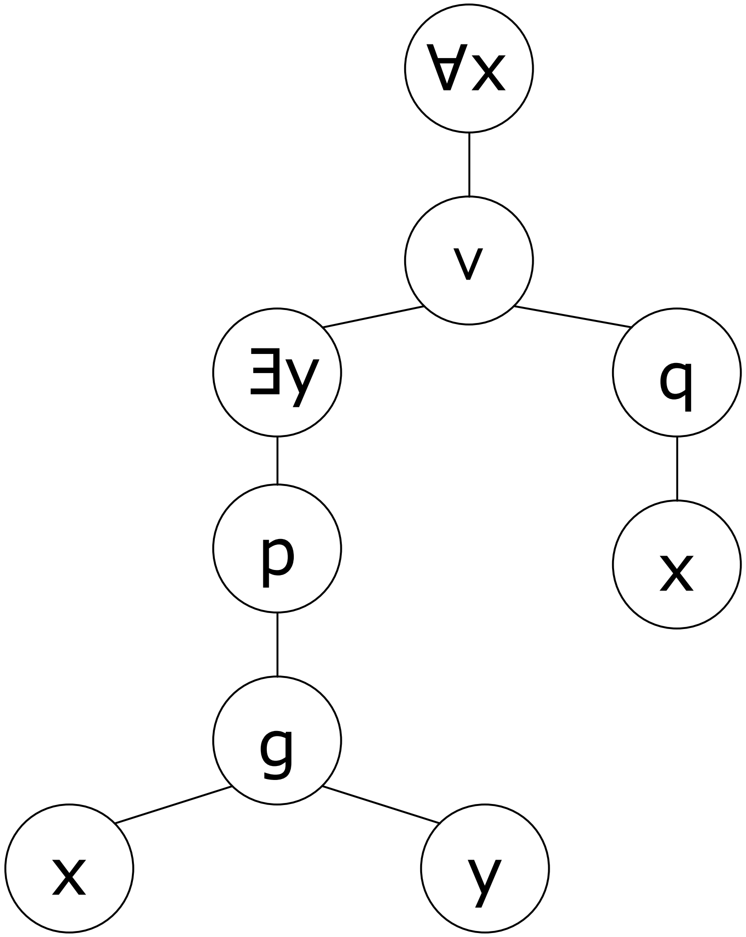
  [existence of inverse]

Examples of Groups: The formulas are true for all cases

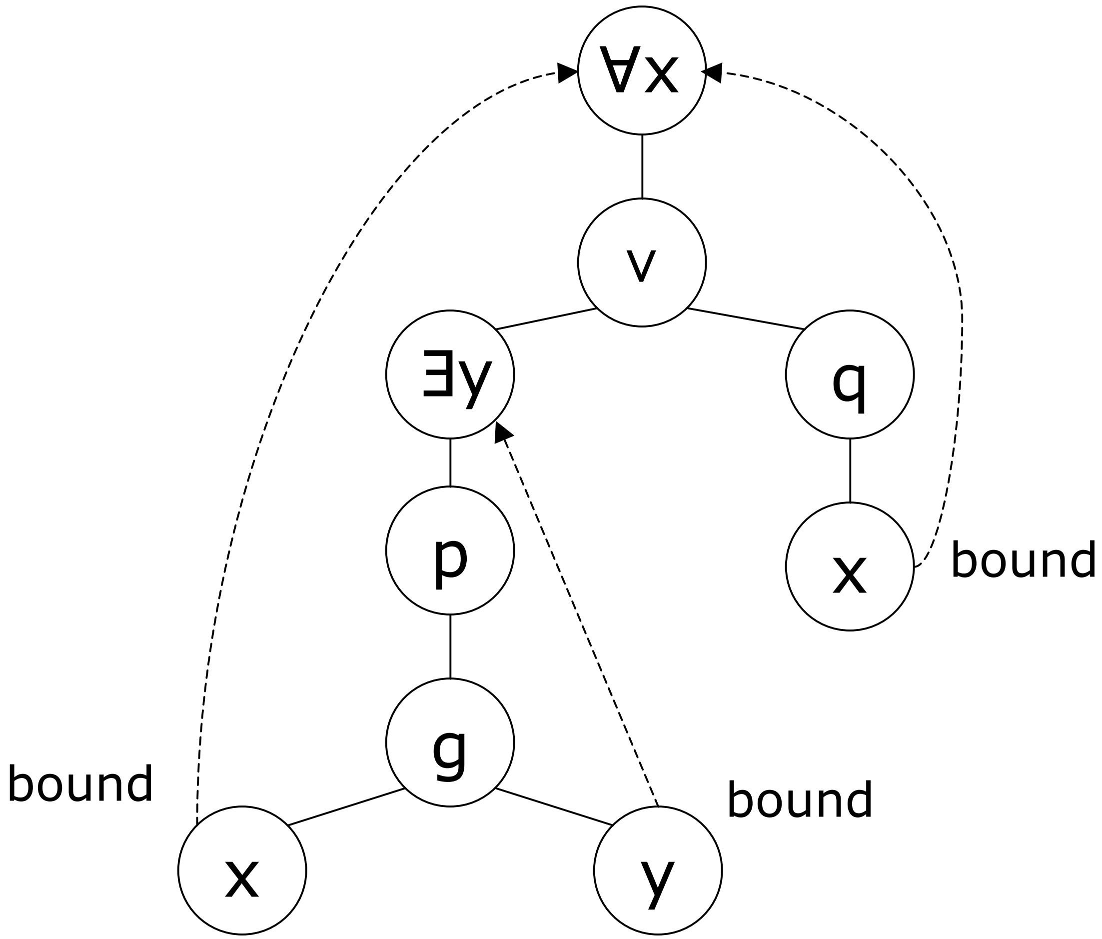
- {0} $u = 0$, $f(0, 0) = 0$
- {0, 1} $u = 0$, $f(x, y) = x + y \pmod 2$
- {0, 1, …, p-1} for any prime p,
  $u = 0$, $f(x, y) = x + y \pmod p$
- Tire rotations
- Rubick’s cube twists
- Particle spins (physics)
- Many others

Syntax Trees (or “Parse” Trees)

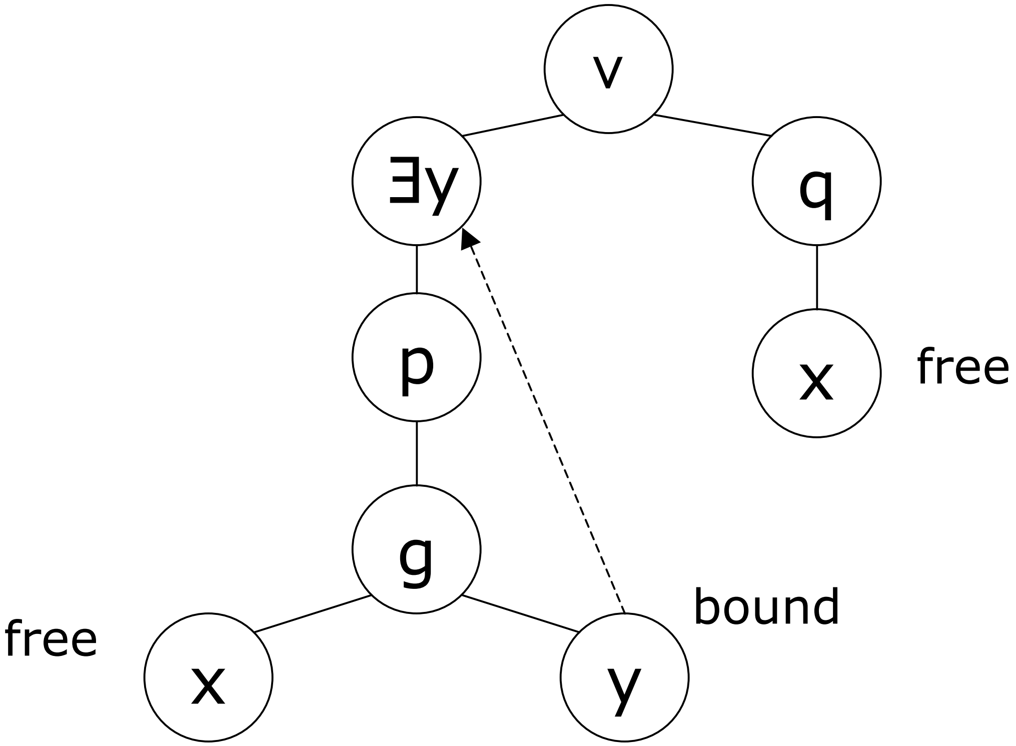
- We are assuming familiarity with syntax trees from CS 60.
- (∀x) (∃x) are treated as **1-ary operators**.
- Example: (∀x) ((∃y) p(g(x, y))) ∨ q(x))



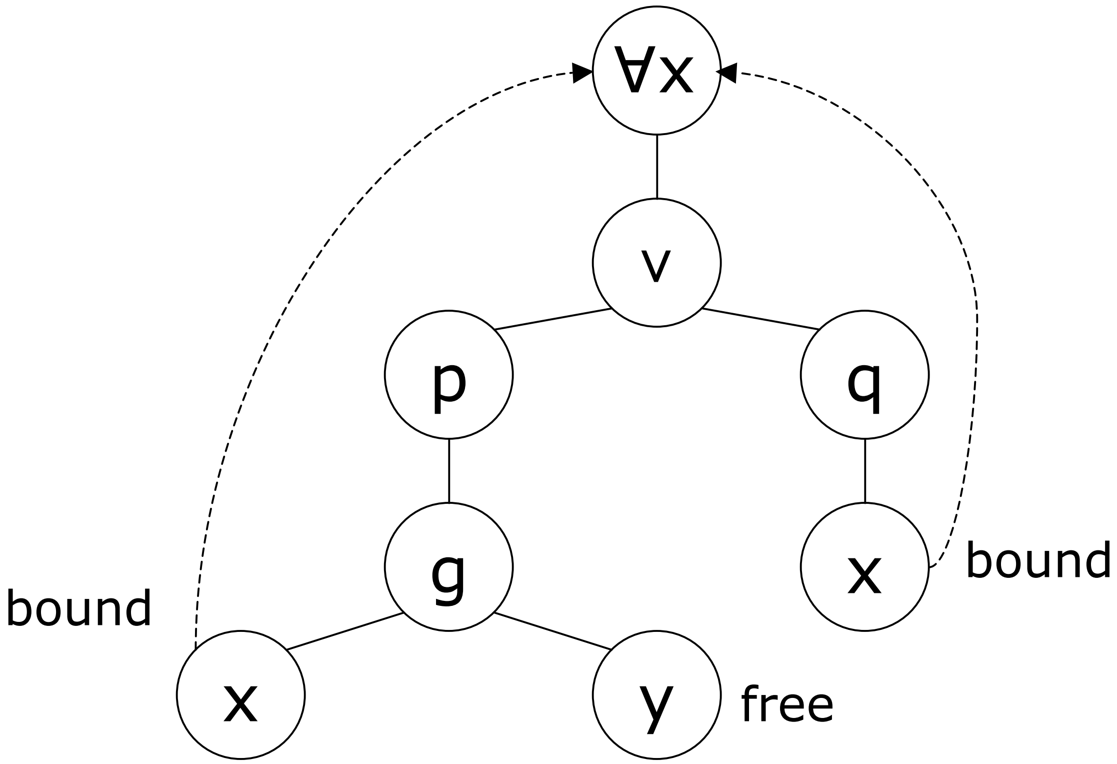
Free and Bound Variable Instances



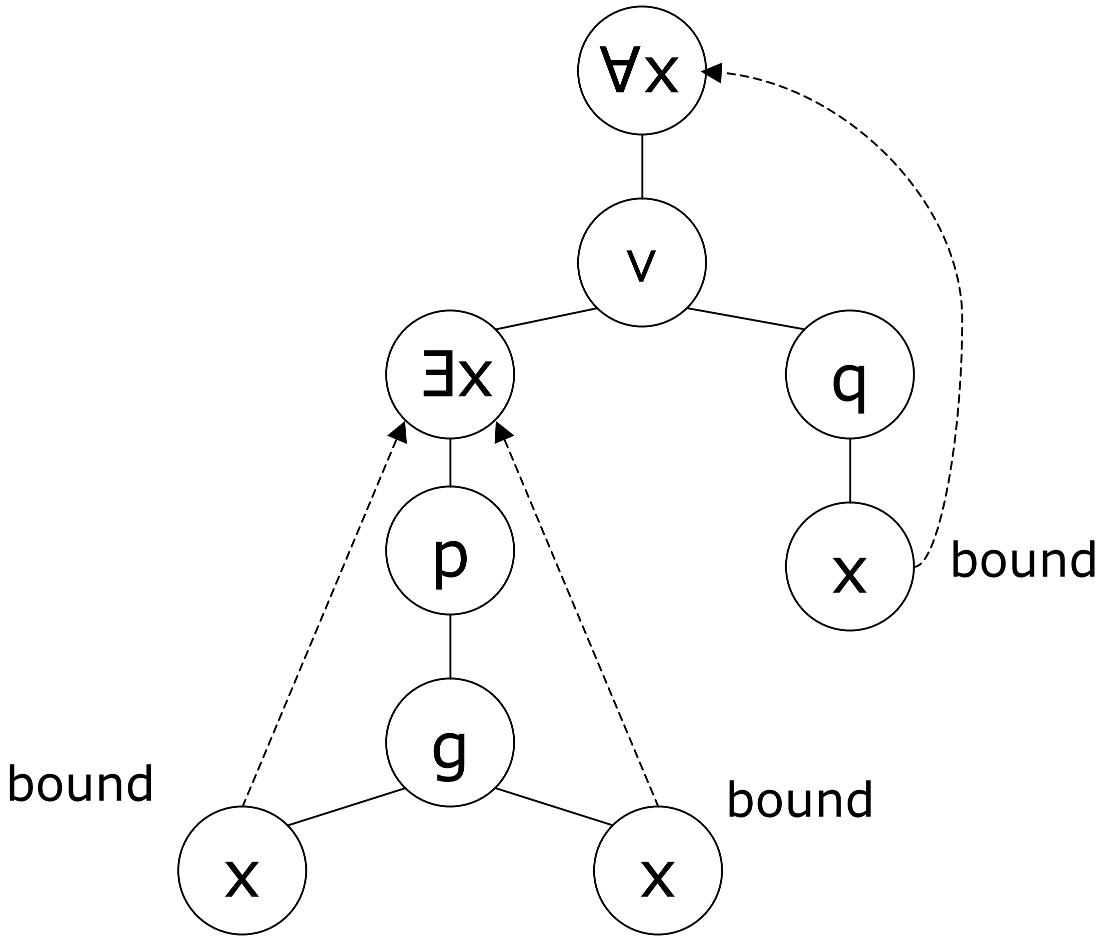
Free and Bound Variable Instances



Free and Bound Variable Instances



Free and Bound Variable Instances



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

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

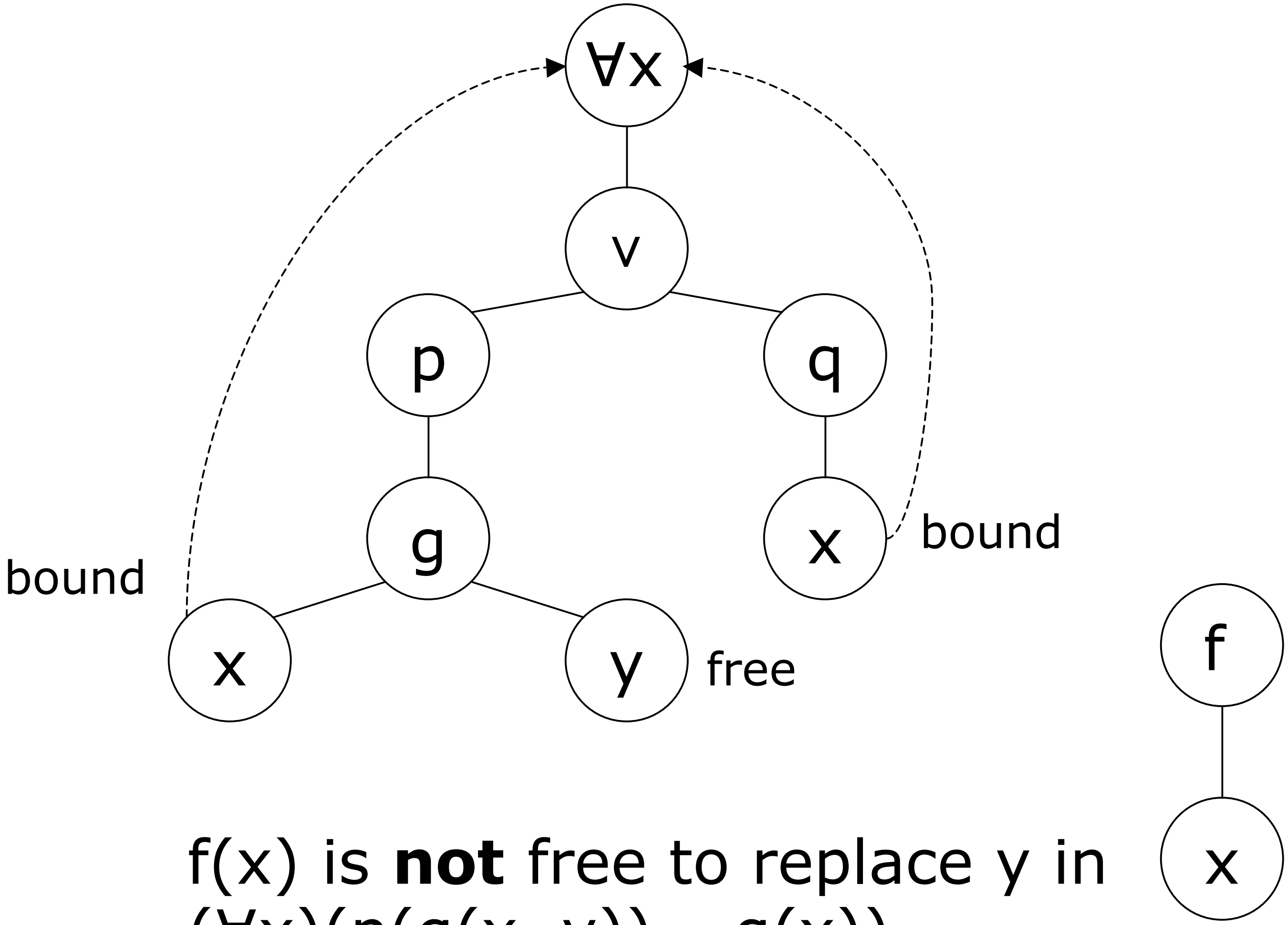
- While this is easy syntactically, there is a semantic restriction that must be observed:
  - In substituting a term for a variable within a formula, **no variables *within* the term can become bound** as a result of the substitution.

- 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”

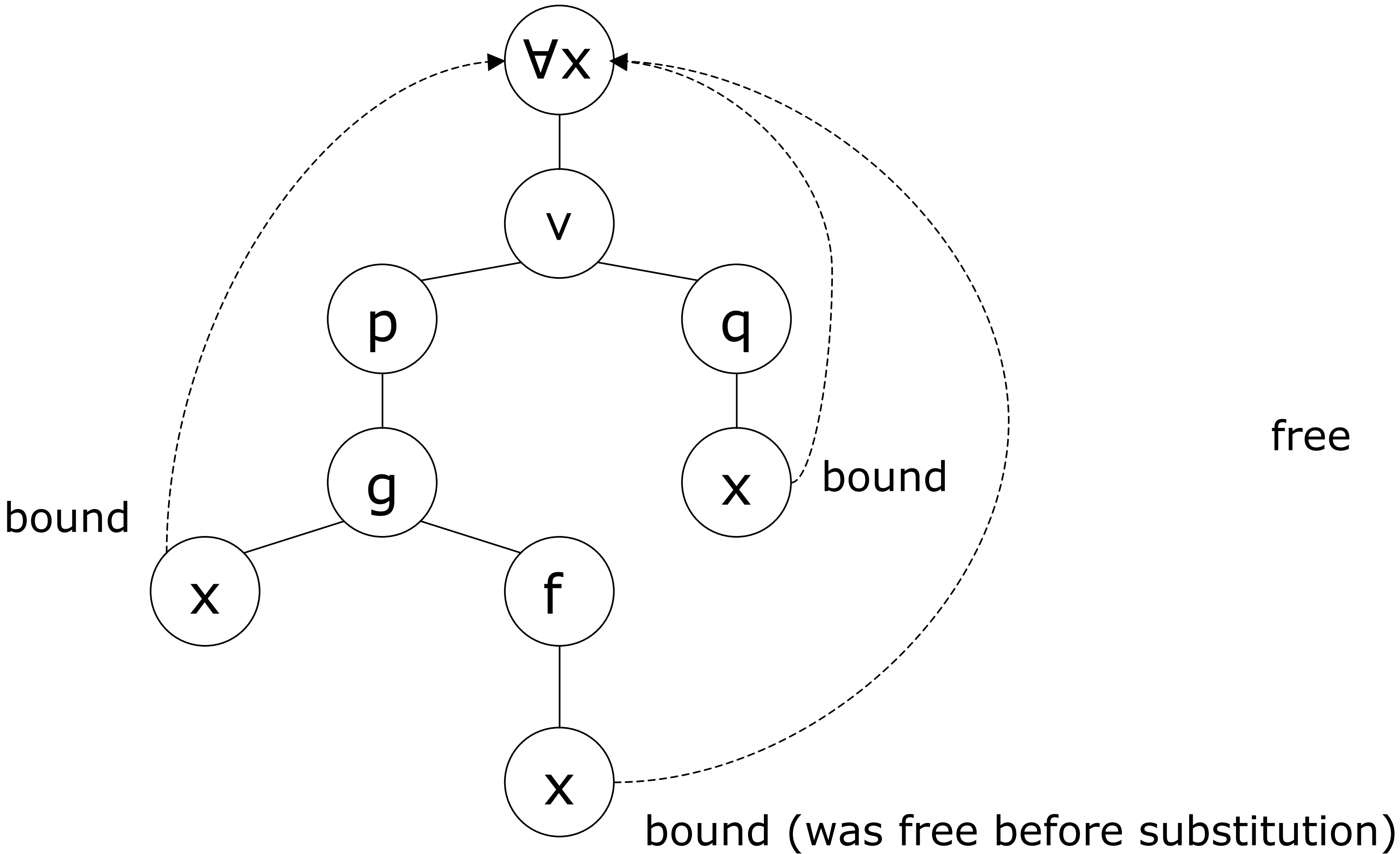
  (or more conventionally, **“t is free for v in F”)**

Non-Substitutability Example



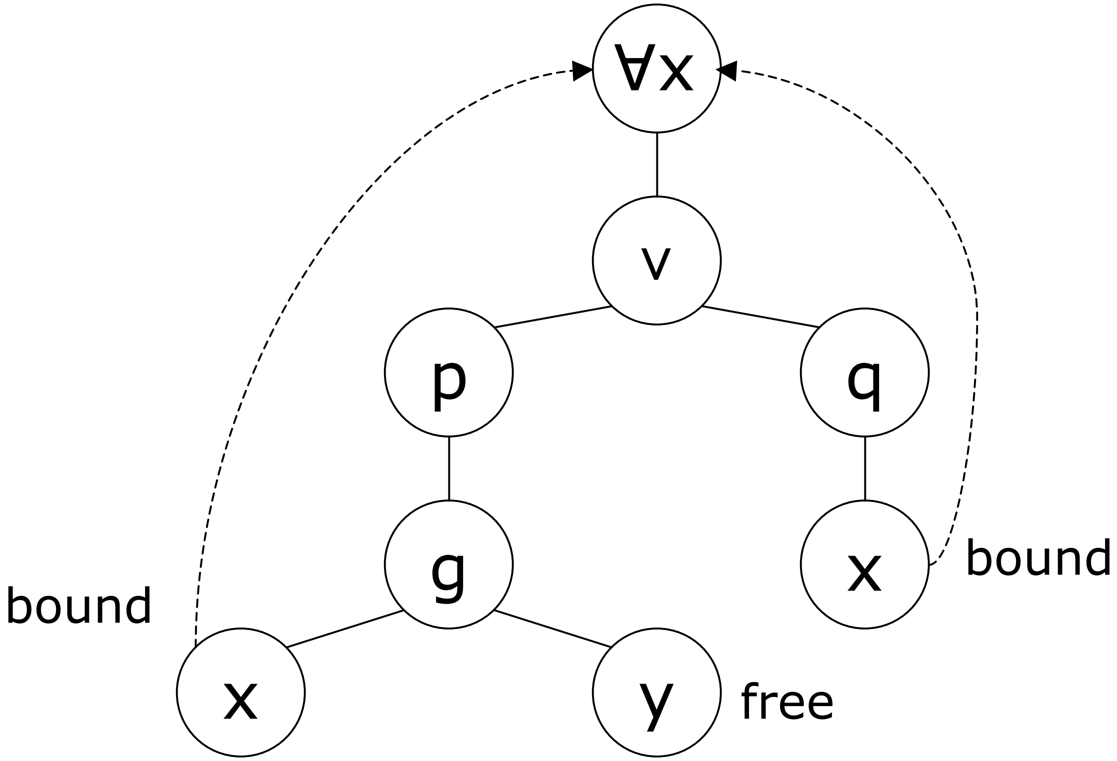
f(x) is **not** free to replace y in
(∀x)(p(g(x, y)) ∨ q(x))

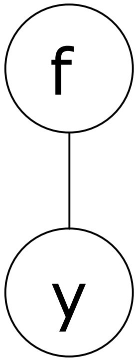
Non-Substitutability Example



(∀x)(p(g(x, **f(x)**)) v q(x))

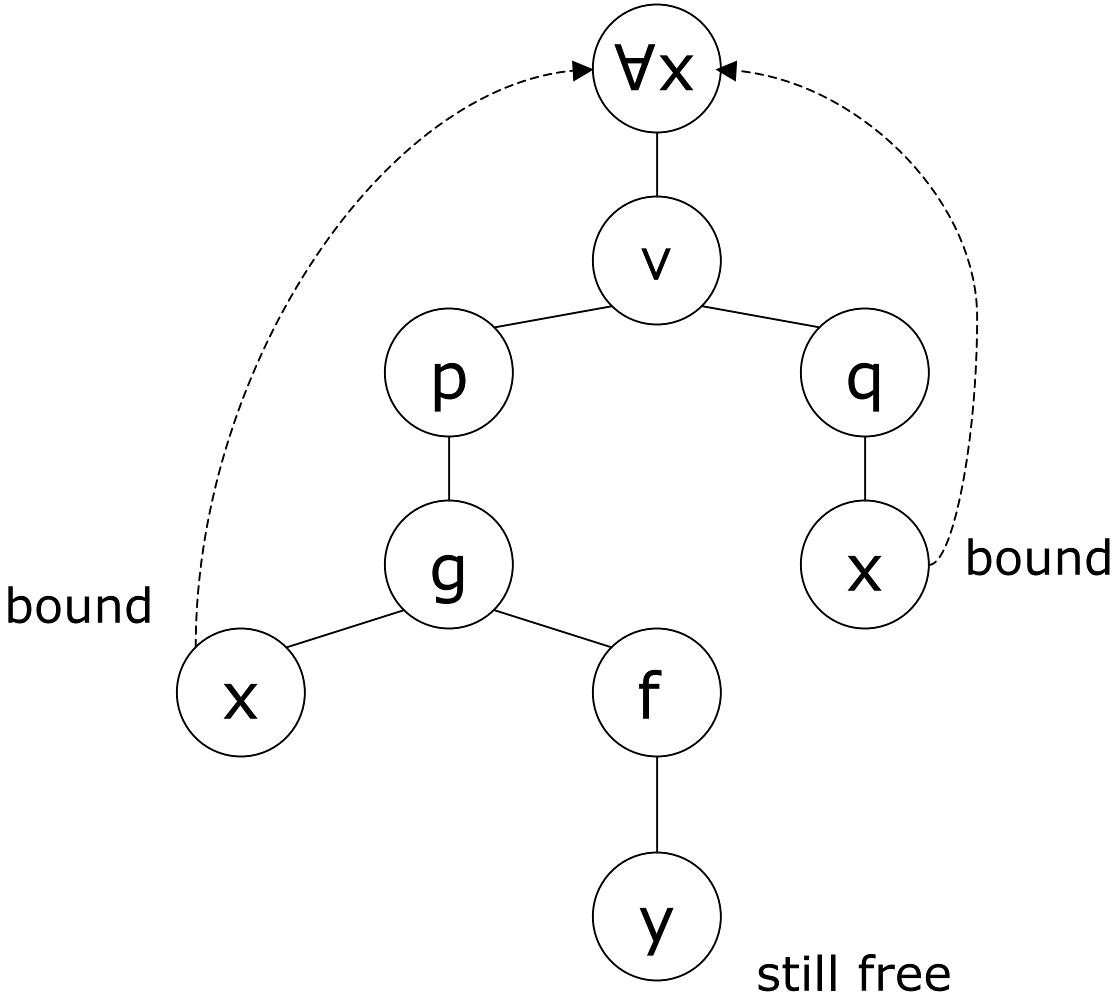
Substitutability Example





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]   (The Hein book uses F(v/t).)

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

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

Substitution Notation Example

Let F be the formula

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

Let v be the variable x.

Let t be the term f(y).

f(y) **is** free to replace x in $(\forall x)(p(g(x, y)) \vee q(x))$; there are no free instances of x.

F[f(y)/y] is the same as F.

Natural Deduction Rules

- We need introduction and elimination rules for both:
  - ∀
  - ∃

∀-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 derivable.

Why the 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**.

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

- Rather 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	
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		
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

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

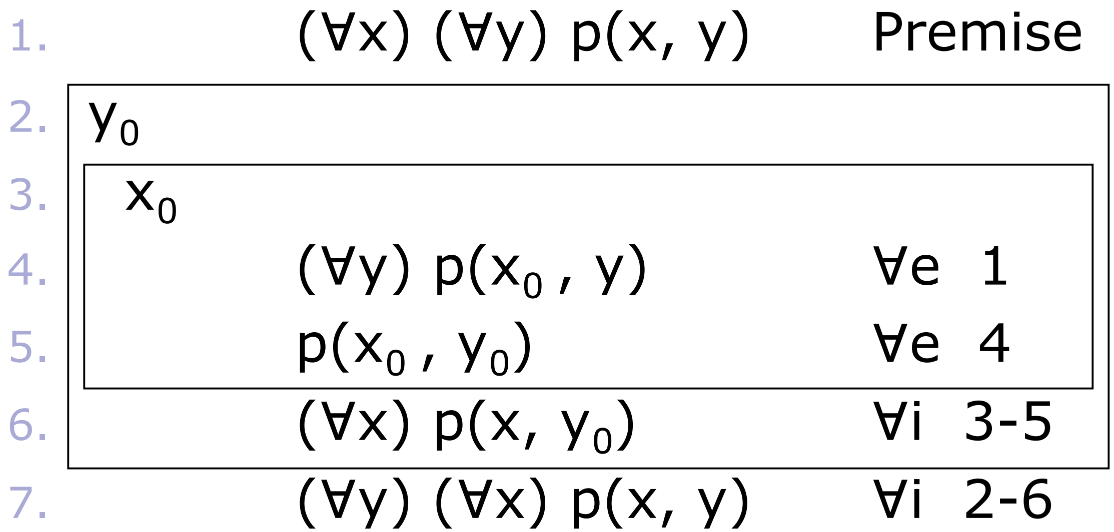
∀e ∀i Example (Tricky)

- 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. x_0

$\forall$e $\forall$i Example (Tricky)

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

∃-Introduction Rule (∃i)

- $$\frac{\varphi[t/x]}{(\exists x)\varphi} \quad (\exists i)$$

  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)

- $$\frac{\varphi[t/x]}{(\exists x)\varphi} \quad (\exists i)$$

  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.

∀e ∃i Example

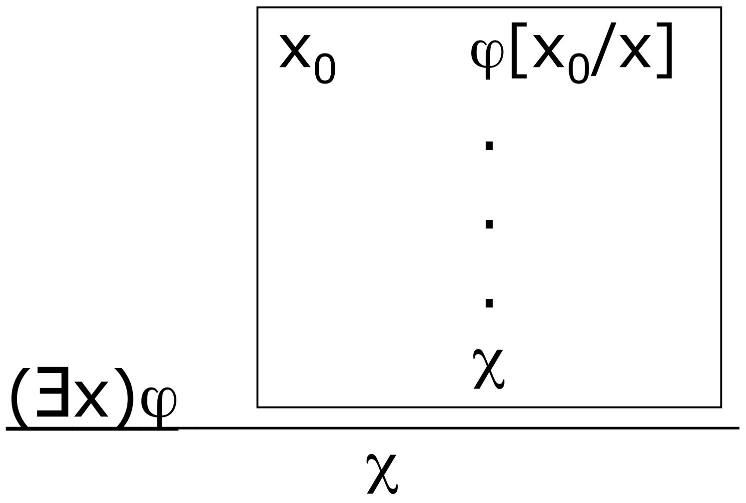
- Derive (∀x)p(x) ├─ (∃x) p(x) :

1.	(∀x) p(x)	Premise
2.	p(x)	∀e 1
3.	(∃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)



x_0 $\varphi[x_0/x]$

.

.

.

χ

$(\exists x)\varphi$

χ

(∃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.

∃-Elimination Rule (∃x e)

$$\frac{(\exists x)\varphi \qquad \boxed{\begin{array}{ll} x_0 & \varphi[x_0/x] \\ & \cdot \\ & \cdot \\ & \cdot \\ & \chi \end{array}}}{\chi} \qquad (\exists e)$$

- **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	$p(x_0)$	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 3-6

- In the ∃x 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 *original* x in 2 *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** 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	$p(x_0)$	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.