Predicate Logic

Robert Keller
February 2010

Predicate Calculus Language

- E is the start symbol
- E ::= A

```
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

- $\exists x \; p(x)$
- $\forall y \; (p(y) \land q(y))$
- $\forall y \; \exists x \; (p(f(x, y)) \rightarrow q(y))$
- $\forall x \; (p(f(x, y)) \lor q(x))$

- (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.]

- $\forall m\ (\neg(m = 0) \to (\exists n)\ (m = s(n)))$

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

- $\forall m\ \forall n\ ((s(m) = s(n)) \to m = n)$

  [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]

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

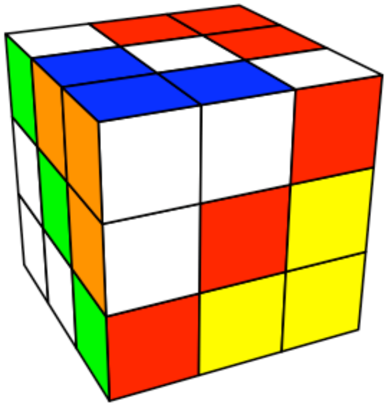
  [f is associative]

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

  [existence of inverse]

Examples of Groups

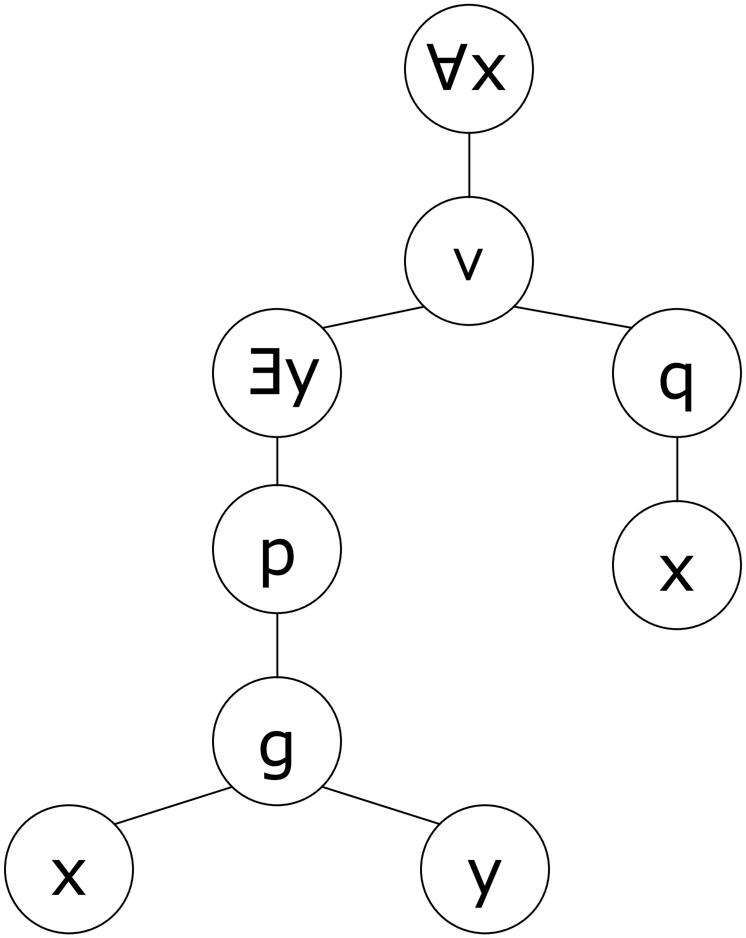
- Trivial group: {0} u = 0, f(0, 0) = 0
- 2-element group: {0, 1} u = 0, f(x, y) = x + y (mod 2)
- p-element group: {0, 1, …, p-1} for any prime p,
  u = 0, f(x, y) = x + y (mod p)
- Rubik's cube twists
- Particle spins (physics)
- Tire rotations
- Many others



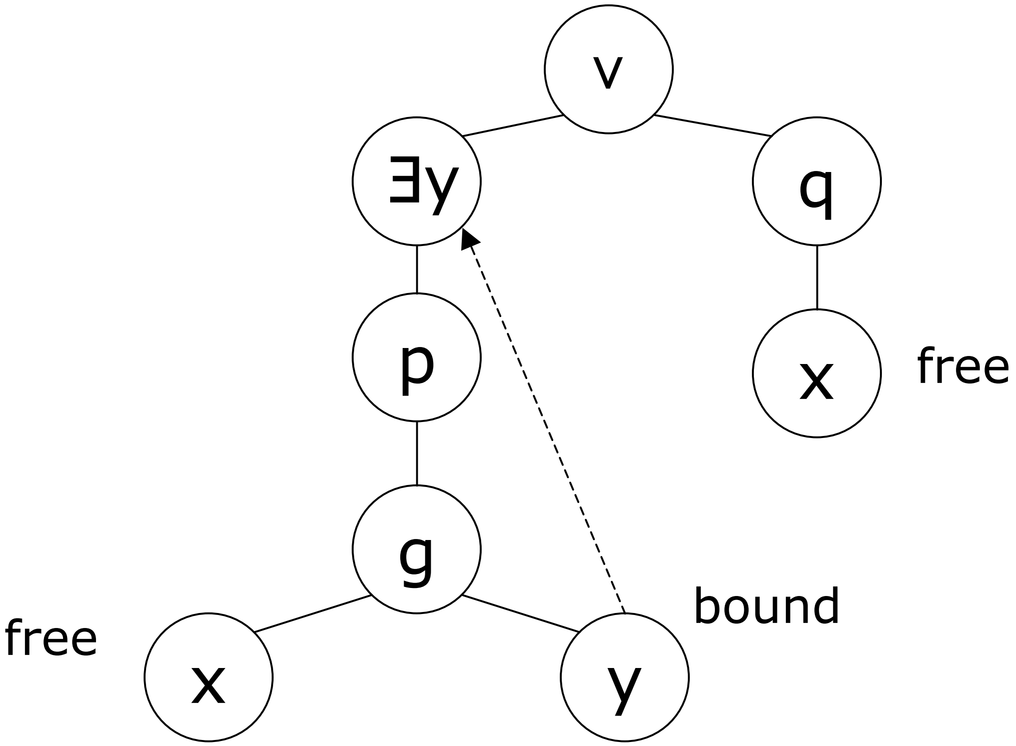
43252003274489856000 positions

Syntax Trees (or "Parse" Trees)

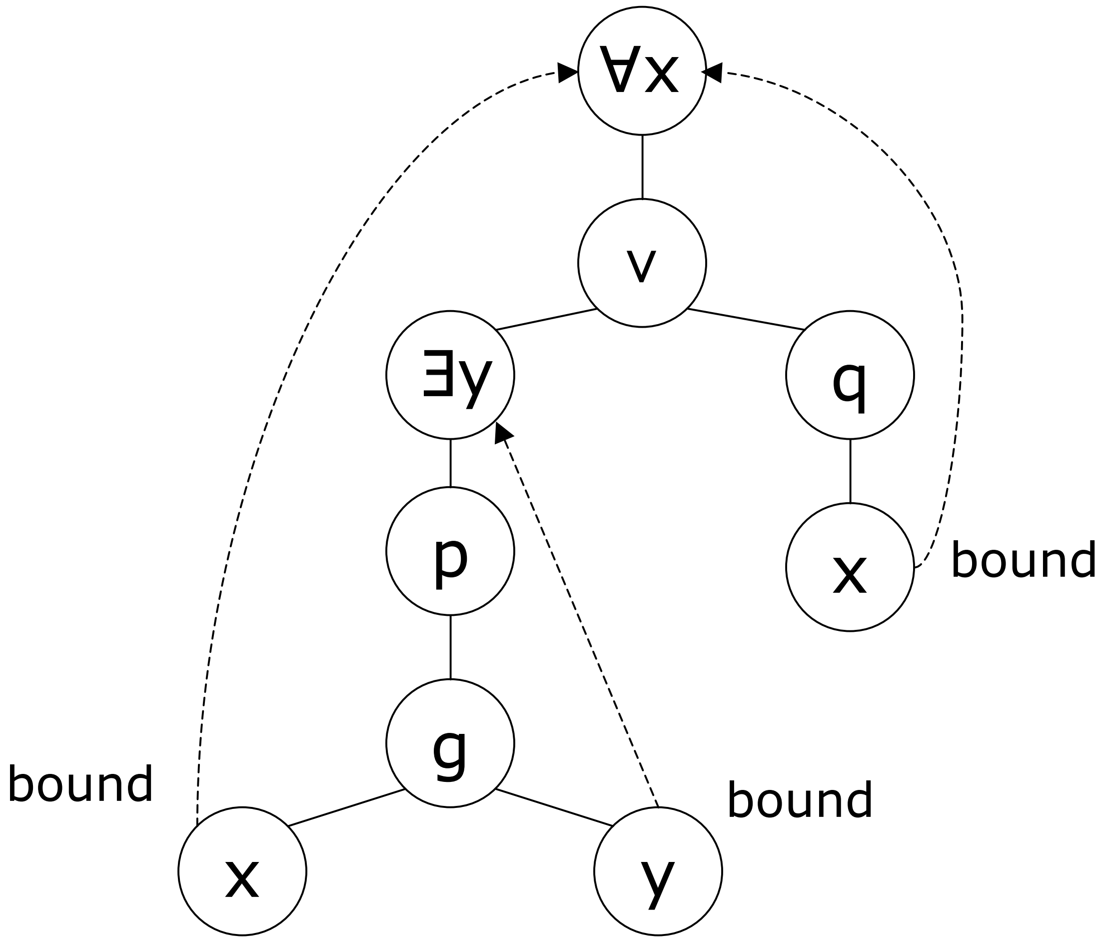
- We are assuming familiarity with syntax trees from CS 60.
- ∀x, ∃x are treated as if **1-ary operators**.
- Example: ∀x ((∃y p(g(x, y))) ∨ q(x))



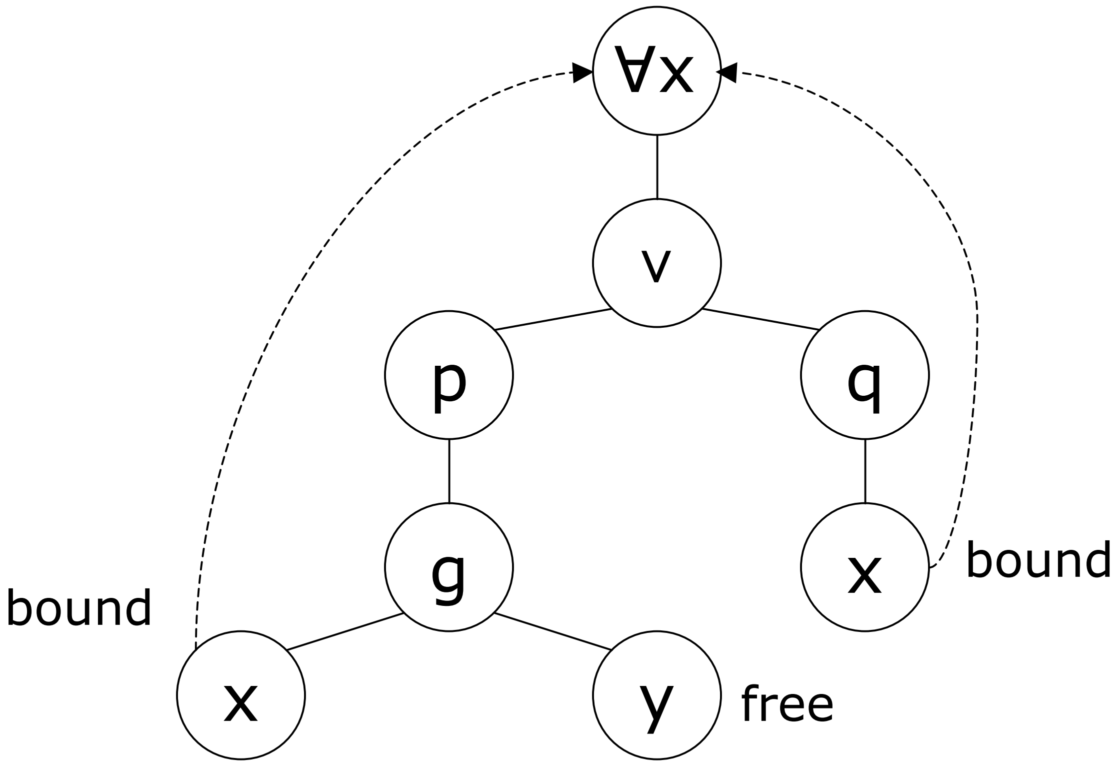
Free and Bound Variable Instances



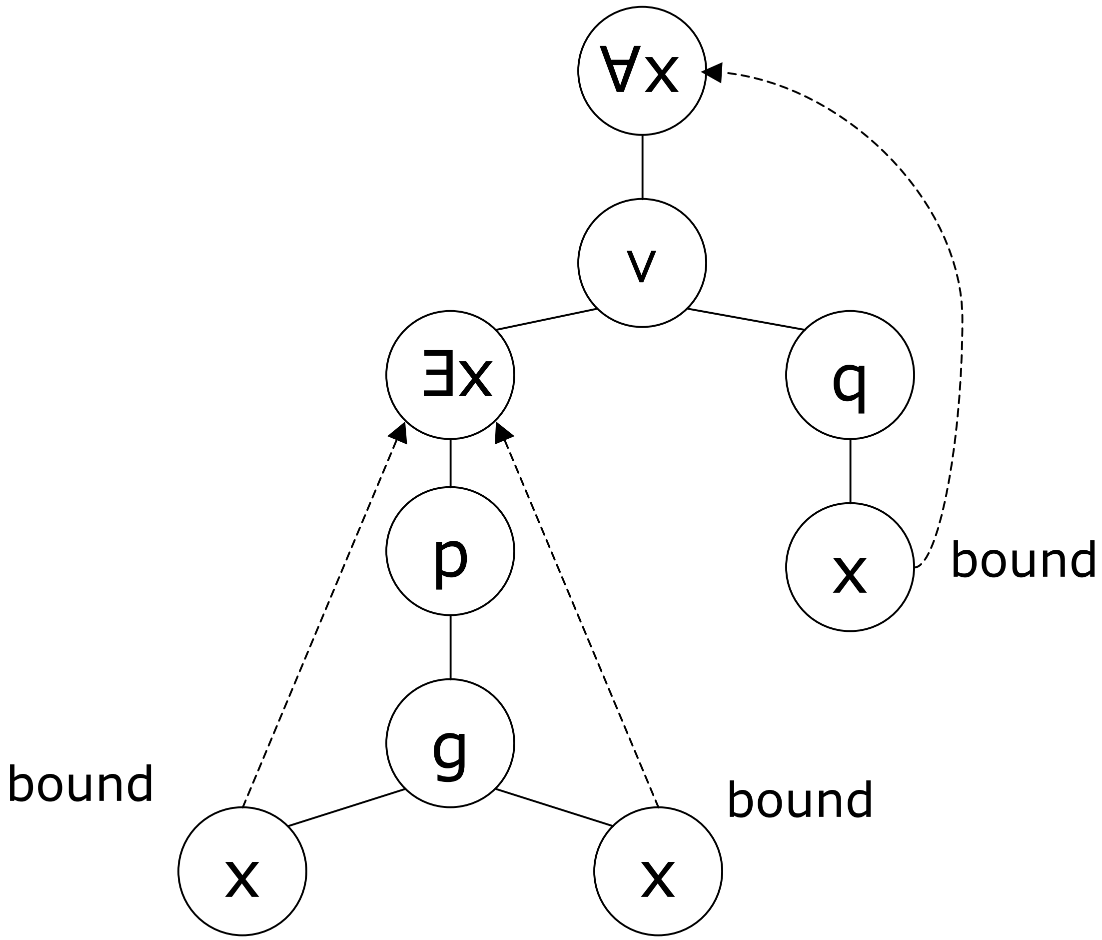
Free and Bound Variable Instances



Free and Bound Variable Instances



Free and Bound Variable Instances

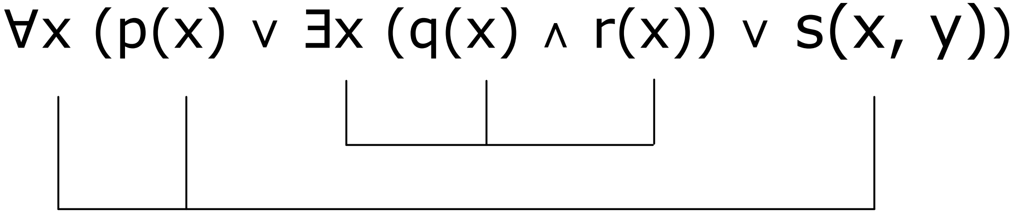


Scope of Variables

- 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 ∀x E, or ∃x E, the scope of x extends only inside E, and not beyond.

Scope Defined Inductively

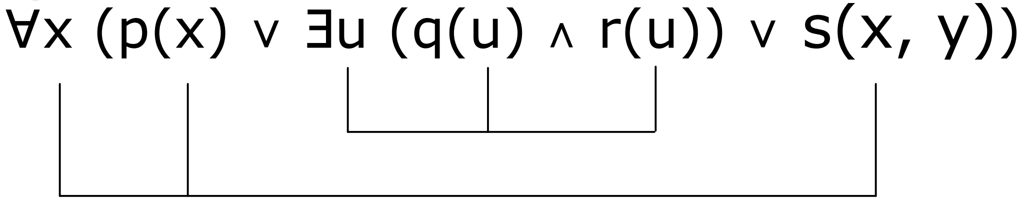
- For a quantifier-free formula, the scope of each variable is the entire formula.
- For ∀x E, or ∃x E, the scope of x is inside E, but not inside any quantification of the same variable inside E.
- Example: Two distinct scopes of x:



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

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:



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

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.

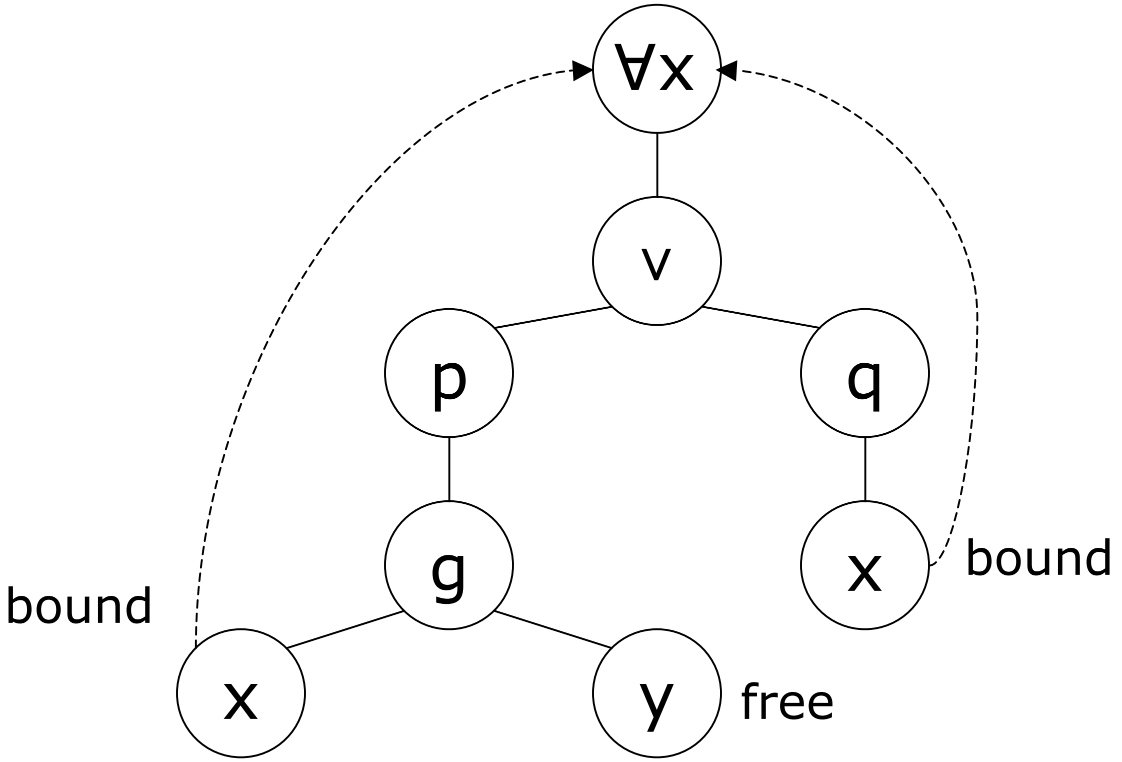
- While this is easy syntactically, there is a semantic restriction that must be observed:
  - In substituting a term for a variable within a formula, **no variables *within* the term can become bound** as a result of the substitution.

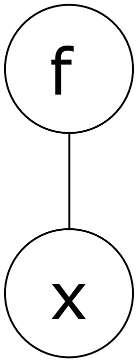
- If t is a term, v is a variable, and F is a formula, and the above restriction applies, we say that

"t is free to replace v in F"

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

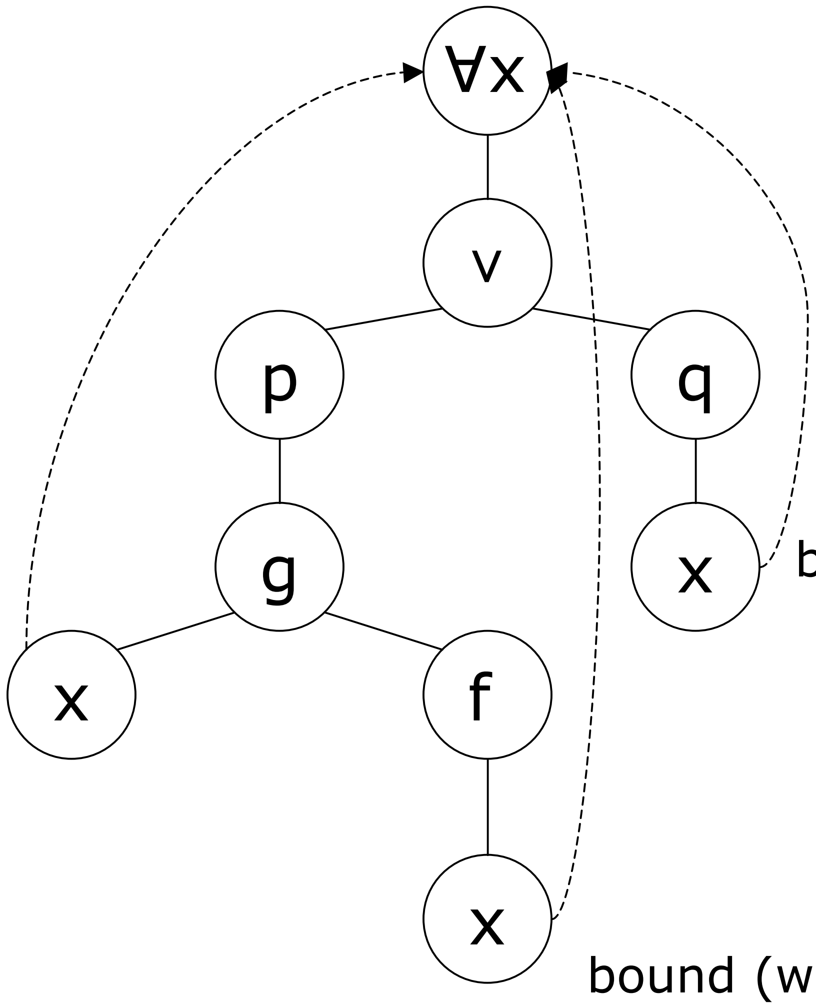
Non-Substitutability Example





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

Non-Substitutability Example



free

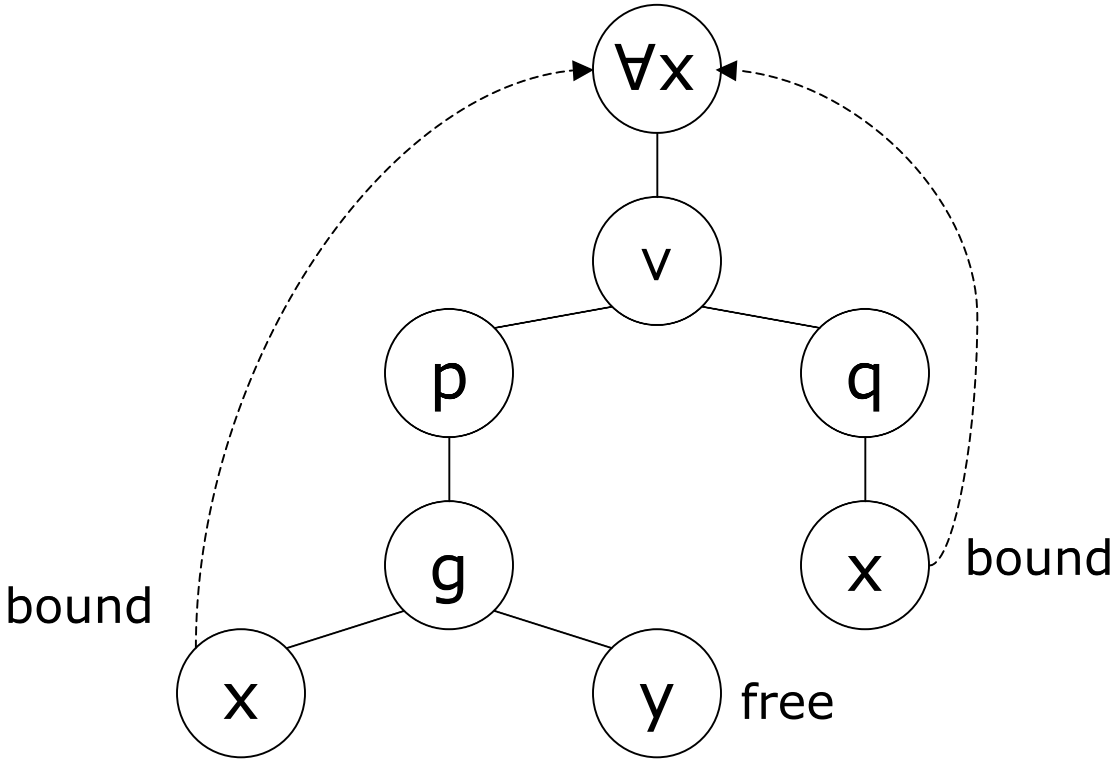
bound

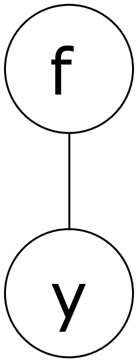
bound

bound (was free before substitution)

∀x (p(g(x, **f(x)**)) v q(x))

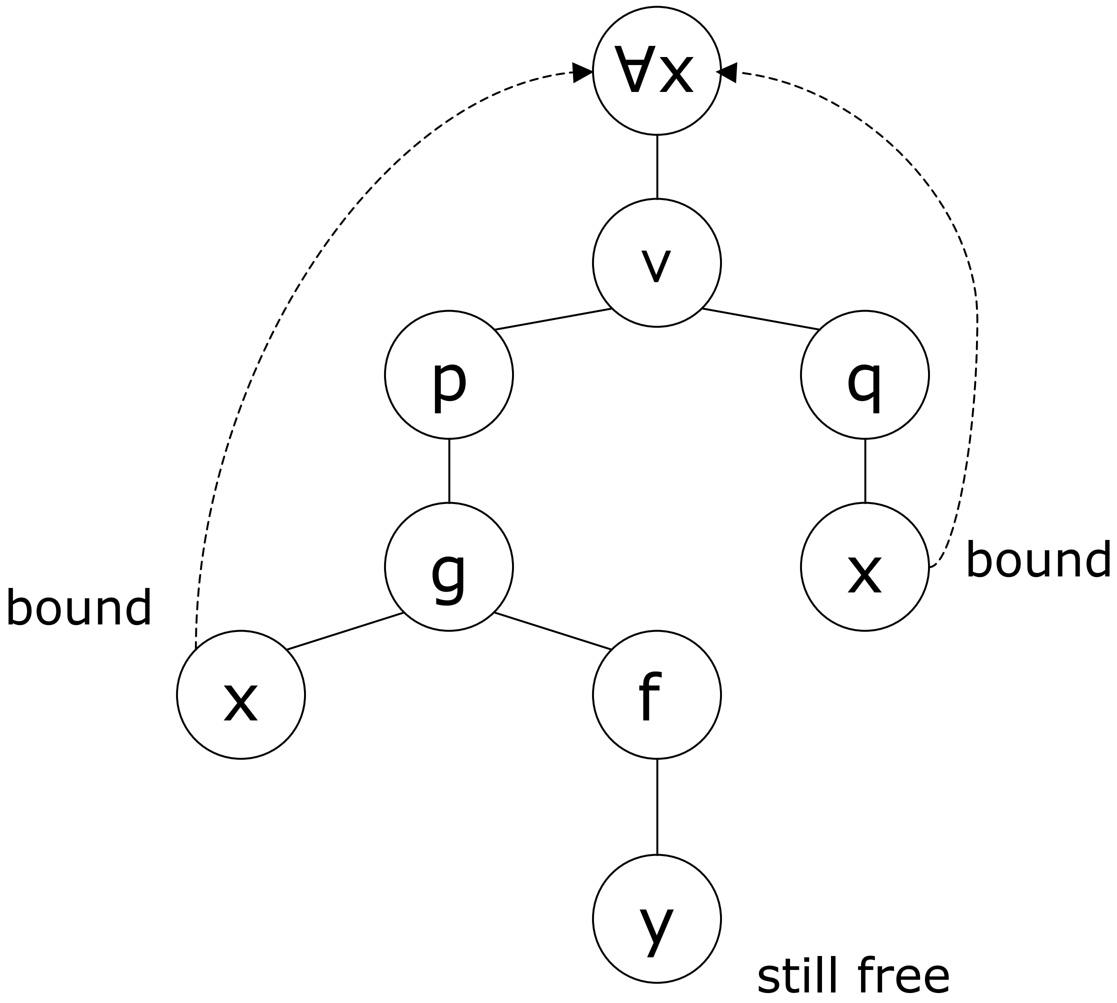
Substitutability Example





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

Substitutability Example



∀x (p(g(x, **f(y)**)) v 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

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

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.

∀-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.

Natural Deduction Rules

- We need introduction and elimination rules for both:
  - ∀
  - ∃
- These will be added to our propositional natural deduction rules.

∀-Introduction Rule (∀I)

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

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

- 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.

∀E ∀I Example

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

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

$\forall$E $\forall$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.	$p(x_0) \rightarrow q(x_0)$	$\forall$E 1
5.	$p(x_0)$	$\forall$E 2
6.	$q(x_0)$	$\rightarrow$E 4, 5
7.	$\forall x \, q(x)$	$\forall$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)

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

  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)

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

  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 $\vee$-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.

Sanity Check

- As with ∨ Introduction, ∃ Introduction is almost never the last line of a proof that two formulas are equivalent.

∀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)

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

- 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.

∃-Elimination Rule (∃x E)

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

- **What this rule says:**
- Assume that we have derived ∃x φ. 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

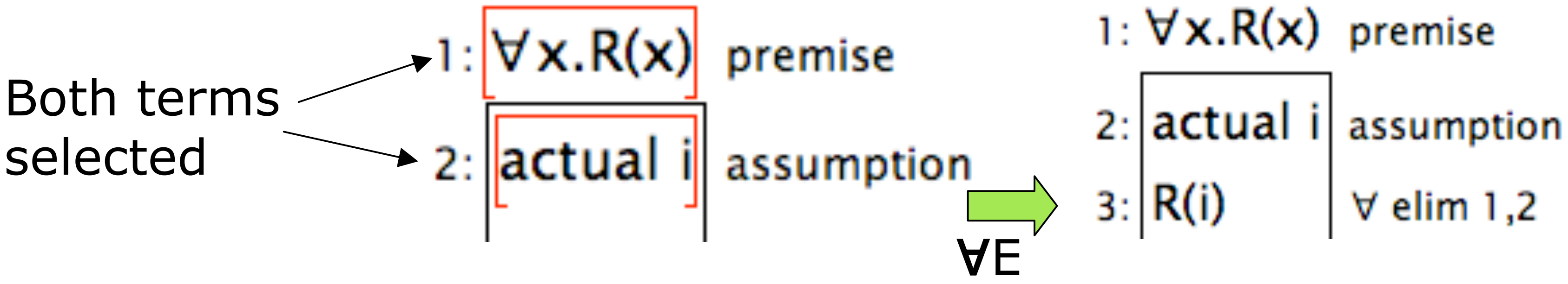
- Normally, ∃E can only be used to introduce a variable once. You cannot use it to introduce a second distinct variable.
- In other words, ∃xφ says that **an** x exists, but not necessarily more than one.
- In contrast, you can use ∃I as many times as you want (not that it well help).

Quantifier rule summary

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

JAPE Examples

- **∀ Elimination** (working ***forward***) instantiates a ∀-quantified variable with **a term that already exists** (in this case, **i**).
- **Both** the term and the ∀ formula must be selected (using shift-click to add one or the other):



Both terms selected

1: ∀x.R(x) premise
2: actual i assumption

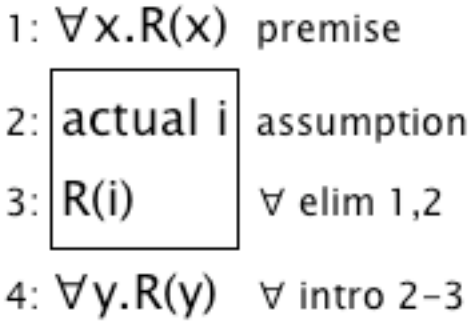
∀E

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

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.

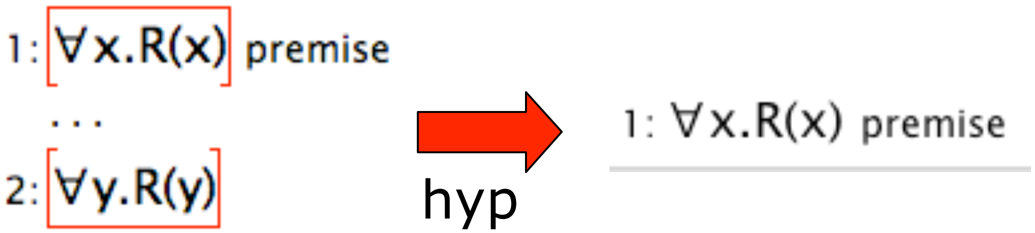
JAPE Examples

- ∀ 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.

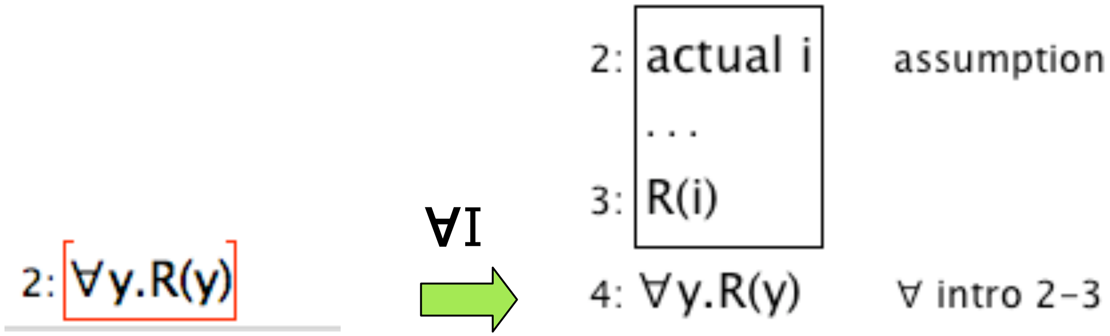


1: ∀x.R(x) premise
. . .
2: ∀y.R(y)

1: ∀x.R(x) premise

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**.



2: ∀y.R(y)

2: actual i — assumption

...

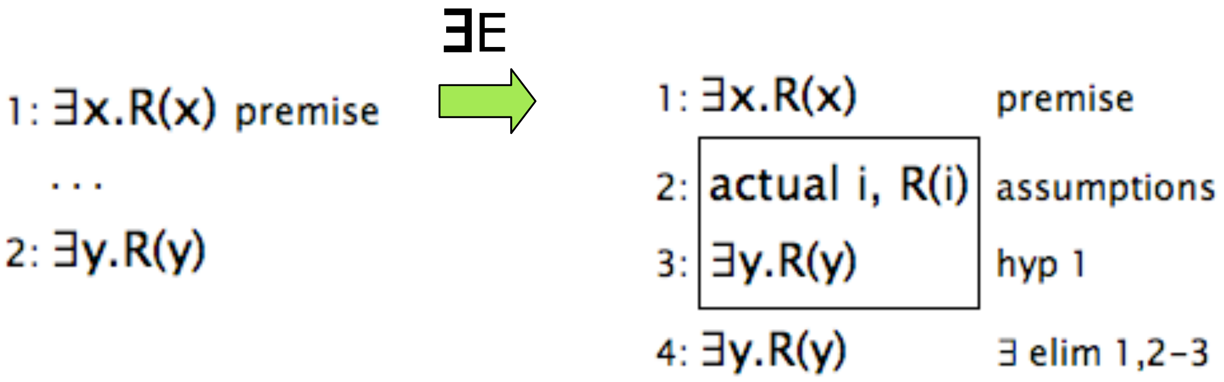
3: R(i)

4: ∀y.R(y) — ∀ intro 2-3

i is meaningless out here

JAPE Examples

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



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

...

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

∃E

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

2: actual i, $R(i)$ assumptions

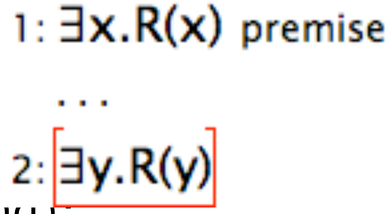
3: $\exists y.R(y)$ hyp 1

4: $\exists y.R(y)$ ∃ elim 1,2-3

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

JAPE Examples

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

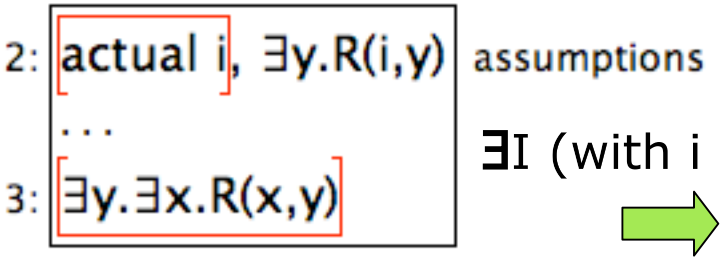


1: ∃x.R(x) premise

...

2: ∃y.R(y)

- Here is an example wi no variable ***n*** be used
  (but leads to a dead e):

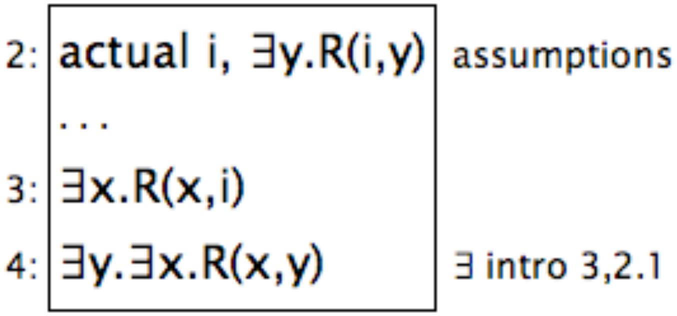


2: actual i, ∃y.R(i,y) assumptions

...

3: ∃y.∃x.R(x,y)

∃I (with i for y) →



2: actual i, ∃y.R(i,y) assumptions

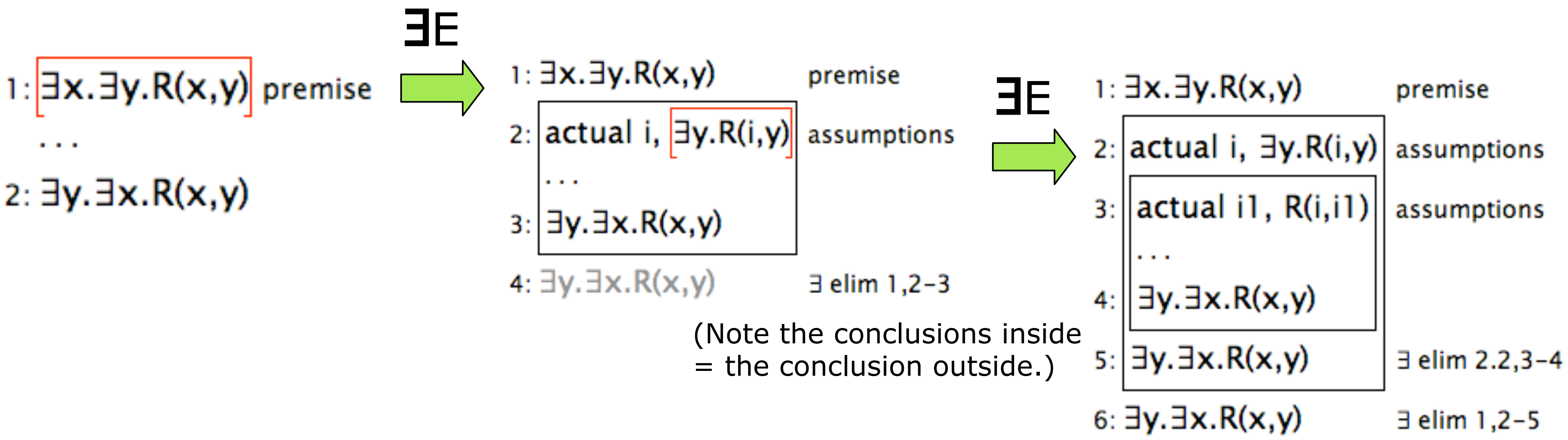
...

3: ∃x.R(x,i)

4: ∃y.∃x.R(x,y) ∃ intro 3,2.1

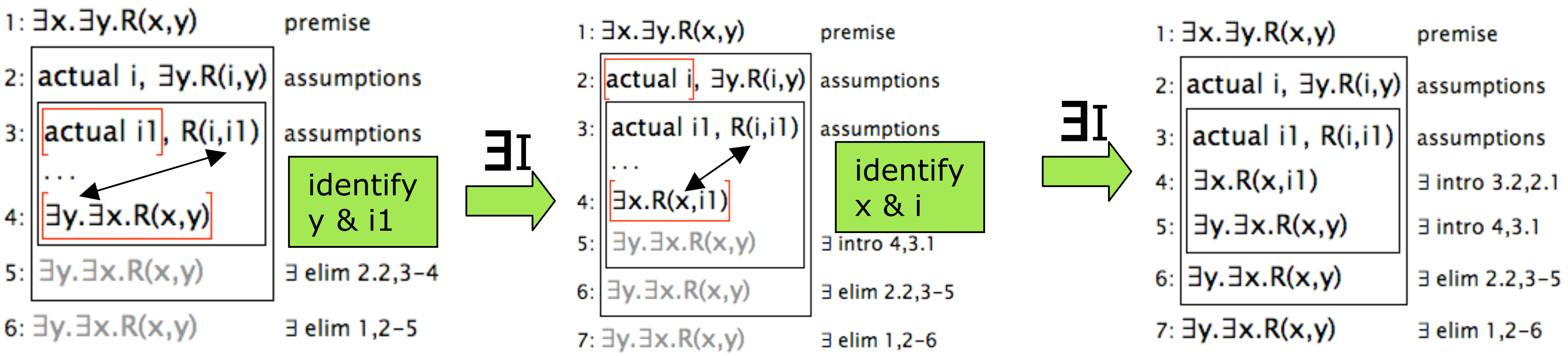
Proof of a sequent using ∃E and ∃I

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

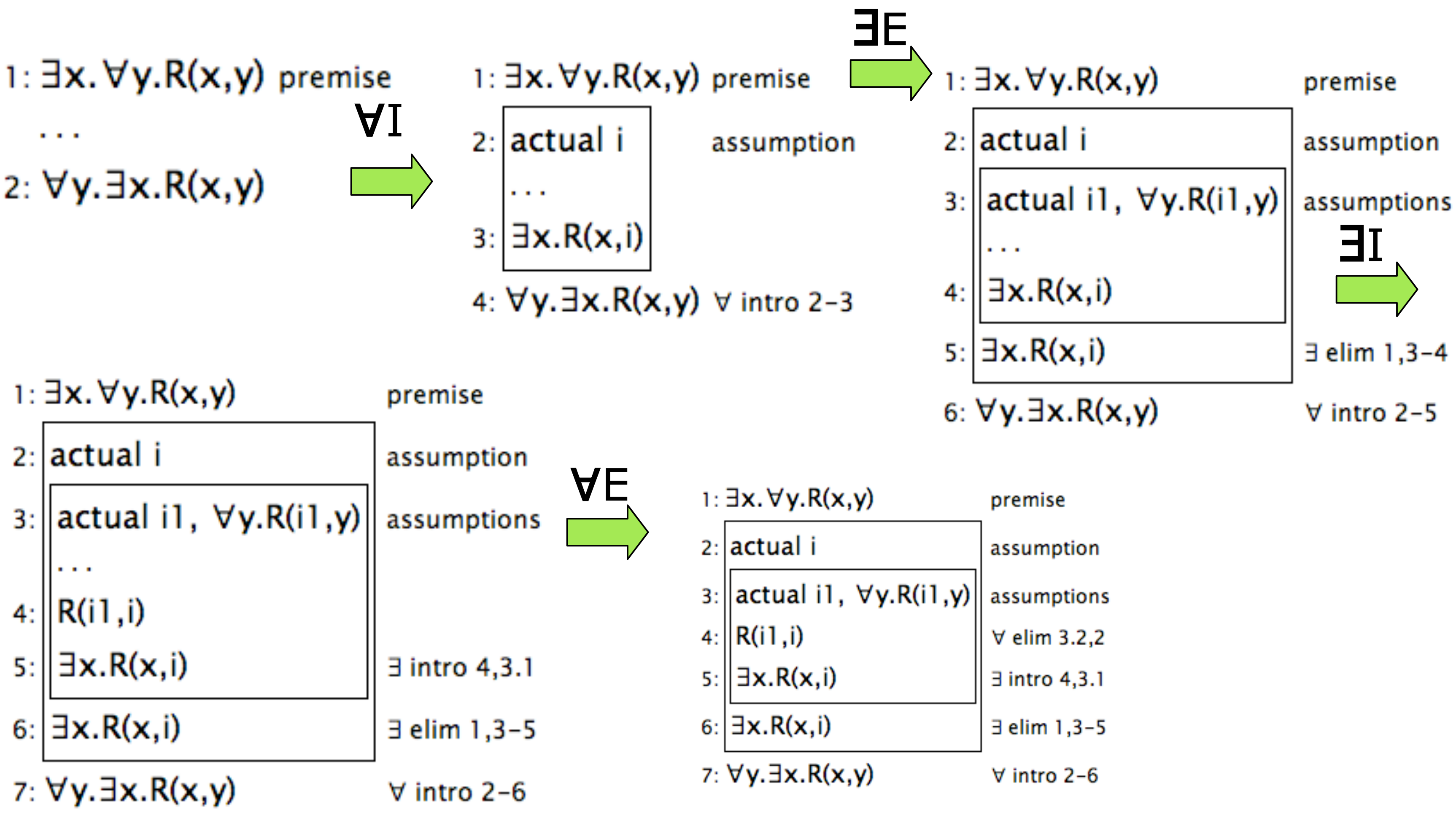


(Note the conclusions inside = the conclusion outside.)

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

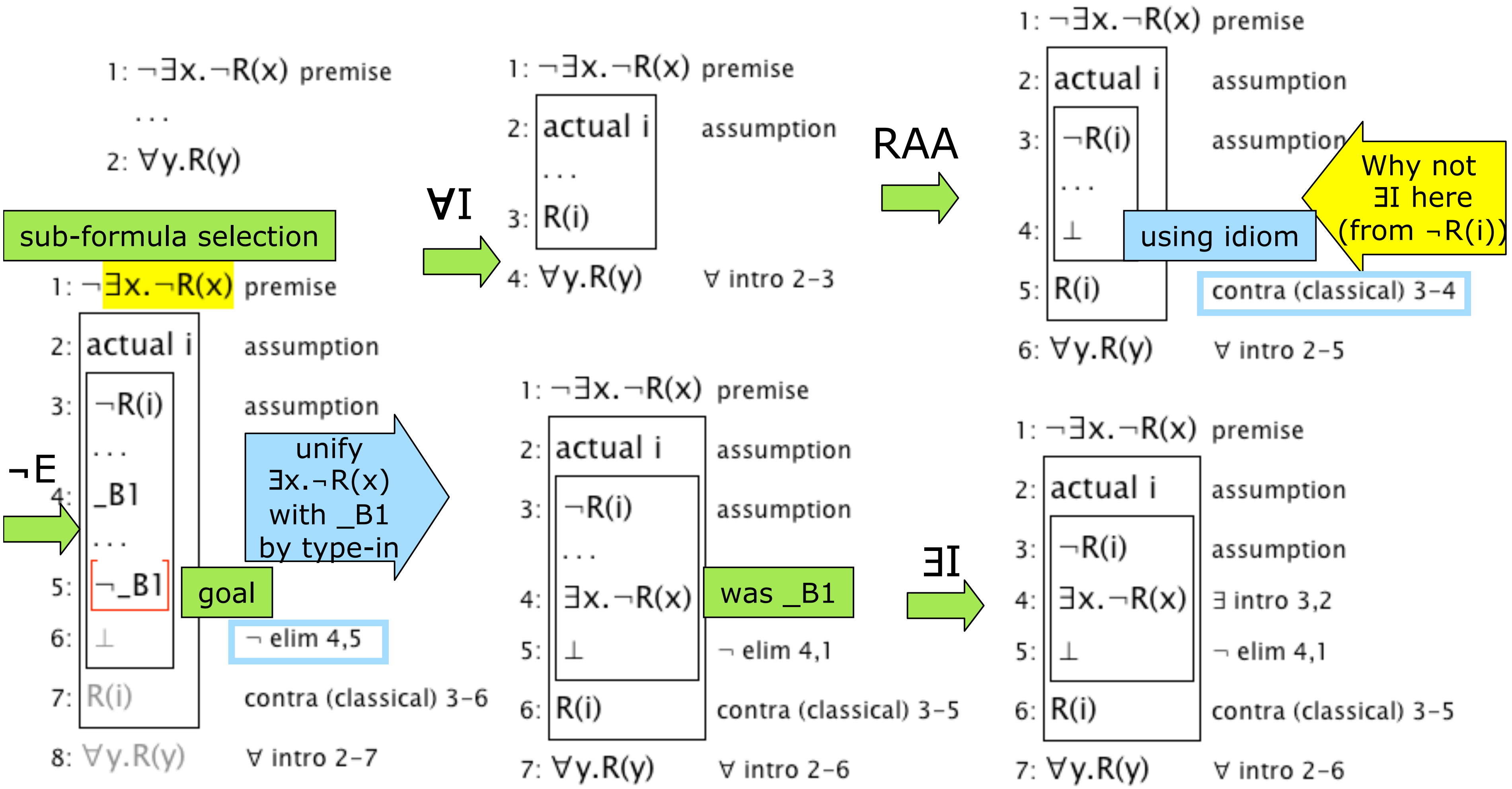
1: ∃x.∀y.R(x,y) premise
2: actual i assumption
3: actual i1, ∀y.R(i1,y) assumptions
4: R(i1,i) ∀ elim 3.2,2
5: ∃x.R(x,i) ∃ intro 4,3.1
6: ∃x.R(x,i) ∃ elim 1,3–5
7: ∀y.∃x.R(x,y) ∀ intro 2–6

A Common Idiom

- 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.



1: ¬∃x.¬R(x) premise
...
2: ∀y.R(y)

∀I

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

RAA

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

using idiom

Why not ∃I here (from ¬R(i))

sub-formula selection

¬E

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

goal

unify ∃x.¬R(x) with _B1 by type-in

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

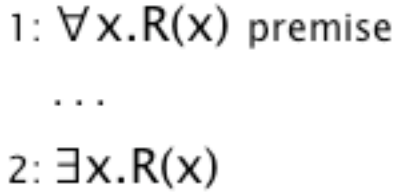
was _B1

∃I

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 assumption 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.
- Proved in textbooks, but not provable in JAPE:

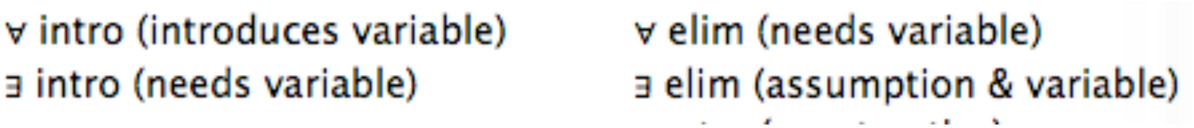


1: ∀x.R(x) premise

...

2: ∃x.R(x)

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



∀ intro (introduces variable)

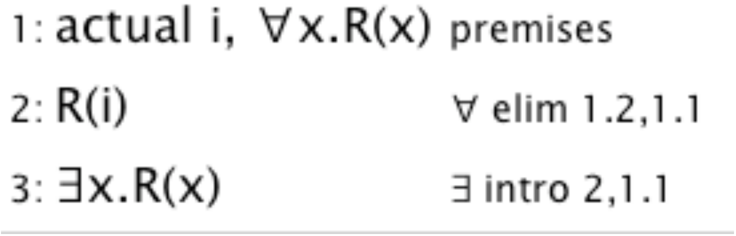
∃ intro (needs variable)

∀ elim (needs variable)

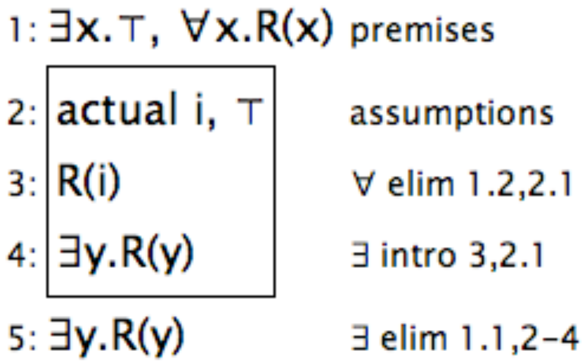
∃ elim (assumption & variable)

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).



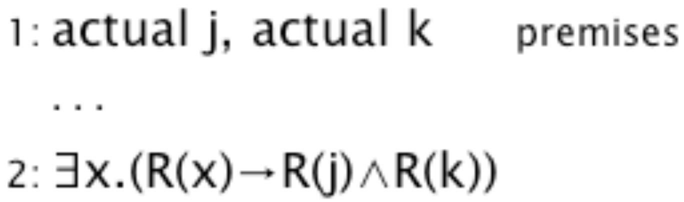
1: actual i, ∀x.R(x) premises
2: R(i) ∀ elim 1.2,1.1
3: ∃x.R(x) ∃ intro 2,1.1



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

- See Bornat’s book “Proof and Dispro 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: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

∨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

. . .

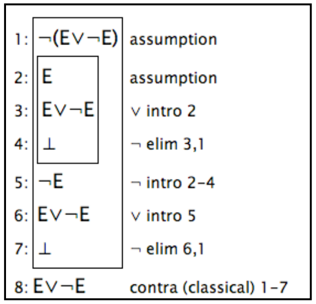
6: $\exists x.(R(x) \rightarrow R(j) \wedge R(k))$

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

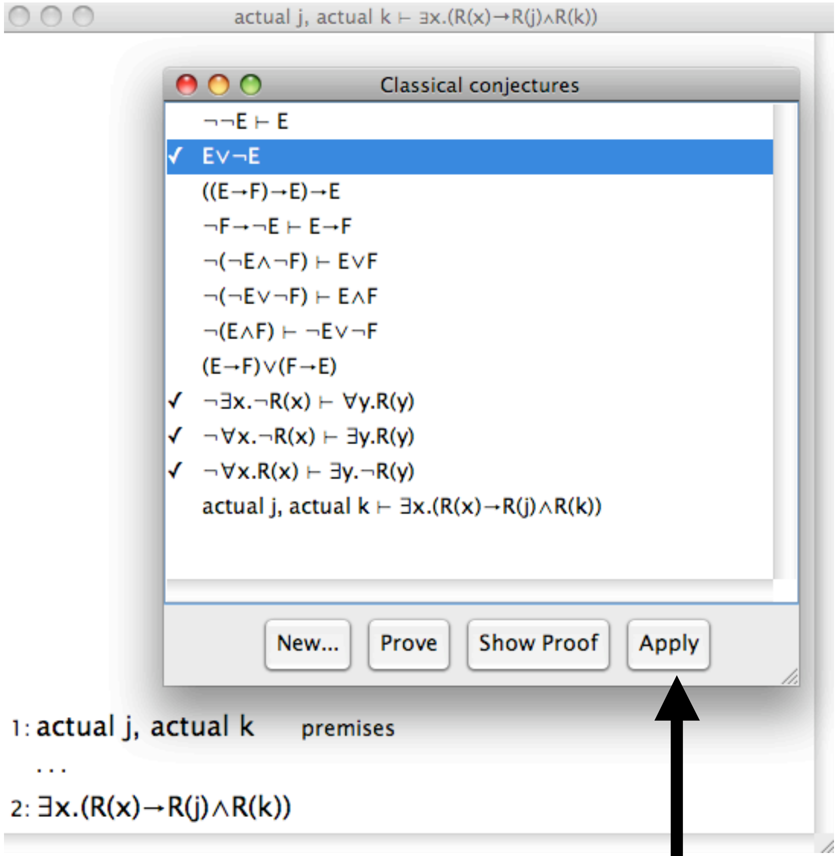
What x would make this work?

What x would make this work?

How to introduce LEM
(it must be proved first)

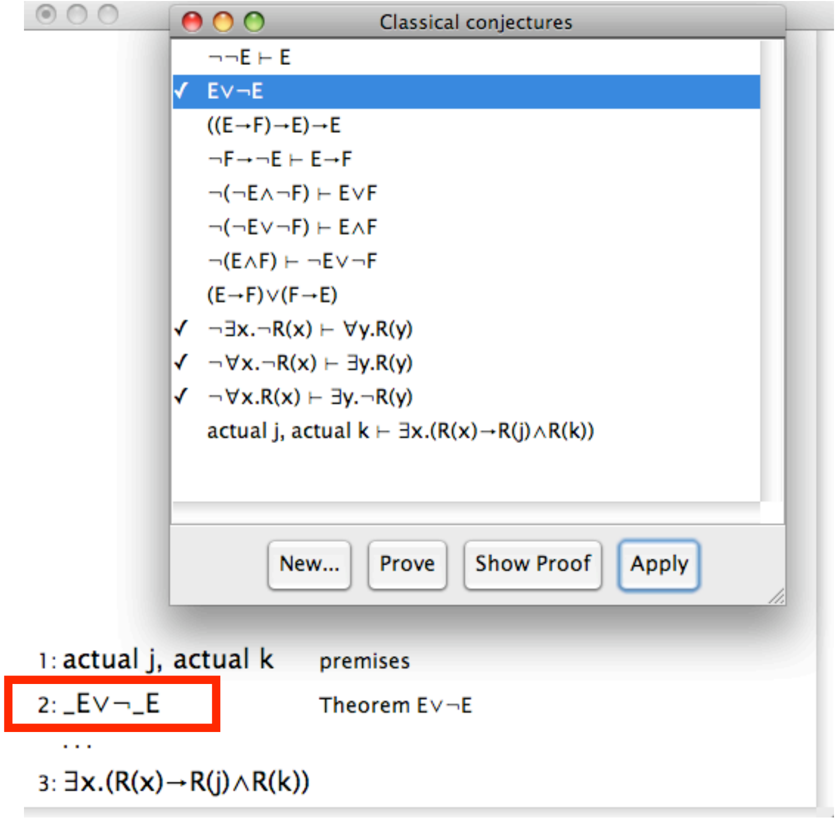


1:	¬(E∨¬E)	assumption
2:	E	assumption
3:	E∨¬E	∨ intro 2
4:	⊥	¬ elim 3,1
5:	¬E	¬ intro 2–4
6:	E∨¬E	∨ intro 5
7:	⊥	¬ elim 6,1
8:	E∨¬E	contra (classical) 1–7



¬¬E ⊢ E
✓ E∨¬E
((E→F)→E)→E
¬F→¬E ⊢ E→F
¬(¬E∧¬F) ⊢ E∨F
¬(¬E∨¬F) ⊢ E∧F
¬(E∧F) ⊢ ¬E∨¬F
(E→F)∨(F→E)
✓ ¬∃x.¬R(x) ⊢ ∀y.R(y)
✓ ¬∀x.¬R(x) ⊢ ∃y.R(y)
✓ ¬∀x.R(x) ⊢ ∃y.¬R(y)
actual j, actual k ⊢ ∃x.(R(x)→R(j)∧R(k))

1: actual j, actual k premises
...
2: ∃x.(R(x)→R(j)∧R(k))



¬¬E ⊢ E
✓ E∨¬E
((E→F)→E)→E
¬F→¬E ⊢ E→F
¬(¬E∧¬F) ⊢ E∨F
¬(¬E∨¬F) ⊢ E∧F
¬(E∧F) ⊢ ¬E∨¬F
(E→F)∨(F→E)
✓ ¬∃x.¬R(x) ⊢ ∀y.R(y)
✓ ¬∀x.¬R(x) ⊢ ∃y.R(y)
✓ ¬∀x.R(x) ⊢ ∃y.¬R(y)
actual j, actual k ⊢ ∃x.(R(x)→R(j)∧R(k))

1: actual j, actual k premises
2: _E∨¬_E Theorem E∨¬E
...
3: ∃x.(R(x)→R(j)∧R(k))

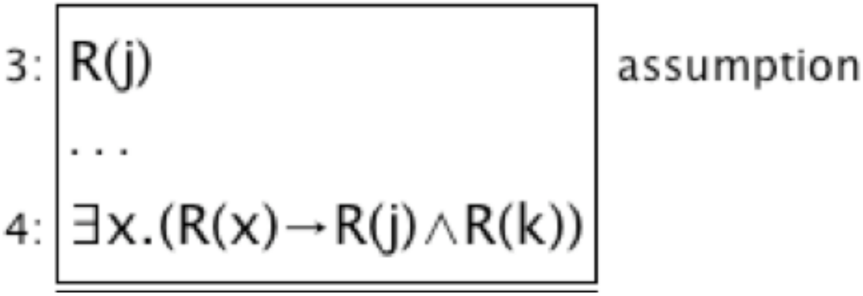
Click to apply as lemma

Voila!

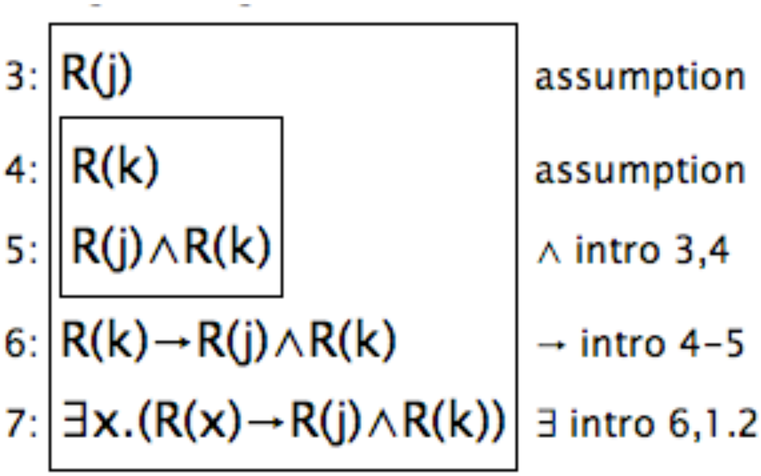
Continuing the tricky proof …

For the Top Box

x = k (an actual) will enable ∃I



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

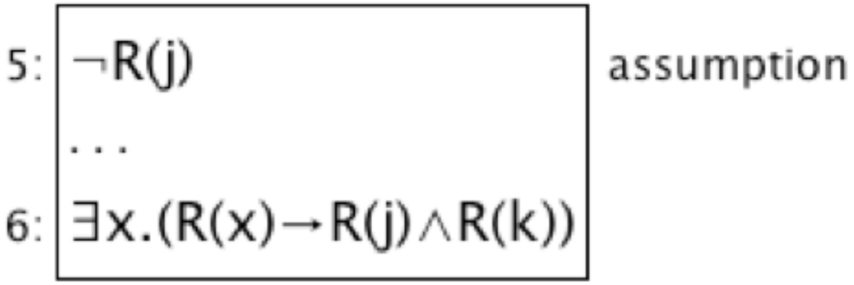


3:	$R(j)$	assumption
4:	$R(k)$	assumption
5:	$R(j) \wedge R(k)$	∧ intro 3,4
6:	$R(k) \rightarrow R(j) \wedge R(k)$	→ intro 4-5
7:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	∃ intro 6,1.2

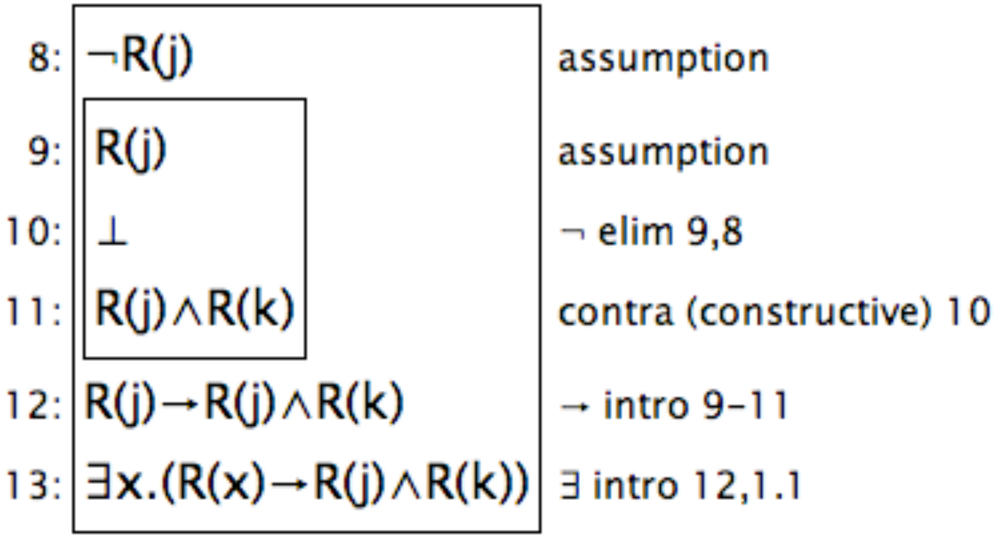
Continuing the tricky proof …

For the Bottom Box

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

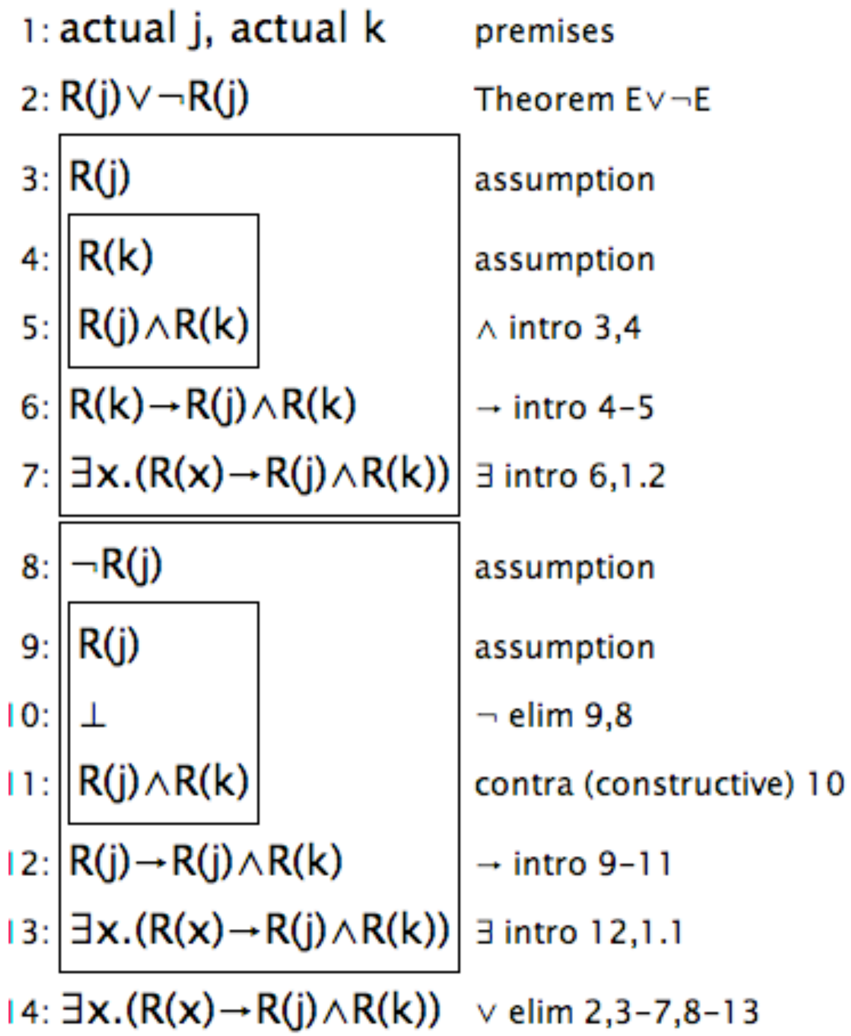


5:	$\neg R(j)$	assumption
	. . .	
6:	$\exists x.(R(x) \to R(j) \wedge R(k))$	



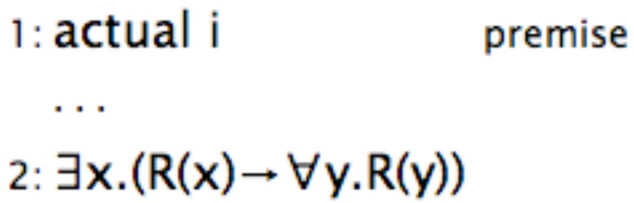
8:	$\neg R(j)$	assumption
9:	$R(j)$	assumption
10:	$\bot$	¬ elim 9,8
11:	$R(j) \wedge R(k)$	contra (constructive) 10
12:	$R(j) \to R(j) \wedge R(k)$	→ intro 9–11
13:	$\exists x.(R(x) \to R(j) \wedge R(k))$	∃ intro 12,1.1

Completed Proof



1:	actual j, actual k	premises
2:	$R(j) \vee \neg R(j)$	Theorem E∨¬E
3:	$R(j)$	assumption
4:	$R(k)$	assumption
5:	$R(j) \wedge R(k)$	∧ intro 3,4
6:	$R(k) \rightarrow R(j) \wedge R(k)$	→ intro 4-5
7:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	∃ intro 6,1.2
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
14:	$\exists x.(R(x) \rightarrow R(j) \wedge R(k))$	∨ elim 2,3-7,8-13

An Analogous Sequent



1: actual i premise

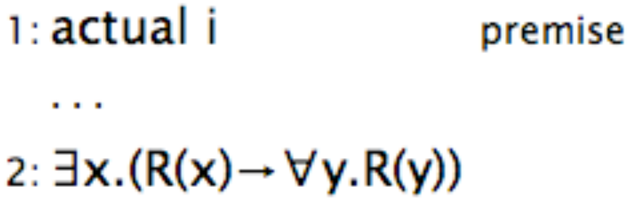
...

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

DeMorgan’s Rules for Quantifiers

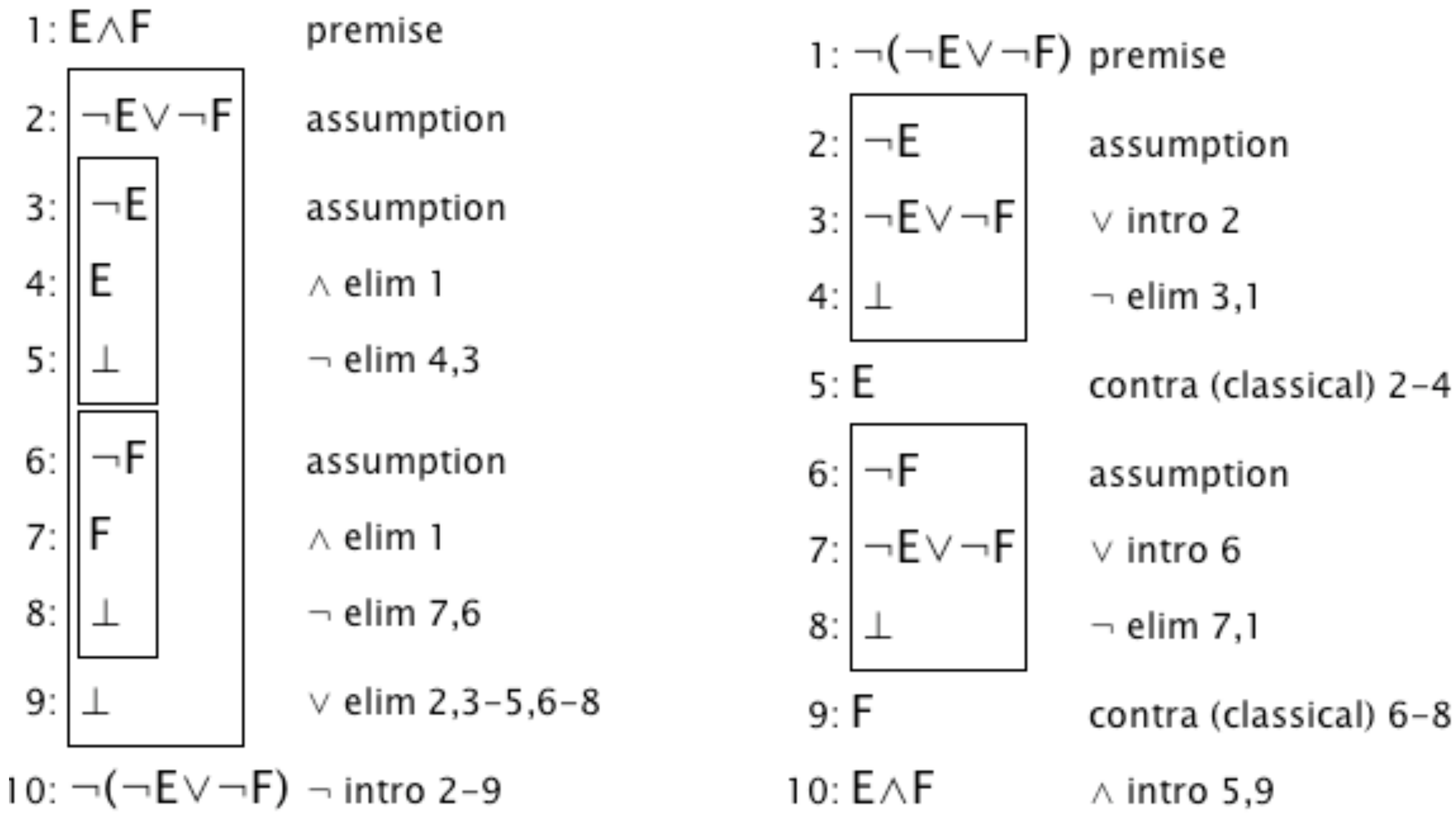
- Recall DeMorgan’s rules for propositions
  - $(p \wedge q) \Leftrightarrow \neg(\neg p \vee \neg q)$ $(\neg p \vee \neg q) \Leftrightarrow \neg(p \wedge q)$
  - $(p \vee q) \Leftrightarrow \neg(\neg p \wedge \neg q)$ $\neg(p \vee q) \Leftrightarrow (\neg p \wedge \neg q)$

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

  - $\exists x\, P(x) \Leftrightarrow \neg(\forall x\, \neg P(x))$ $\neg \exists x\, P(x) \Leftrightarrow \forall x\, \neg P(x)$

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

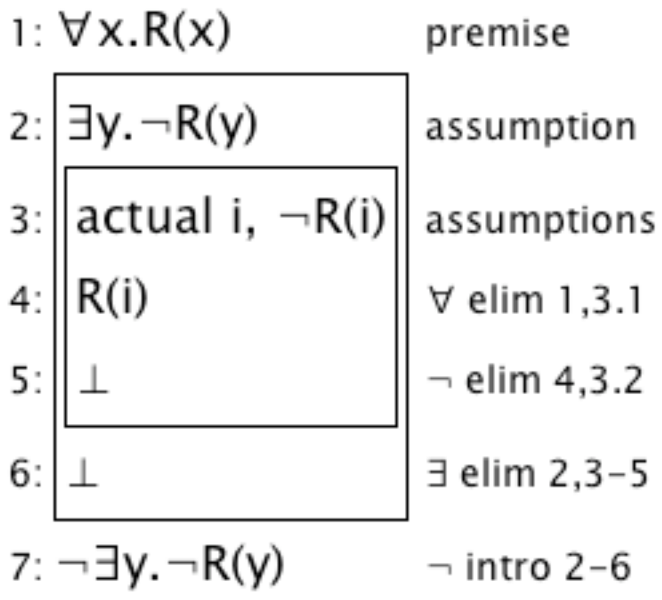
Constructive → vs. Classical ←



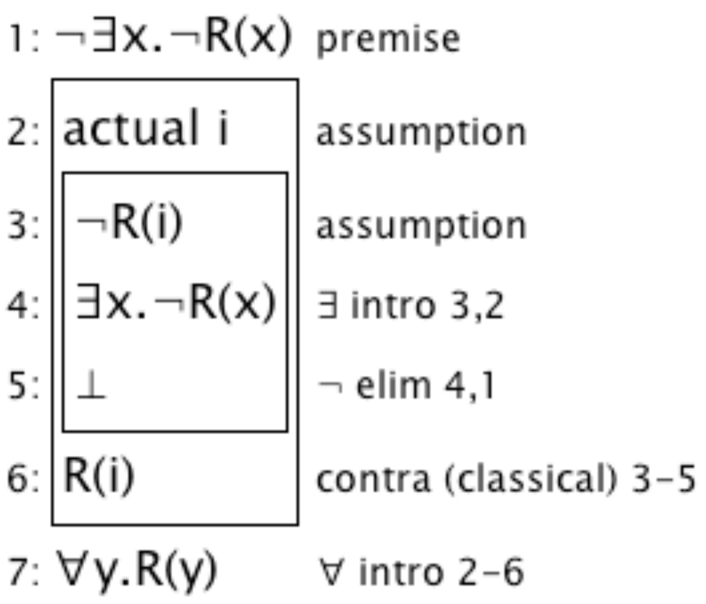
1:	$E \wedge F$	premise
2:	$\neg E \vee \neg F$	assumption
3:	$\neg E$	assumption
4:	E	∧ elim 1
5:	$\bot$	¬ elim 4,3
6:	$\neg F$	assumption
7:	F	∧ elim 1
8:	$\bot$	¬ elim 7,6
9:	$\bot$	∨ elim 2,3–5,6–8
10:	$\neg(\neg E \vee \neg F)$	¬ intro 2–9

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

Constructive → vs. Classical ←

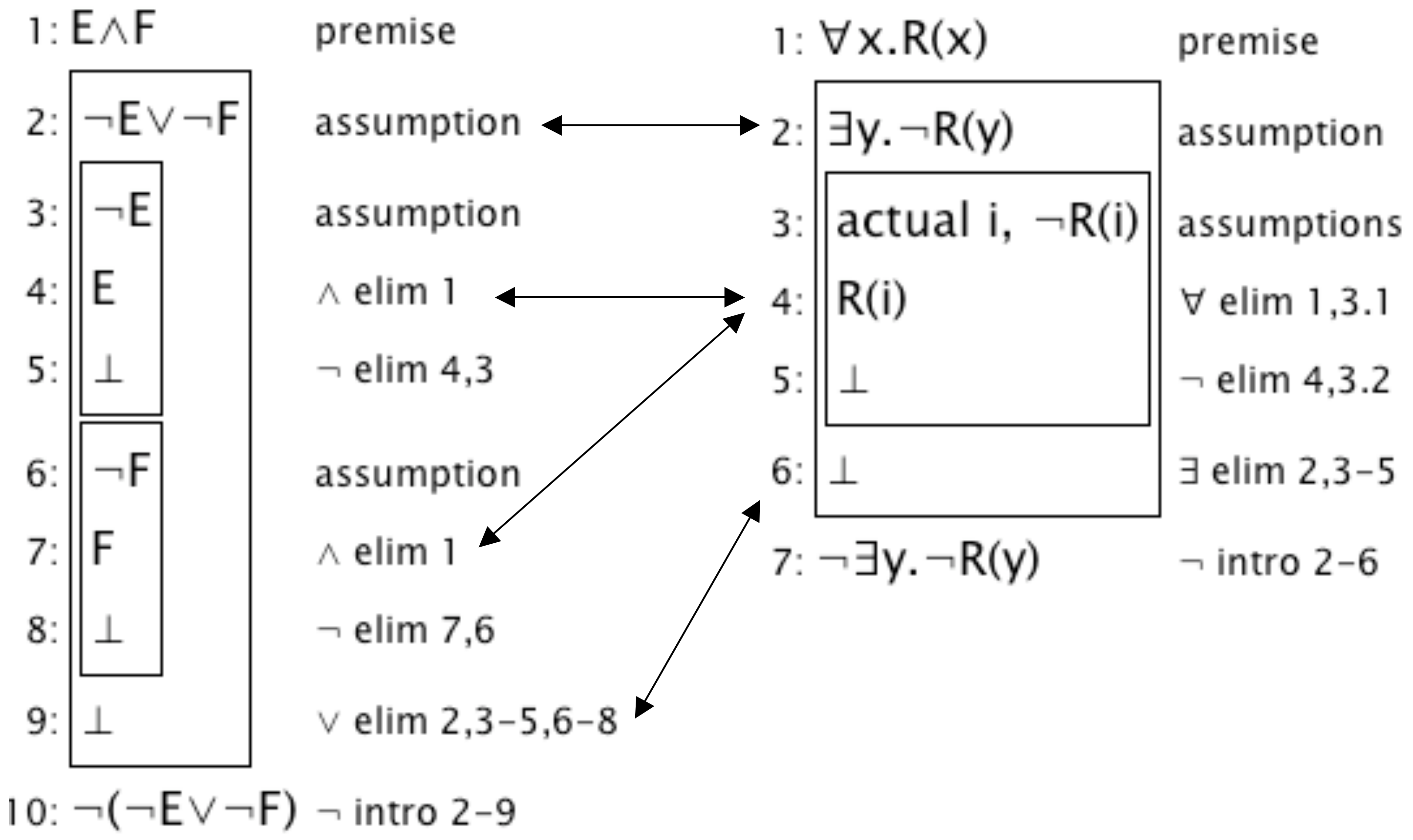


1:	$\forall x.R(x)$	premise
2:	$\exists y.\neg R(y)$	assumption
3:	actual i, $\neg R(i)$	assumptions
4:	$R(i)$	∀ elim 1,3.1
5:	$\bot$	¬ elim 4,3.2
6:	$\bot$	∃ elim 2,3–5
7:	$\neg\exists y.\neg R(y)$	¬ intro 2–6



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

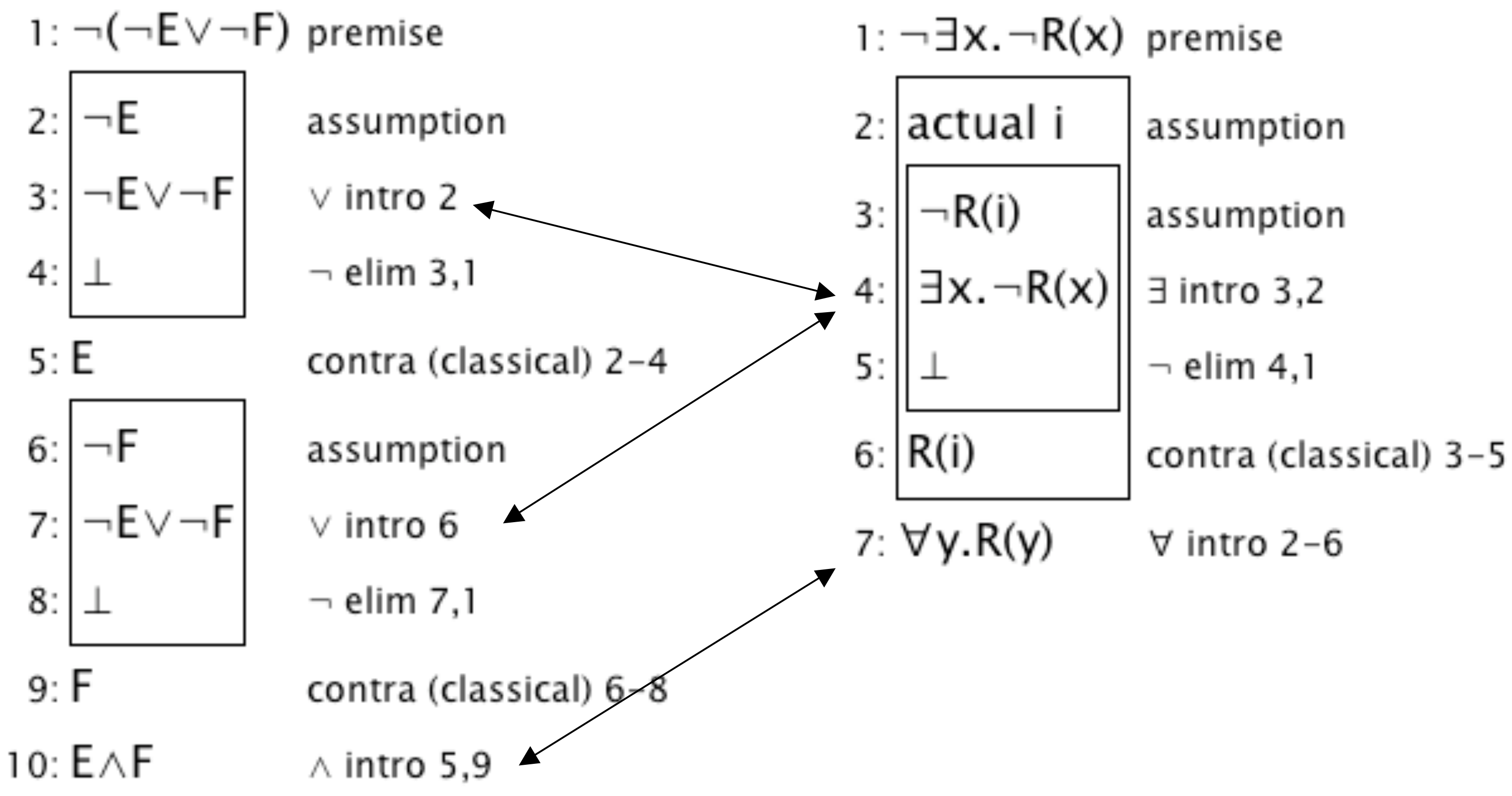
Note Rule Parallels



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

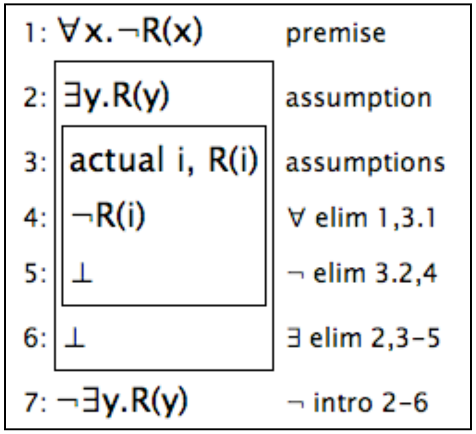
A Clue

- If it “resembles DNE” (double-negation elimination), it requires a classical proof.

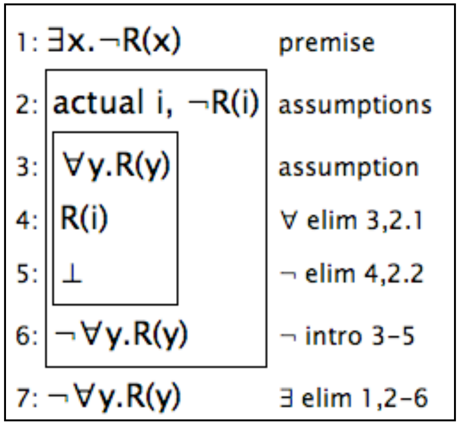
Classical	**Constructive**
$\neg(\neg p \vee \neg q) \rightarrow (p \wedge q)$	$(p \wedge q) \rightarrow \neg(\neg p \vee \neg q)$
$\neg(\neg p \wedge \neg q) \rightarrow (p \vee q)$	$(p \vee q) \rightarrow \neg(\neg p \wedge \neg q)$
$\neg(\exists x\ \neg P(x)) \rightarrow \forall x\ P(x)$	$\forall x\ P(x) \rightarrow \neg(\exists x\ \neg P(x))$
$\neg(\forall x\ \neg P(x)) \rightarrow \exists x\ P(x)$	$\exists x\ P(x) \rightarrow \neg(\forall x\ \neg P(x))$

But what about “hybrids” such as $(\exists x\ \neg P(x)) \rightarrow \neg \forall x\ P(x)$?

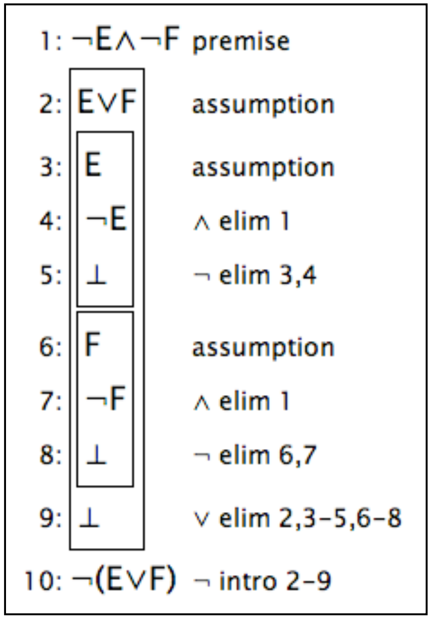
Hybrid Examples (Intuitionistic: use ¬Intro)



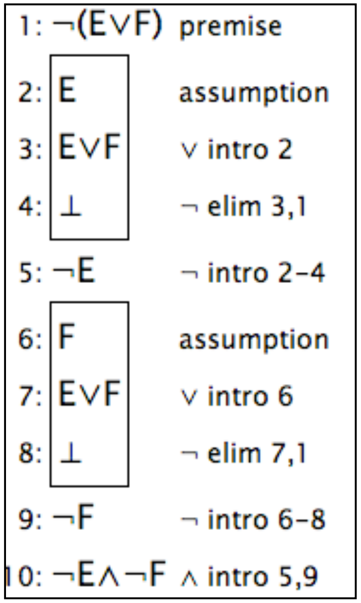
1:	∀x.¬R(x)	premise
2:	∃y.R(y)	assumption
3:	actual i, R(i)	assumptions
4:	¬R(i)	∀ elim 1,3.1
5:	⊥	¬ elim 3.2,4
6:	⊥	∃ elim 2,3-5
7:	¬∃y.R(y)	¬ intro 2-6



1:	∃x.¬R(x)	premise
2:	actual i, ¬R(i)	assumptions
3:	∀y.R(y)	assumption
4:	R(i)	∀ elim 3,2.1
5:	⊥	¬ elim 4,2.2
6:	¬∀y.R(y)	¬ intro 3-5
7:	¬∀y.R(y)	∃ elim 1,2-6



1:	¬E∧¬F	premise
2:	E∨F	assumption
3:	E	assumption
4:	¬E	∧ elim 1
5:	⊥	¬ elim 3,4
6:	F	assumption
7:	¬F	∧ elim 1
8:	⊥	¬ elim 6,7
9:	⊥	∨ elim 2,3-5,6-8
10:	¬(E∨F)	¬ intro 2-9



1:	¬(E∨F)	premise
2:	E	assumption
3:	E∨F	∨ intro 2
4:	⊥	¬ elim 3,1
5:	¬E	¬ intro 2-4
6:	F	assumption
7:	E∨F	∨ intro 6
8:	⊥	¬ elim 7,1
9:	¬F	¬ intro 6-8
10:	¬E∧¬F	∧ intro 5,9

ND Equality Rules

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

  The second rule is the familiar "equals substitute for equals" property.

Equality Properties ("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

Semantics

Interpretations of Formulas

- Interpretations are the way meanings are given to 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.
- “Theorems” are then derived from the axioms, and assuming that the axioms are correct about the structure, and the rules are sound, the theorems will also be correct.

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 $\mu(f)\colon \Delta^n \rightarrow \Delta$.
  - For each n-ary **predicate symbol** p, a function $\mu(p)\colon \Delta^n \rightarrow \{T, F\}$.
- 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.

Predicate Calculus “with Equality”

- The equality symbol always has a fixed interpretation as the identity predicate.

Assignments

- An interpretation does not fix the value of free variables in a formula.
- To specify these, we need the idea of an assignment.
- An **assignment** for an interpretation I = (Δ, μ) is a function α that determines a value

  $$\alpha[x] \in \Delta$$

  for each variable symbol x.

The **value of a term** under an interpretation and assignment.

- An interpretation I = (Δ, μ), together with an assignment α, determines a value $I_\alpha[t] \in \Delta$ of each **term** recursively:
  - If t is a constant symbol c, then $I_\alpha[t] = \mu(c)$, the assigned value in Δ.
  - If t is a variable symbol v, then $I_\alpha[t] = \alpha[v]$, the assigned value in α.
  - If t is $f(t_1, t_2, \ldots, t_n)$ where the t_i are terms, then

    $$I_\alpha[t] = \mu(f)(I_\alpha[t_1], I_\alpha[t_2], \ldots I_\alpha[t_n])$$

    recalling that $\mu(f)$ is the **function** I assigns the function symbol f.
- Notation: Stuff inside […] is ***syntactic***, not an expression in the meta-language. So it is not pre-evaluated as if a function.

The value of **atomic formulas** under an interpretation

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

    $$I_\alpha[E] = \mu(p)(I_\alpha[t_1], I_\alpha[t_2], \ldots I_\alpha[t_n]) \in \{T, F\}$$

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

    [using the **value of terms** definition presented earlier]

Example

- (Atomic) formula E is: q(f(f(x)), c), where q is a predicate symbol, f is a function symbol, and c is a constant.

- ***An*** interpretation I might assign
  - $\Delta = \{0, 1, 2\}$ domain
  - $\mu(c) = 0$ constant
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$ function
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$ predicate
    the set of pairs for which $\mu(q)$ is T
- An assignment for I might assign
  - $\alpha[x] = 2$
- Thus:

  - $I_\alpha[f(f(x)] = \mu(f)(I_\alpha[f(x)]) = \mu(f)(\mu(f)(I_\alpha[x])) = \mu(f)(\mu(f)(\alpha[x])) = 1$

  - $I(q(f(f(x)), c)) = \mu(q)(I[f(f(x)], \mu(c)) = \mu(q)(1, 0) = T$

Example

- (Atomic) formula E is: q(f(f(x)), c), where q is a predicate symbol, f is a function symbol, and c is a constant.
- Consider the same interpretation I:
  - $\Delta = \{0, 1, 2\}$ domain
  - $\mu(c) = 0$ constant
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$ function
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$ predicate
    the set of pairs for which $\mu(q)$ is T
- An ***different*** assignment for I might assign
  - $\alpha[x] = 0$ <<< different
- Thus:
  - $I_\alpha[f(f(x)] = \mu(f)(I_\alpha[f(x)]) = \mu(f)(\mu(f)(I_\alpha[x])) = \mu(f)(\mu(f)(\alpha[x])) = 2$ <<< different
  - $I(q(f(f(x)), c)) = \mu(q)(I[f(f(x)], \mu(c)) = \mu(q)(2, 0) = F$ <<< different

The value of **composite formulas, connected using propositional connectives** under an interpretation

- An interpretation $I = (\Delta, \mu)$, together with an assignment α, determines a value
  $I_\alpha[t] \in \{T, F\}$ of each formula E **recursively**:
  - If E is E_1 o E_2, where o is one of $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, then
    $I_\alpha[E] = h_o(I_\alpha[E_1], I_\alpha[E_2])$
  - If E is $\neg E_1$, then $I_\alpha[E] = h_\neg(I_\alpha[E_1])$
- These are the same functions $\mathbf{h_o}$, $\mathbf{h_\neg}$ as used to define the semantics of propositional logic.
- The values $I_\alpha[t]$ are analogous to **valuations** in the propositional case.

The value of **quantified formulas**, under an interpretation

- An interpretation $I = (\Delta, \mu)$, together with an assignment α, determines a value $I_\alpha[t] \in \{T, F\}$ of each formula E **recursively**:
  - If E is $\forall x\ B$,

    then $I_\alpha[E] = T$

    iff $I_\beta[B] = T$ for ***every*** assignment β that agrees with α on free[E].

  - If E is $\exists x\ B$,

    then $I_\alpha[E] = T$
  - iff $I_\beta[B] = T$ for ***some*** assignment β that agrees with α on free[E].
- Two assignments **agree on a set** of variables if they assign the same values to every variable in the set.

Example

- Formula E is: $\exists x\ q(f(f(x)), y)$ free[E] = $\{y\}$
- An interpretation I is
  - $\Delta = \{0, 1, 2\}$
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$
    the set of pairs for which $\mu(q)$ is T
- Consider an assignment α having $\alpha[y] = 0$.
- Thus:
  - $I_\alpha[\exists x\ q(f(f(x)), y)] = T$ iff for *some* assignment β agreeing with α on $\{y\}$, i.e. $\beta[y] = 0$ such that
    $$I_\beta[q(f(f(x)), y)] = T.$$
  - **There is** such a β, namely one where $\beta[x] = 2$.
  - So $I[\exists x\ q(f(f(x)), y)] = T$.

Example

- Formula E is: $\forall x\ q(f(f(x)), y)$ free[E] = $\{y\}$
- An interpretation I is
  - $\Delta = \{0, 1, 2\}$
  - $\mu(f) = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 0\}$
  - $\mu(q) = \{(0, 2), (1, 0), (2, 1)\}$
    the set of pairs for which $\mu(q)$ is T
- Again consider an assignment α having $\alpha[y] = 0$.
- Thus:
  - $I_\alpha[\forall x\ q(f(f(x)), y)] = T$ iff for ***every*** assignment β agreeing with α on $\{y\}$, i.e. $\beta[y] = 0$ such that

    $$I_\beta[q(f(f(x)), y)] = T.$$

  - **But there is** a β, namely one where $\beta[x] = 0$, such that $I_\beta[q(f(f(x)), y)] = F$.
  - So $I_\alpha[\forall x\ q(f(f(x)), y)] =$ **F**.

Moot Assignments

- If a formula has **no free variables**, the value for a given interpretation is the same regardless of assignment.

- All assignments agree on the set of free variables, which is the empty set.
- In this case we *drop* the subscript from I_α as the specific assignment doesn't matter.

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
- There are no free variables.
- Thus:
  - $I[\forall x\ q(f(f(x)), c)] = T$ iff for *all* β, $I_\beta[q(f(f(x)), c)] = T$.
  - There is a β for which $I_\beta[q(f(f(x)), c)] = \mathbf{F}$, namely one where $\beta[x] = 0$.
  - So $I[\forall x\ q(f(f(x)), c)] = F$.

Satisfaction

- An interpretation, together with an assignment, I_α, ***satisfies*** a formula E iff $I_\alpha[E] = T$.

  We also say that E ***is valid under*** I_α.
  We also say that I_α is a ***model*** for E.

- For formulas with no free variables, we can drop the subscript.

- A formula is ***satisfiable*** iff there is an I_α that satisfies it, otherwise it is ***unsatisfiable***.

Validity

- A formula E is **valid for an interpretation** I provided that for every assignment α, $I_\alpha[E] = T$.
- A formula is (universally) **valid** if it is valid for every interpretation.

Formalizing Semantic Entailment |=

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

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

  means:

  For every interpretation with assignment I_α that satisfies each of the formulas $\varphi_1, \ldots, \varphi_n$

  I_α also satisfies ψ.

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

  Every model I_α for Γ is also a model for ψ.

- $\models \psi$ is the special case where Γ is empty.

|= in predicate calculus vs. propositional

- The predicate version of |= ψ is a **very broad** statement:
  - The **set** of applicable interpretations is often **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 |= ψ for predicate calculus in the way there is for the propositional calculus.

Examples of Valid and Invalid Formulas

- Check that most of the predicate logic formulas proved so far are valid.

- Examples of **invalid** formulas include:
  - $(\exists x\ P(x)) \wedge (\exists x\ Q(x)) \rightarrow \exists x\ (P(x) \wedge Q(x))$
  - $\forall x\ (P(x) \vee Q(x)) \rightarrow (\forall x\ P(x)) \vee (\forall x\ Q(x))$

Invalid Formulas Valid Under Specific Interpretations

- Consider any algebraic structure, such as a group.

- There are formulas that characterize the structure, such as the associative law, and so are called the **axioms** for the structure.

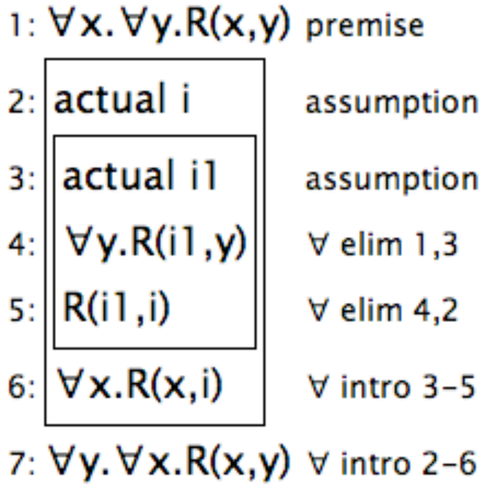
- But those formulas are not true for (e.g. non-group) interpretations in general.

Showing a Formula Valid from “First Principles” using Interpretations

- Argue that the value of the formula is T **regardless of interpretation** (and assignment, if there are free variables).
- Example: $\forall x\ \forall y\ P(x, y) \leftrightarrow \forall y\ \forall x\ P(x, y)$
- Suppose I is an interpretation such that
  $I[\forall x\ \forall y\ P(x, y)] = T$ (1)
- We must show that
  $I[\forall y\ \forall x\ P(x, y)] = T$ (2) and conversely.
- (1) is equivalent to (1’): for every assignment β, $I_\beta[\forall y\ P(x, y)] = T$.
- (1’) is equivalent to (1’’): for every assignment γ such that $\gamma[x] = \beta[x]$, $I_\gamma[P(x, y)] = T$.
- But β was **arbitrary**, so (1) is equivalent to:
  (*) for every assignment γ, ***without qualification***, $I_\gamma[P(x, y)] = T$.
- By a similar chain of reasoning from (2), we also arrive at (*).
- Hence (1) and (2) are equivalent.

The Complexity of a First-Principles Argument

- The Complexity of a First-Principles Argument is one of the motivating factors for using a logical calculus: It makes fine-detailed thinking unnecessary, replacing it with symbol manipulation.



1:	$\forall x.\forall y.R(x,y)$	premise
2:	actual i	assumption
3:	actual i1	assumption
4:	$\forall y.R(i1,y)$	$\forall$ elim 1,3
5:	$R(i1,i)$	$\forall$ elim 4,2
6:	$\forall x.R(x,i)$	$\forall$ intro 3–5
7:	$\forall y.\forall x.R(x,y)$	$\forall$ intro 2–6

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:
    - $\Delta = \{1, 2\}$
    - $\mu(A) = \{2\}$
    - $\mu(B) = \{2\}$

Interpretation-Specific Proofs

- To make ND proofs for specific interpretations, add premises that are true only for those interpretations.
- Example: To prove things that are true for interpretations that are groups, include group axioms as premises.

Example: Proof about Fields

- Field Axioms:
  - F1: $\forall x\ \forall y\ \forall z\ ((x+y)+z) = (x+(y+z))$
  - F2: $\forall x\ (x+0) = x$
  - F3: $\forall x \exists y\ (x+y) = 0$
  - F4: $\forall x\ \forall y\ (x+y) = (y+x)$
  - F5: $\forall x\ \forall y\ \forall z\ ((x*y)*z) = (x*(y*z))$
  - F6: $\forall x\ (x*1) = x$
  - F7: $\forall x\ \neg(x=0) \rightarrow \exists z\ (x*z) = 1$
  - F8: $\forall x\ \forall y\ (x*y) = (y*x)$
  - F9: $\forall x\ \forall y\ \forall z\ (x*(y+z)) = (x*y)+(x*z)$
  - F10: $\neg(1=0)$
- Prove:
  - $\forall x\ \forall y\ \forall z\ ((y+z)*x) = (y*x)+(z*x)$

Proof Outline

1.	x_0, y_0, z_0	Fresh variables
2.	$((y_0+z_0)*x_0) = (x_0*(y_0+z_0))$	F8, ∀E
3.	$(x_0*(y_0+z_0)) = (x_0*y_0)+(x_0*z_0)$	F9, ∀E
4.	$(y_0+z_0)*x_0 = (x_0*y_0)+(x_0*z_0)$	2, 3, Equality
5.	$(x_0*y_0) = (y_0*x_0)$	F8, ∀E
6.	$(y_0+z_0)*x_0 = (y_0*x_0)+(x_0*z_0)$	4, 5, Equality
7.	$(x_0*z_0) = (z_0*x_0)$	F8, ∀E
8.	$(y_0+z_0)*x_0 = (y_0*x_0)+(z_0*x_0)$	6, 7, Equality
9.	$\forall x\ \forall y\ \forall z\ ((y+z)*x) = (y*x)+(z*x)$	2-8, ∀I

F8: $\forall x\ \forall y\ (x*y) = (y*x)$
F9: $\forall x\ \forall y\ \forall z\ (x*(y+z)) = (x*y)+(x*z)$

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 $\Gamma \vdash \psi$

Completeness Theorem

- Natural deduction for predicate logic is both sound and complete.
- This was first shown (for Hilbert-Ackermann systems) by Gödel in 1929.
- cf. http://en.wikipedia.org/wiki/Original_proof_of_G%C3%B6del%27s_completeness_theorem
- Leon Henkin published an easier proof in 1949, which has become the standard approach.
- Henkin, Leon. 1949. "The Completeness of the First-Order Functional Calculus", Journal of Symbolic Logic. 14: 159–166.
- There is a proof for natural deduction in van Dalen’s book, “Logic and Structure”.

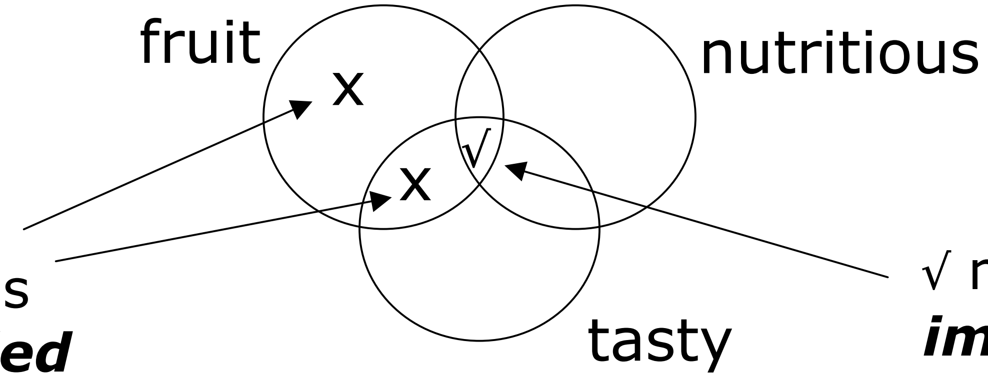
Syllogisms (WP)

- 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.	$\exists x\ (S(x) \wedge P(x))$
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.

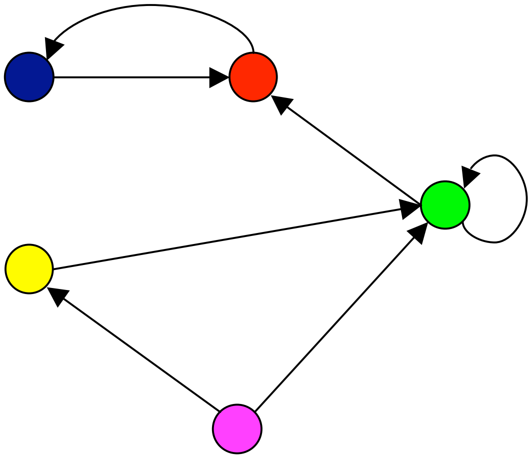
Example: Translate this syllogism, then try to prove it.

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

Fun with Relations

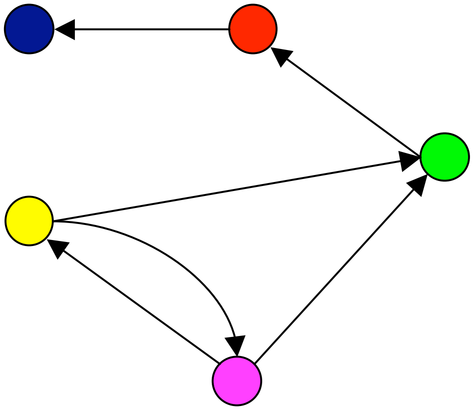
- A 2-ary predicate represents a binary relation, i.e. a set of pairs of domain elements.
- Various properties of relations can be expressed using predicate logic formulas.
- In the following, what formula characterizes each relation represented by predicate L (sometimes using “loves” for analogy), and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

Everybody loves somebody.

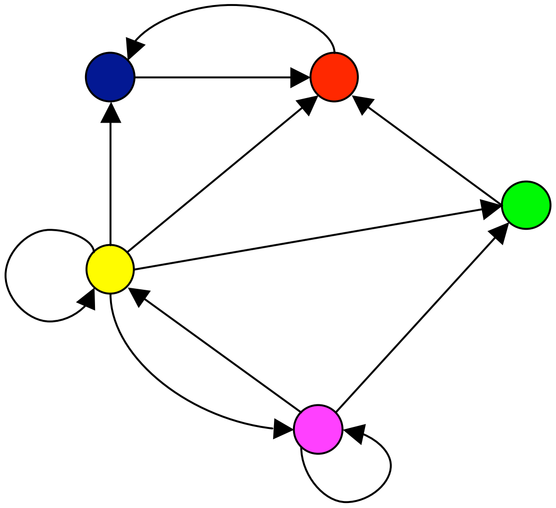


$\forall x \, \exists y \; L(x, y)$

Everybody is loved by somebody.

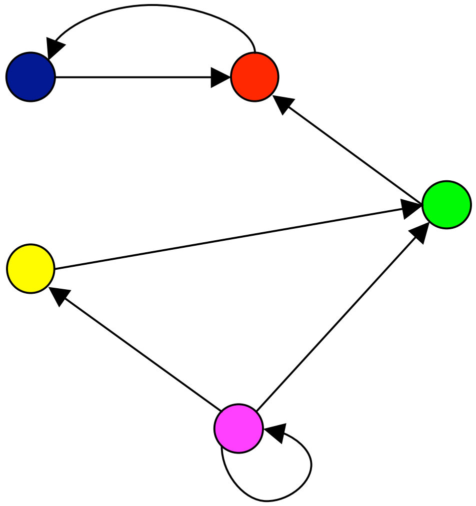


Somebody loves everybody.

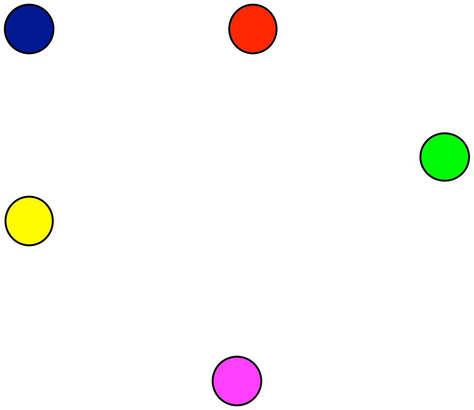


"Pollyanna"

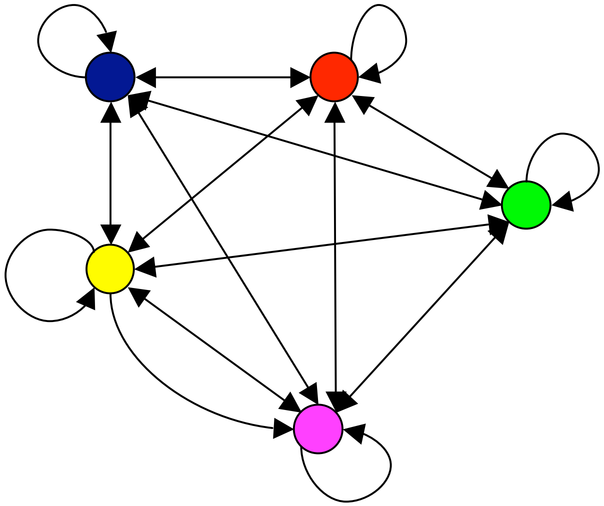
Nobody loves everybody.



Nobody loves somebody.

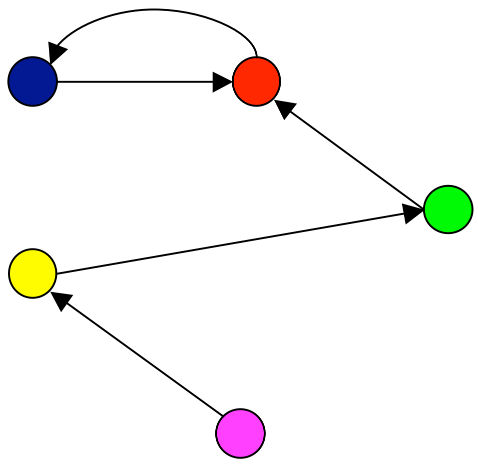


Everybody loves everybody.

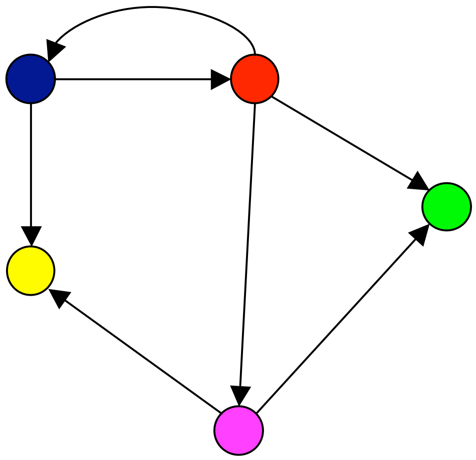


“Commune”

Everybody loves exactly one.

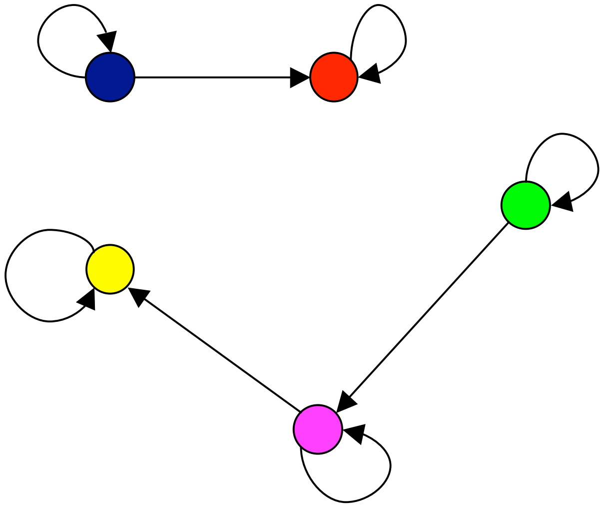


Nobody loves exactly one.



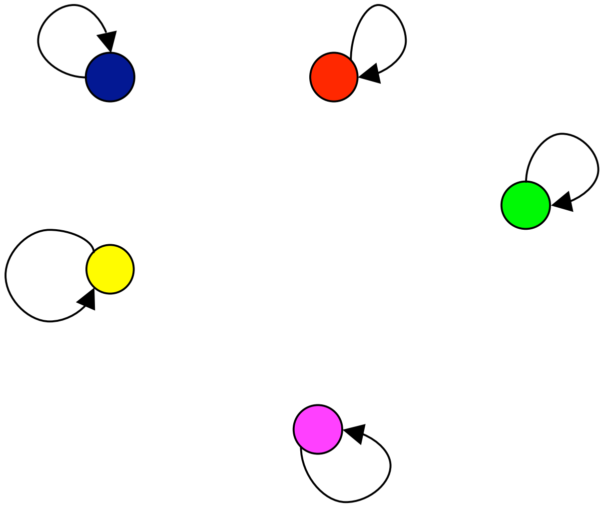
Everybody loves him/herself.

“Reflexive”

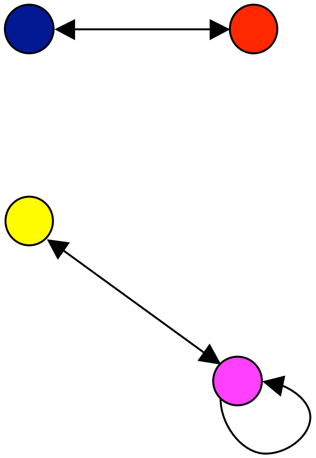


Everybody loves her/himself and only her/himself.

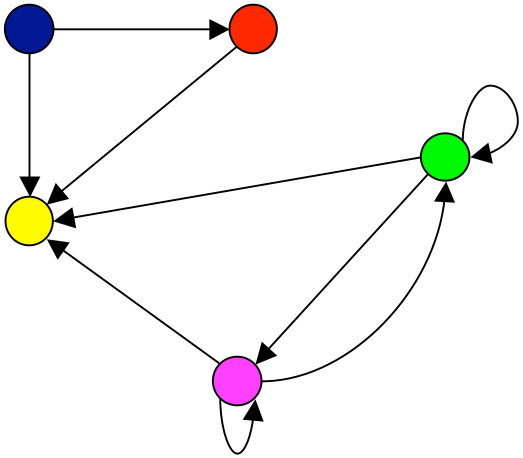
“Isolationist”



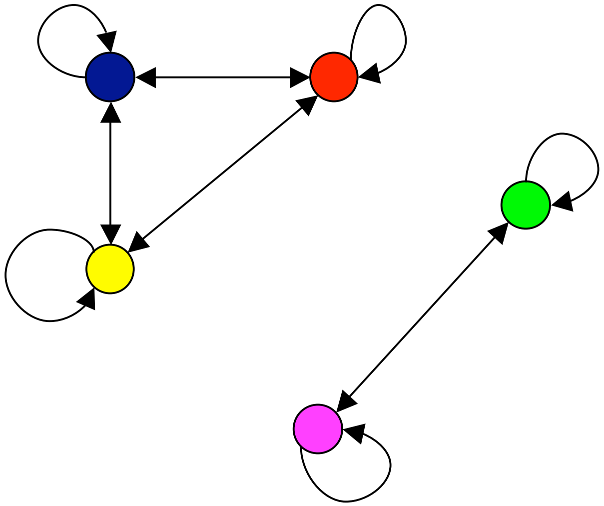
L is symmetric.



L is transitive.

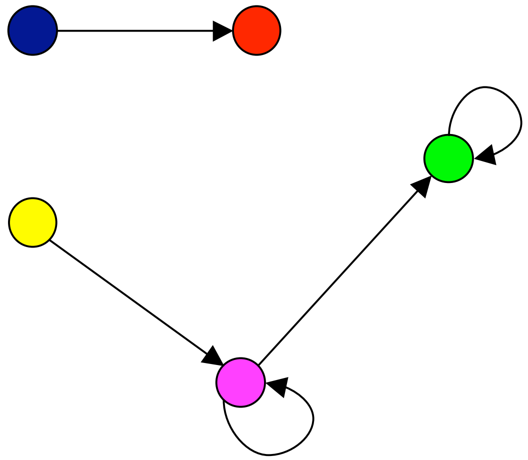


L is an equivalence relation

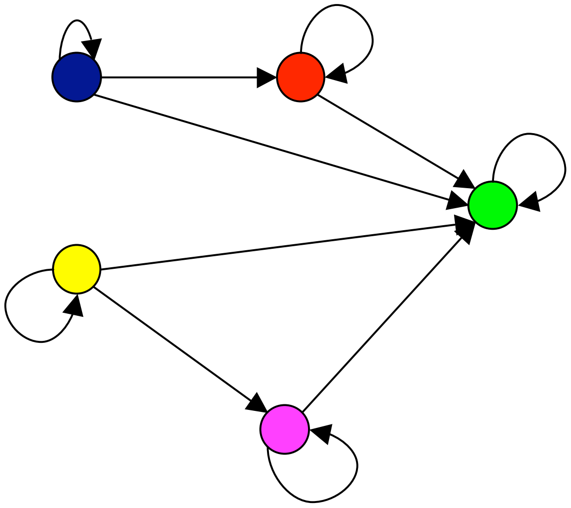


Reflexive,
symmetric,
transitive

L is antisymmetric.

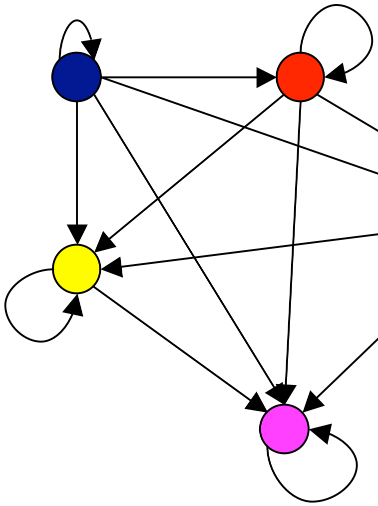


L is a partial order (“poset”).



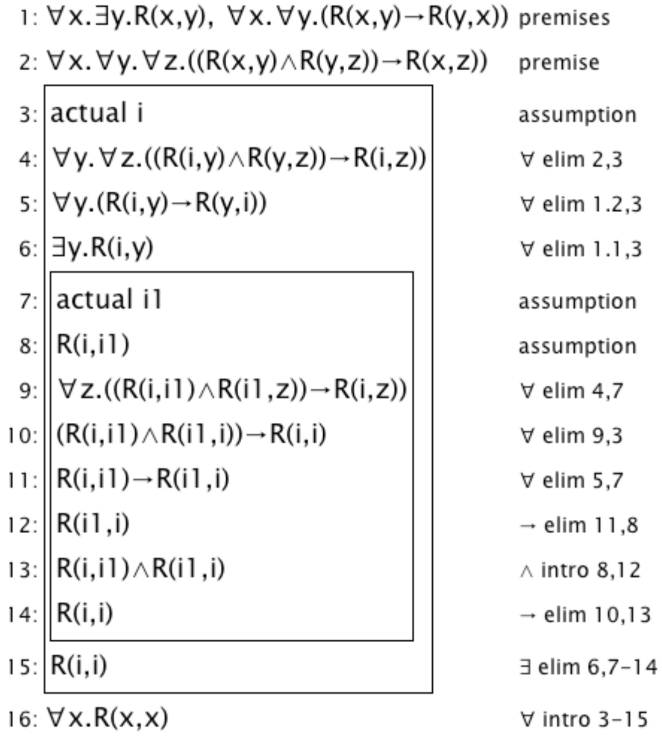
Reflexive,
Antisymmetric,
Transitive

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:	$\forall y.\forall z.((R(i,y)\wedge R(y,z))\rightarrow R(i,z))$	∀ elim 2,3
5:	$\forall y.(R(i,y)\rightarrow R(y,i))$	∀ elim 1.2,3
6:	$\exists y.R(i,y)$	∀ elim 1.1,3
7:	actual i1	assumption
8:	$R(i,i1)$	assumption
9:	$\forall z.((R(i,i1)\wedge R(i1,z))\rightarrow R(i,z))$	∀ elim 4,7
10:	$(R(i,i1)\wedge R(i1,i))\rightarrow R(i,i)$	∀ elim 9,3
11:	$R(i,i1)\rightarrow R(i1,i)$	∀ elim 5,7
12:	$R(i1,i)$	→ elim 11,8
13:	$R(i,i1)\wedge R(i1,i)$	∧ intro 8,12
14:	$R(i,i)$	→ elim 10,13
15:	$R(i,i)$	∃ elim 6,7-14
16:	$\forall x.R(x,x)$	∀ intro 3-15

How to do without function symbols (use only predicate 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 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