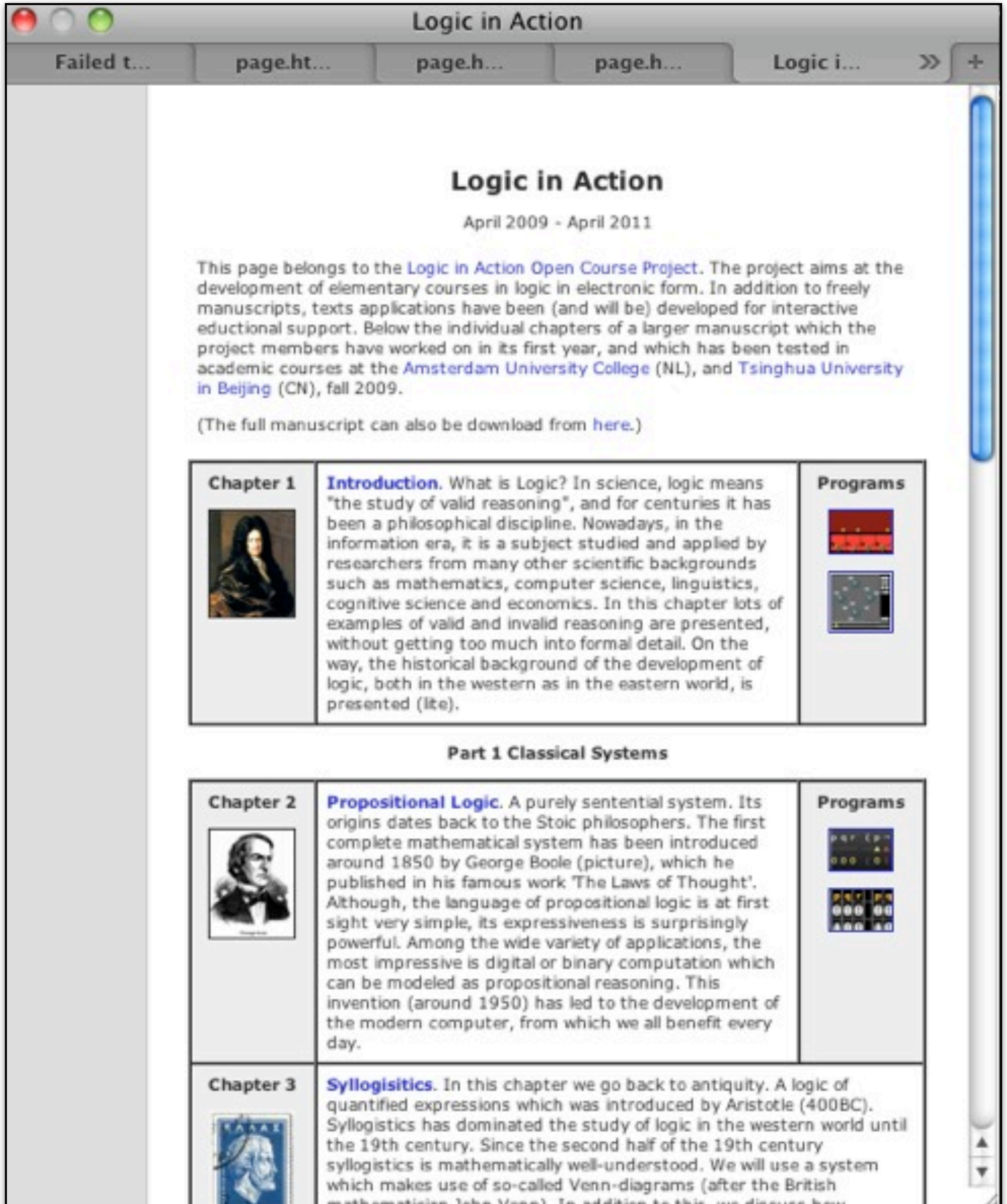
Logic in Action

Johan van Benthem, Jan van Eijck and Jan Jaspars

23rd ESSLLI, Ljubljana, Aug 1-5 2011



Logic in Action

Failed t... | page.ht... | page.h... | page.h... | Logic i...

Logic in Action

April 2009 - April 2011

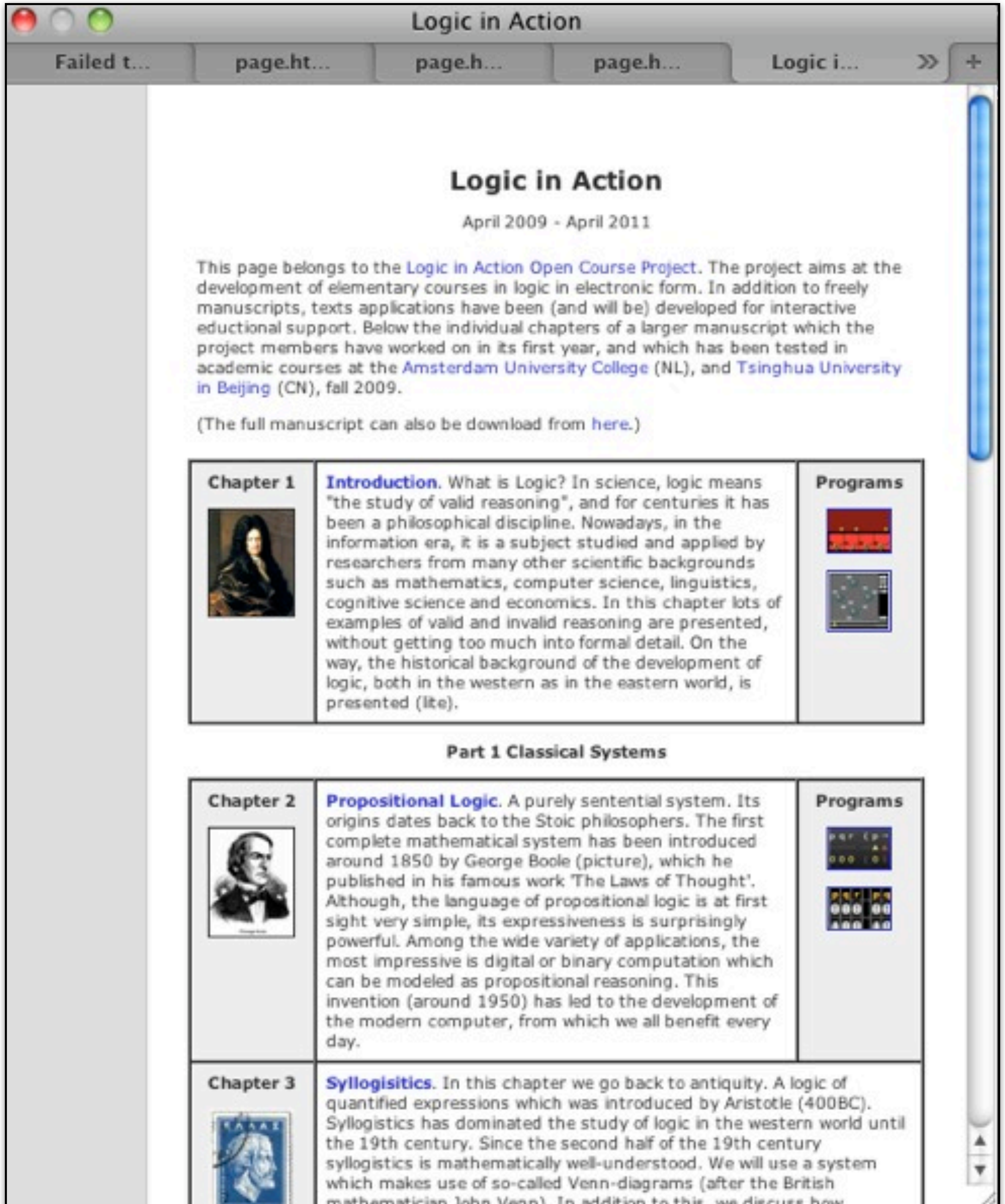
This page belongs to the Logic in Action Open Course Project. The project aims at the development of elementary courses in logic in electronic form. In addition to freely manuscripts, texts applications have been (and will be) developed for interactive eductional support. Below the individual chapters of a larger manuscript which the project members have worked on in its first year, and which has been tested in academic courses at the Amsterdam University College (NL), and Tsinghua University in Beijing (CN), fall 2009.

(The full manuscript can also be download from here.)

Chapter 1	**Introduction**. What is Logic? In science, logic means "the study of valid reasoning", and for centuries it has been a philosophical discipline. Nowadays, in the information era, it is a subject studied and applied by researchers from many other scientific backgrounds such as mathematics, computer science, linguistics, cognitive science and economics. In this chapter lots of examples of valid and invalid reasoning are presented, without getting too much into formal detail. On the way, the historical background of the development of logic, both in the western as in the eastern world, is presented (lite).	Programs

Part 1 Classical Systems

Chapter 2	**Propositional Logic**. A purely sentential system. Its origins dates back to the Stoic philosophers. The first complete mathematical system has been introduced around 1850 by George Boole (picture), which he published in his famous work 'The Laws of Thought'. Although, the language of propositional logic is at first sight very simple, its expressiveness is surprisingly powerful. Among the wide variety of applications, the most impressive is digital or binary computation which can be modeled as propositional reasoning. This invention (around 1950) has led to the development of the modern computer, from which we all benefit every day.	Programs
Chapter 3	**Syllogisitics**. In this chapter we go back to antiquity. A logic of quantified expressions which was introduced by Aristotle (400BC). Syllogistics has dominated the study of logic in the western world until the 19th century. Since the second half of the 19th century syllogistics is mathematically well-understood. We will use a system which makes use of so-called Venn-diagrams (after the British mathematician John Venn). In addition to this, we discuss how	



logicinaction.org

All politicians are rich
No student is rich

No student is politician

All politicians are rich
No student is politician

No student is rich

All ∞℘ŧb are ♬

No ☀伍 is ♬

No ☀伍 is ∞℘ŧb

All ∞℘ŧb are ♬

No ☀伍 is ∞℘ŧb

No ☀伍 is ♬

All ∞℘ƀ are ♬

No ☀伍 is ♬

No ☀伍 is ∞℘ƀ

All ∞℘ŧb are ♬

No ☀伍 is ♬

No ☀伍 is ∞℘ŧb

Calculemus

Calculemus

dx/dy $\int$

Calculemus

Calculemus

O I

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31
100000	32
&c.	&c.

bres entiers au-dessous du double du plus haut degré. Car icy, c'est comme si on disoit, par exemple, que 111 ou 7 est la somme de quatre, de deux & d'un.

100	4
10	2
1	1
111	7

Et que 1101 ou 13 est la somme de huit, quatre & un. Cette proprieté sert aux Essayeurs pour peser toutes sortes de masses avec peu de poids, & pourroit servir dans les monnoyes pour donner plusieurs valeurs avec peu de pieces.

1000	8
100	4
1	1
1101	13

Cette expression des Nombres étant établie, sert à faire tres-facilement toutes sortes d'operations.

Pour l'*Addition* par exemple. ☽

110	6	101	5	1110	14
111	7	1011	11	10001	17
1101	13	10000	16	11111	31

Pour la *Soustraction.*

1101	13	10000	16	11111	31
111	7	1011	11	10001	17
110	6	101	5	1110	14

Pour la *Multiplication.* ⊙

11	3	101	5	101	5
11	3	11	3	101	5
11		101		101	
11		101		1010	
1001	9	1111	15	11001	25

Pour la *Division.*

15	1111	101	5
3	11		

Et toutes ces operations sont si aisées, qu'on n'a jamais besoin de rien essayer ni deviner, comme il faut faire dans la division ordinaire. On n'a point besoin non-plus de rien apprendre par cœur icy, comme il faut faire dans le calcul ordinaire, où il faut sçavoir, par exemple, que 6 & 7 pris ensemble font 13; & que 5 multiplié par 3 donne 15, suivant la Table *d'une fois un est un*, qu'on appelle Pythagorique. Mais icy tout cela se trouve & se prouve de source, comme l'on voit dans les exemples precedens sous les signes ☽ & ⊙.

"... I'm not good at mathematics, but I don't bother. Numbers, triangles, functions .. it's just *not* my world. But, yes, logic I *have to* be good at it. It's about my own thoughts!"

Student of the Amsterdam University College.

"... I'm not good at mathematics, but I don't bother. Numbers, triangles, functions .. it's just *not* my world. But, yes, logic I *have to* be good at it. It's about my own thoughts!"

Student of the Amsterdam University College.

All men
are mortal

All men
are mortal

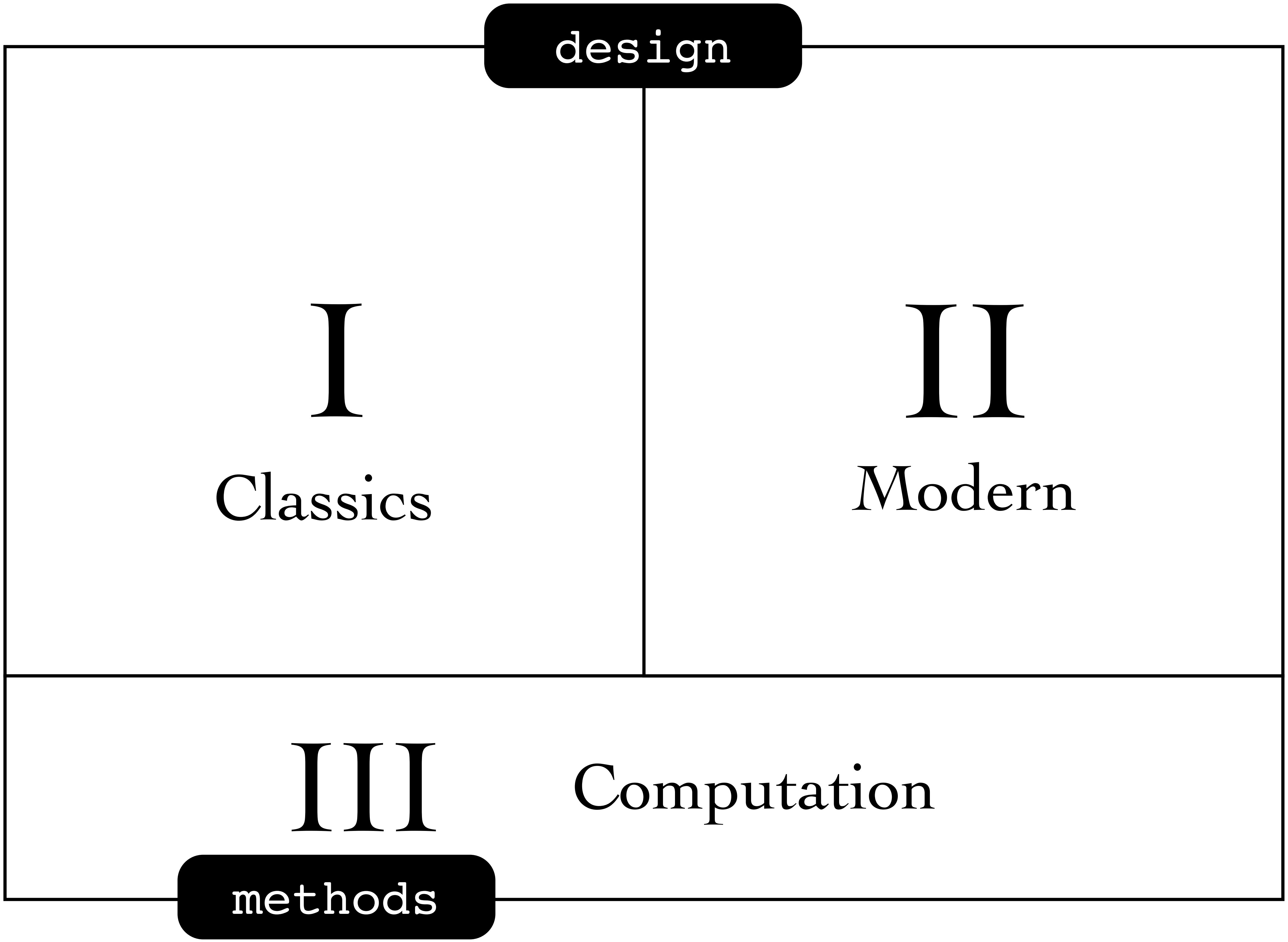
Socrates is a
man

Socrates is mortal

All men are mortal

Socrates is a man

Socrates is mortal


design
I
Classics
II
Modern
III
Computation
methods

design

I

- ✤Propositional Logic
- ✤Syllogistics (& sets)
- ✤First Order Logic

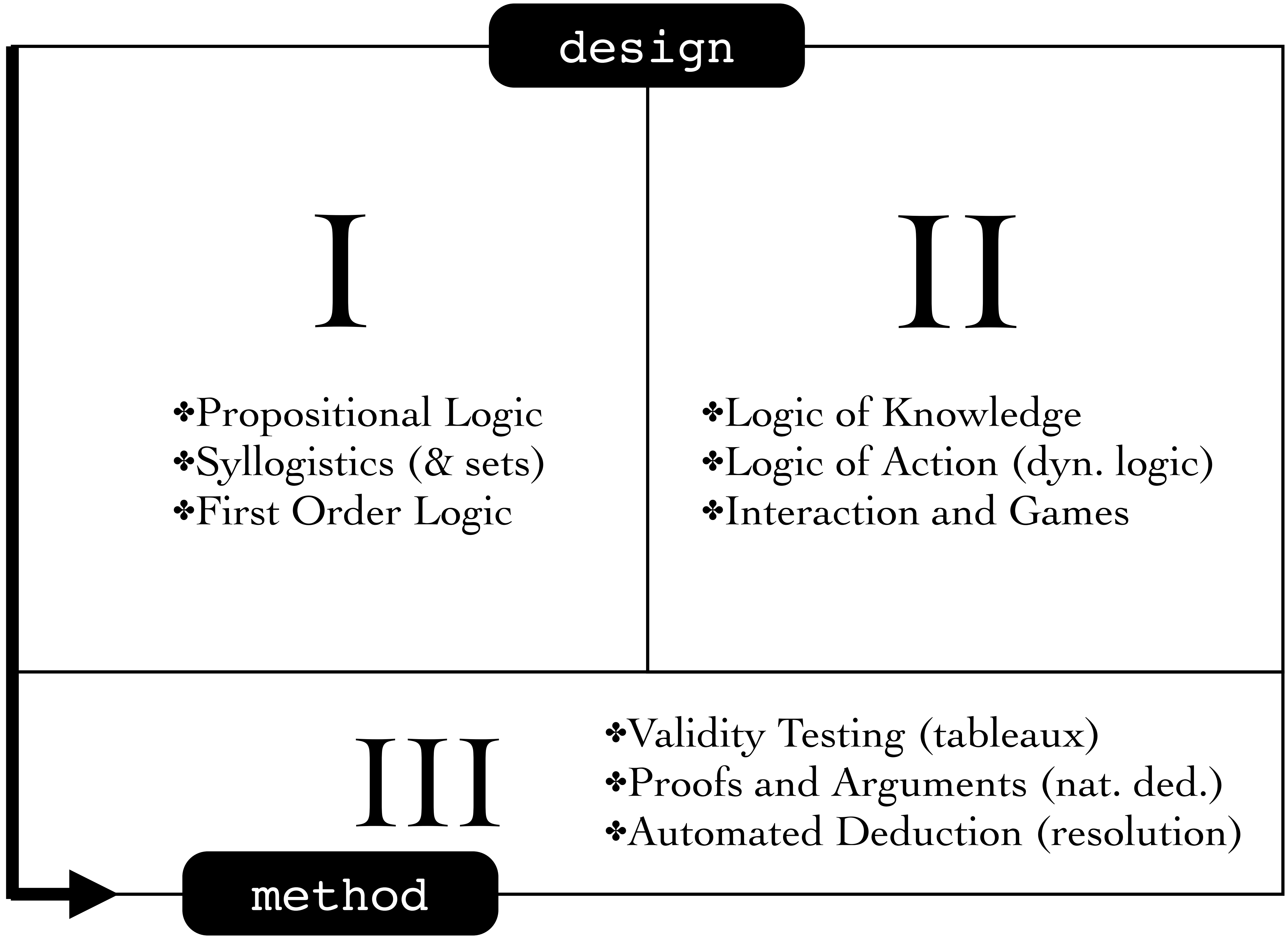
II

- ✤Logic of Knowledge
- ✤Logic of Action (dyn. logic)
- ✤Interaction and Games

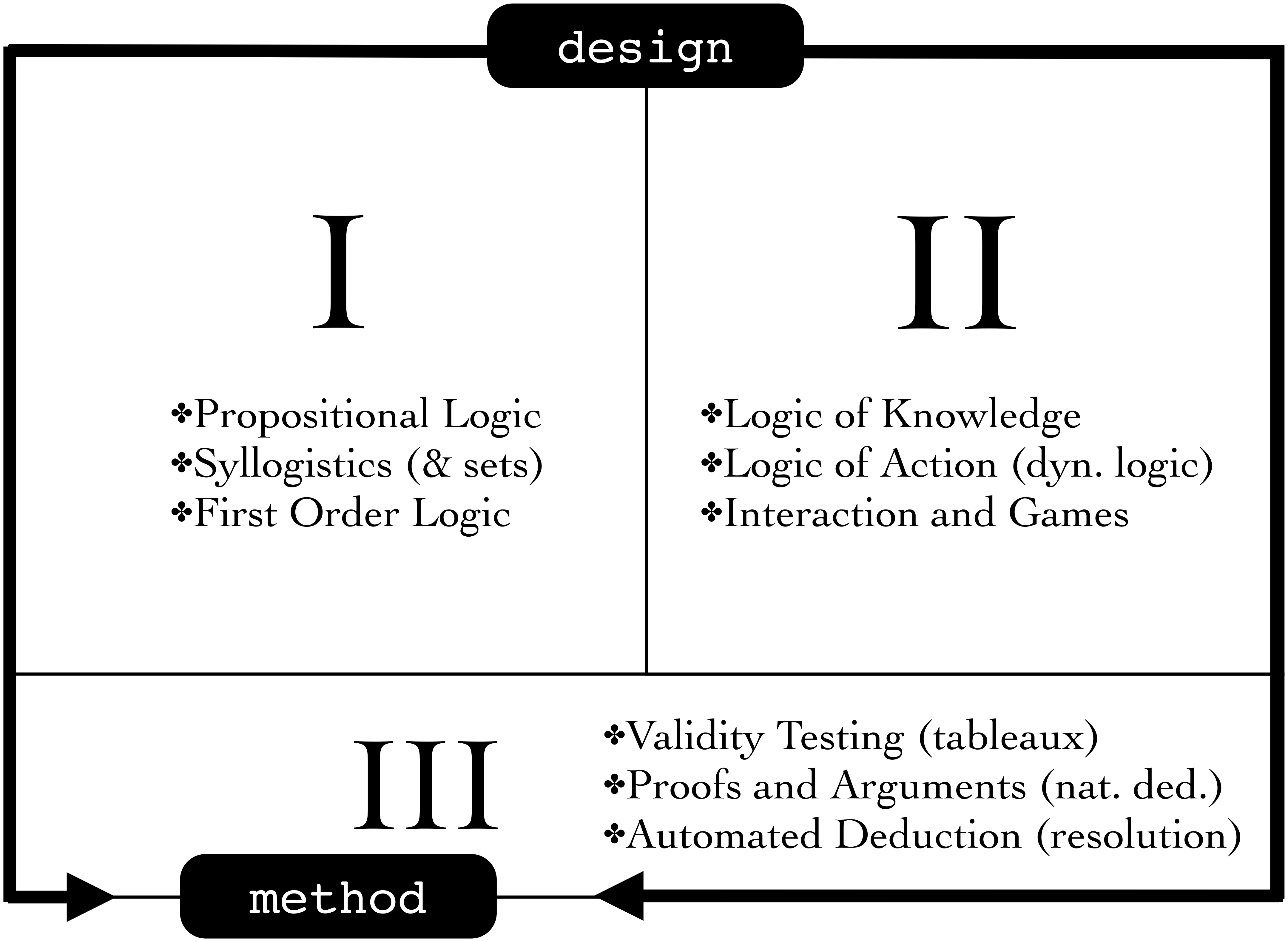
III

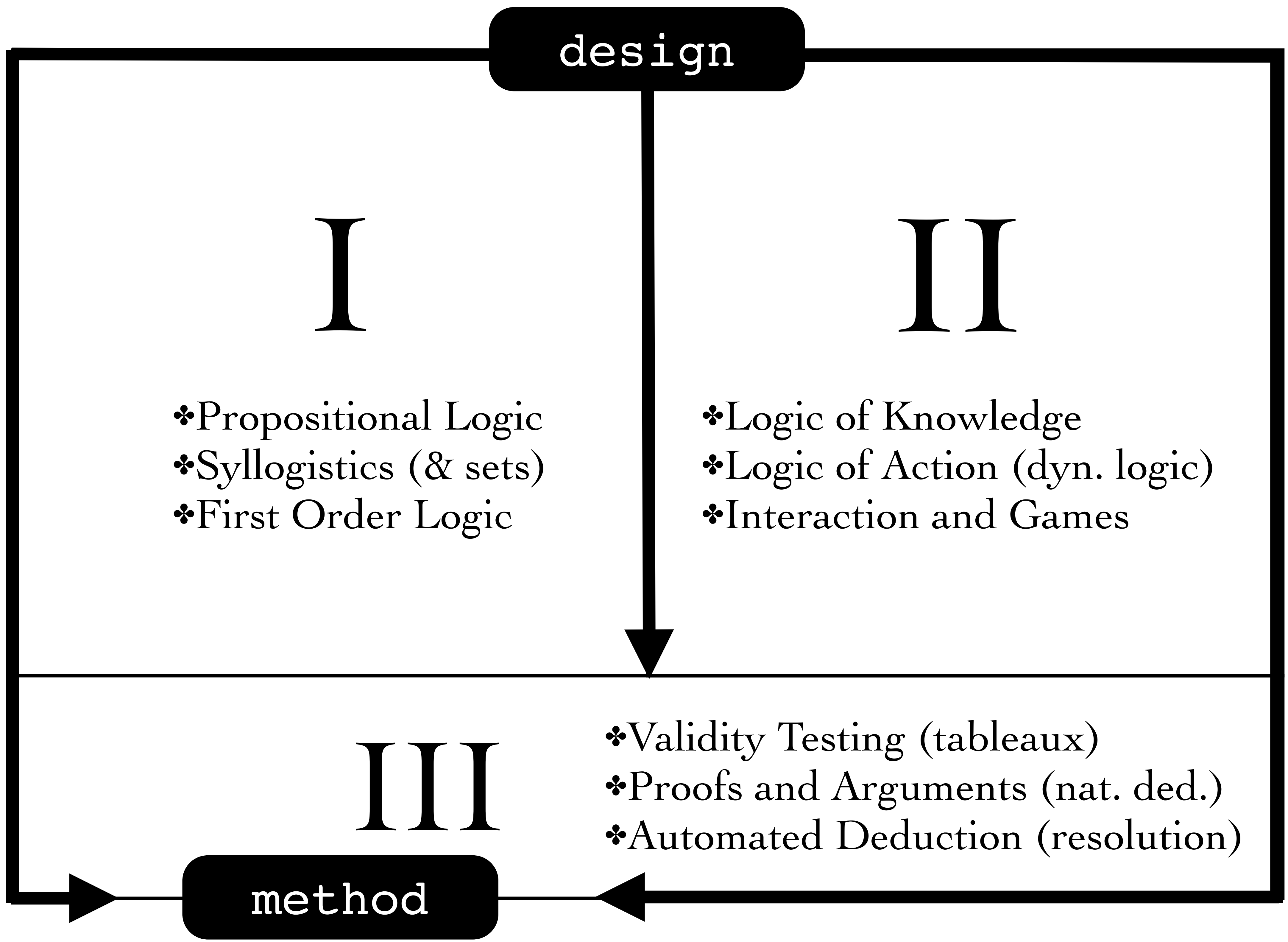
- ✤Validity Testing (tableaux)
- ✤Proofs and Arguments (nat. ded.)
- ✤Automated Deduction (resolution)

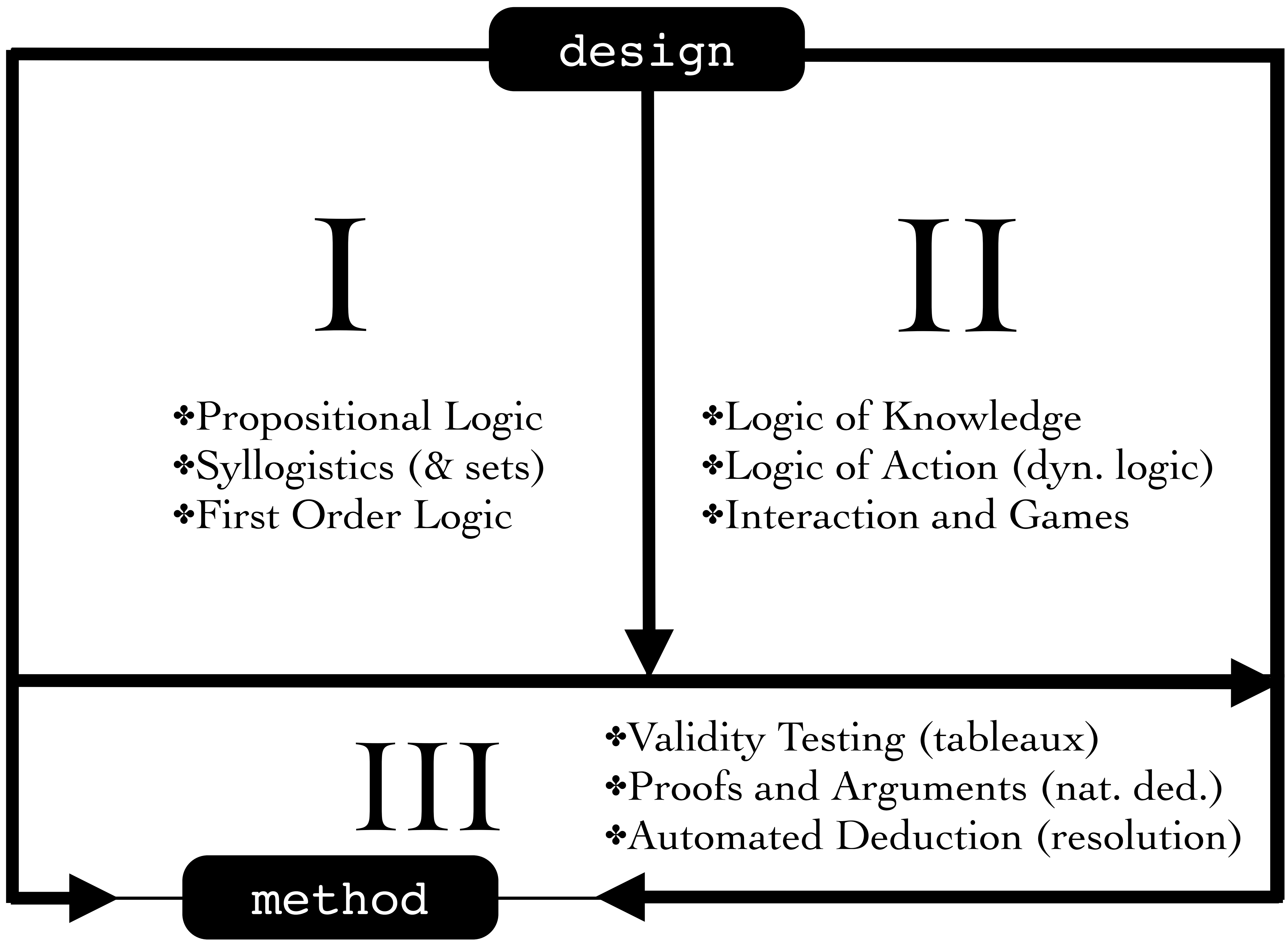
method



design
I
✤Propositional Logic
✤Syllogistics (& sets)
✤First Order Logic
II
✤Logic of Knowledge
✤Logic of Action (dyn. logic)
✤Interaction and Games
III
✤Validity Testing (tableaux)
✤Proofs and Arguments (nat. ded.)
✤Automated Deduction (resolution)
method


design
I
✤Propositional Logic
✤Syllogistics (& sets)
✤First Order Logic
II
✤Logic of Knowledge
✤Logic of Action (dyn. logic)
✤Interaction and Games
III
✤Validity Testing (tableaux)
✤Proofs and Arguments (nat. ded.)
✤Automated Deduction (resolution)
method


design
I
✤Propositional Logic
✤Syllogistics (& sets)
✤First Order Logic
II
✤Logic of Knowledge
✤Logic of Action (dyn. logic)
✤Interaction and Games
III
✤Validity Testing (tableaux)
✤Proofs and Arguments (nat. ded.)
✤Automated Deduction (resolution)
method


design
I
✤Propositional Logic
✤Syllogistics (& sets)
✤First Order Logic
II
✤Logic of Knowledge
✤Logic of Action (dyn. logic)
✤Interaction and Games
III
✤Validity Testing (tableaux)
✤Proofs and Arguments (nat. ded.)
✤Automated Deduction (resolution)
method

1. At least one of them is guilty.
2. Not all of them are guilty.
3. Mrs White is guilty only if Colonel Mustard helped her (is guilty too).
4. If Miss Scarlet is innocent then so is Colonel Mustard.

1. $w \vee s \vee m$
2. $\neg(w \wedge s \wedge m)$
3. $w \rightarrow m$
4. $\neg s \rightarrow \neg m$

innocent	innocent	innocent
innocent	innocent	guilty
innocent	guilty	innocent
innocent	guilty	guilty
guilty	innocent	innocent
guilty	innocent	guilty
guilty	guilty	innocent
guilty	guilty	guilty


Mrs White
Mrs White
Miss Scarlet
Miss Scarlet
Colonel Mustard
Colonel Mustard

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$\neg(w \wedge s \wedge m)$

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

$w \vee s \vee m$

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

$w \vee s \vee m$

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

$w \vee s \vee m$

~~0~~	~~0~~	~~0~~
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

~~0~~	~~0~~	~~0~~
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

w → m

~~0~~	~~0~~	~~0~~
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
~~1~~	~~1~~	~~1~~

~~0~~	~~0~~	~~0~~
0	0	1
0	1	0
0	1	1
~~1~~	~~0~~	~~0~~
1	0	1
~~1~~	~~1~~	~~0~~
~~1~~	~~1~~	~~1~~

w → m

~~0~~	~~0~~	~~0~~
0	0	1
0	1	0
0	1	1
~~1~~	~~0~~	~~0~~
1	0	1
~~1~~	~~1~~	~~0~~
~~1~~	~~1~~	~~1~~

$\neg s \rightarrow \neg m$

~~0~~	~~0~~	~~0~~
0	0	1
0	1	0
0	1	1
~~1~~	~~0~~	~~0~~
1	0	1
~~1~~	~~1~~	~~0~~
~~1~~	~~1~~	~~1~~

$\neg s \rightarrow \neg m$

~~0~~	~~0~~	~~0~~
~~0~~	~~0~~	~~1~~
0	1	0
0	1	1
~~1~~	~~0~~	~~0~~
~~1~~	~~0~~	~~1~~
~~1~~	~~1~~	~~0~~
~~1~~	~~1~~	~~1~~

~~0~~	~~0~~	~~0~~
~~0~~	~~0~~	~~1~~
0	1	0
0	1	1
~~1~~	~~0~~	~~0~~
~~1~~	~~0~~	~~1~~
~~1~~	~~1~~	~~0~~
~~1~~	~~1~~	~~1~~

Validity

An inference is *valid* if and only if its **conclusion** is **true** in every situation at which **all the assumptions** (premises) **hold**.

Counter-example

A ***counter-example*** of an inference is a situation at which **all the premises hold** but the **conclusion does not**.

Invalidity

An inference is ***invalid*** if and only if there **exists a counter-example** of it.

Invalidity

An inference is ***invalid*** if and only if there **exists a counter-example** of it.

$$\not\models$$

Invalidity

An inference is ***invalid*** if and only if there **exists a counter-example** of it.

$$\phi_1, \ldots, \phi_n \not\models \psi$$

Validity

An inference is ***valid*** if and only if it has **no counter-examples**.

$$\phi_1,\ldots,\phi_n \models \psi$$

Logical Equivalence

Two propositions/formulas are ***logically equivalent*** if and only if they are true under **the same circumstances**.

Logical Equivalence

Two propositions/formulas are ***logically equivalent*** if and only if they are true under **the same circumstances**.

$$\phi \equiv \psi$$

Logical Equivalence

Two propositions/formulas are ***logically equivalent*** if and only if they are true under **the same circumstances**.

$$\phi \models \psi \quad \textit{and} \quad \psi \models \phi$$

$$\phi \equiv \psi$$

Valid and satisfiable formulas

A formula is *valid* if it is true under all circumstances.

A formula is *satisfiable* if it is true under certain circumstances.

Valid and satisfiable formulas

A formula is *valid* if it is true under all circumstances.

$$\models \psi$$

A formula is *satisfiable* if it is true under certain circumstances.

Valid and satisfiable formulas

A formula is *valid* if it is true under all circumstances.

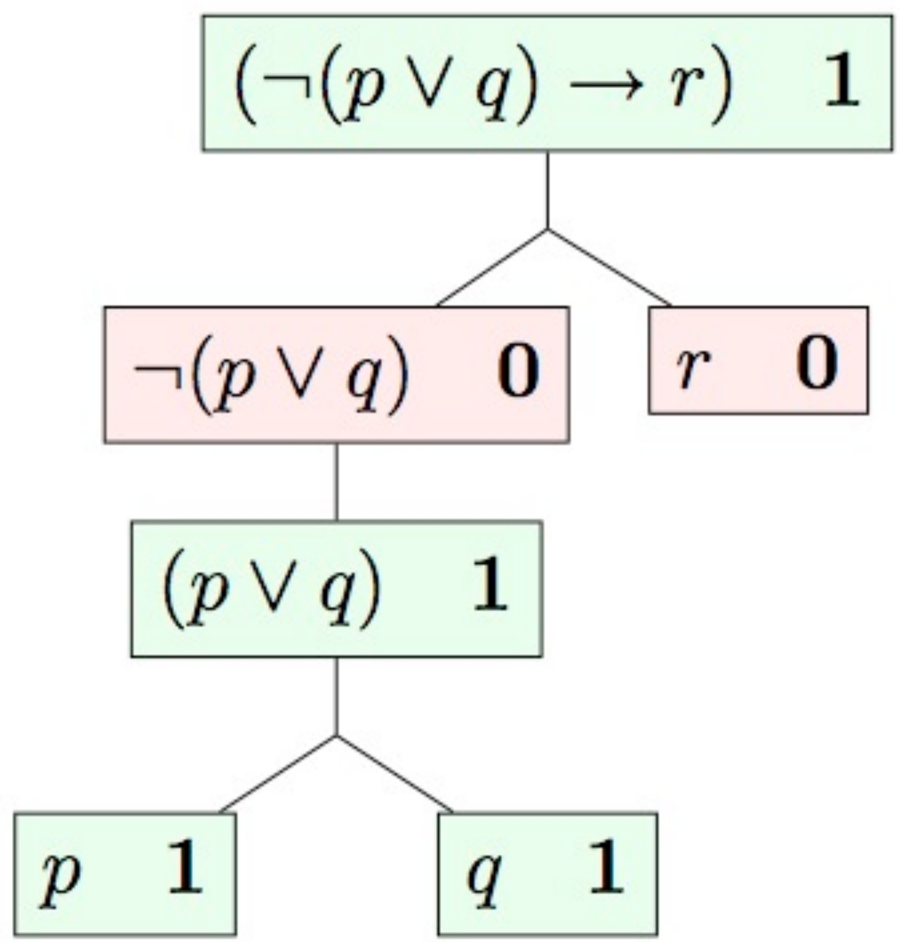
$$\models \psi$$

A formula is *satisfiable* if it is true under certain circumstances.

$$\not\models \neg\psi$$

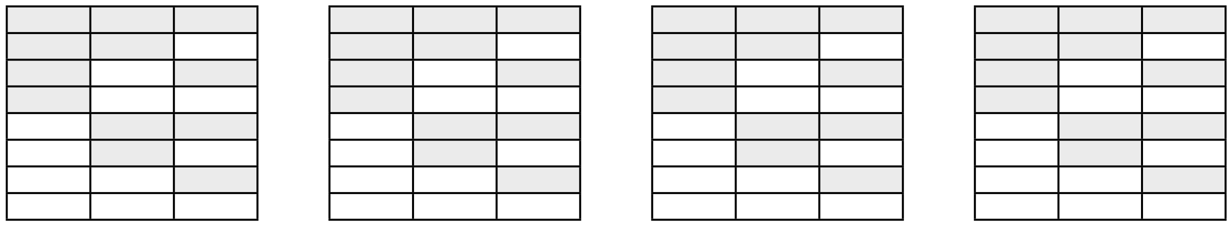
I – 1. Propostional Logic

design



I-1. Propostional Logic

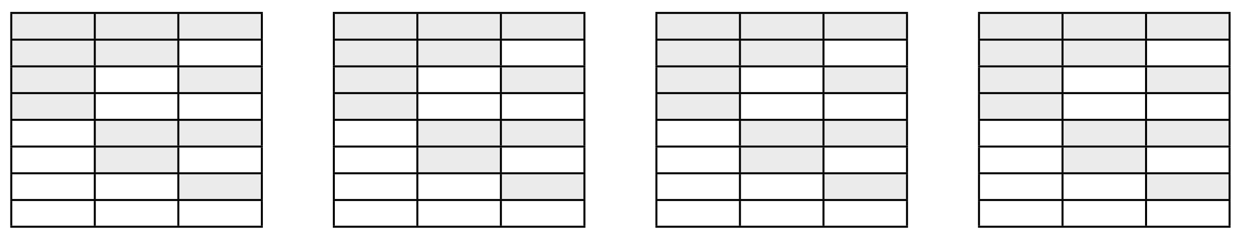
design



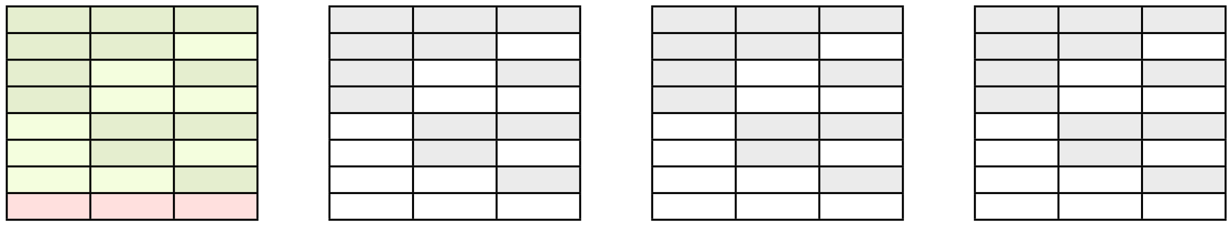
I – 1. Propostional Logic

design

¬(w∧m∧s)



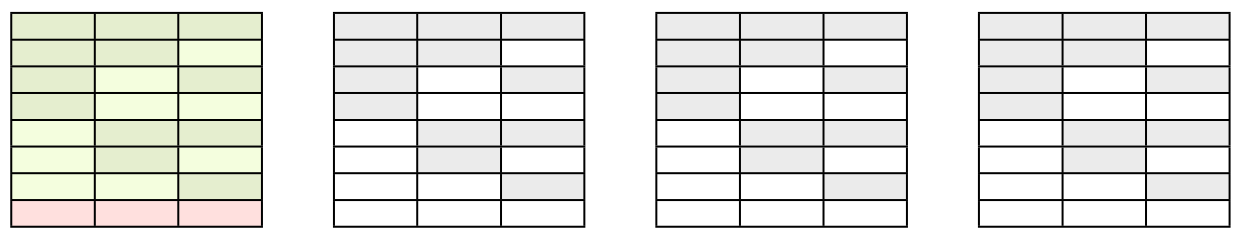
$\neg(w \wedge m \wedge s)$



I - 1. Propostional Logic

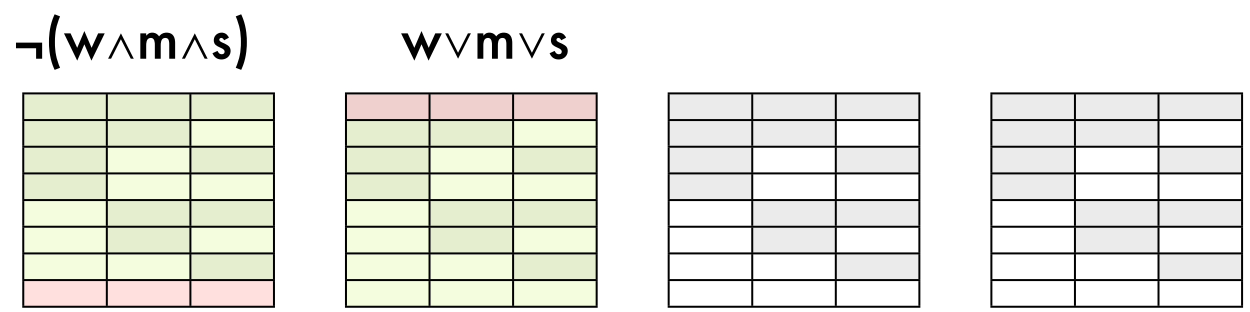
design

$\neg(w \wedge m \wedge s)$ $w \vee m \vee s$



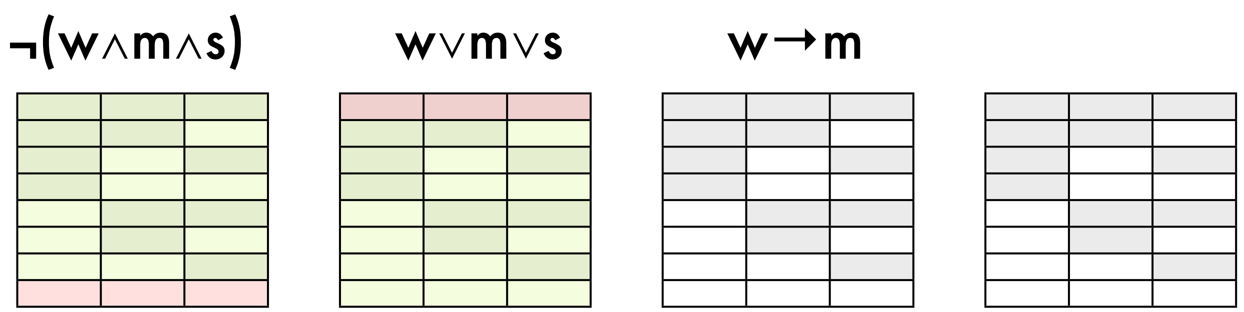
I -1. Propostional Logic

design



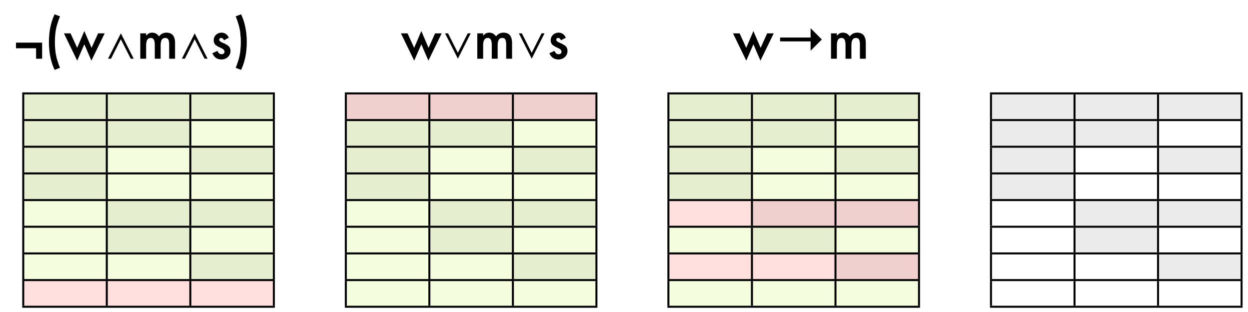
I-1. Propostional Logic

design



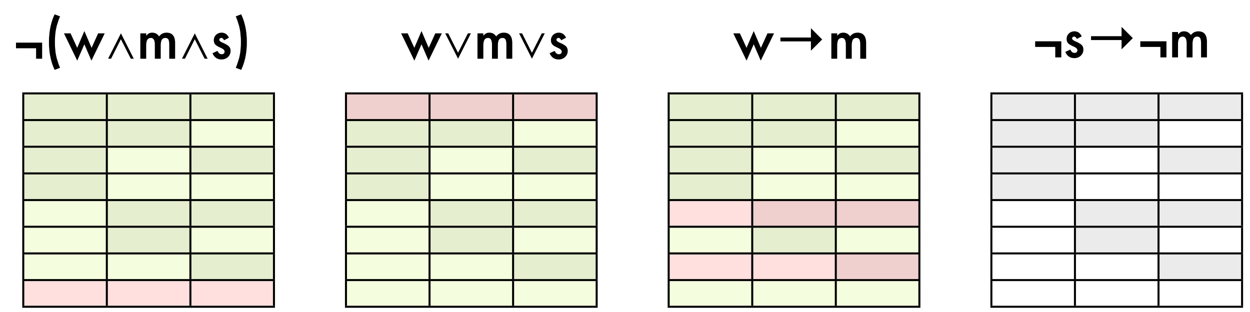
I-1. Propostional Logic

design



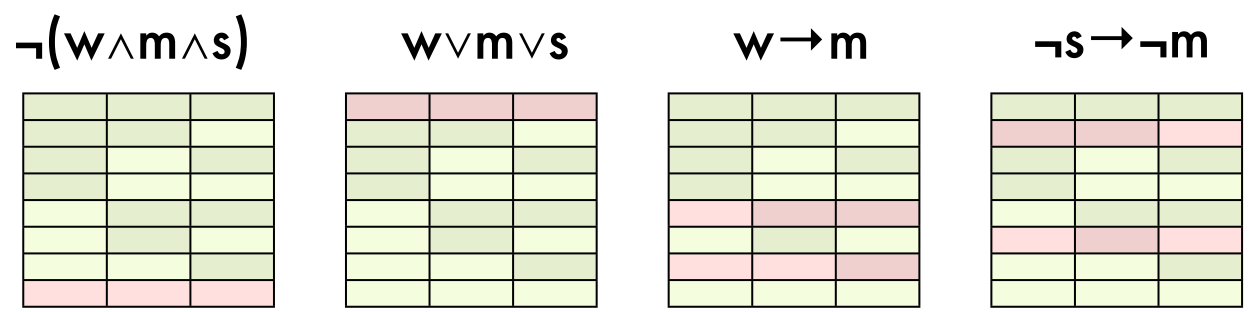
I -1. Propostional Logic

design



I-1. Propostional Logic

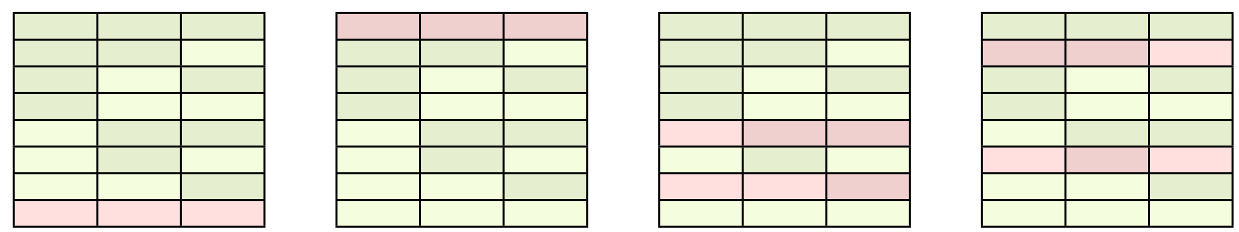
design



I-1. Propostional Logic

design

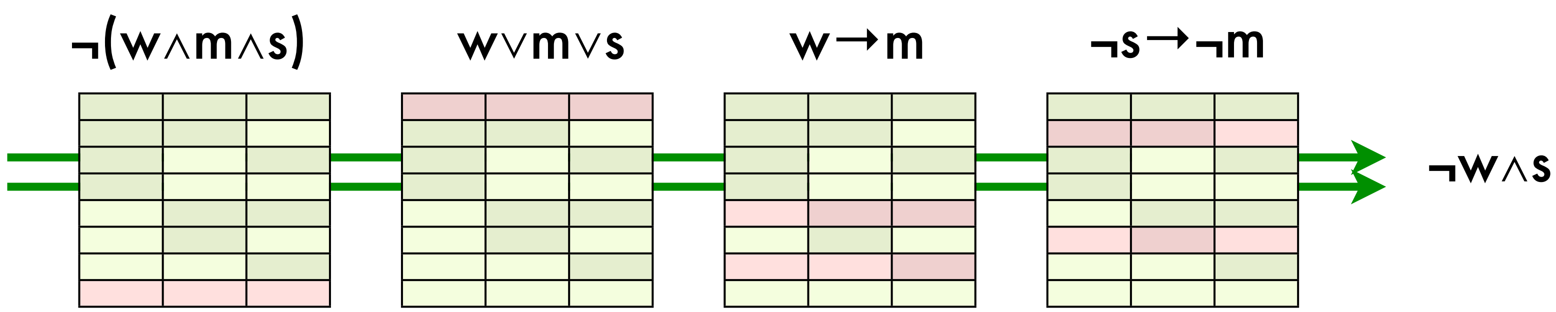
¬(w∧m∧s)　　w∨m∨s　　w→m　　¬s→¬m



¬w∧s

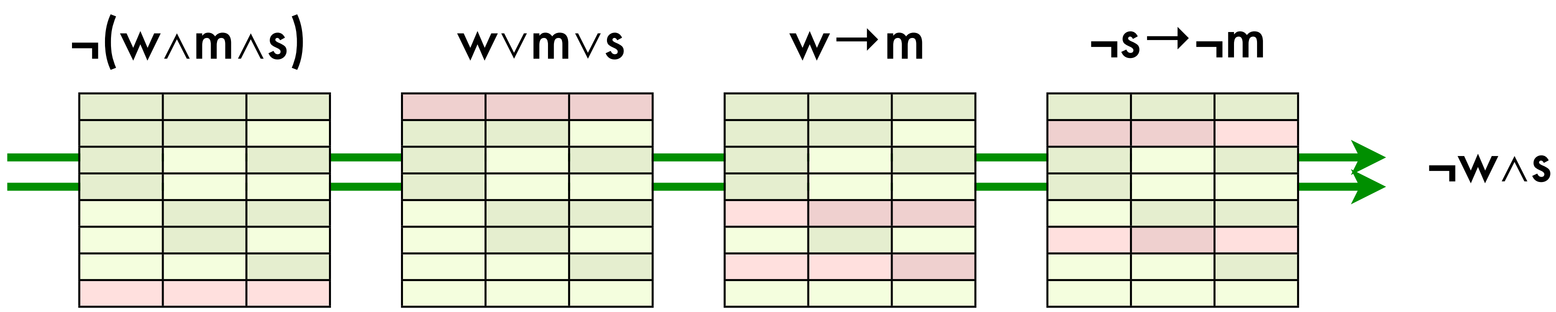
I-1. Propostional Logic

design



I – 1. Propostional Logic

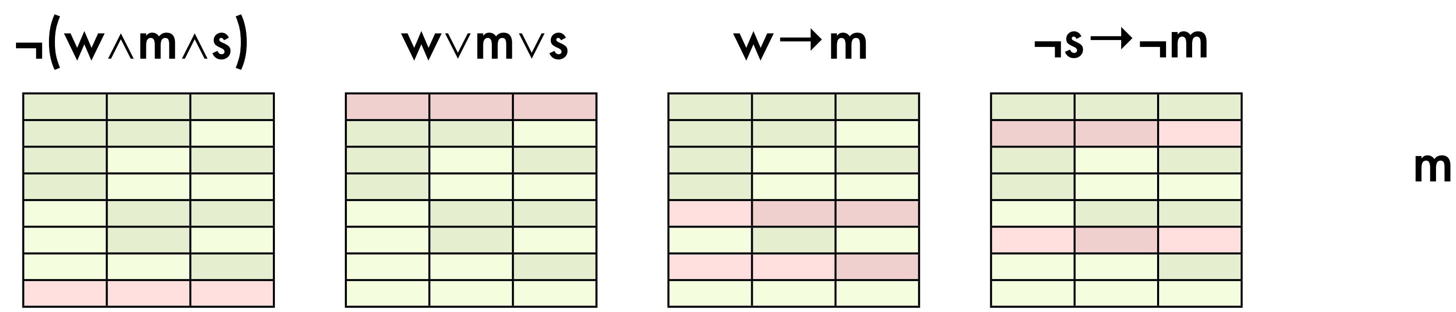
design



$\neg(w \wedge m \wedge s),\ w \vee m \vee s,\ w \rightarrow m,\ \neg s \rightarrow \neg m \models \neg w \wedge s$

I-1. Propositional Logic

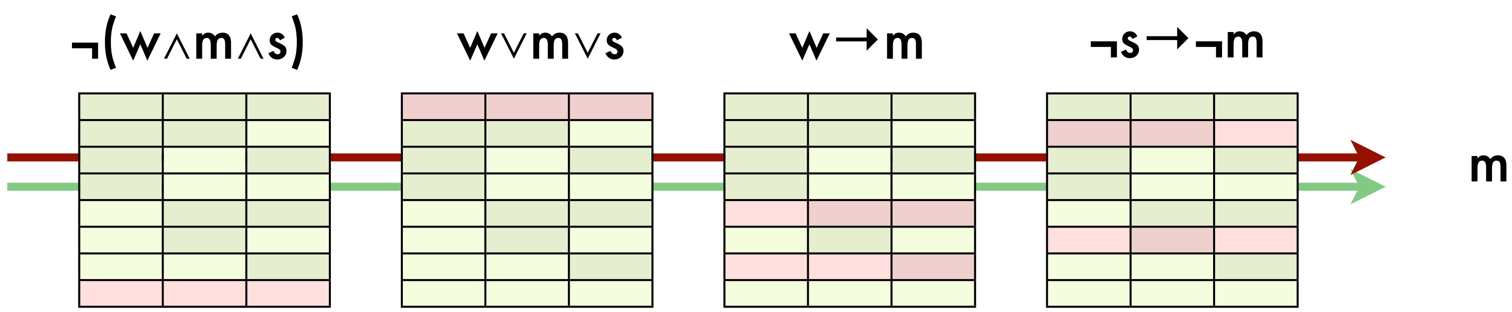
design



$$\neg(w \wedge m \wedge s),\ w \vee m \vee s,\ w \rightarrow m,\ \neg s \rightarrow \neg m \not\models m$$

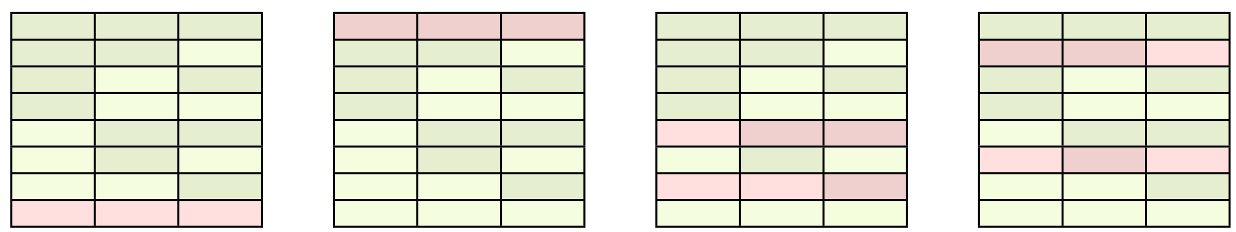
I – 1. Propositional Logic

design



$$\neg(w \land m \land s),\ w \lor m \lor s,\ w \rightarrow m,\ \neg s \rightarrow \neg m \not\models m$$

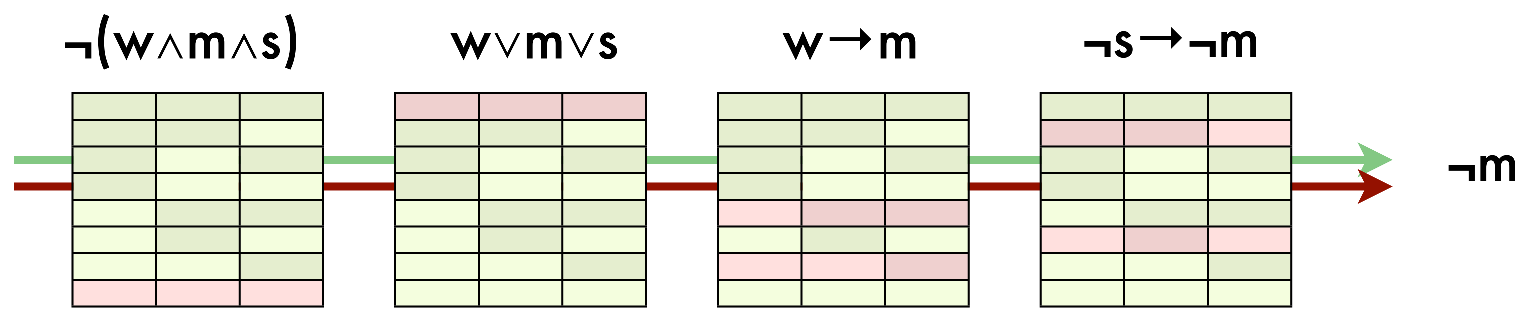
I-1. Propostional Logic



$$\neg(w \wedge m \wedge s),\ w \vee m \vee s,\ w \rightarrow m,\ \neg s \rightarrow \neg m \nvDash \neg m$$

I – 1. Propostional Logic

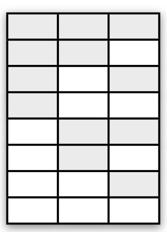
design



$$\neg(w \wedge m \wedge s),\ w \vee m \vee s,\ w \rightarrow m,\ \neg s \rightarrow \neg m \not\models \neg m$$

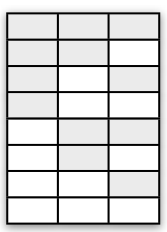
I – 1. Propostional Logic

design



I – 1. Propostional Logic

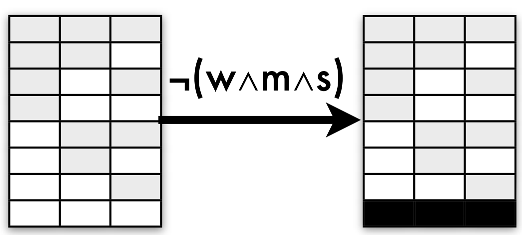
design



$\neg(w \wedge m \wedge s)$

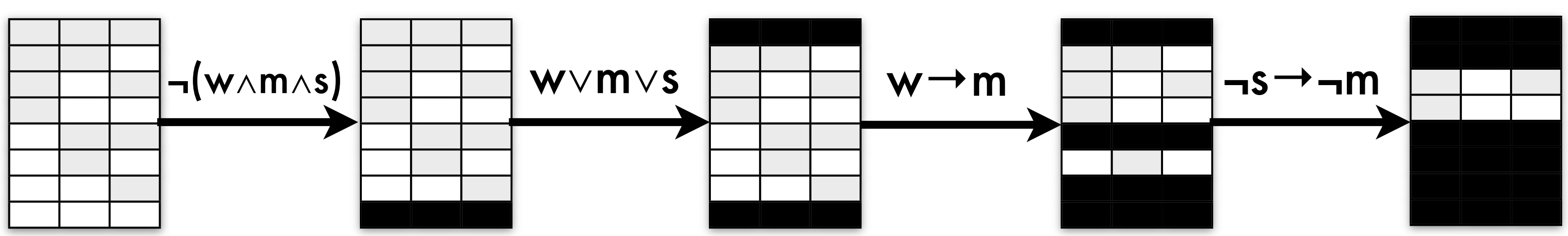
I – 1. Propostional Logic

design



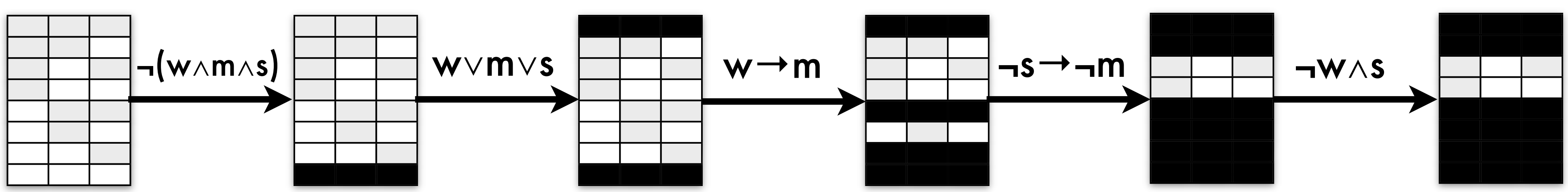
I - 1. Propostional Logic

design



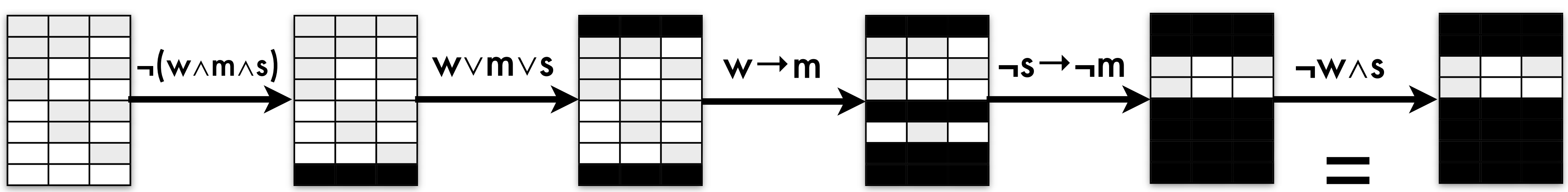
I - 1. Propostional Logic

design



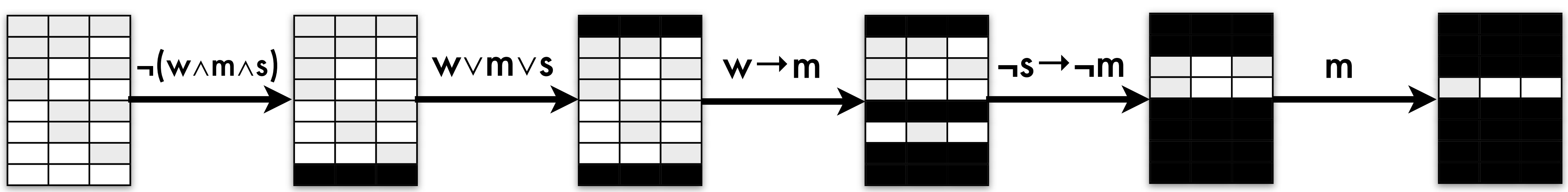
I - 1. Propostional Logic

design



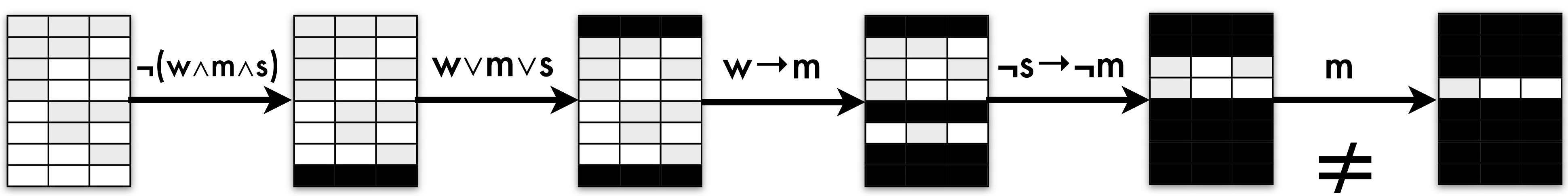
I - 1. Propostional Logic

design

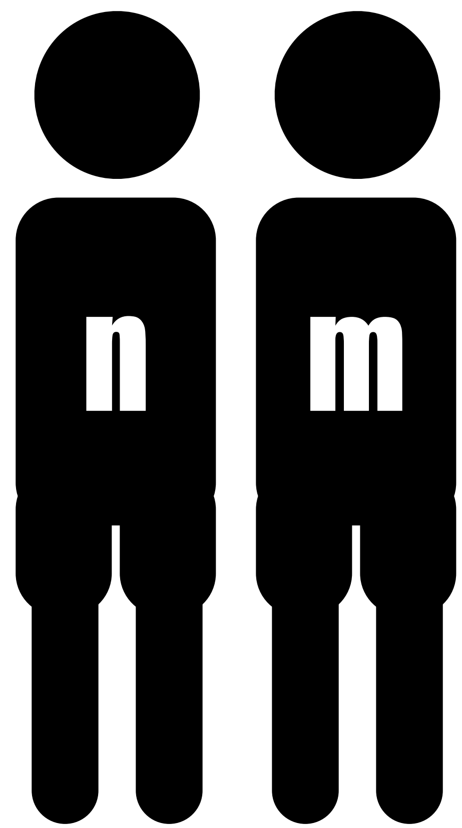


I - 1. Propostional Logic

design



design

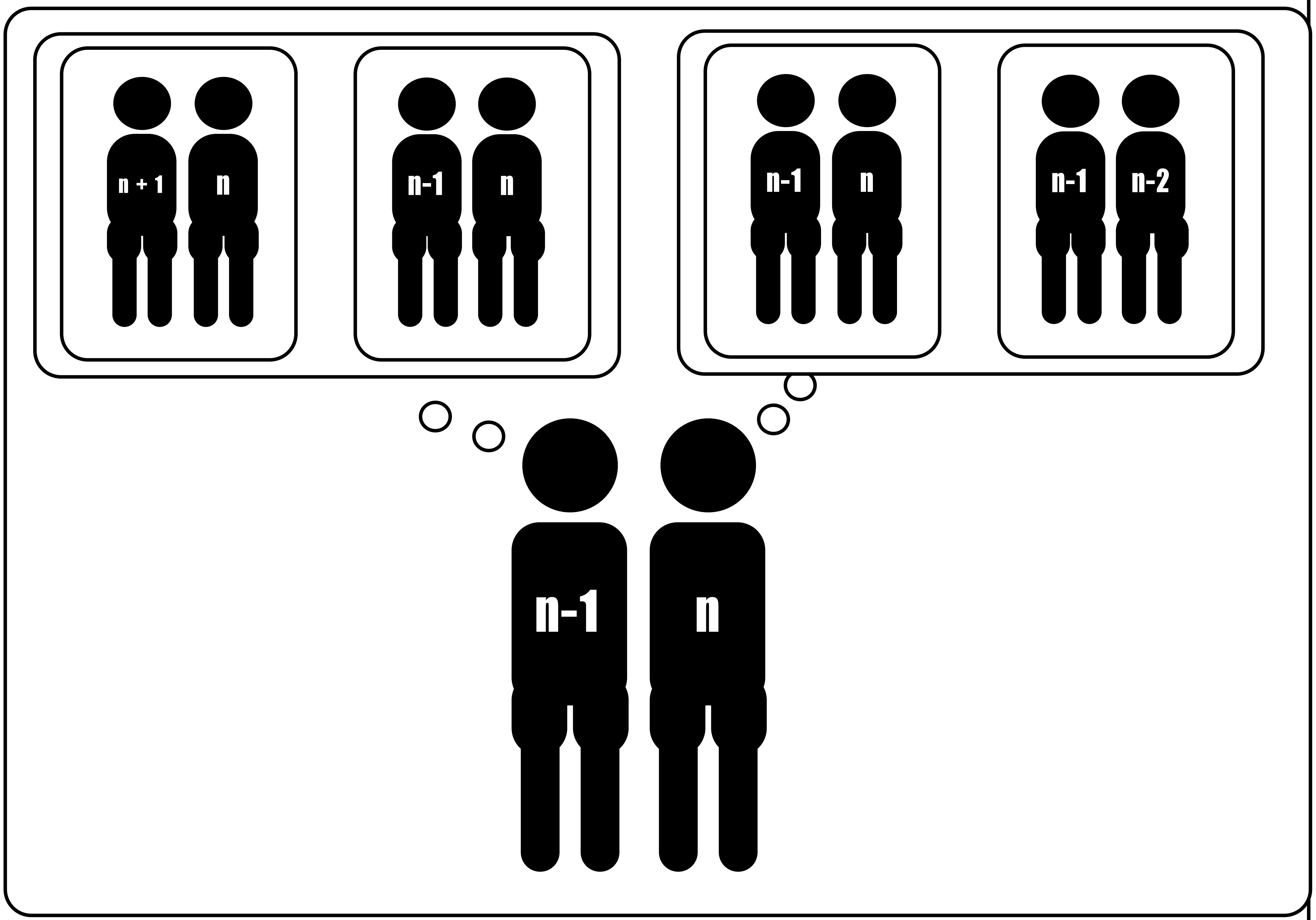


$$m = n \pm 1$$

$$m,n = 1, 2, 3, \ldots.$$

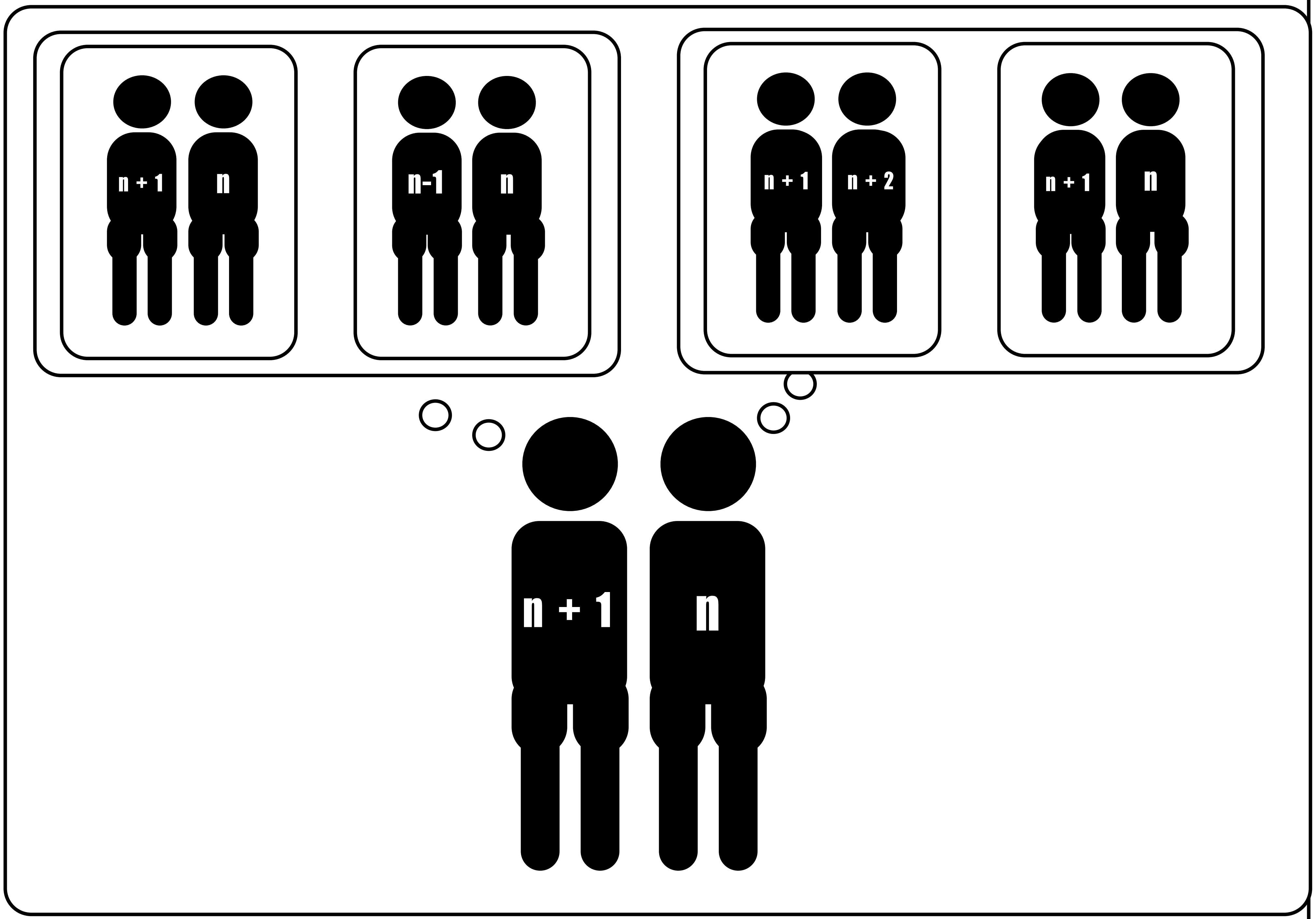
II-4. Epistemic Logic

design



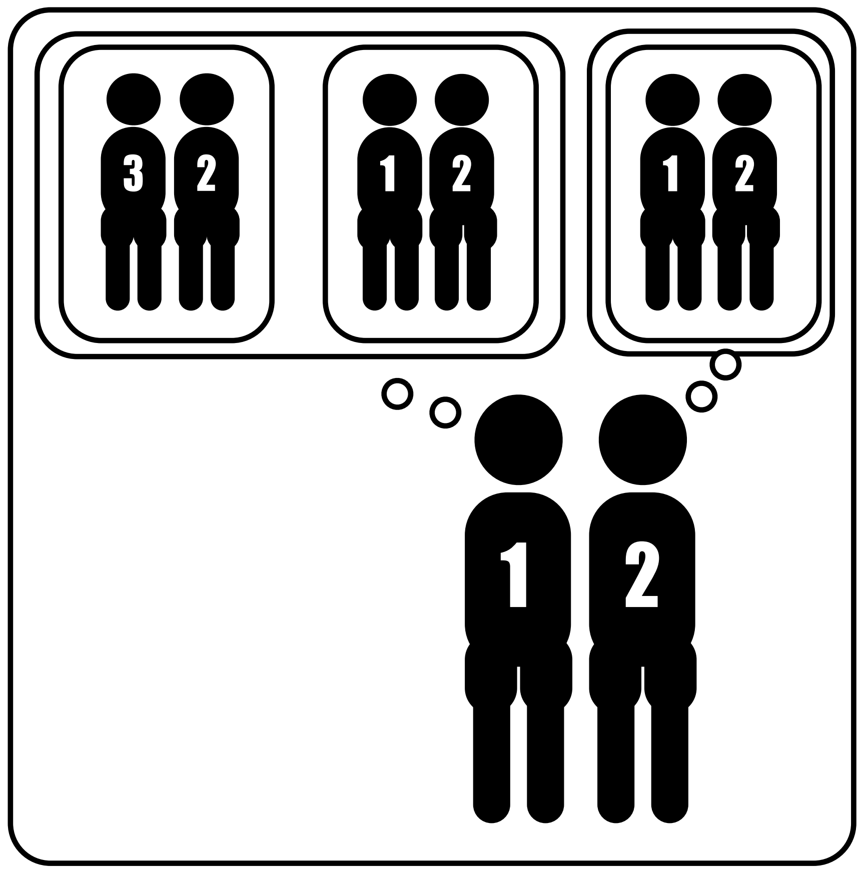
II -4. Epistemic Logic

design

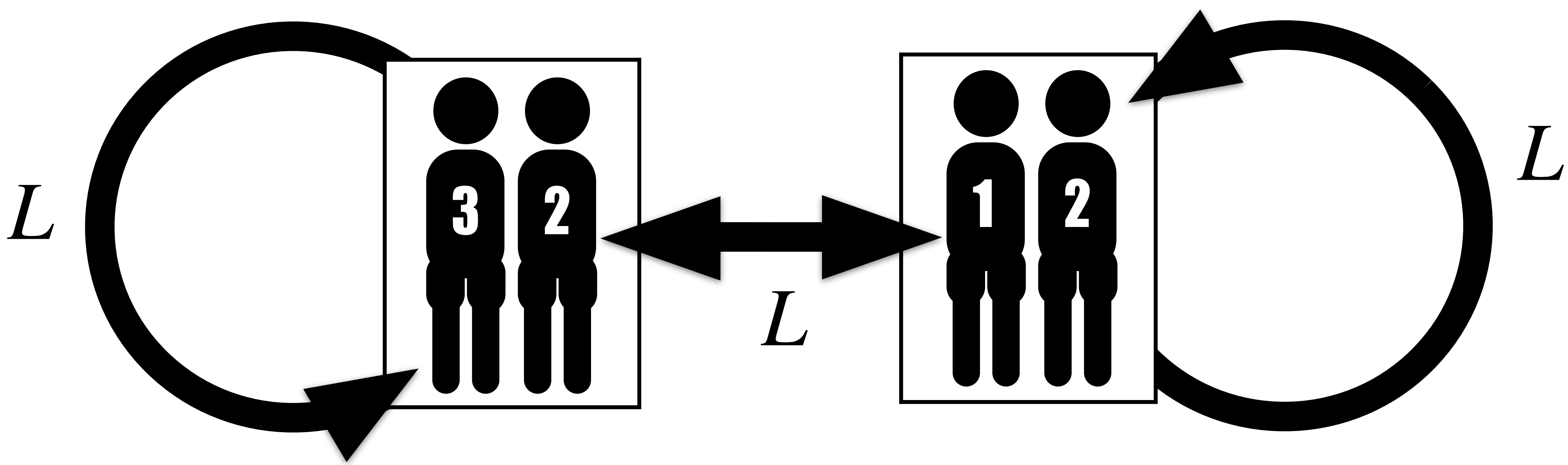


II -4. Epistemic Logic

design

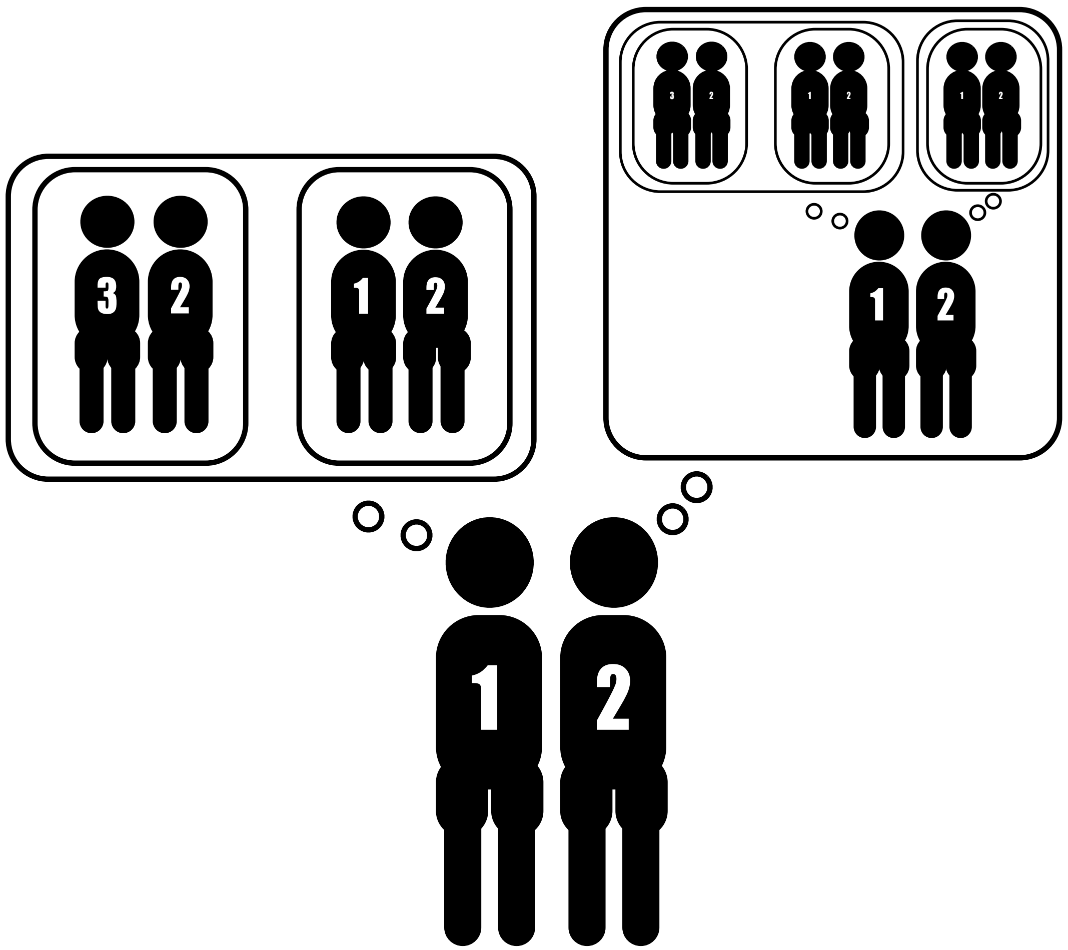


design



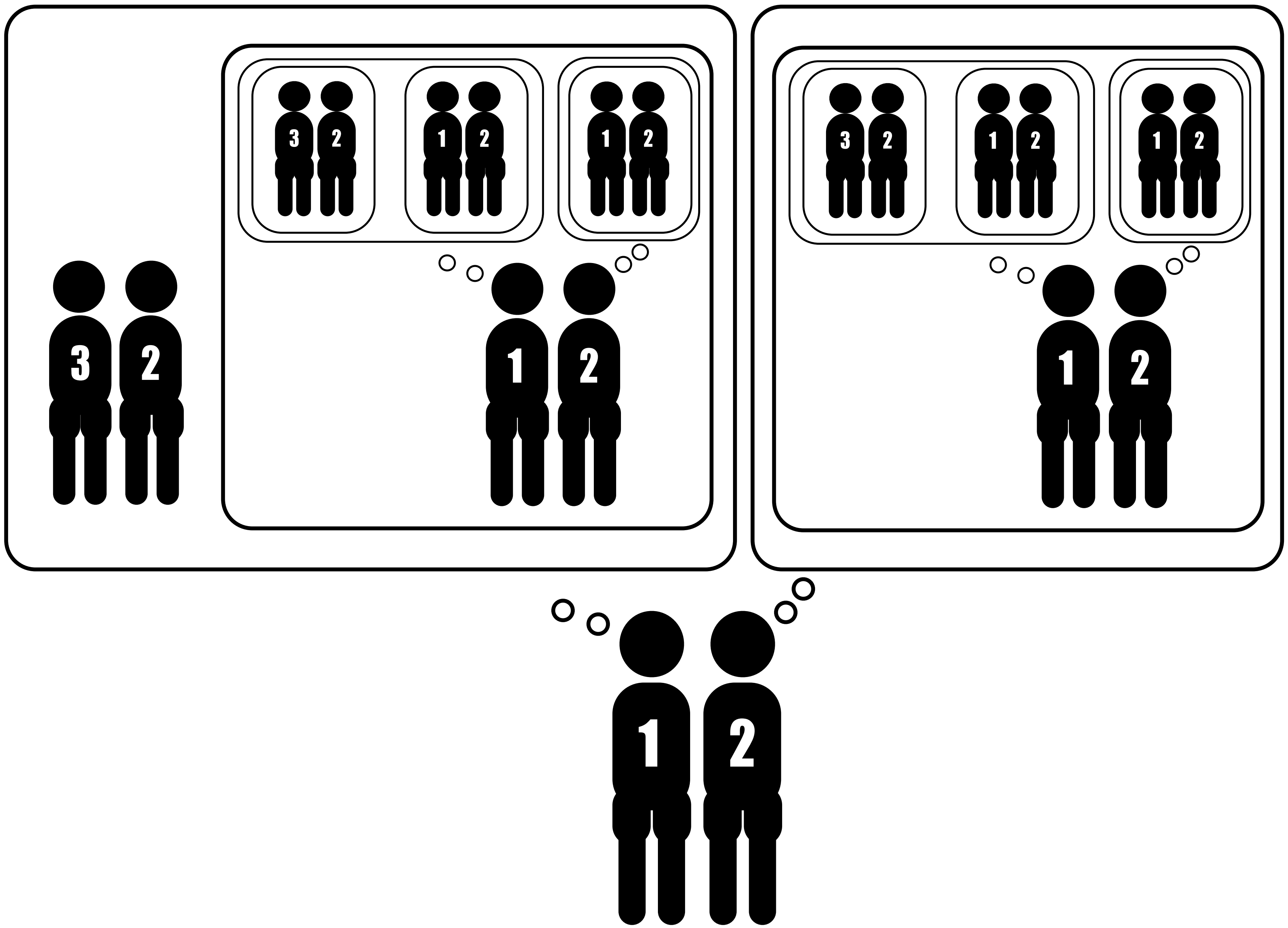
II -4. Epistemic Logic

design



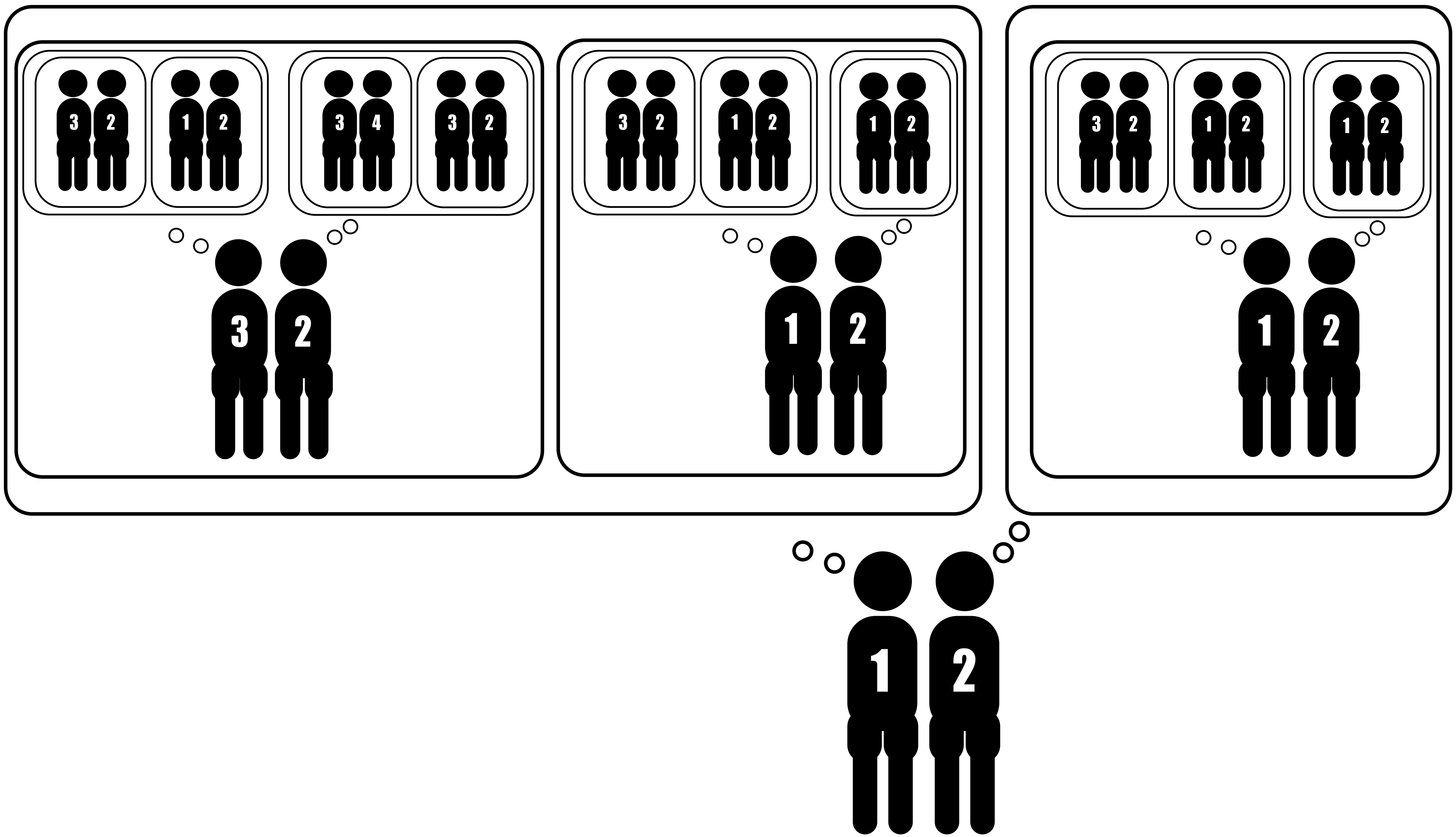
II-4. Epistemic Logic

design



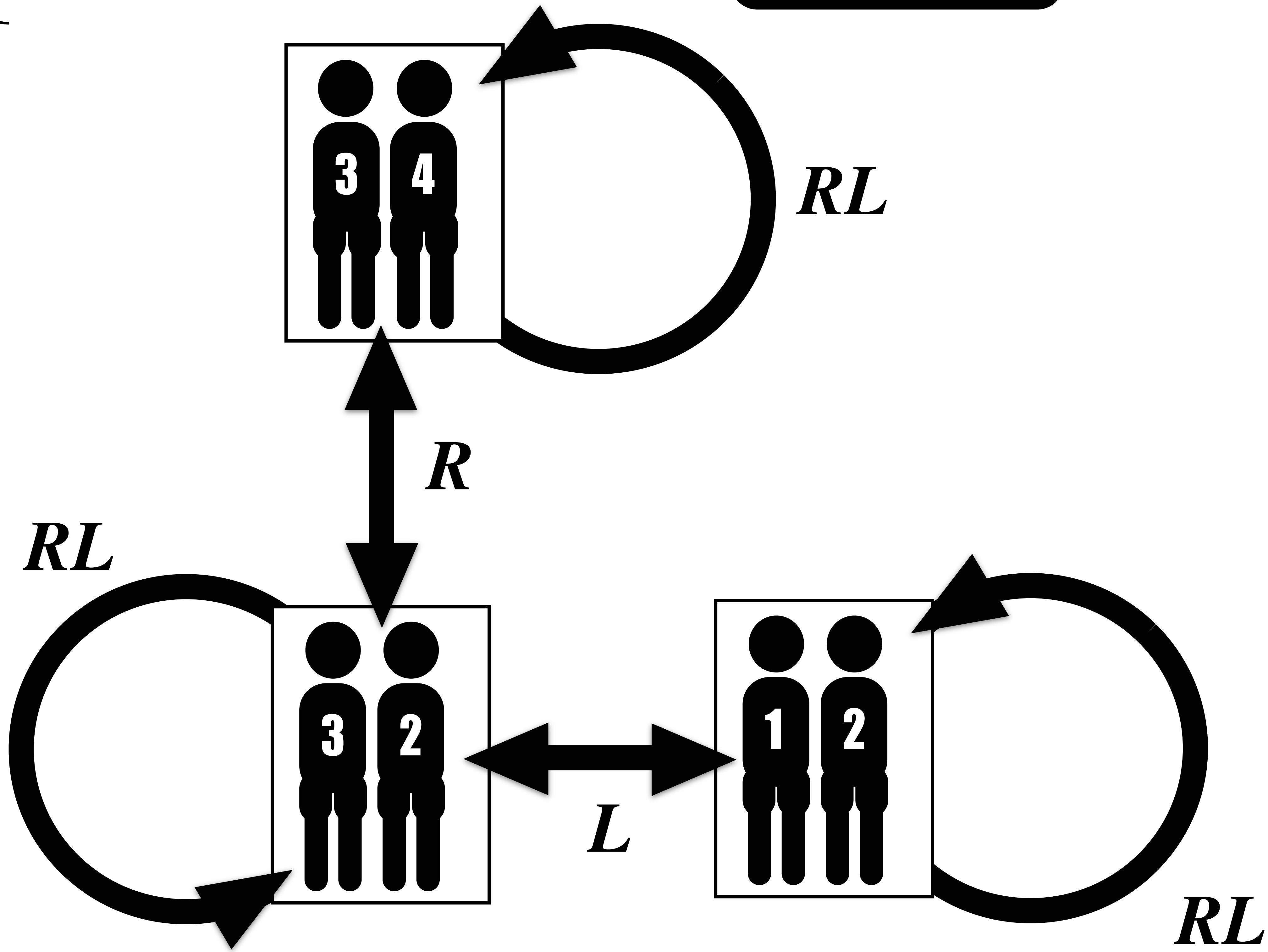
II-4. Epistemic Logic

design



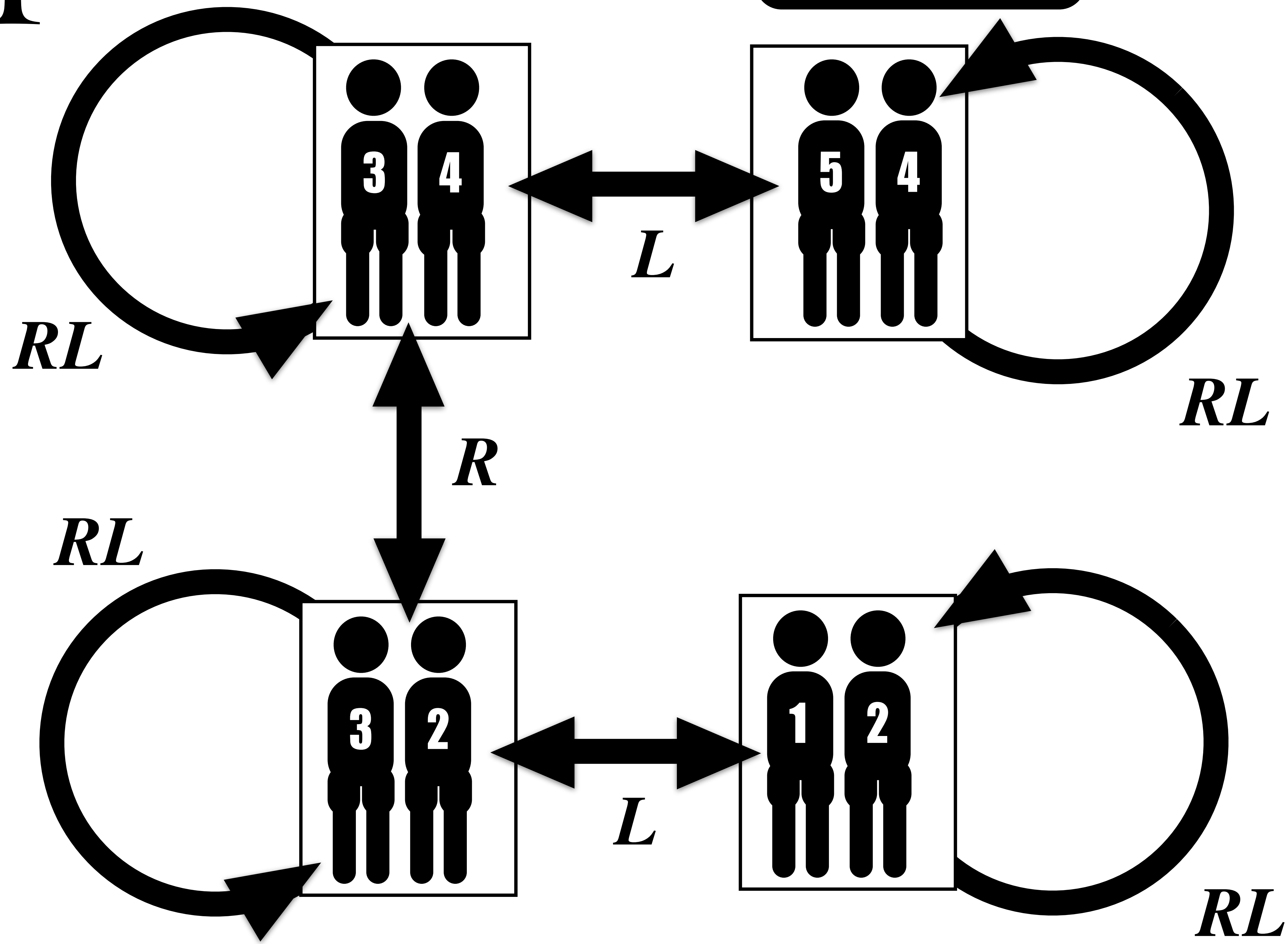
II-4. Epistemic Logic

design



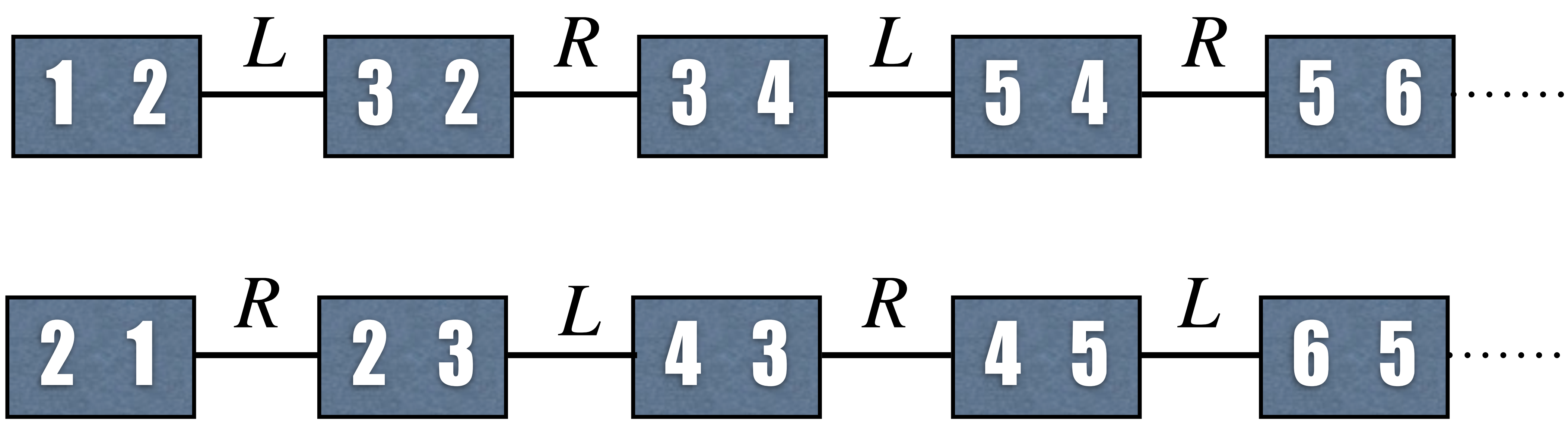
II-4. Epistemic Logic

design



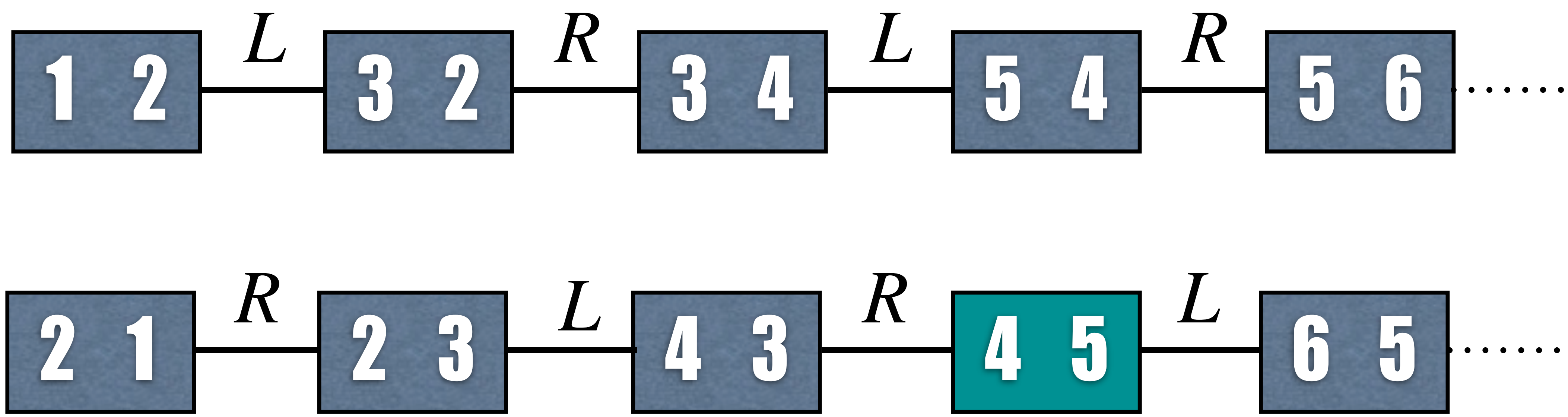
II-4. Epistemic Logic

design



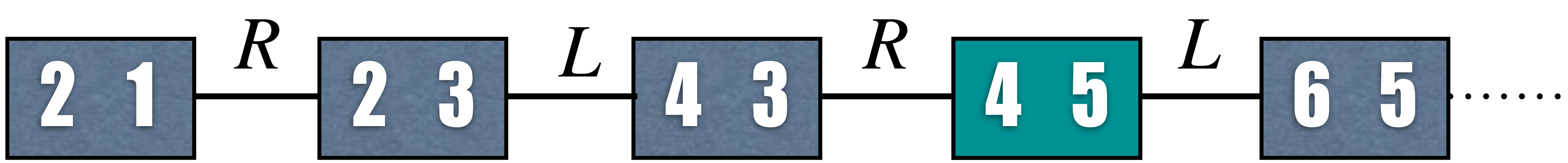
II-4. Epistemic Logic

design



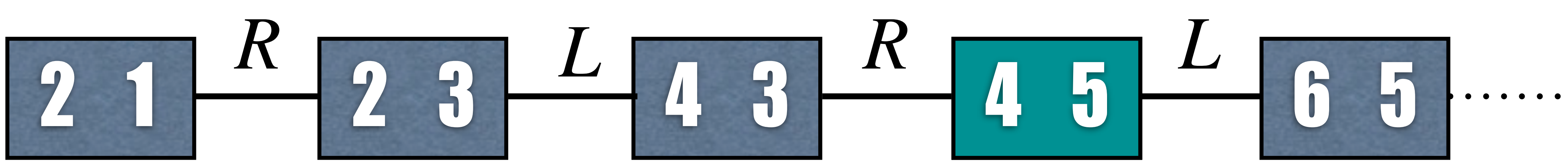
II-4. Epistemic Logic

design



design

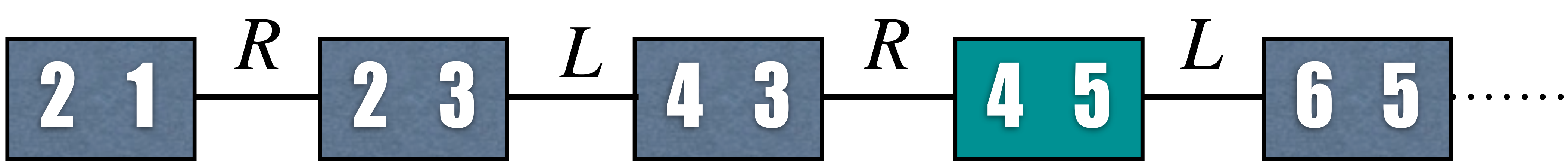
L and R say: "no"



design

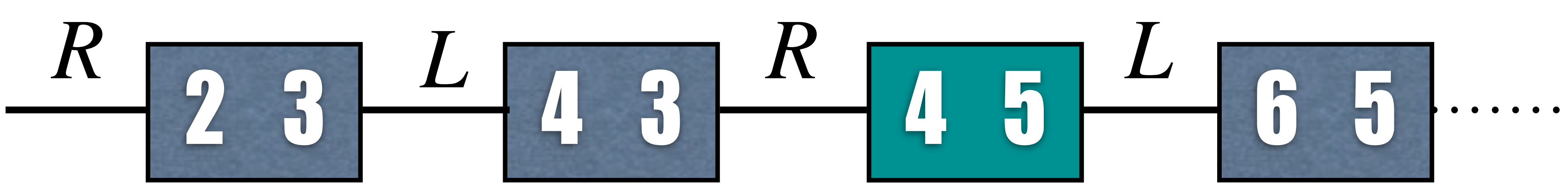
L and R say: "no"

once



L and R say: "no"

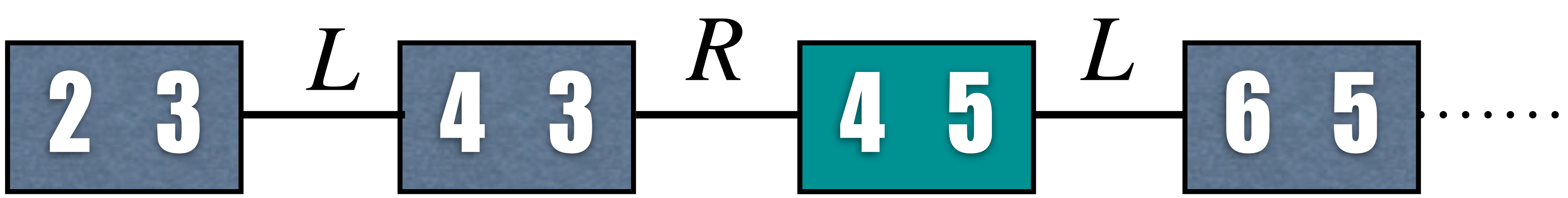
once



design

L and R say: "no"

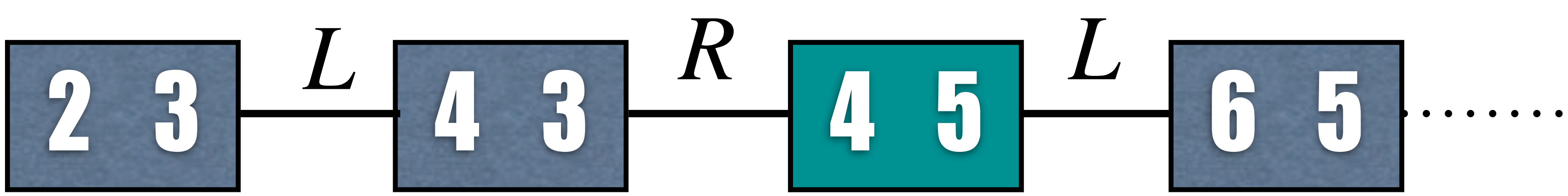
once



design

L and *R* say: "no"

once , twice



design

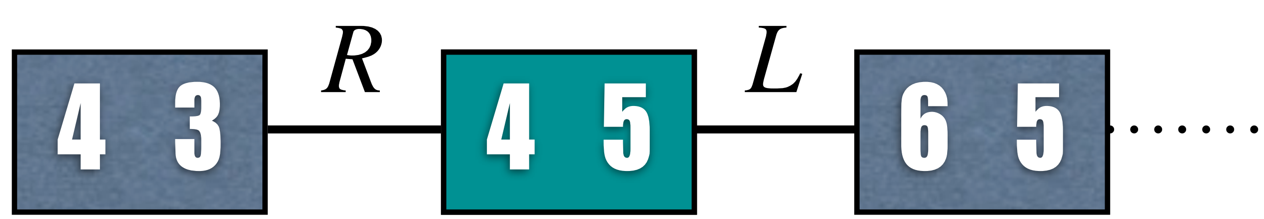
L and *R* say: "no"

once , twice

design

L and *R* say: "no"

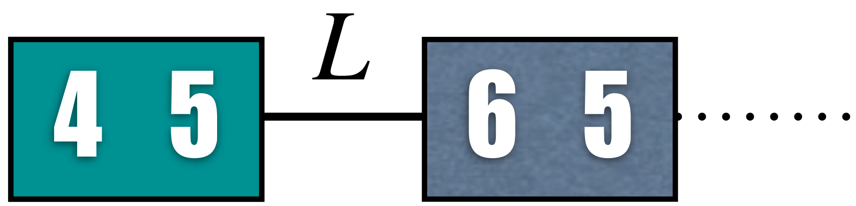
once , twice , three times ....



design

L and *R* say: “no”

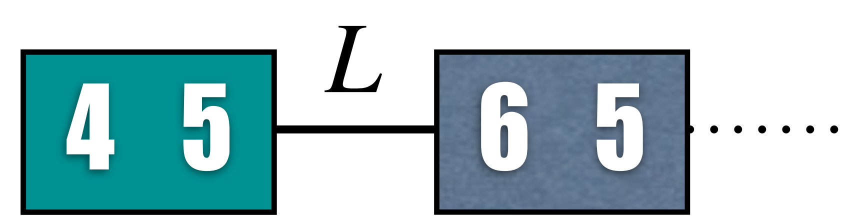
once , twice , three times ....



design

L and *R* say: “no”

once , twice , three times ....



... then *R* says: “yes”.

design

L and *R* say: “no”

once , twice , three times ....

4 5

....

... then *R* says: “yes”.

design

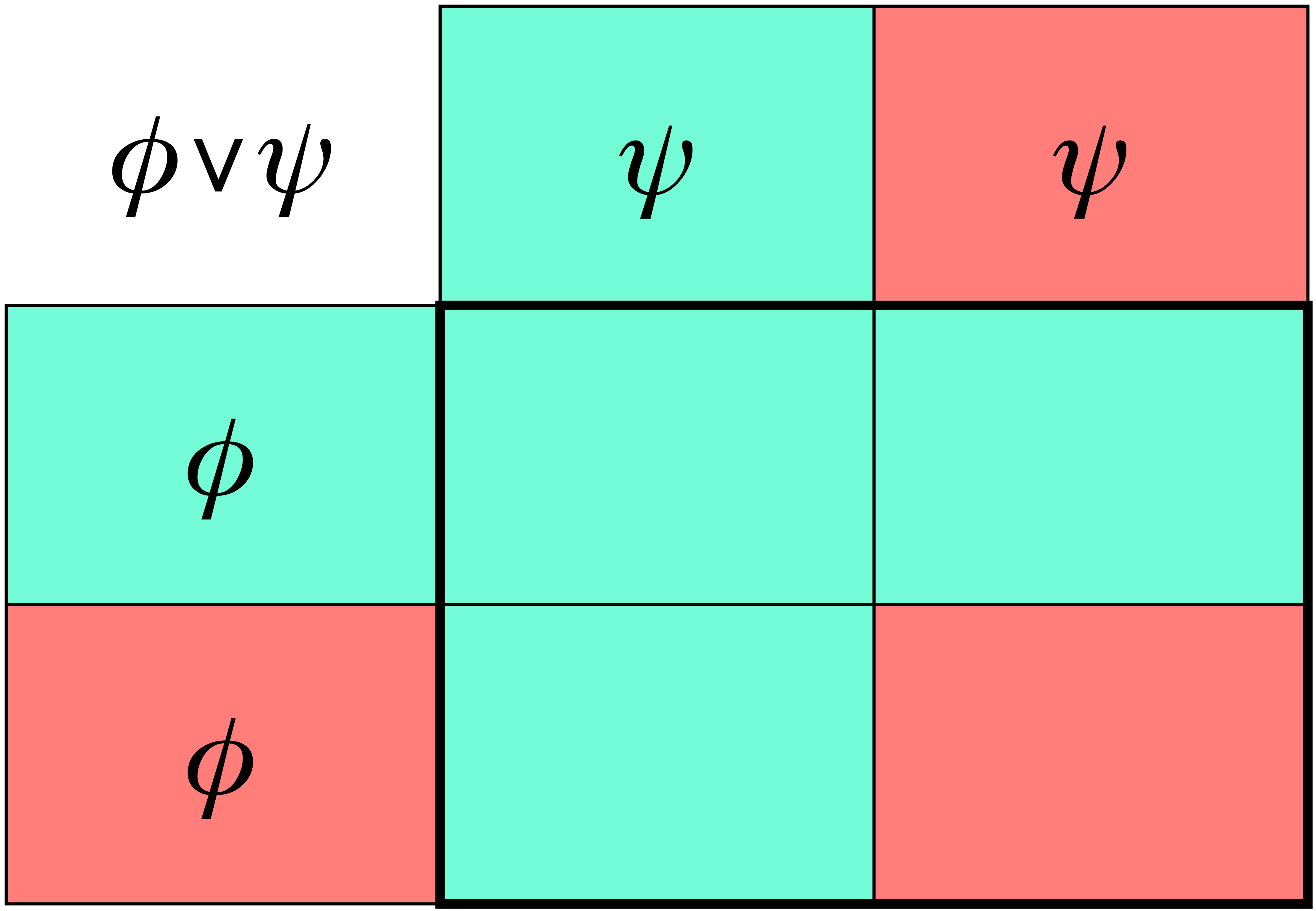
L and *R* say: "no"

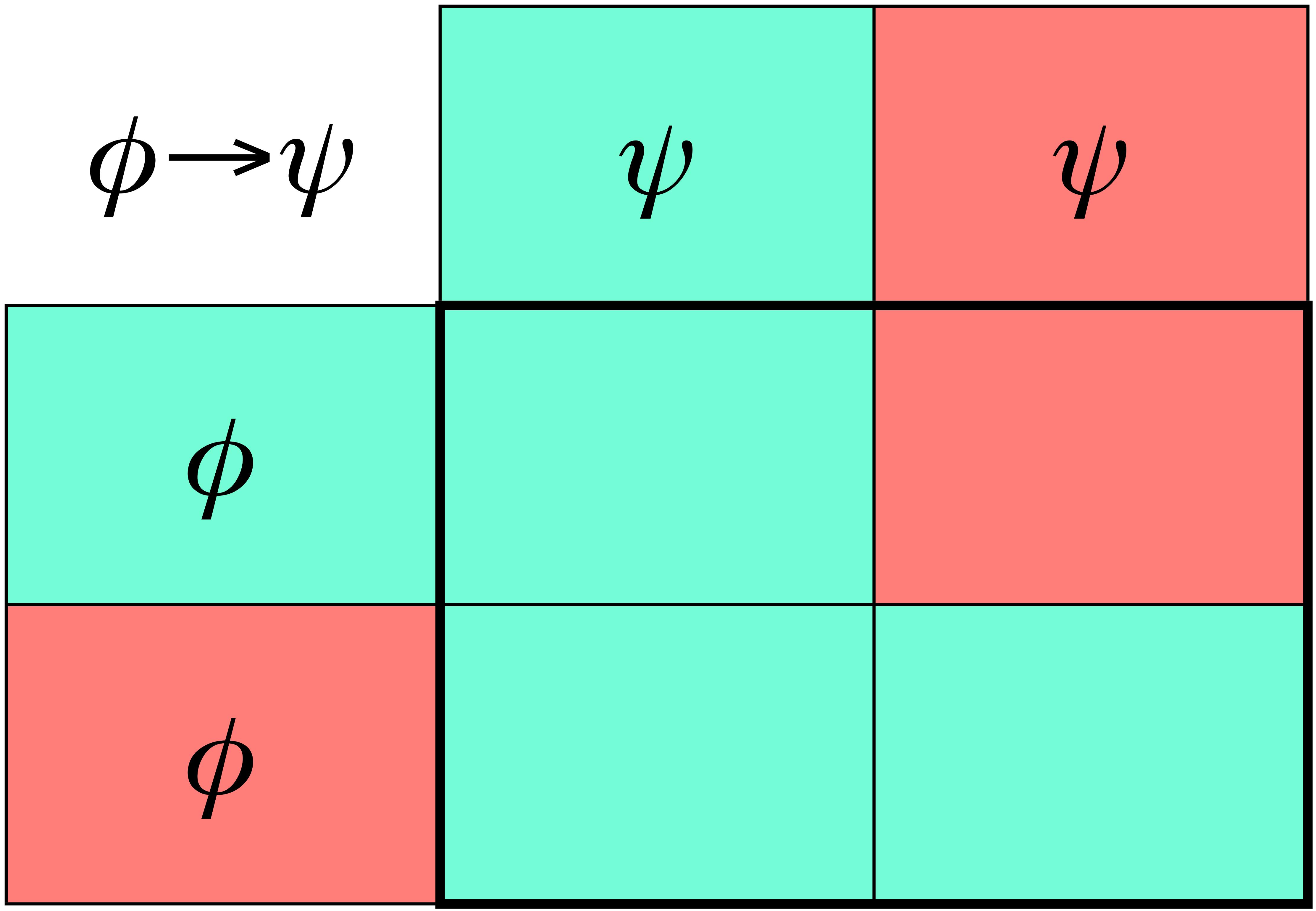
once , twice , three times ....

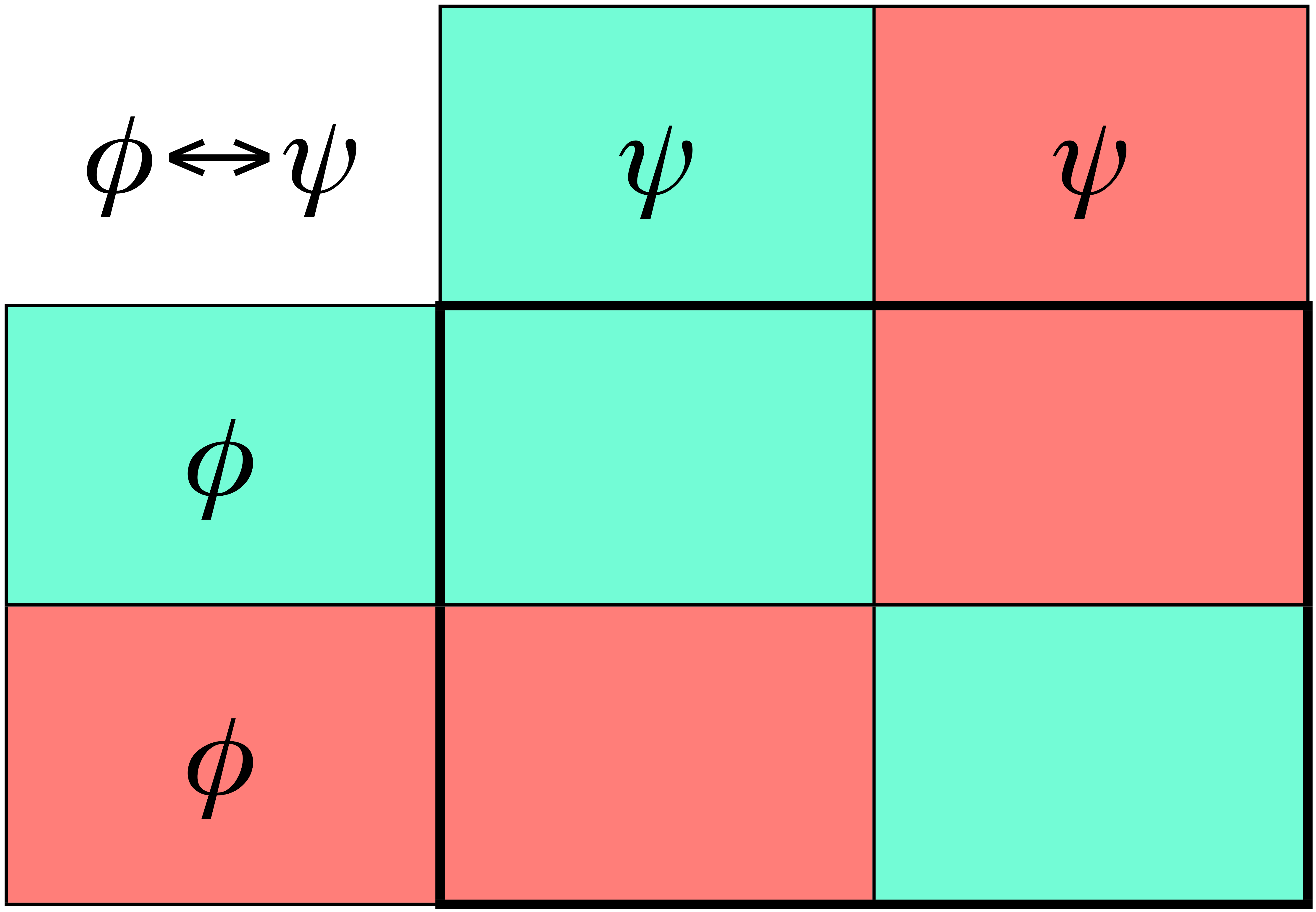
4 5

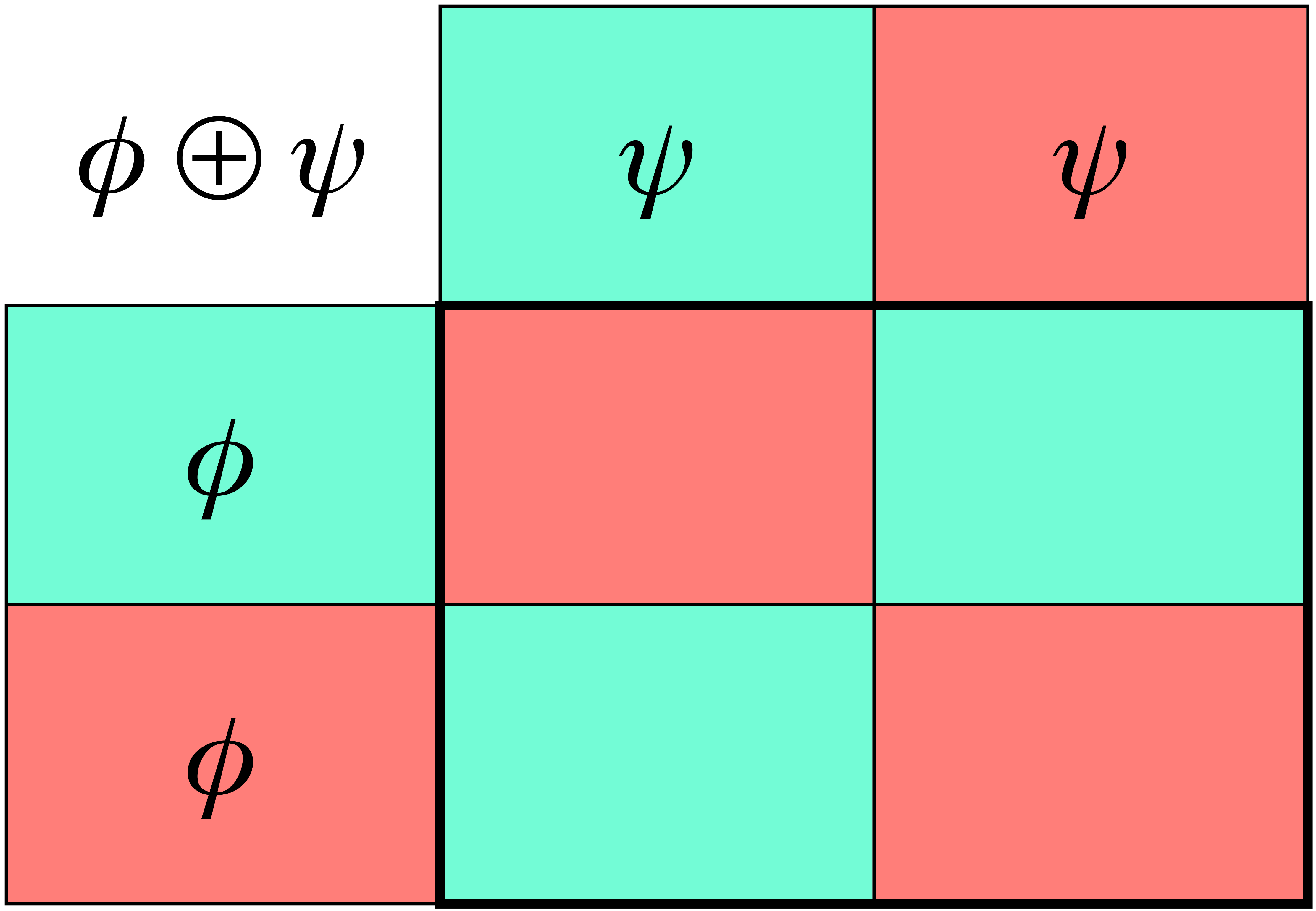
... then *R* says: "yes".

$\phi \wedge \psi$	ψ	ψ
ϕ	$\phi \wedge \psi$	$\phi \wedge \psi$
ϕ	$\phi \wedge \psi$	$\phi \wedge \psi$

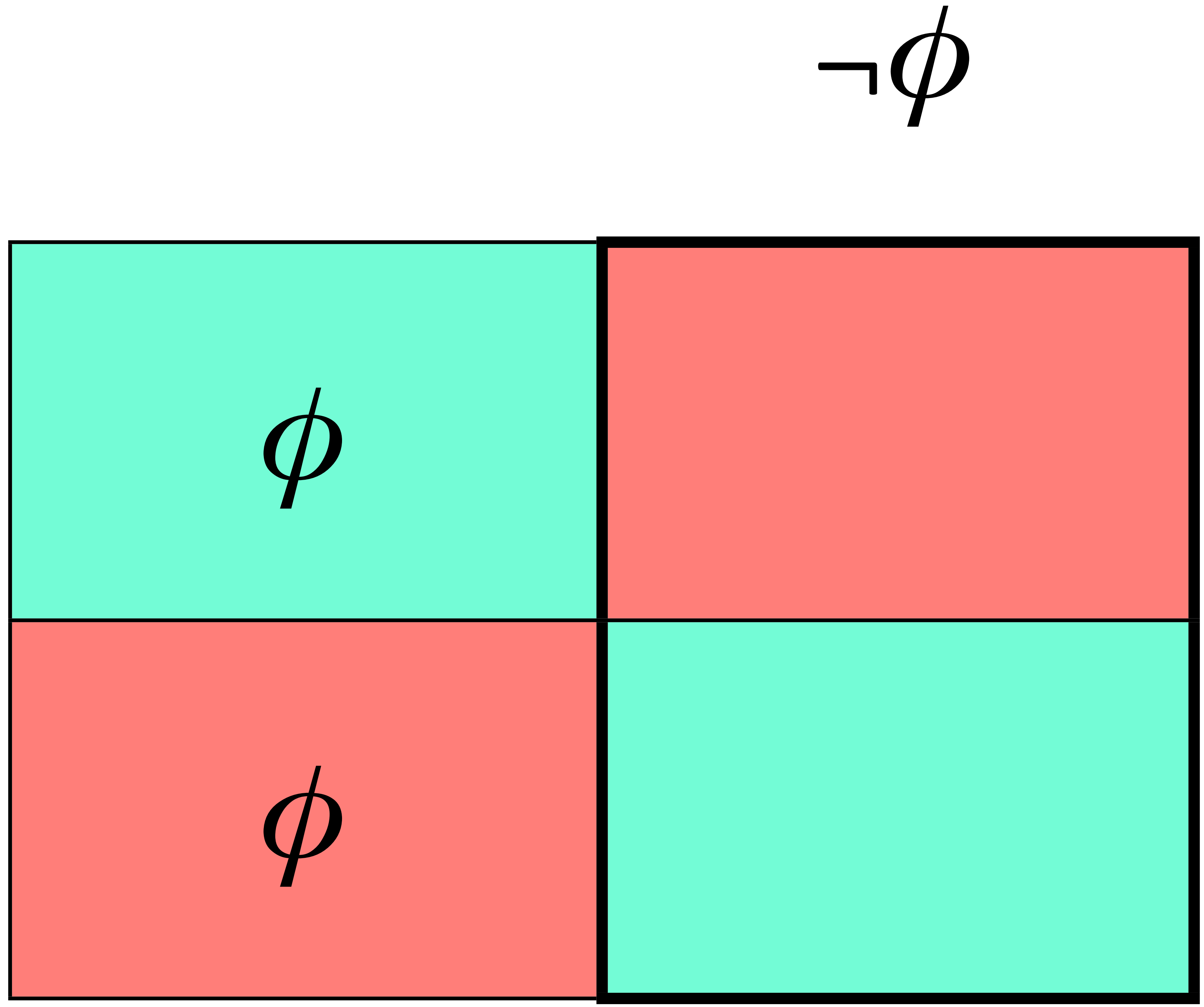

$\phi \vee \psi$
ψ
ψ
ϕ
ϕ


$\phi \rightarrow \psi$
ψ
ψ
ϕ
ϕ

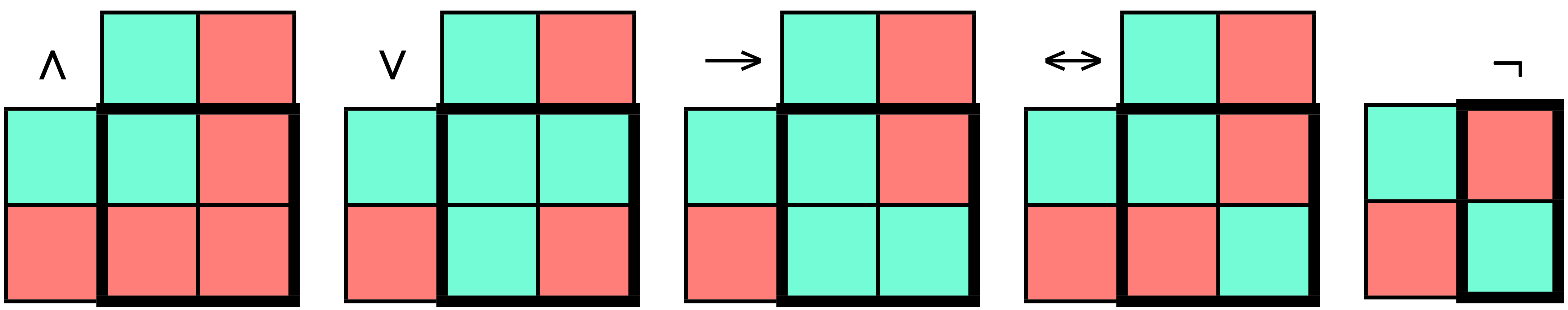

φ⇔ψ
ψ
ψ
φ
φ



$\phi \oplus \psi$
ψ
ψ
ϕ
ϕ



¬φ
φ
φ



∧	I	O
I	I	O
O	O	O

∨	I	O
I	I	I
O	I	O

→	I	O
I	I	O
O	I	I

↔	I	O
I	I	O
O	O	I

	¬
I	O
O	I

∧	1	0
1	**1**	**0**
0	**0**	**0**

∨	1	0
1	**1**	**1**
0	**1**	**0**

→	1	0
1	**1**	**0**
0	**1**	**1**

↔	1	0
1	**1**	**0**
0	**0**	**1**

	¬
1	**0**
0	**1**

Logical Equivalence

Logical Equivalence

$$\phi \oplus \psi \equiv \neg(\phi \leftrightarrow \psi) \equiv \neg\phi \leftrightarrow \psi$$

Logical Equivalence

$$\phi \oplus \psi \equiv \neg(\phi \leftrightarrow \psi) \equiv \neg\phi \leftrightarrow \psi$$

$$\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$$

Logical Equivalence

$$\phi \oplus \psi \equiv \neg(\phi \leftrightarrow \psi) \equiv \neg\phi \leftrightarrow \psi$$

$$\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$$

$$\phi \rightarrow \psi \equiv \neg\phi \vee \psi \equiv \neg(\phi \wedge \neg\psi)$$

Logical Equivalence

$$\phi \oplus \psi \equiv \neg(\phi \leftrightarrow \psi) \equiv \neg\phi \leftrightarrow \psi$$

$$\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$$

$$\phi \rightarrow \psi \equiv \neg\phi \vee \psi \equiv \neg(\phi \wedge \neg\psi)$$

$$\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$$

Expressivity

Expressivity

Every boolean function can be defined by using ¬, ∧ and ∨ only.

Expressivity

Every boolean function can be defined by using ¬, ∧ and ∨ only.

Also by ¬ and ∧.

Expressivity

Every boolean function can be defined by using $\neg$, $\wedge$ and $\vee$ only.

Also by $\neg$ and $\wedge$.

Also by $\neg$ and $\rightarrow$.

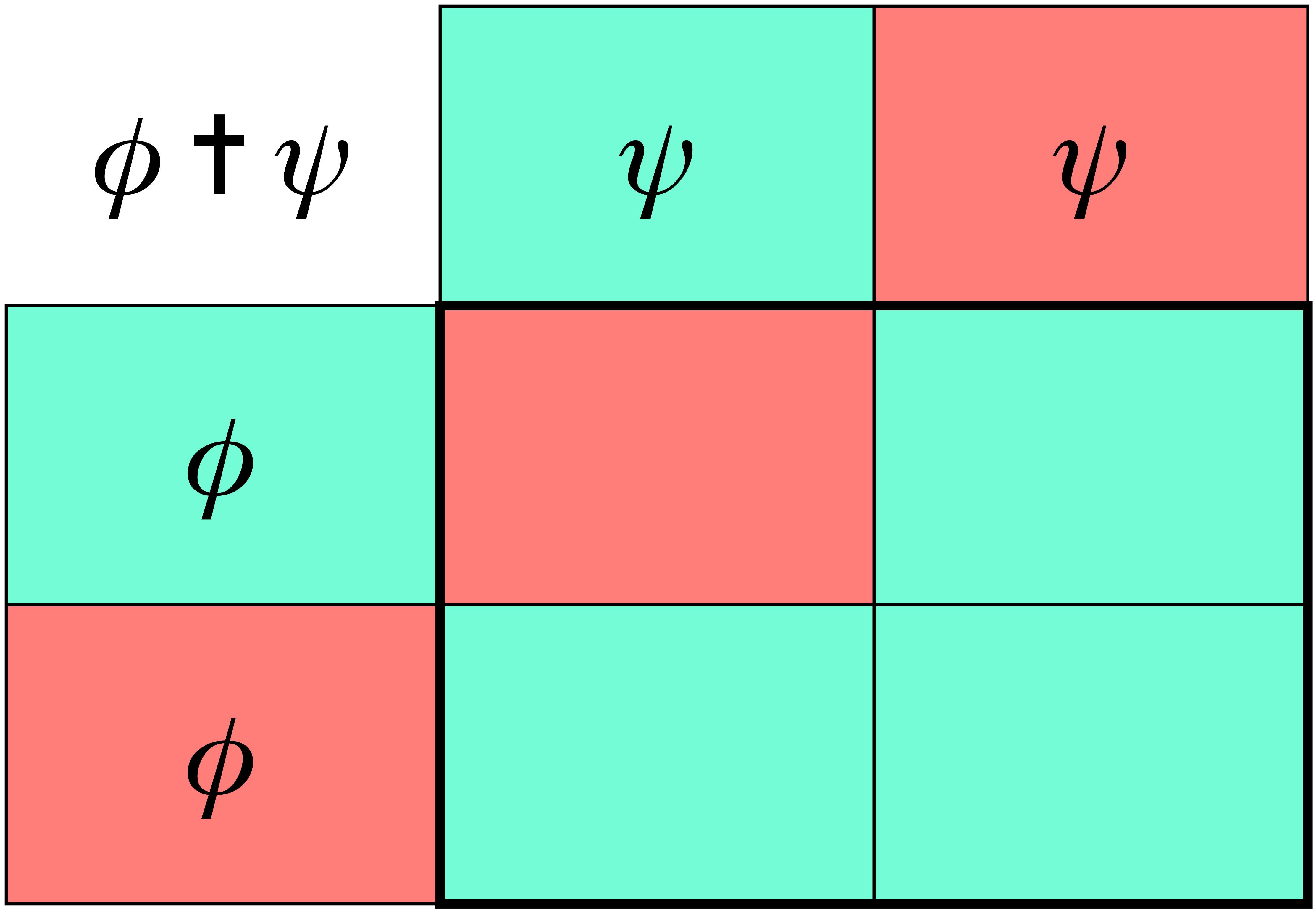
Expressivity

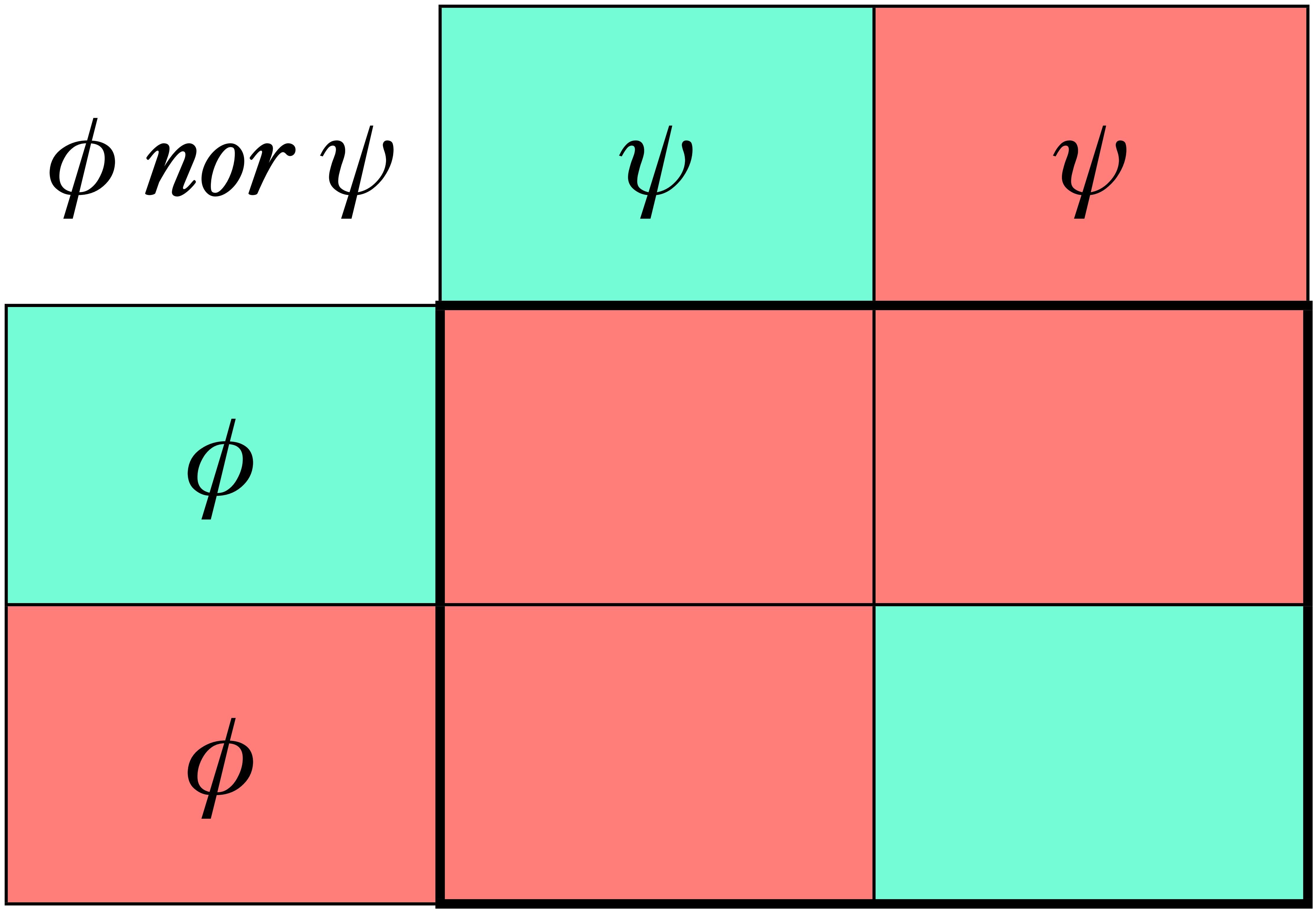
Every boolean function can be defined by using ¬, ∧ and ∨ only.

Also by ¬ and ∧.

Also by ¬ and →.

Also by †!


$\phi^\dagger\psi$
ψ
ψ
ϕ
ϕ


ϕ nor ψ
ψ
ψ
ϕ
ϕ

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

p	∨	(q	∧	r)
0	0	0	0	0
0	0	0	0	1
0	0	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	1	0	0
1	1	1	1	1

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0		0		0
0		0		1
0		1		0
0		1		1
1		0		0
1		0		1
1		1		0
1		1		1

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0	0	0		0
0	0	0		1
0	1	1		0
0	1	1		1
1	1	0		0
1	1	0		1
1	1	1		0
1	1	1		1

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0	0	0	0	0
0	0	0	0	1
0	1	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

p	∨	(q	∧	r)
0	0	0	0	0
0	0	0	0	1
0	0	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	1	0	0
1	1	1	1	1

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0	0	0	0	0
0	0	0	0	1
0	1	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

p	∨	(q	∧	r)
0	0	0	0	0
0	0	0	0	1
0	0	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	1	0	0
1	1	1	1	1
	2		1	

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0	0	0	0	0
0	0	0	0	1
0	1	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1
	1		2	

Truth tables

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

p	∨	(q	∧	r)
0	0	0	0	0
0	0	0	0	1
0	0	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	1	0	0
1	1	1	1	1
	2		1	

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0	0	0	0	0
0	0	0	0	1
0	1	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1
	1		2	

$$(p \vee q) \wedge r \vDash p \vee (q \wedge r)$$

Truth tables

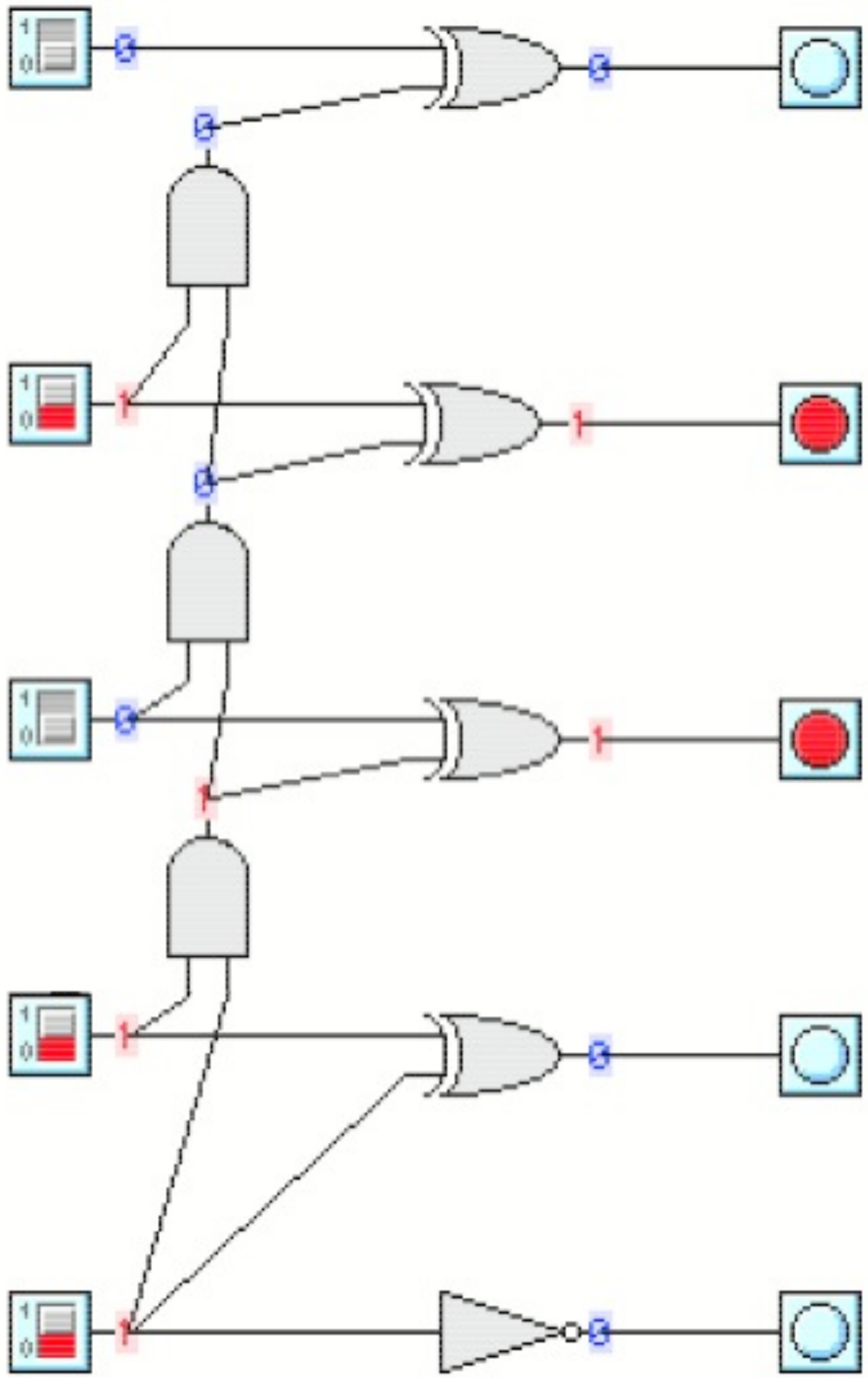
p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

p	∨	(q	∧	r)
0	0	0	0	0
0	0	0	0	1
0	0	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	1	0	0
1	1	1	1	1
	2		1	

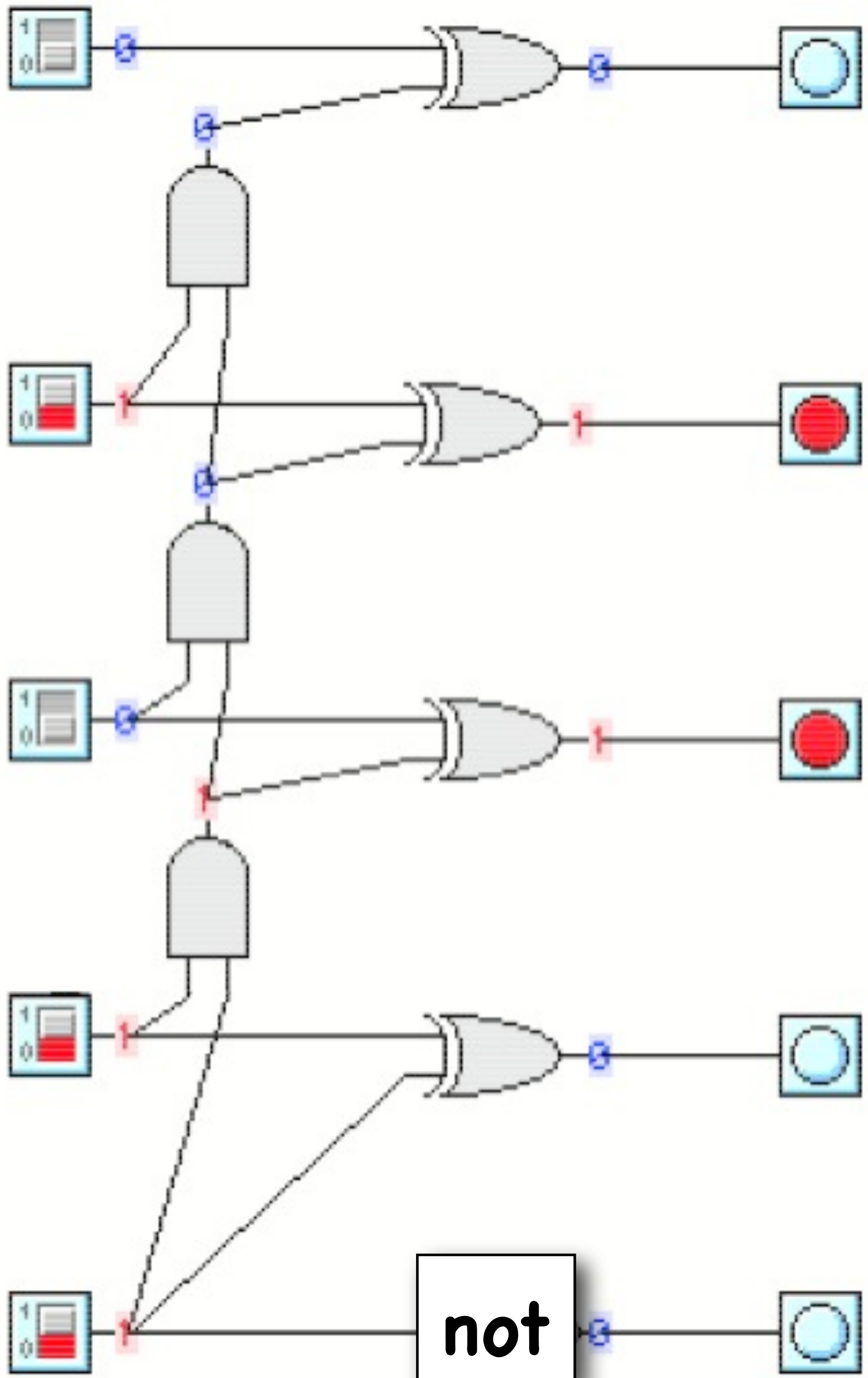
p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

(p	∨	q)	∧	r
0	0	0	0	0
0	0	0	0	1
0	1	1	0	0
0	1	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1
	1		2	

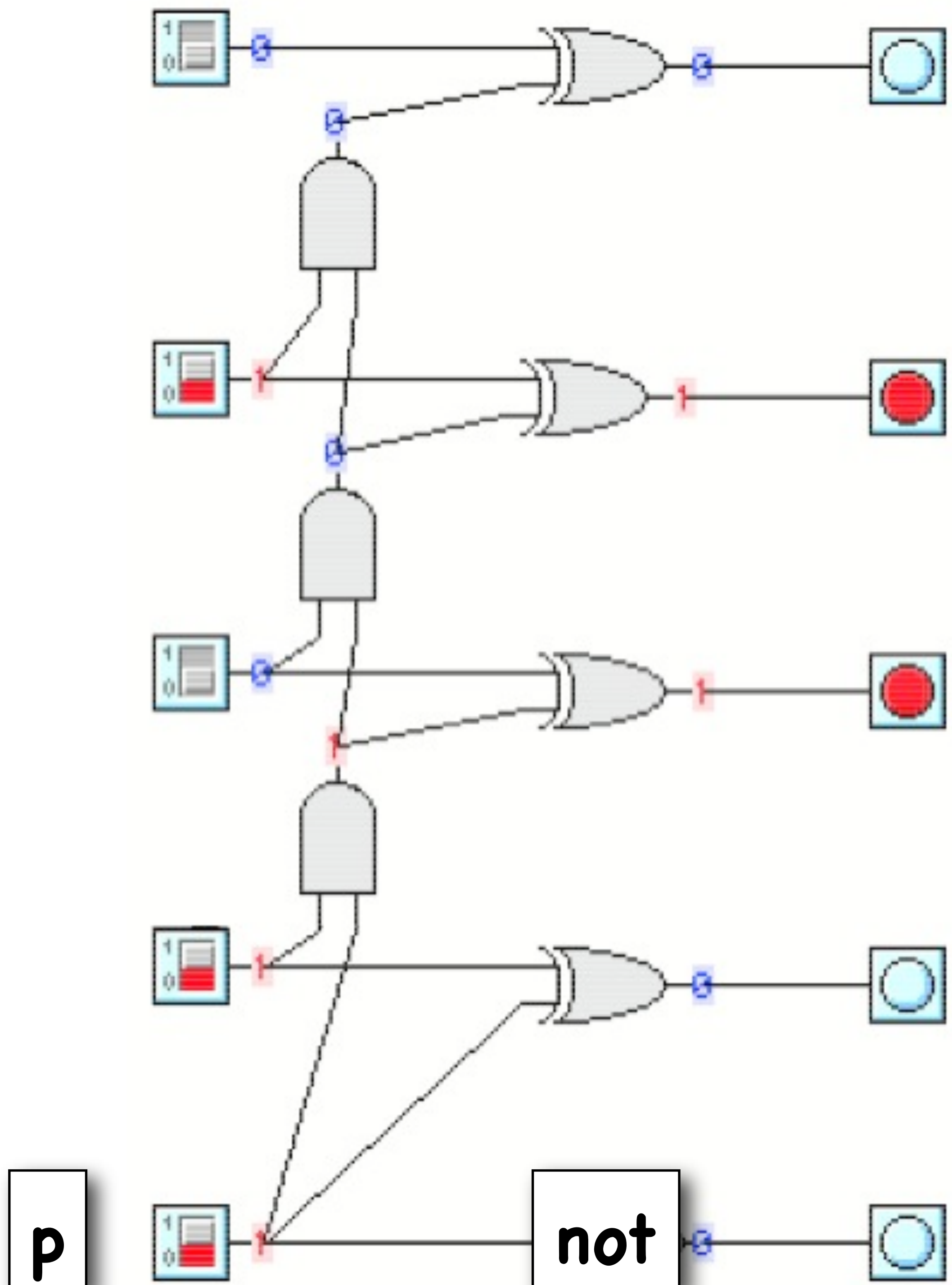
$$p \lor (q \land r) \not\models (p \lor q) \land r$$



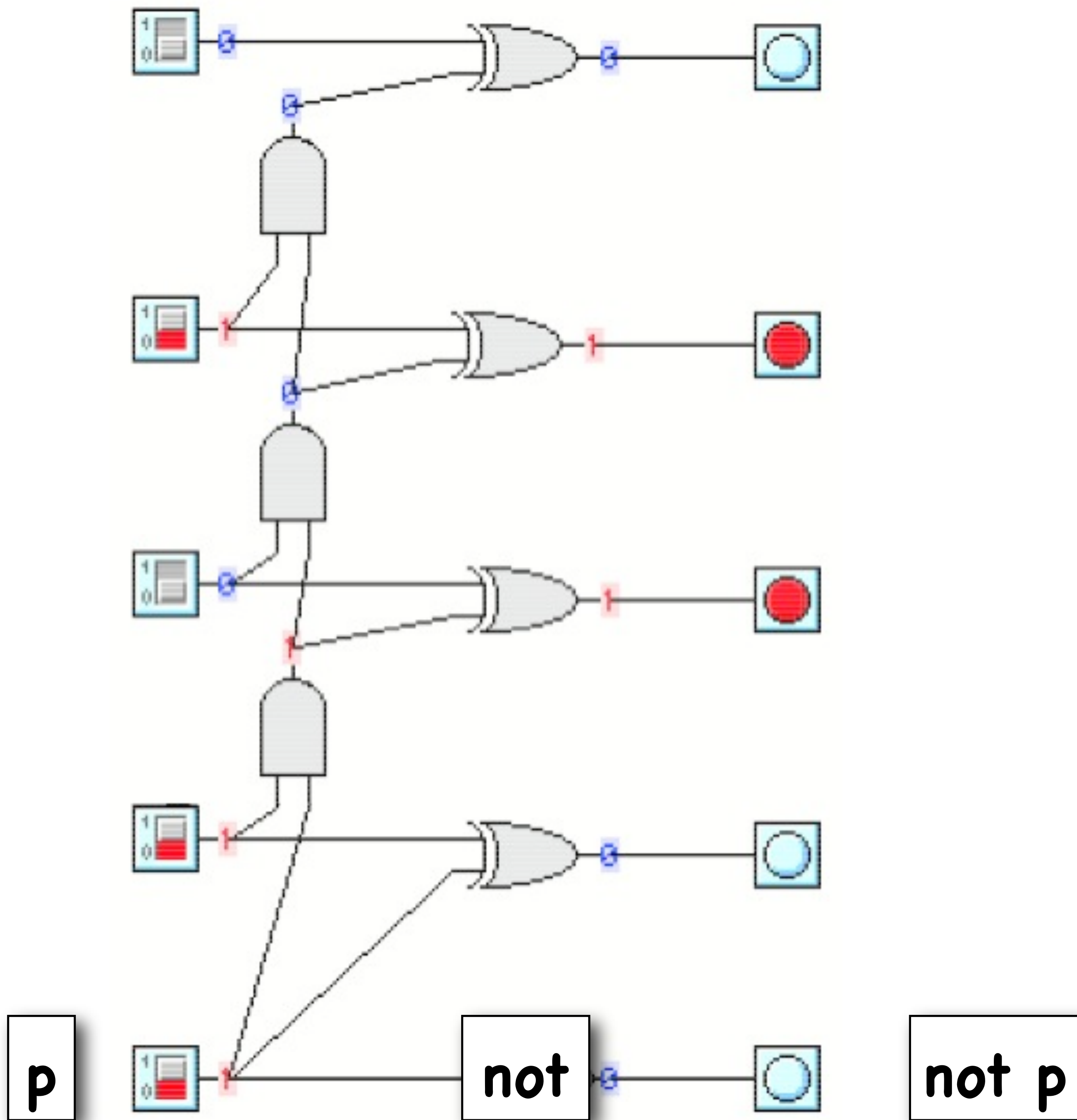
1948 (Shannon)



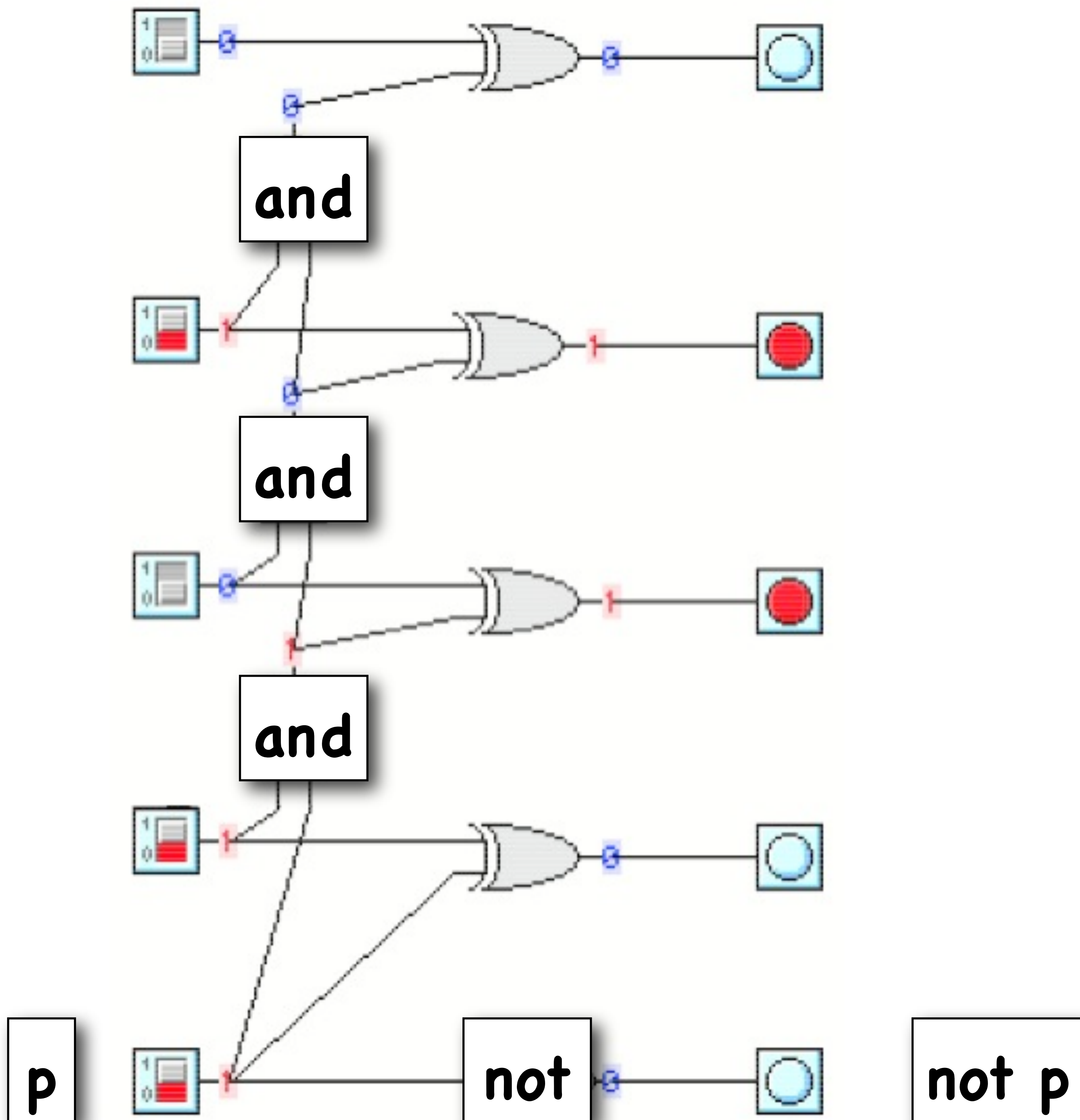
1948 (Shannon)



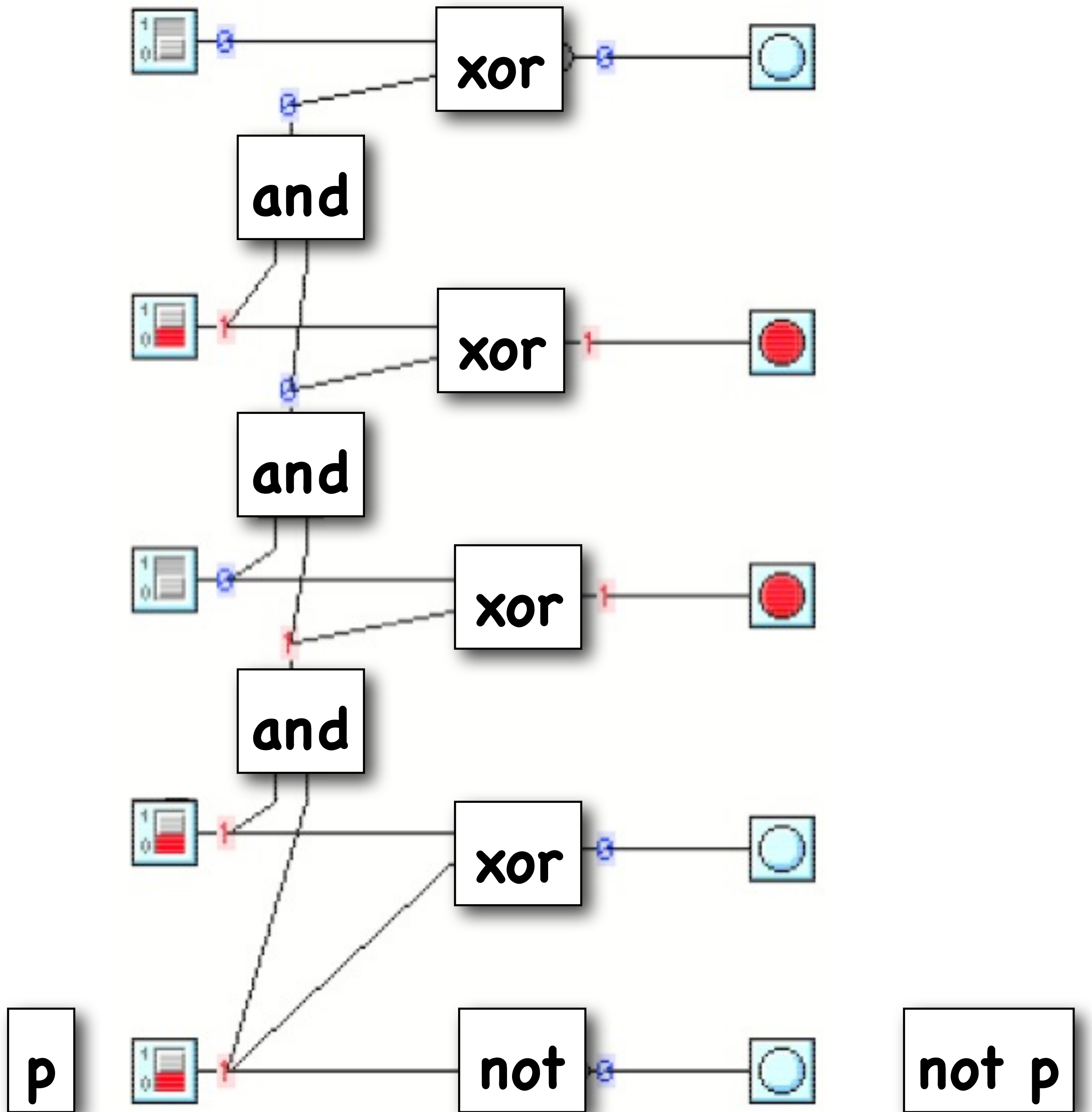
1948 (Shannon)



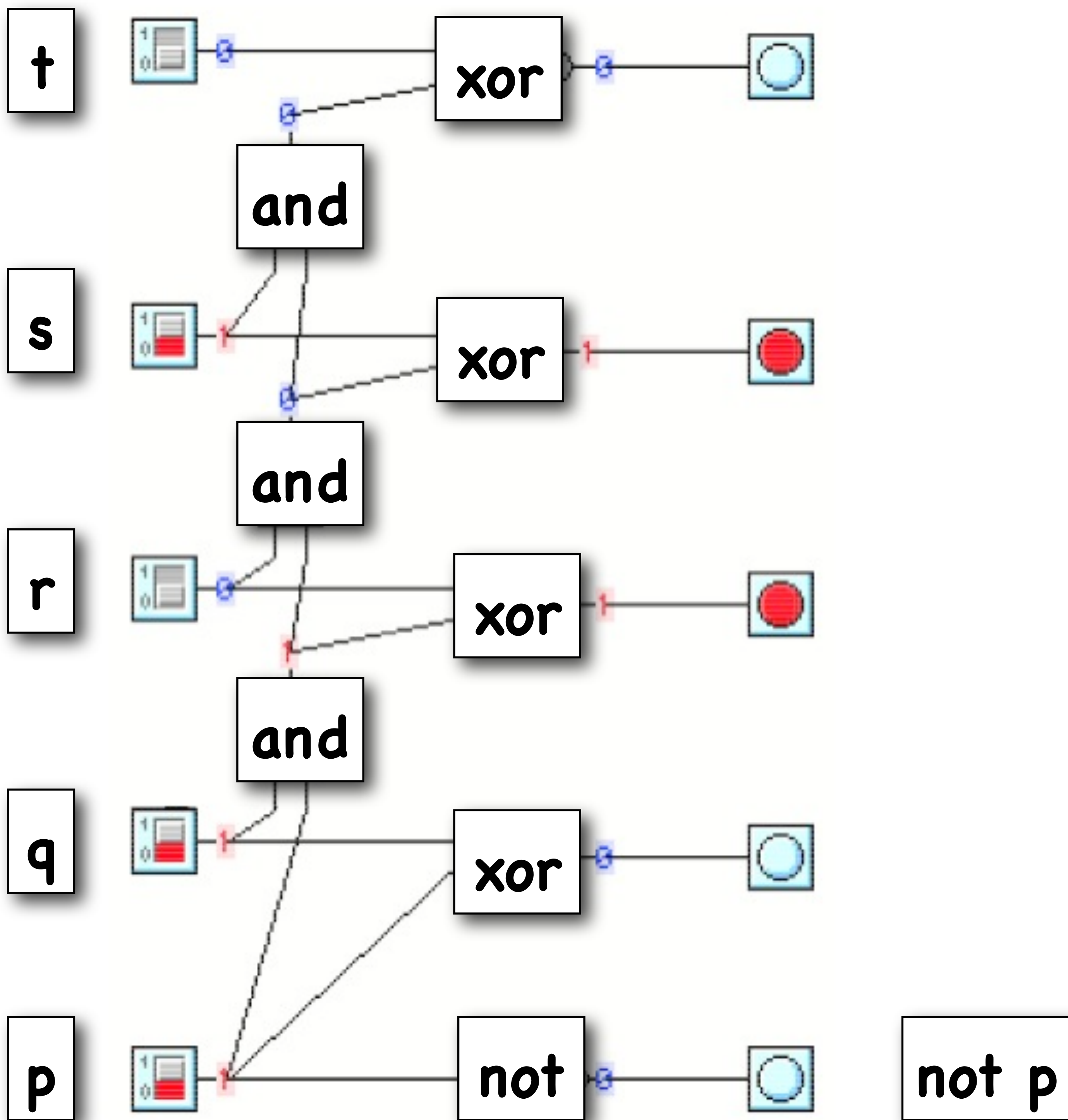
1948 (Shannon)



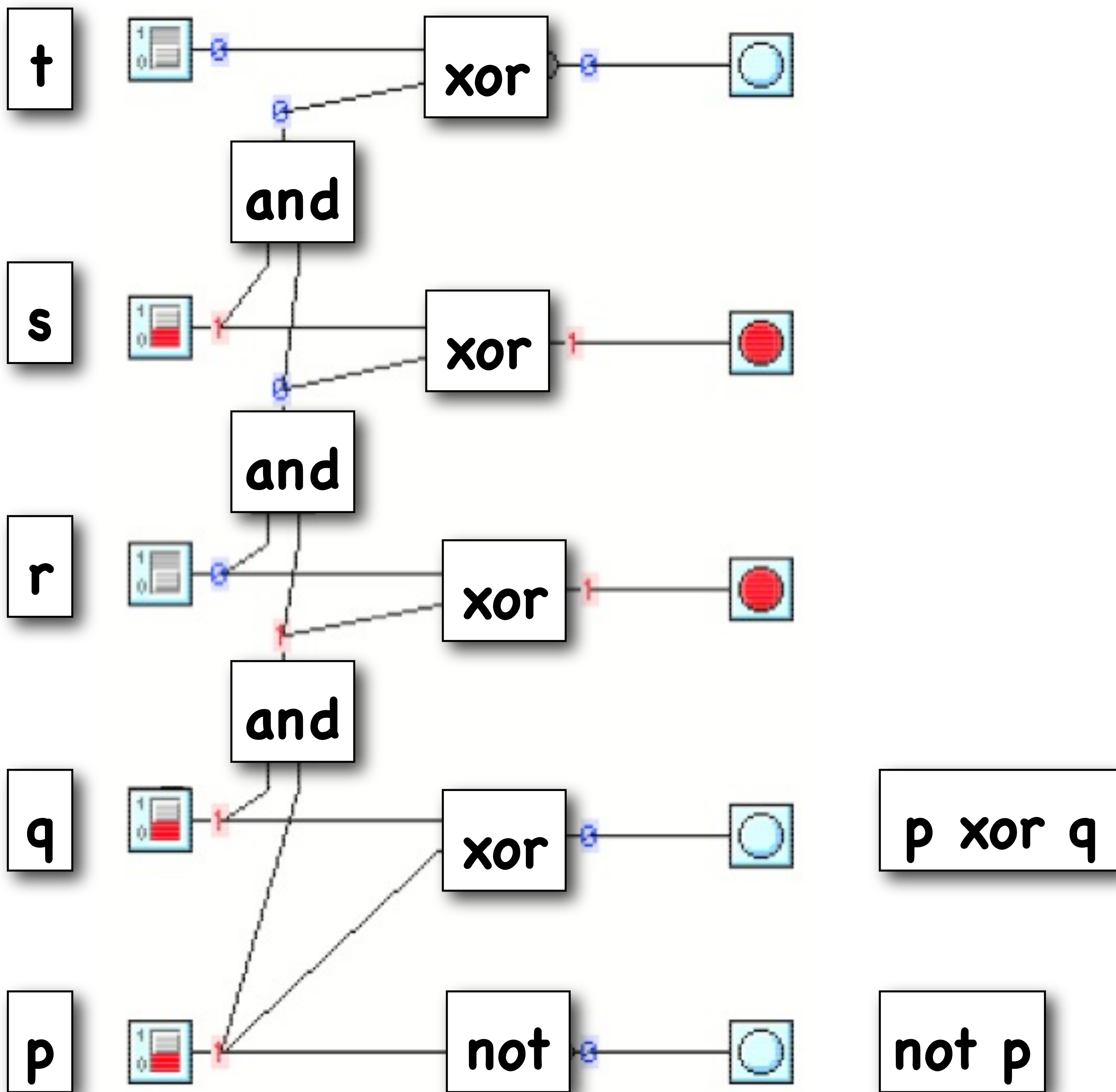
1948 (Shannon)



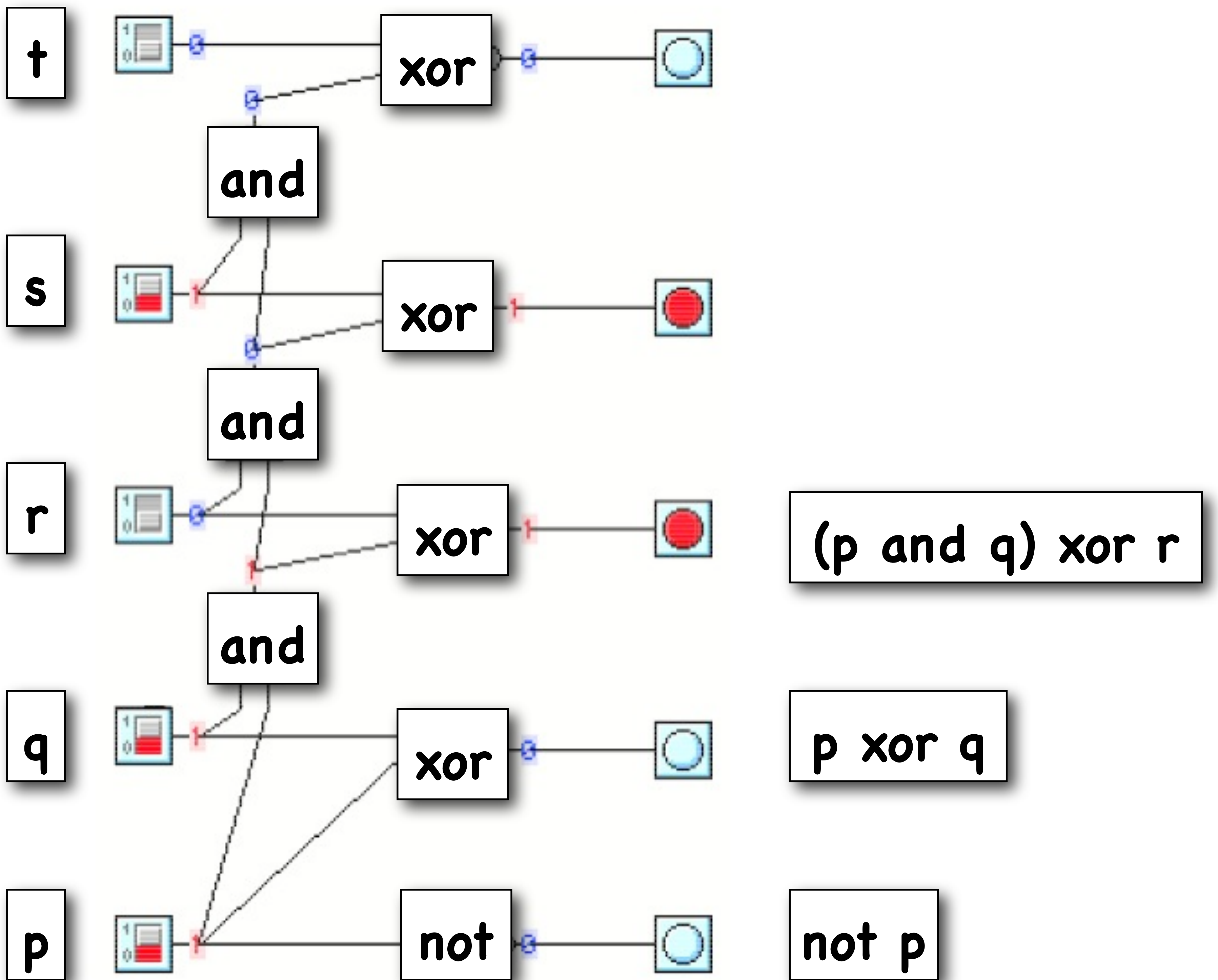
1948 (Shannon)



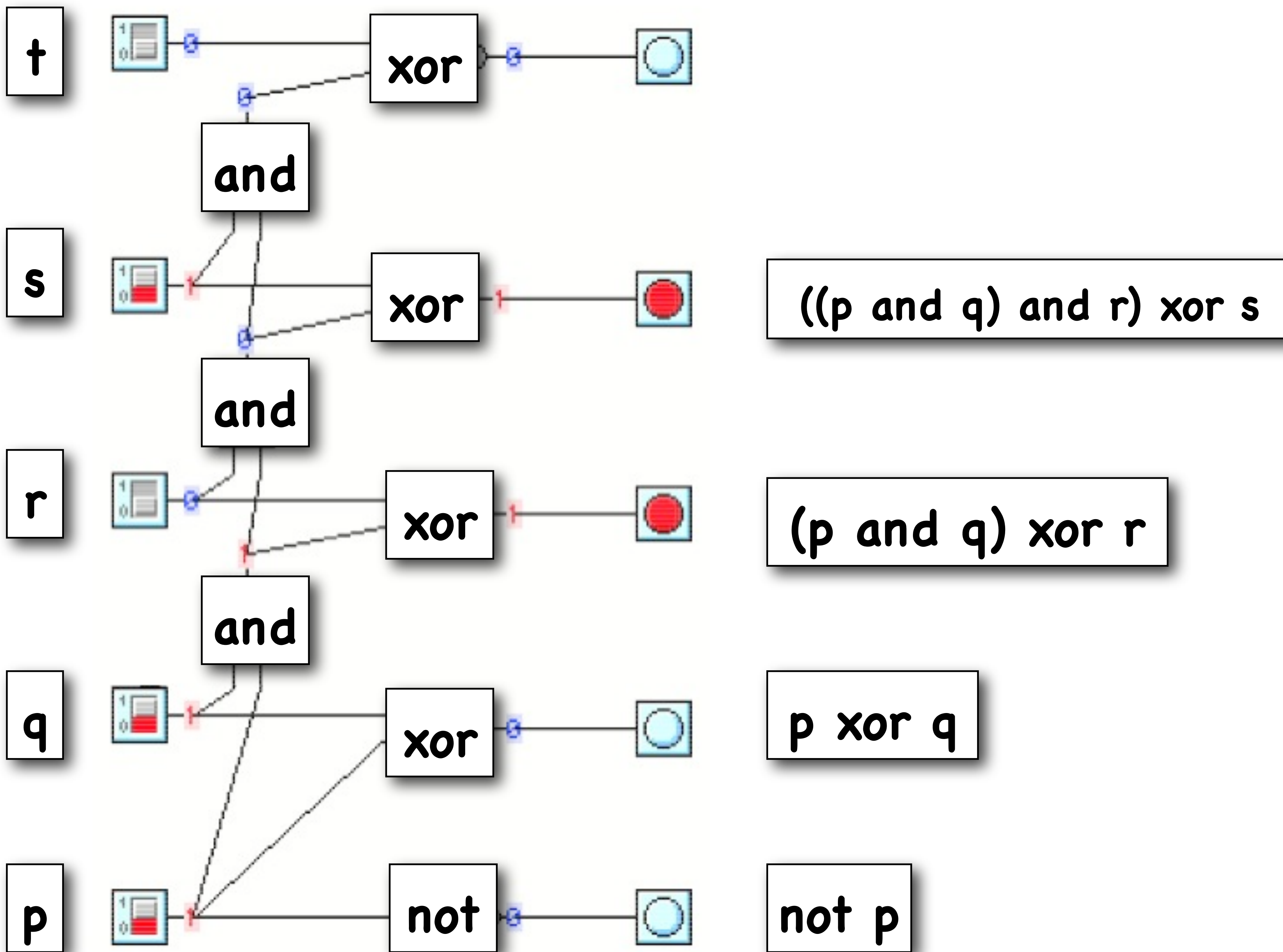
1948 (Shannon)



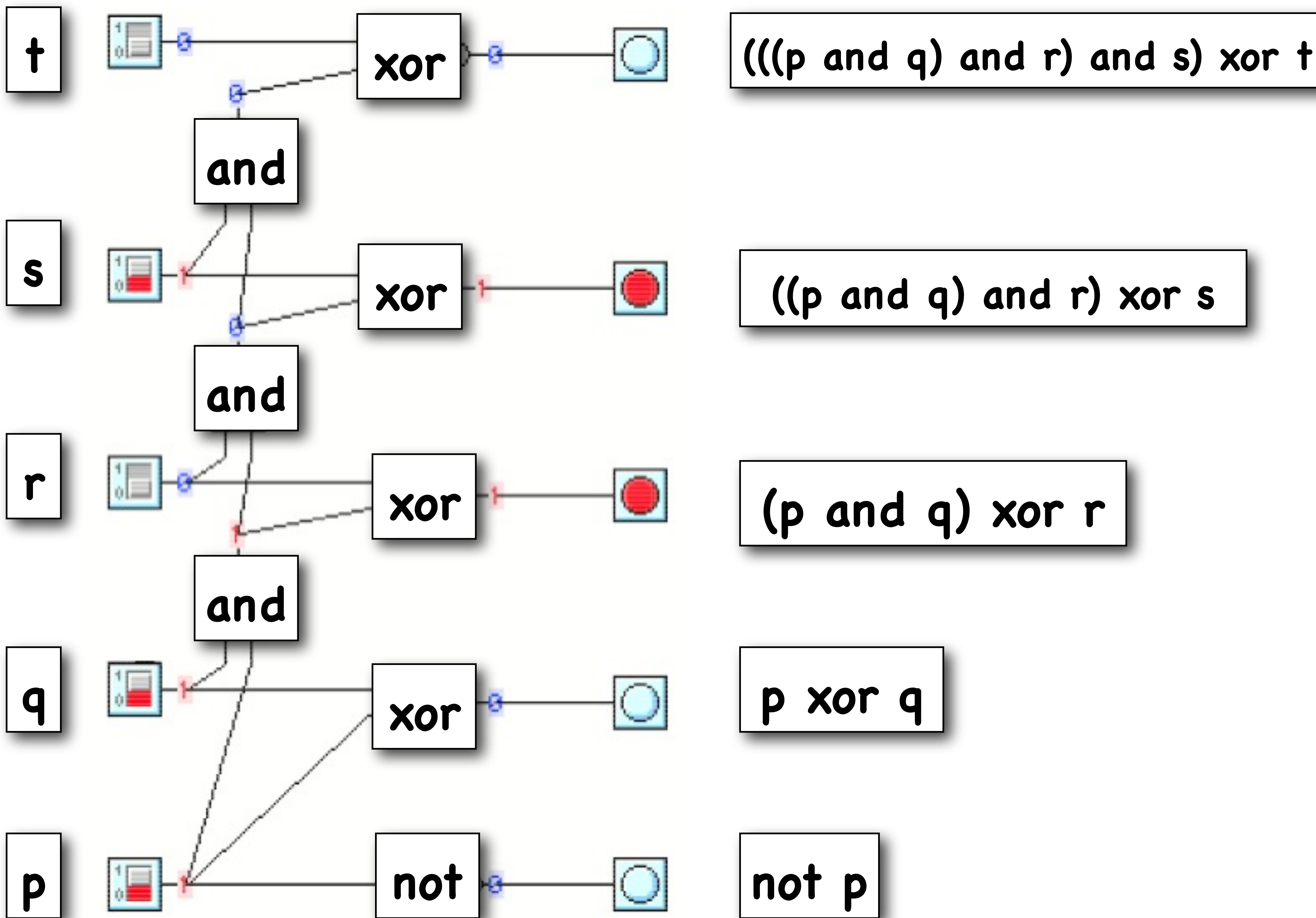
1948 (Shannon)



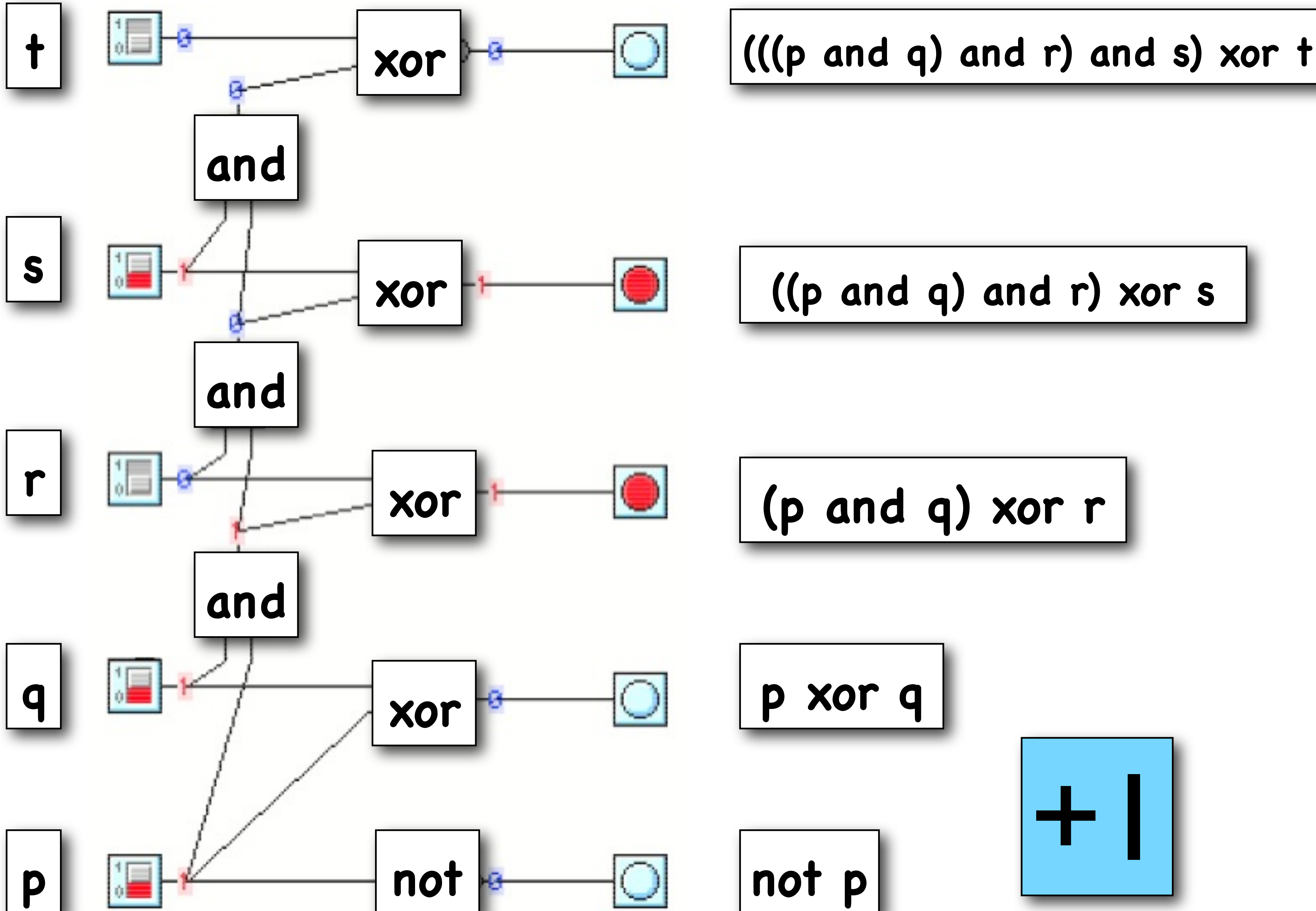
1948 (Shannon)



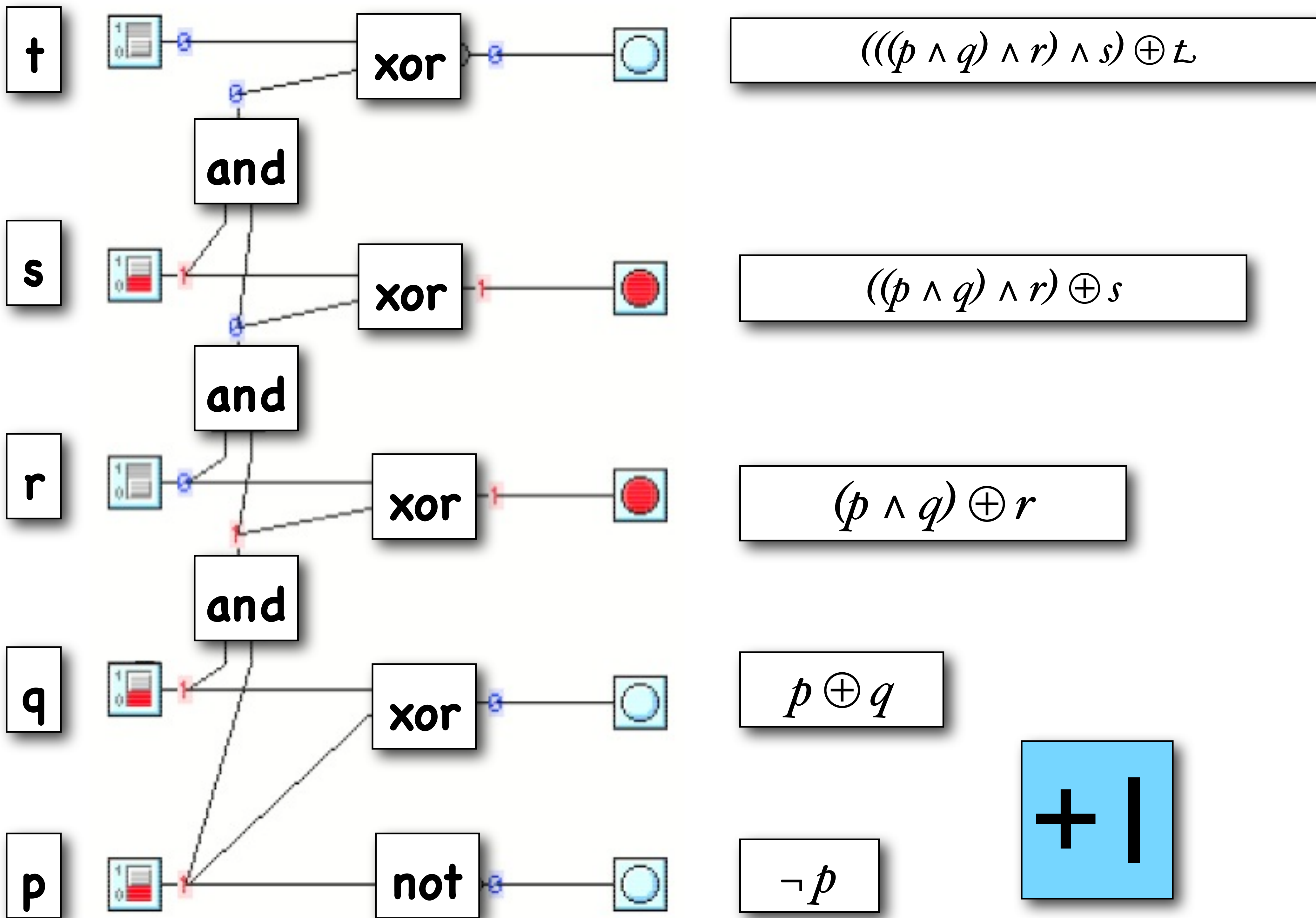
1948 (Shannon)



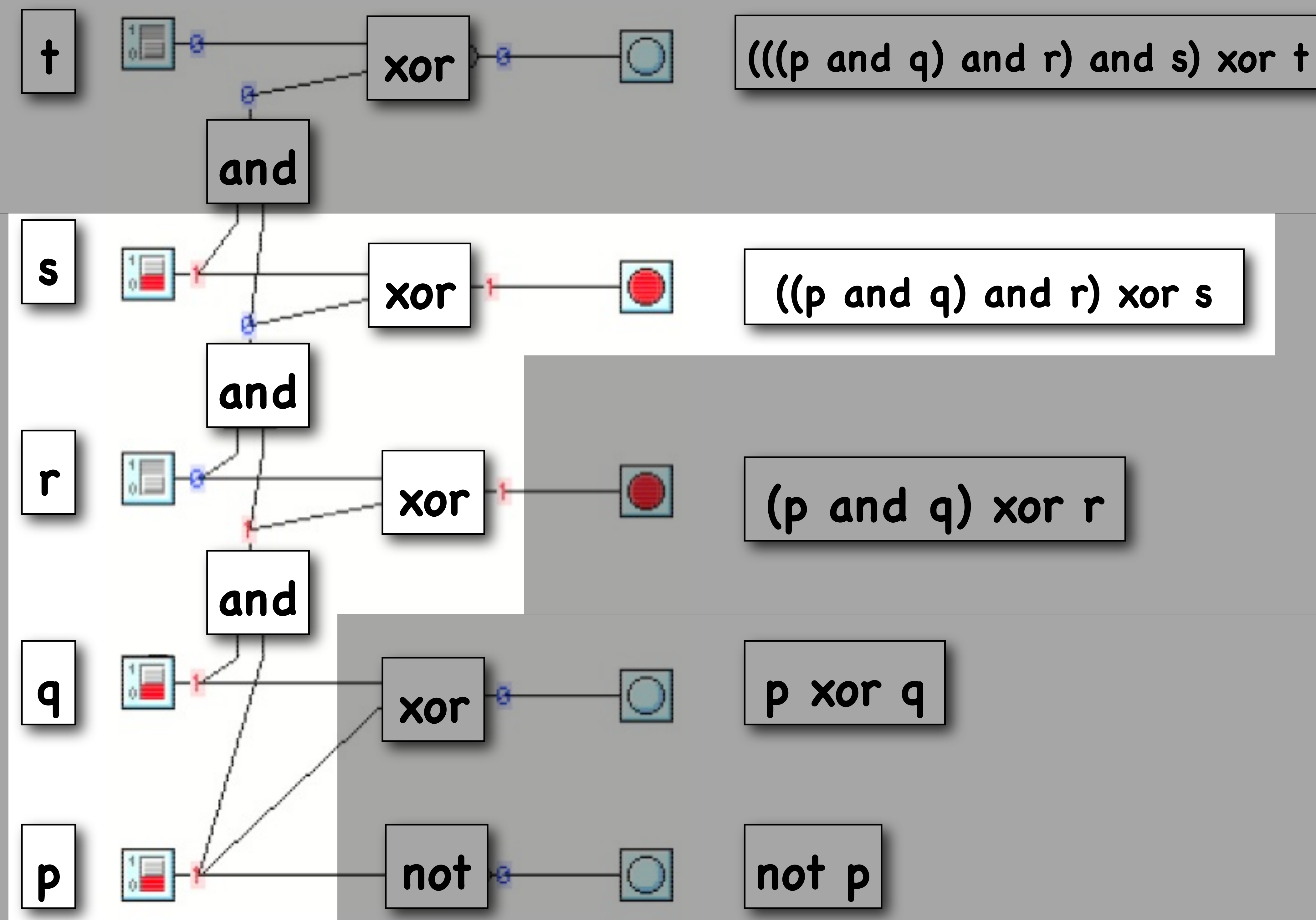
1948 (Shannon)



1948 (Shannon)



1948 (Shannon)

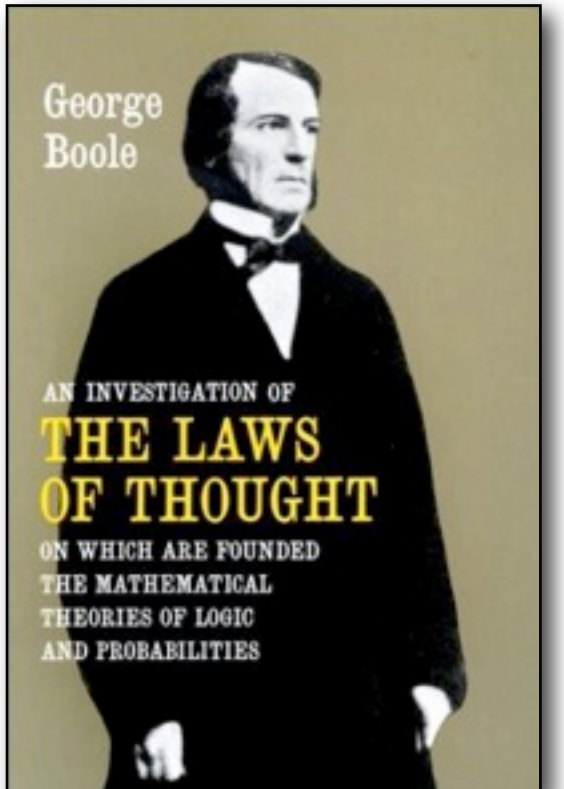

t
xor
(((p and q) and r) and s) xor t
and
s
xor
((p and q) and r) xor s
and
r
xor
(p and q) xor r
and
q
xor
p xor q
p
not
not p

p_1	p_2	...	p_n	f
0	0	...	0	1
0	0	...	1	1
...	...	...	0	0
...	...	...	...	...
1	1	...	0	0
1	1	...	1	1

p_1	p_2	...	p_n	f
0	0	...	0	1
0	0	...	1	1
...	...	...	0	0
...	...	...	...	...
1	1	...	0	0
1	1	...	1	1

$$\phi_f = (\neg p_1 \wedge \ldots \wedge \neg p_n) \vee$$
$$(\neg p_1 \wedge \ldots \wedge p_n) \vee$$
$$\ldots. \vee$$
$$(p_1 \wedge \ldots \wedge p_n)$$

$$(-x)\,x = 0 \qquad x0 = 0 \qquad x\,1 = x$$
$$(-x) + x = 1 \qquad x + 0 = x \qquad x + 1 = 1$$



$$xx = x \qquad x + x = x$$

Idempotence

$$xy = y\,x \qquad x + y = y + x$$

Commutativity

$$x(yz) = (xy)\,z \qquad x + (y + z) = (x + y) + z$$

Associativity

$$x\,(x + y) = x \qquad x + (x\,y) = x$$

Absorption

$$- - x = x$$

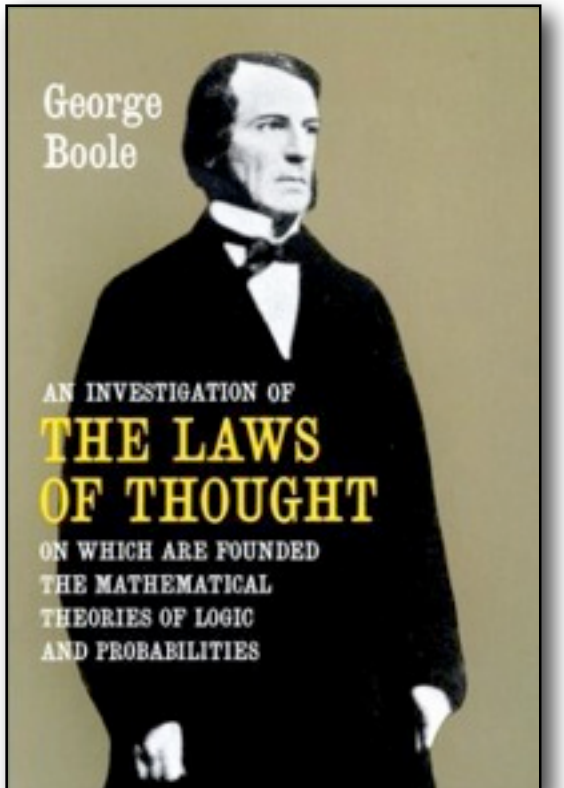
Double negation

$$-(xy) = (-x) + (-y) \qquad -(x + y) = (-x)\,(-y)$$

De Morgan

$$x\,(y + z) = (x\,y) + (x\,z) \qquad x + (y\,z) = (x + y)\,(x + z)$$

Distribution



(not x) and x = false
(not x) or x = true

x and false = false
x or false = x

x and true = x
x or true = true

x and x = x
x or x = x

x and y = y and x
x or y = y or x

x and (y and z) = (x and y) and z
x or (y or z) = (x or y) or z

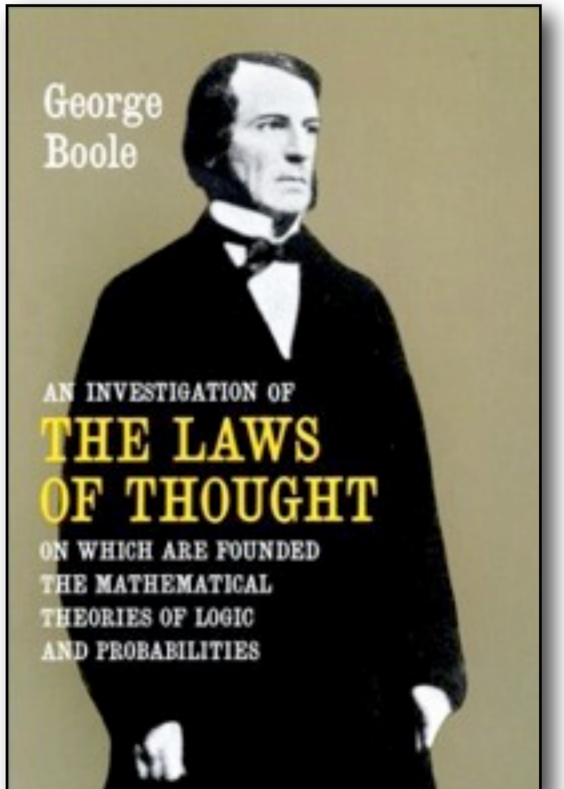
x and (x or y) = x
x or (x and y) = x

not not x = x

not (x and y) = (not x) or (not y)
not (x or y) = (not x) and (not y)

x and (y or z) = (x and y) or (x and z)
x or (y and z) = (x or y) and (x or z)

$$(\neg \phi) \wedge \phi = \mathit{false} \qquad \phi \wedge \mathit{false} = \mathit{false} \qquad \phi \wedge \mathit{true} = \phi$$
$$(\neg \phi) \vee \phi = \mathit{true} \qquad \phi \vee \mathit{false} = \phi \qquad \phi \vee \mathit{true} = \mathit{true}$$



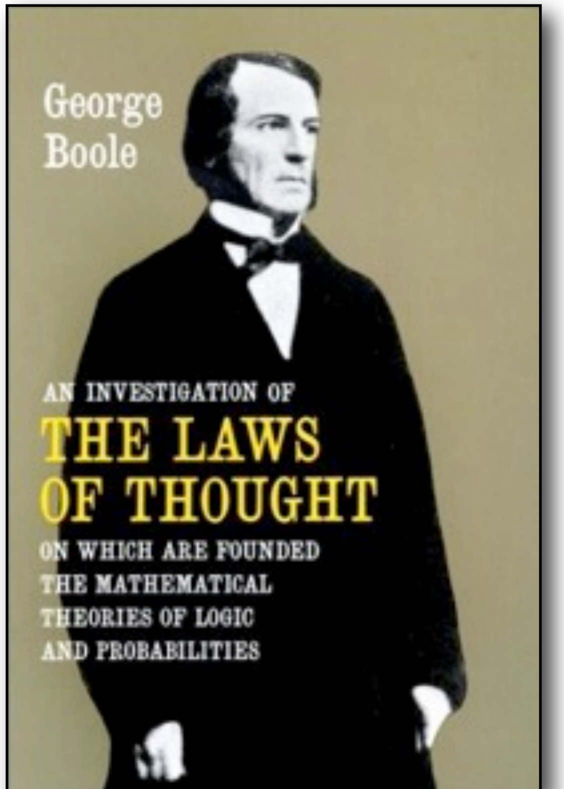
$$\phi \wedge \phi = \phi \qquad \phi \wedge \psi = \psi \wedge \phi$$
$$\phi \vee \phi = \phi \qquad \phi \vee \psi = \psi \vee \phi$$

$$\phi \wedge (\psi \wedge \chi) = (\phi \wedge \psi) \wedge \chi \qquad \phi \wedge (\phi \vee \psi) = \phi$$
$$\phi \vee (\psi \vee \chi) = (\phi \vee \psi) \vee \chi \qquad \phi \vee (\phi \wedge \psi) = \phi$$

$$\neg \neg \phi = \phi$$

$$\neg (\phi \wedge \psi) = (\neg \phi) \vee (\neg \psi) \qquad \phi \wedge (\psi \vee \chi) = (\phi \wedge \psi) \vee (\phi \wedge \chi)$$
$$\neg (\phi \vee \psi) = (\neg \phi) \wedge (\neg \psi) \qquad \phi \vee (\psi \wedge \chi) = (\phi \vee \psi) \wedge (\phi \vee \chi)$$

$$(\neg \phi) \land \phi = \bot \qquad \phi \land \bot = \bot \qquad \phi \land \top = \phi$$
$$(\neg \phi) \lor \phi = \top \qquad \phi \lor \bot = \phi \qquad \phi \lor \top = \top$$



$$\phi \land \phi = \phi \qquad \phi \land \psi = \psi \land \phi$$
$$\phi \lor \phi = \phi \qquad \phi \lor \psi = \psi \lor \phi$$

$$\phi \land (\psi \land \chi) = (\phi \land \psi) \land \chi \qquad \phi \land (\phi \lor \psi) = \phi$$
$$\phi \lor (\psi \lor \chi) = (\phi \lor \psi) \lor \chi \qquad \phi \lor (\phi \land \psi) = \phi$$

$$\neg \neg \phi = \phi$$

$$\neg (\phi \land \psi) = (\neg \phi) \lor (\neg \psi) \qquad \phi \land (\psi \lor \chi) = (\phi \land \psi) \lor (\phi \land \chi)$$
$$\neg (\phi \lor \psi) = (\neg \phi) \land (\neg \psi) \qquad \phi \lor (\psi \land \chi) = (\phi \lor \psi) \land (\phi \lor \chi)$$

$$\phi \wedge \psi \models \phi$$

$$\phi \wedge \psi \models \phi$$

$$(\phi \wedge \psi) \wedge \phi = \phi \wedge \psi$$

$$\phi \wedge \psi \vDash \phi$$

$$(\phi \wedge \psi) \wedge \phi =$$
$$(\psi \wedge \phi) \wedge \phi =$$
$$\psi \wedge (\phi \wedge \phi) =$$
$$\psi \wedge \phi =$$
$$\phi \wedge \psi$$

$$(\phi \wedge \psi) \wedge \phi = \phi \wedge \psi$$

$$\phi \wedge \psi \vDash \phi$$

$$(\phi \wedge \psi) \wedge \phi =$$

$$(\psi \wedge \phi) \wedge \phi =$$

$$\psi \wedge (\phi \wedge \phi) =$$

$$\psi \wedge \phi =$$

$$\phi \wedge \psi$$

$$(\phi \wedge \psi) \wedge \phi = \phi \wedge \psi$$

1. $\phi \rightarrow (\psi \rightarrow \phi)$
2. $(\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi))$
3. $(\neg\phi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \phi)$

1., 2. and 3. are theorems (for every ϕ, ψ and χ)

Every instance of a theorem is a theorem.

If ϕ and $\phi \rightarrow \psi$ are theorems then so is ψ (modus ponens).

Jan Łukasiewicz

Part III chapter 9

Natural Deduction, Proofs and Arguments

Gerhard Gentzen

$$\begin{array}{|c|}\hline \phi \\ \cdot \\ \cdot \\ \psi \\ \hline \end{array}$$

$$\phi \rightarrow \psi$$

Gerhard Gentzen

$$\begin{array}{|c|}\hline \phi \\ \vdots \\ \psi \\ \hline \end{array} \qquad \begin{array}{|c|}\hline \phi \\ \vdots \\ \bot \\ \hline \end{array}$$

$$\phi \rightarrow \psi \qquad \neg \phi$$

Gerhard Gentzen

ϕ	ϕ	$\neg\phi$
$\vdots$	$\vdots$	$\vdots$
ψ	$\perp$	$\perp$
$\phi \rightarrow \psi$	$\neg\phi$	ϕ

Gerhard Gentzen

ϕ	ϕ	$\neg\phi$
.	.	.
.	.	.
ψ	$\bot$	$\bot$
$\phi \rightarrow \psi$	$\neg\phi$	ϕ

+ modus ponens = complete deduction system for propositional logic!

Gerhard Gentzen

ϕ	ϕ	$\neg\phi$
$\vdots$	$\vdots$	$\vdots$
ψ	$\bot$	$\bot$
$\phi \rightarrow \psi$	$\phi \rightarrow \bot$	ϕ

	$\vdots$	
$\vdots$	$\phi \rightarrow \psi$	$\vdots$
ψ	$\vdots$	$\bot$
$\vdots$	ϕ	$\vdots$
$\phi \rightarrow \psi$	$\vdots$	ϕ
	ψ	

$\rightarrow$*Intro* $\rightarrow$*Elim* $\bot$*Elim*

Gerhard Gentzen

$\to$*Intro*	$\to$*Elim*	$\bot$*Elim*
$[\phi \; \vdots \; \psi]$		$[\neg\phi \; \vdots \; \bot]$
$\phi\to\psi$		ϕ
$\vdots$	$\vdots$	$\vdots$
ψ	$\phi\to\psi$	$\bot$
$\vdots$	$\vdots$	$\vdots$
$\phi\to\psi$	ϕ	ϕ
	$\vdots$	
	ψ	

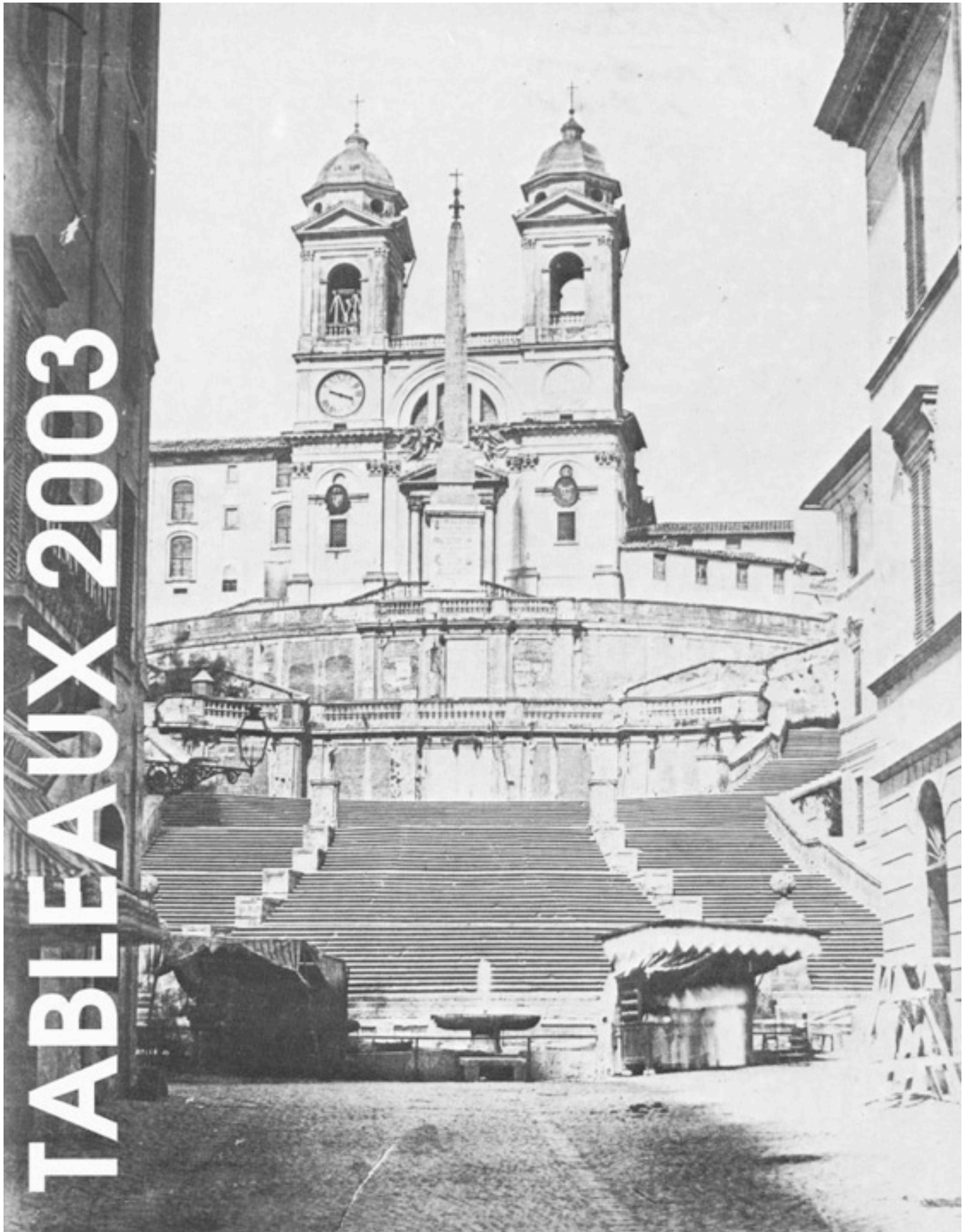
$\to$*Intro* $\to$*Elim* $\bot$*Elim*

Gerhard Gentzen

Part III chapter 8

Tableaux, Testing Validity

Evert Beth


TABLEAUX 2003

$$p \wedge (q \vee r) \overset{?}{\vDash} (p \wedge q) \vee r$$

$$p \land (q \lor r) \bullet (p \land q) \lor r$$

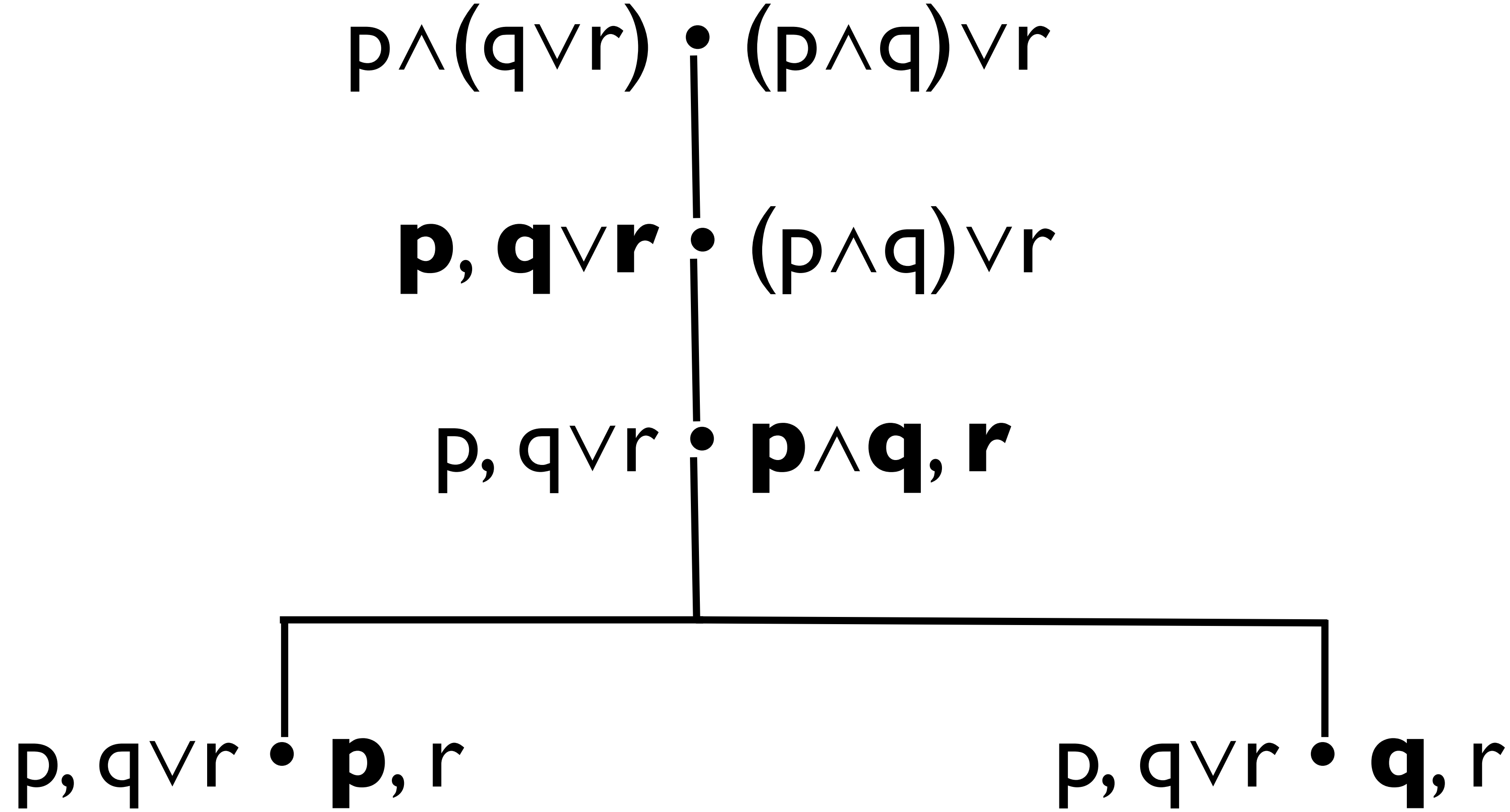
$p \wedge (q \vee r) \bullet (p \wedge q) \vee r$

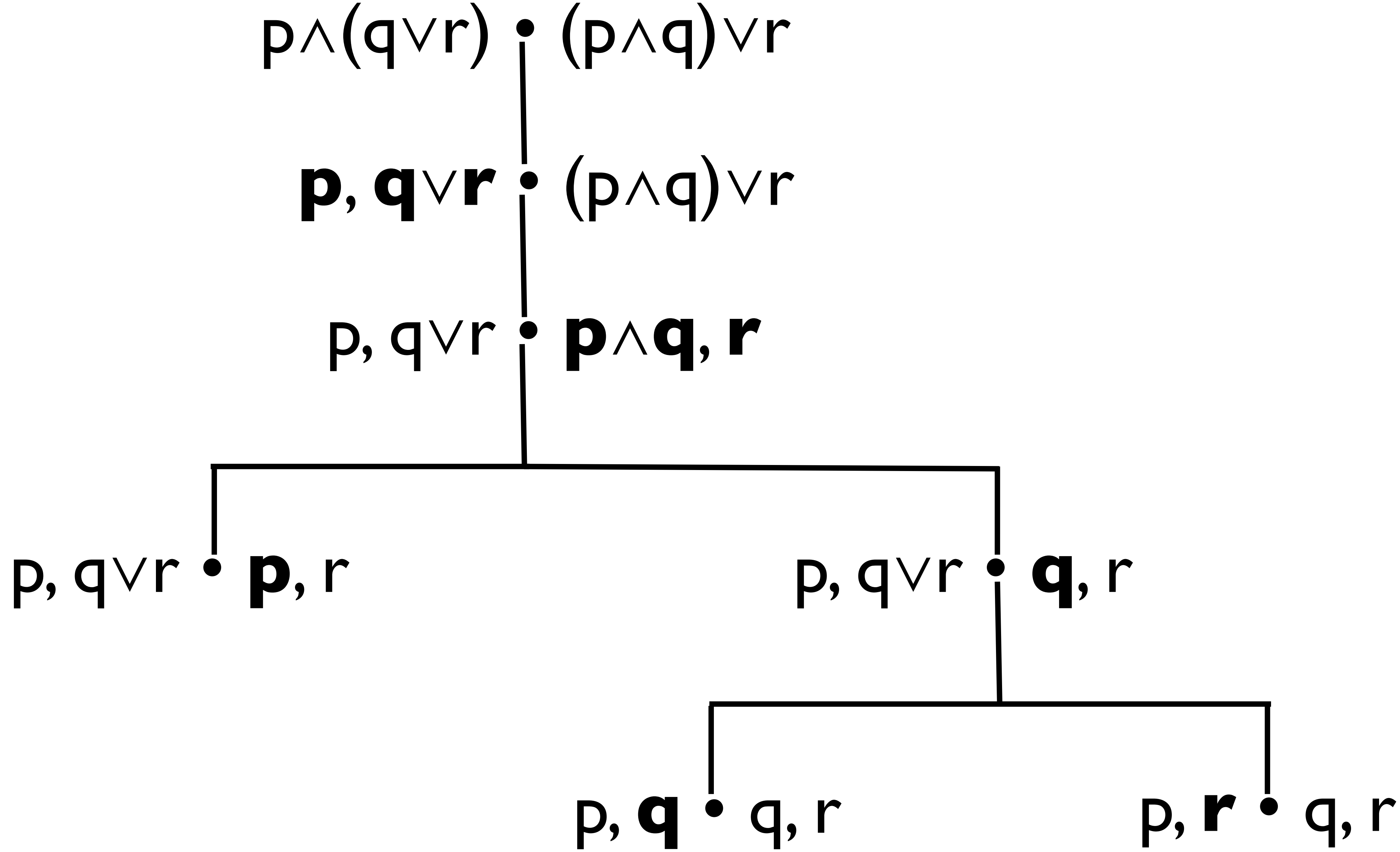
$\mathbf{p}, \mathbf{q} \vee \mathbf{r} \bullet (p \wedge q) \vee r$

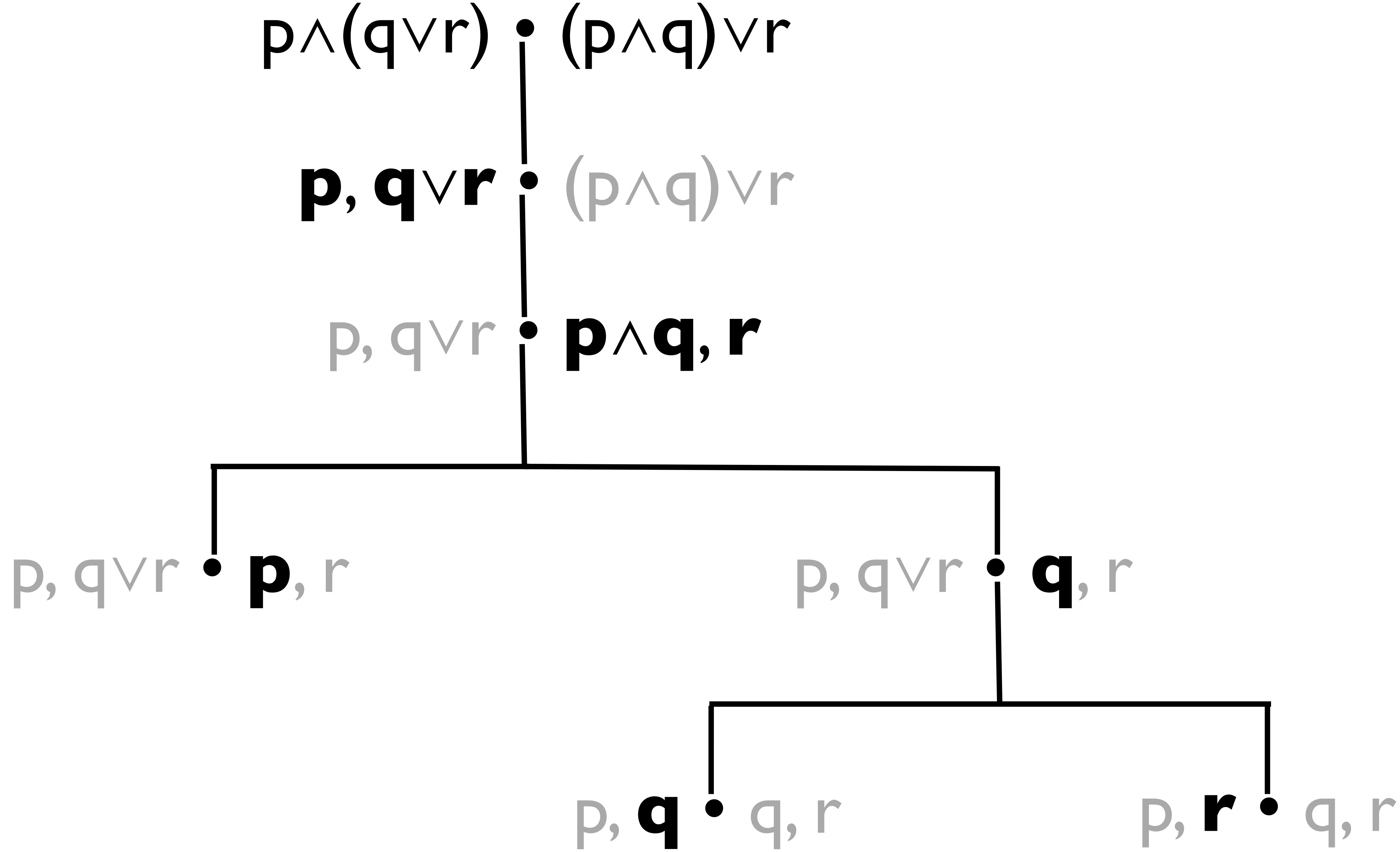
$p \wedge (q \vee r) \bullet (p \wedge q) \vee r$

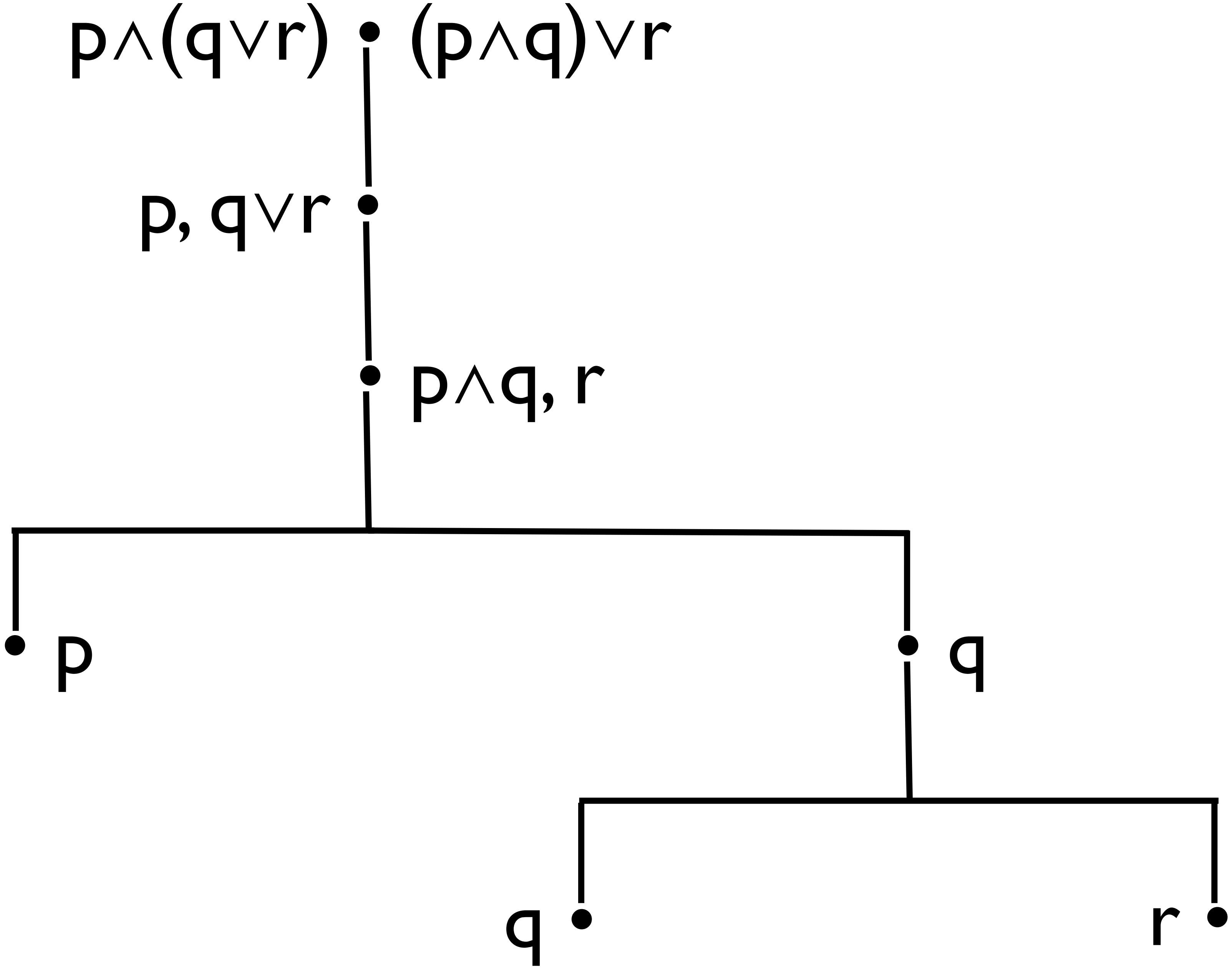
$\mathbf{p}, \mathbf{q} \vee \mathbf{r} \bullet (p \wedge q) \vee r$

$p, q \vee r \bullet \mathbf{p} \wedge \mathbf{q}, \mathbf{r}$


p∧(q∨r) • (p∧q)∨r
p, q∨r • (p∧q)∨r
p, q∨r • p∧q, r
p, q∨r • p, r
p, q∨r • q, r


p∧(q∨r) • (p∧q)∨r
p, **q**∨**r** • (p∧q)∨r
p, q∨r • **p**∧**q**, **r**
p, q∨r • **p**, r
p, q∨r • **q**, r
p, **q** • q, r
p, **r** • q, r


p∧(q∨r) • (p∧q)∨r
p, q∨r • (p∧q)∨r
p, q∨r • p∧q, r
p, q∨r • p, r
p, q∨r • q, r
p, q • q, r
p, r • q, r


p∧(q∨r)
(p∧q)∨r
p, q∨r
p∧q, r
p
q
q
r

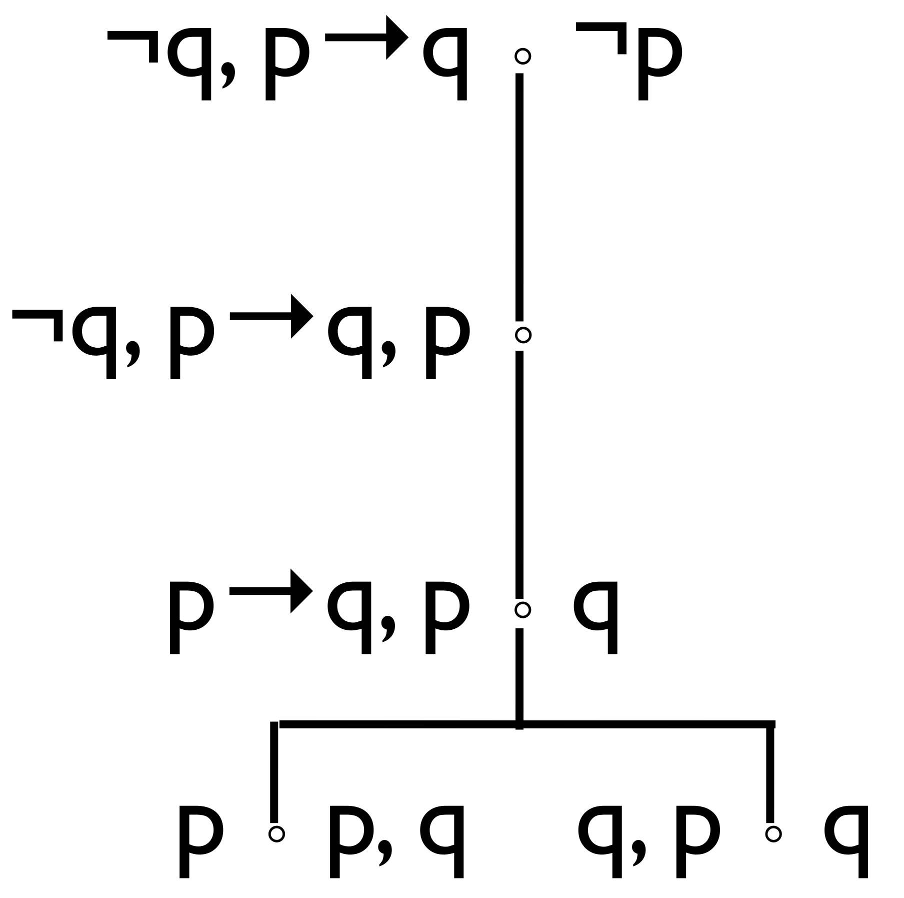
$$\neg q, p \to q \circ \neg p$$

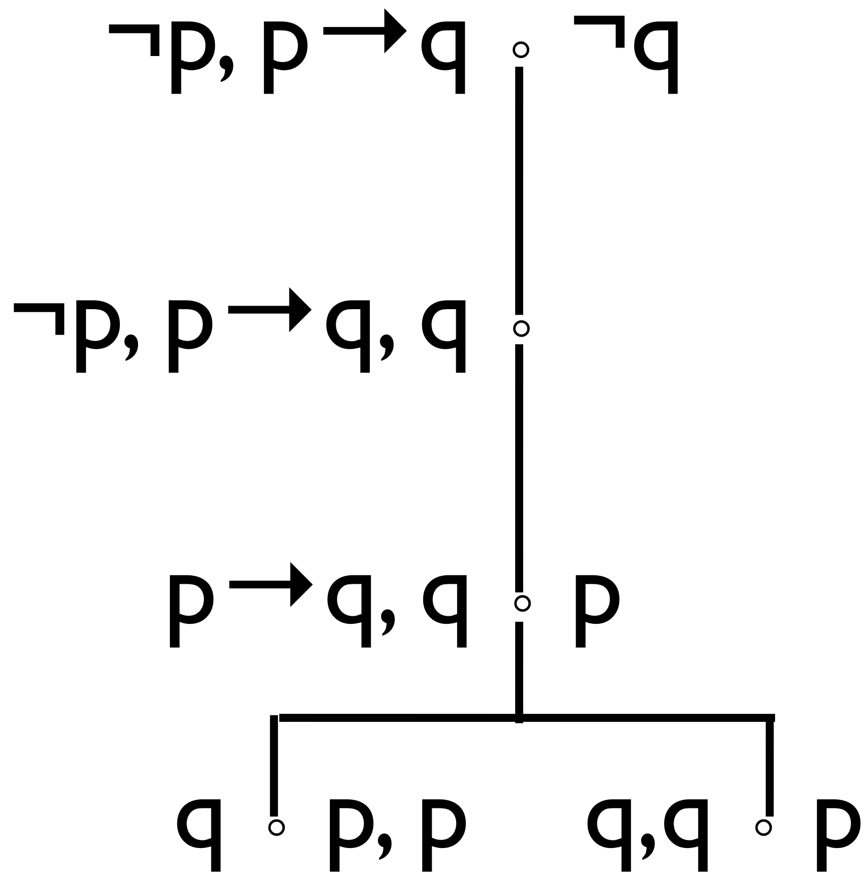
$$\neg q, p \to q, p \circ$$

$$p \to q, p \circ q$$

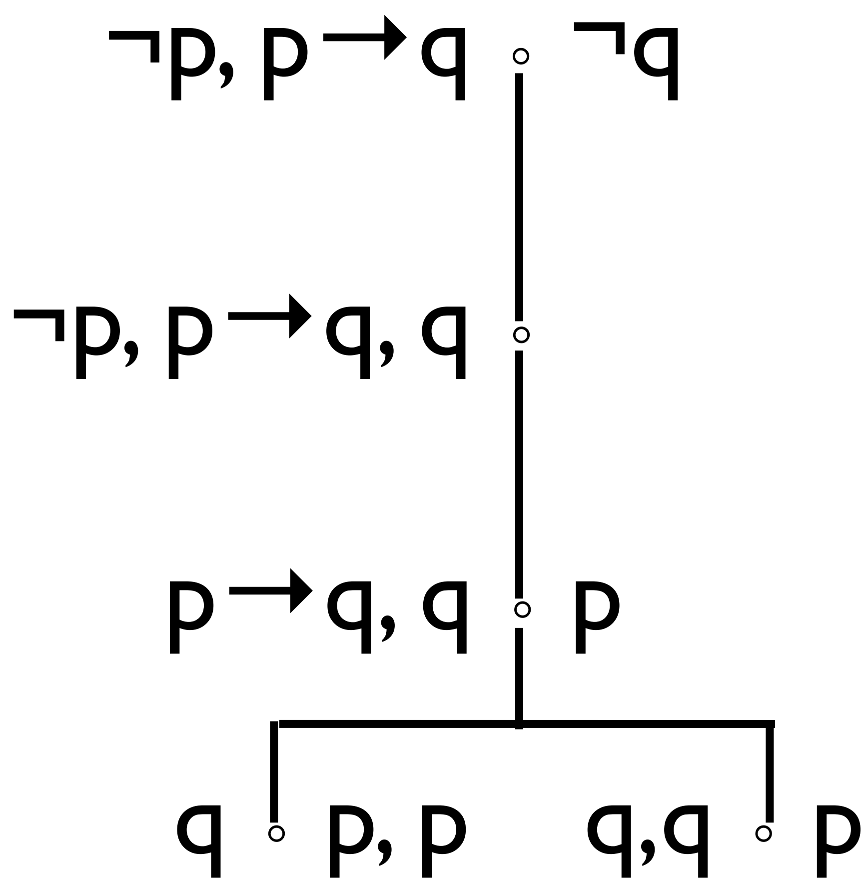
$$\underline{p \circ p, q} \qquad \underline{q, p \circ q}$$

$$\neg q, p \to q \models \neg p$$

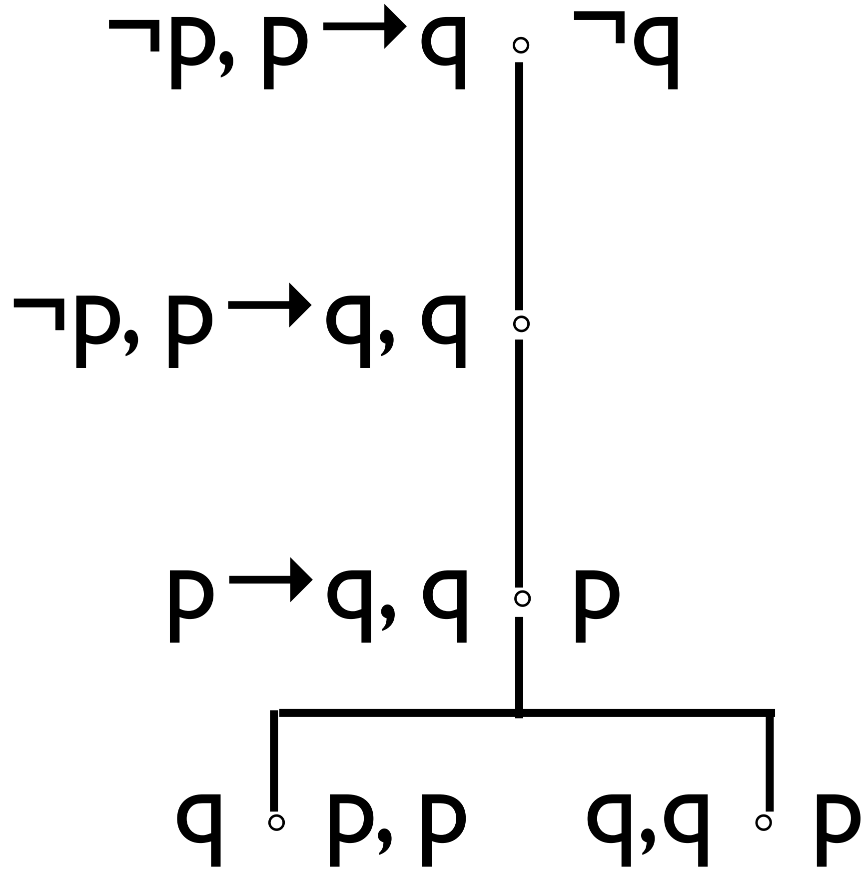



¬p, p→q ∘ ¬q
¬p, p→q, q ∘
p→q, q ∘ p
q ∘ p, p
q,q ∘ p

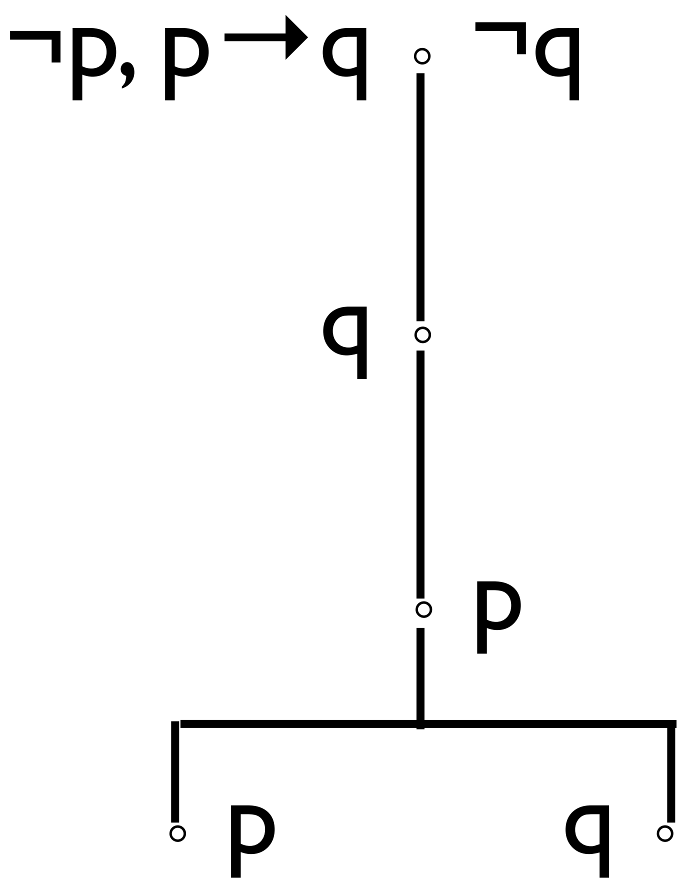
$$\neg p,\ p \to q \not\models \neg q$$



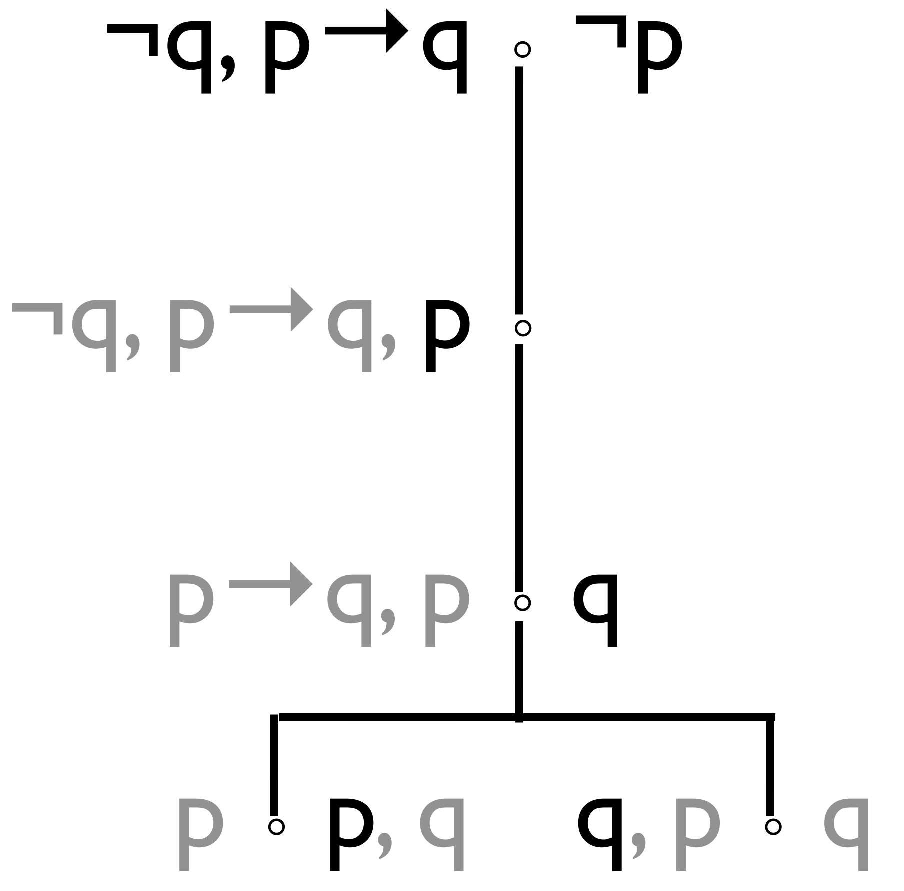
$$\neg p,\ p \to q \not\models \neg q$$



Both branches are open. They represent the same counter-example, i.e., the valuation *V* with *V(q)=1* and *V(p)=0*.



¬p, p→q
¬q
q
p
p
q


¬q, p→q ∘ ¬p
¬q, p→q, p ∘
p→q, p ∘ q
p ∘ p, q
q, p ∘ q

$$\phi_1,\ldots,\phi_n \models \psi$$

if & only if

$$\phi_1,\ldots,\phi_n \circ \psi$$

can be rewritten as
a closing tableau
(no counter-examples)

This property holds for propositional logic, *and also for predicate logic*.

$$\phi_1, \ldots, \phi_n \nvDash \psi$$

if & only if

$$\phi_1, \ldots, \phi_n \circ \psi$$

can be rewritten as
a tableau with an open branch
(repr. a counter-example)

This property holds for propositional logic, *but not for predicate logic*.

Logic in Action

Johan van Benthem, Jan van Eijck and Jan Jaspars

23rd ESSLLI, Ljubljana, Aug 1-5 2011

design

I

- Propositional Logic
- Syllogistics (& sets)
- First Order Logic

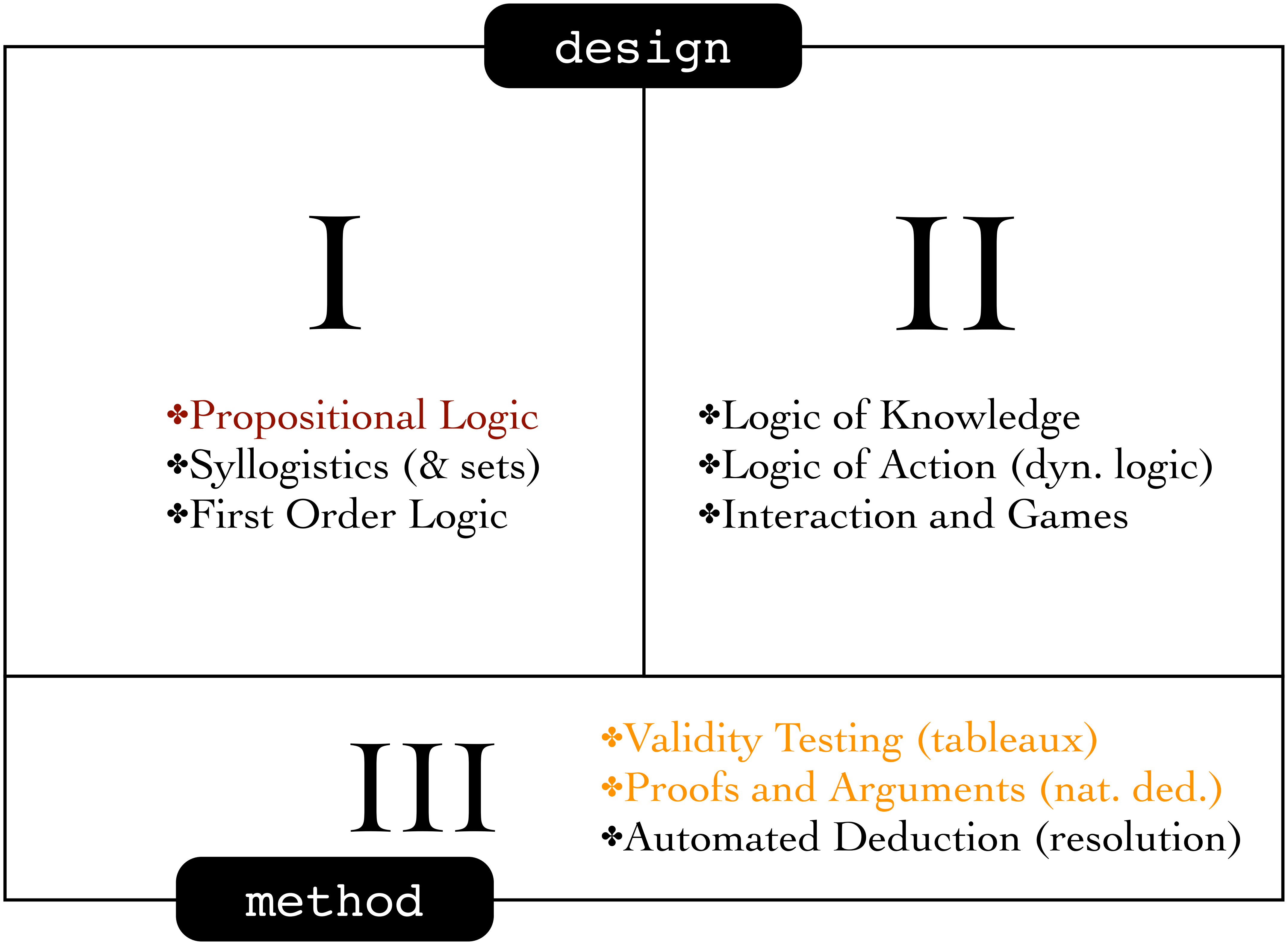
II

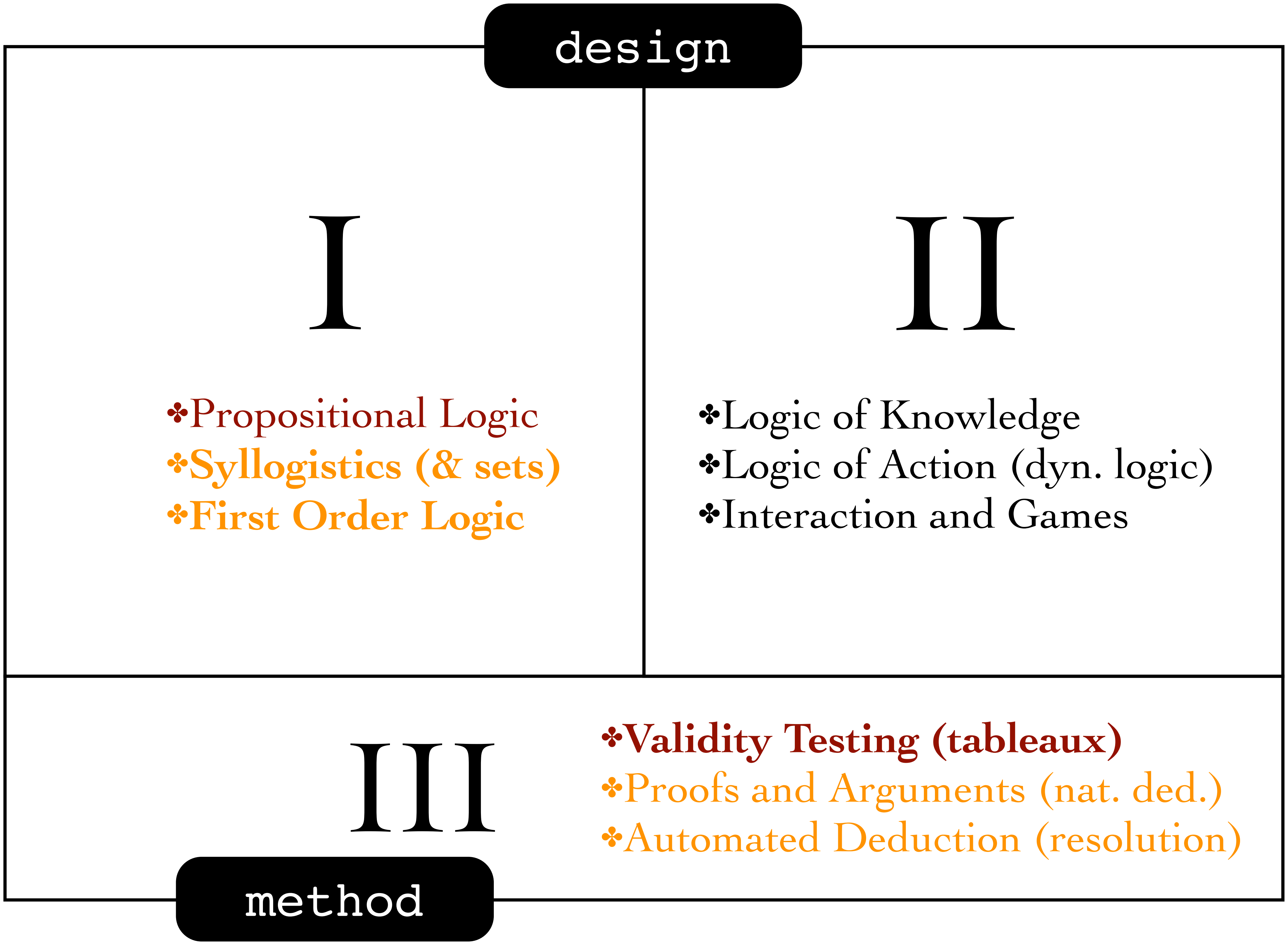
- Logic of Knowledge
- Logic of Action (dyn. logic)
- Interaction and Games

III

- Validity Testing (tableaux)
- Proofs and Arguments (nat. ded.)
- Automated Deduction (resolution)

method


design
I
✤Propositional Logic
✤Syllogistics (& sets)
✤First Order Logic
II
✤Logic of Knowledge
✤Logic of Action (dyn. logic)
✤Interaction and Games
III
✤Validity Testing (tableaux)
✤Proofs and Arguments (nat. ded.)
✤Automated Deduction (resolution)
method


design
I
✤Propositional Logic
✤Syllogistics (& sets)
✤First Order Logic
II
✤Logic of Knowledge
✤Logic of Action (dyn. logic)
✤Interaction and Games
III
✤Validity Testing (tableaux)
✤Proofs and Arguments (nat. ded.)
✤Automated Deduction (resolution)
method

design

I

- Propositional Logic
- Syllogistics (& sets)
- First Order Logic

II

- **(Wed) Logic of Knowledge**
- **(Thu) Logic of Action (dyn. logic)**
- **(Friday) Interaction and Games**

III

- **Validity Testing (tableaux)**
- Proofs and Arguments (nat. ded.)
- Automated Deduction (resolution)

method

All ∞℘ŧƀ are ♬

No ☀伍 is ♬

No ☀伍 is ∞℘ŧƀ

All ∞℘ŧb are ♬

No ☀伍 is ♬

No ☀伍 is ∞℘ŧb

All ∞℘ƀ are ♬

No ☀伍 is ♬

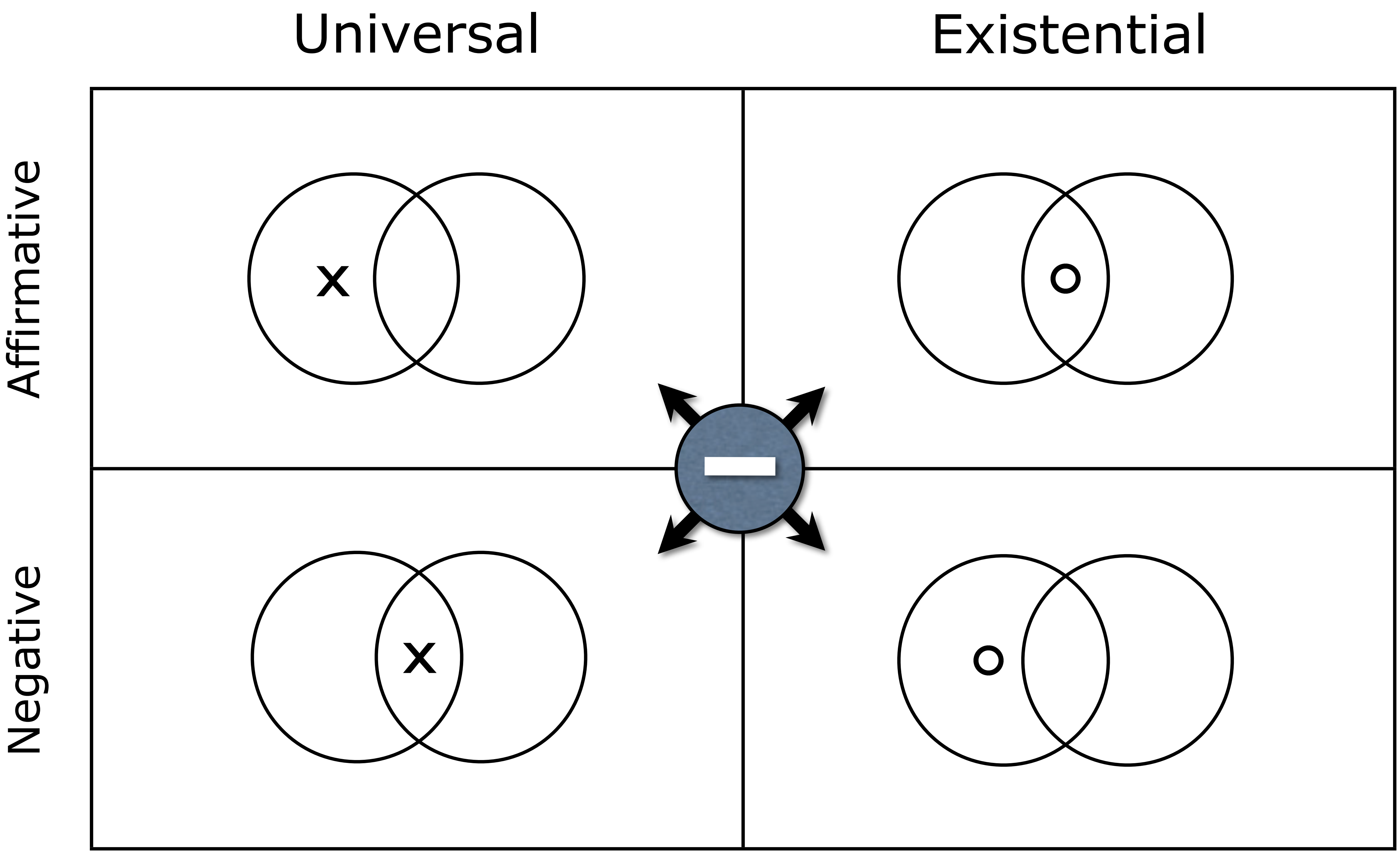
No ☀伍 is ∞℘ƀ

Calculemus

Aristotelian diagram (sem)

	Universal	Existential
Affirmative	All AB	Some AB
Negative	No AB	Not all AB

Aristotelian diagram (sem)



Aristotle

All politicians are rich
No student is rich

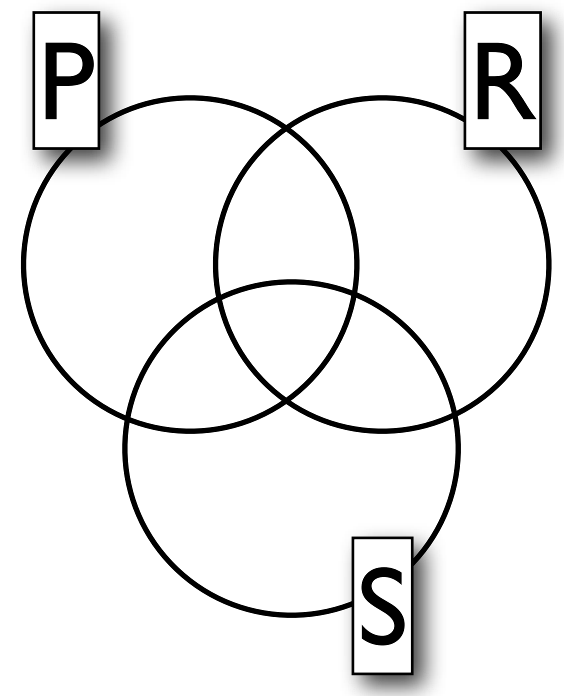
No student is politician

All politicians are rich
No student is politician

No student is rich

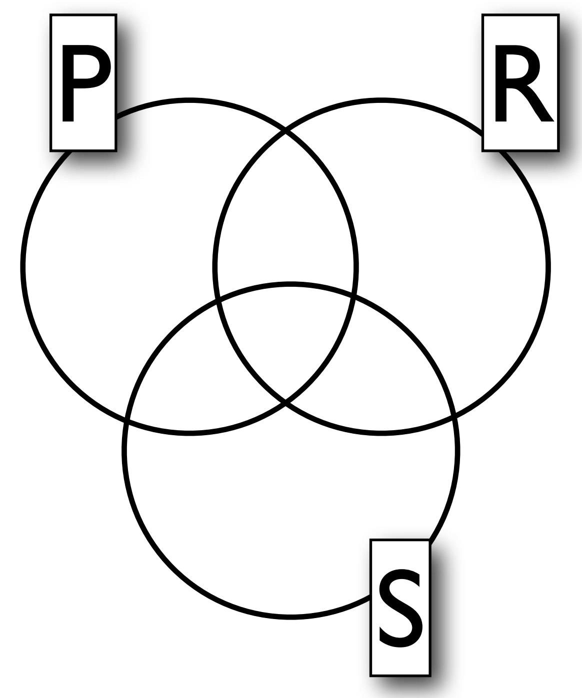
All politicians are rich
No student is rich

No student is politician



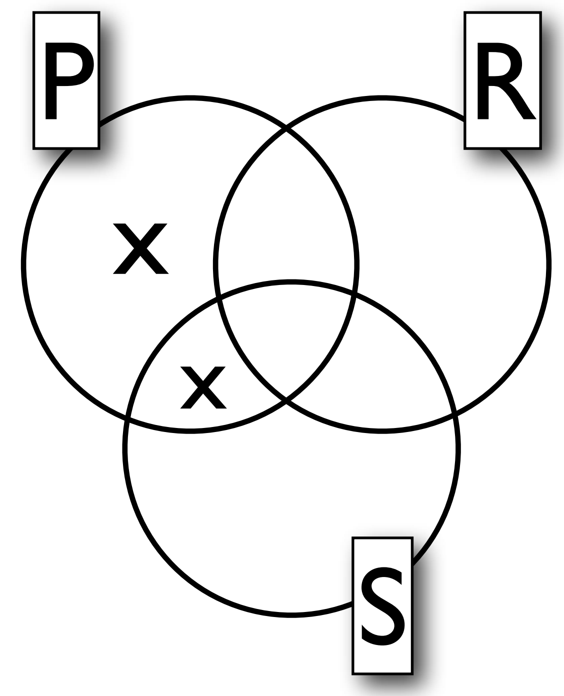
All politicians are rich
No student is politician

No student is rich



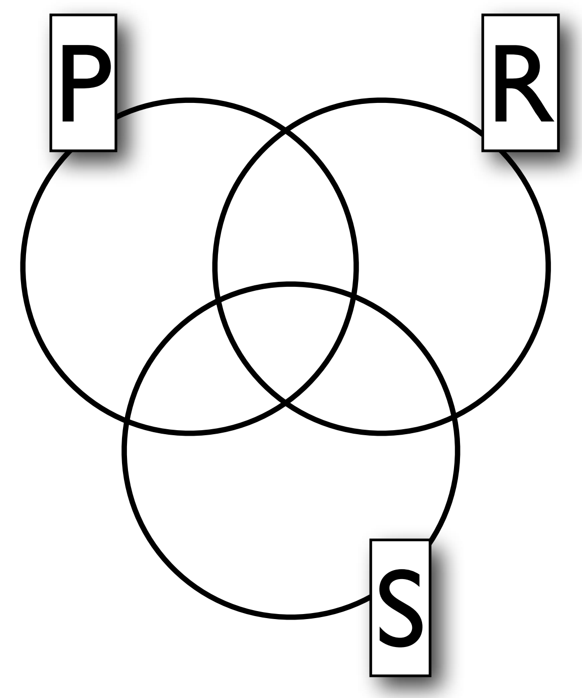
All politicians are rich
No student is rich

No student is politician



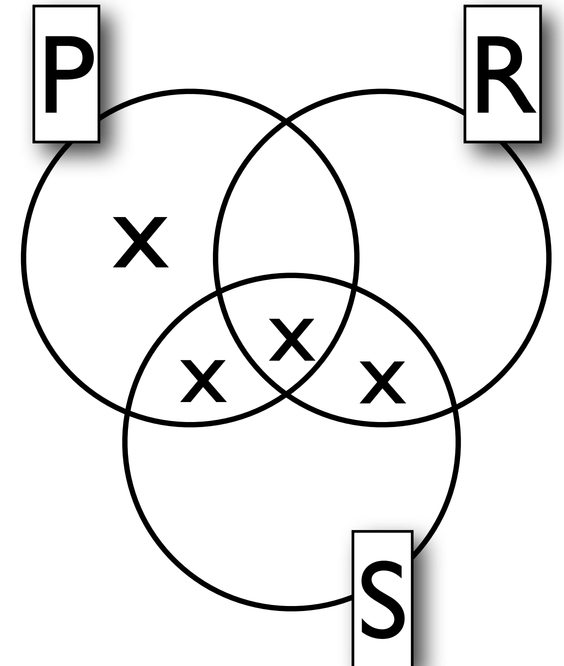
All politicians are rich
No student is politician

No student is rich



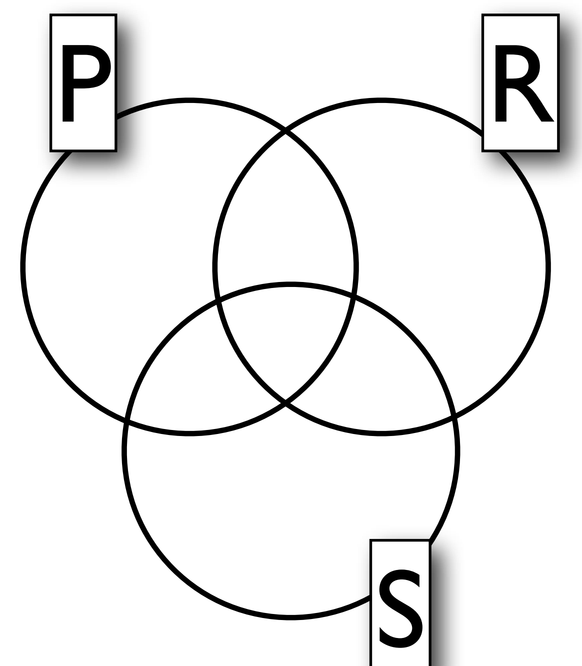
All politicians are rich
No student is rich

No student is politician



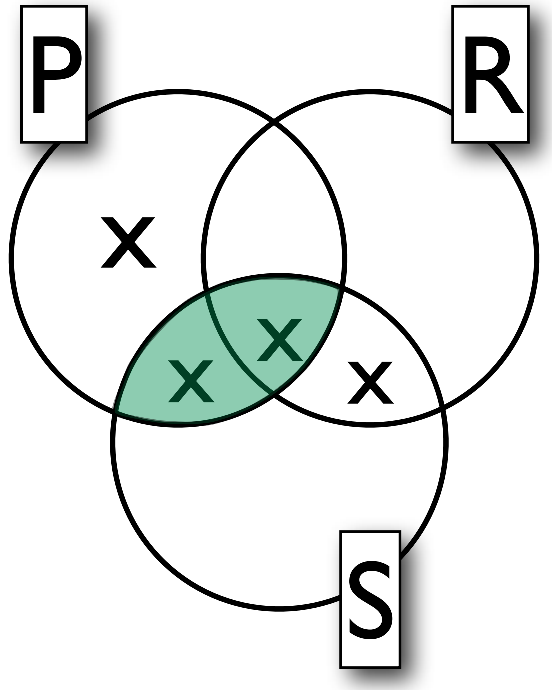
All politicians are rich
No student is politician

No student is rich



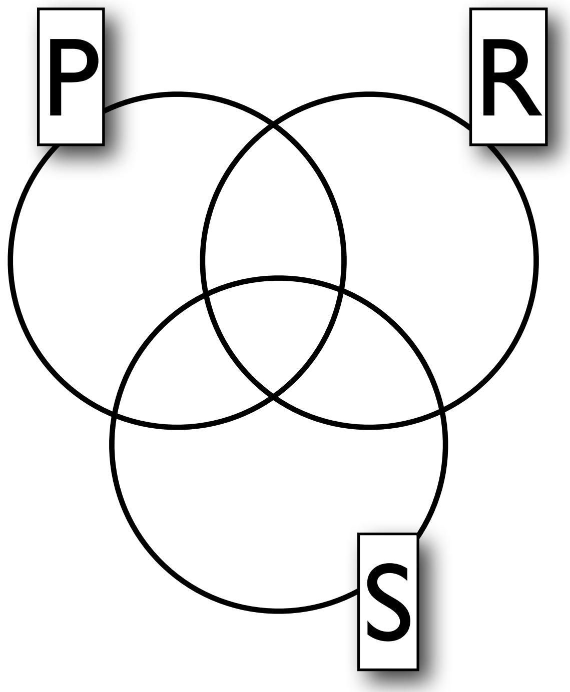
All politicians are rich
No student is rich

No student is politician



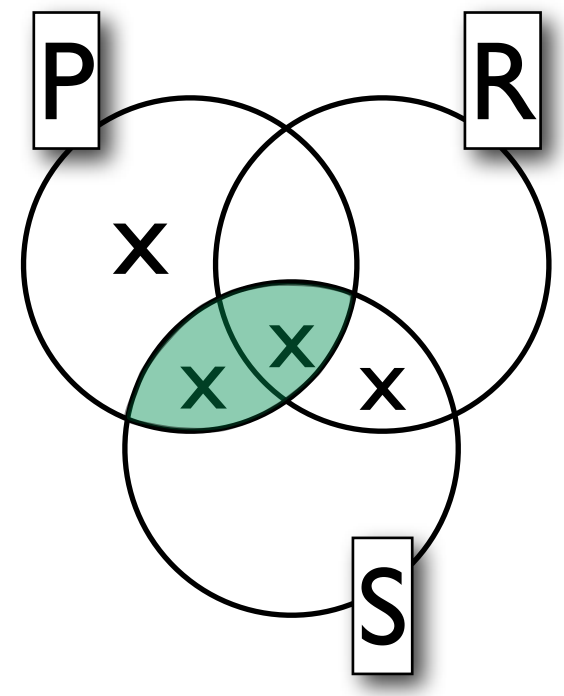
All politicians are rich
No student is politician

No student is rich



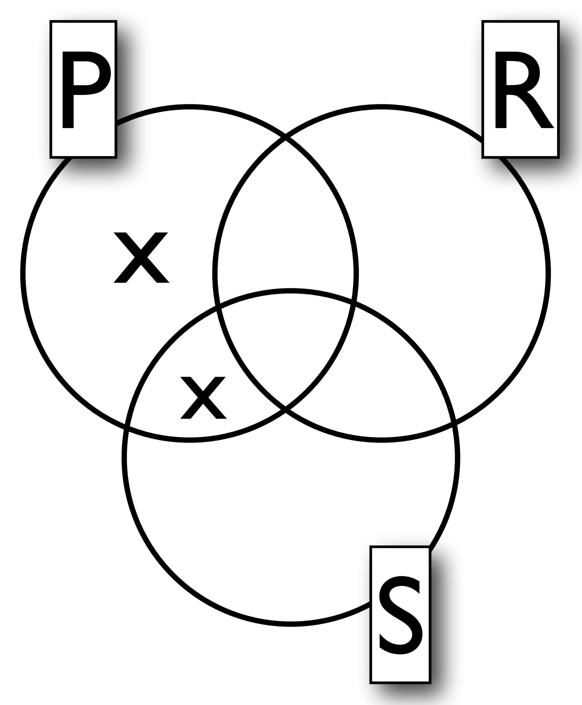
All politicians are rich
No student is rich

No student is politician



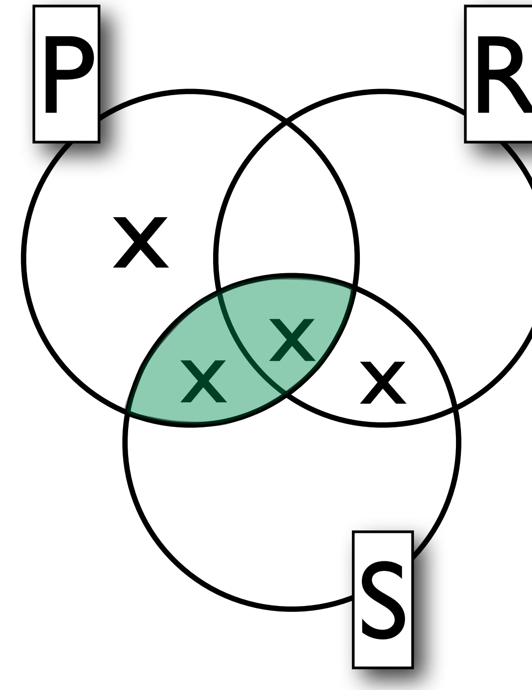
All politicians are rich
No student is politician

No student is rich



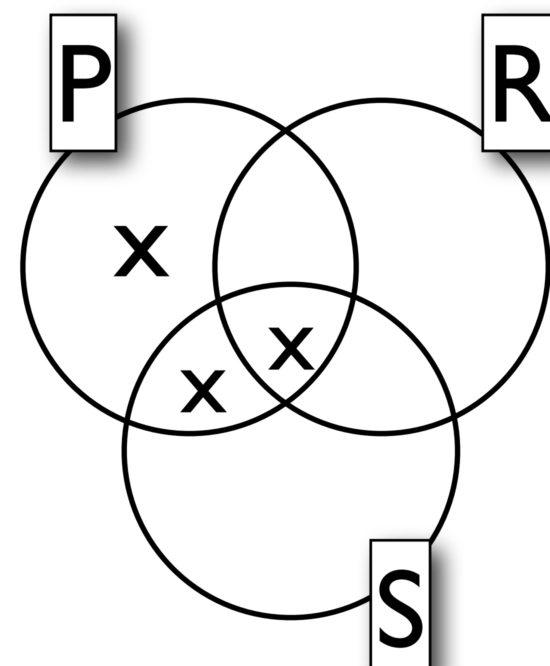
All politicians are rich
No student is rich

No student is politician



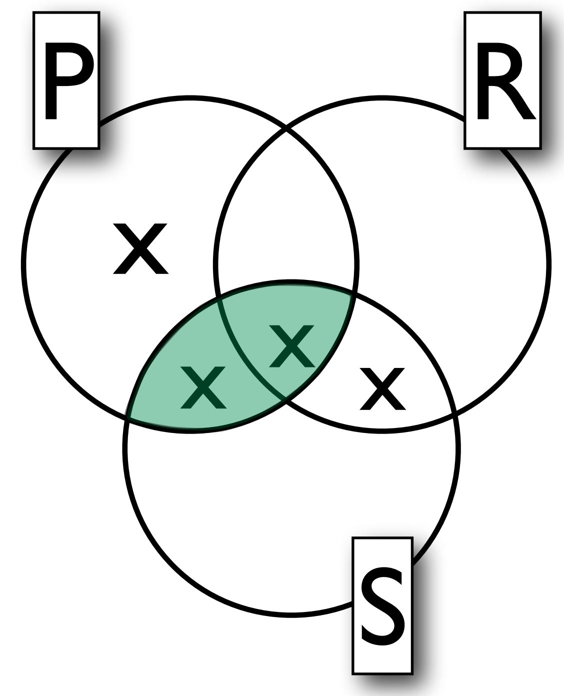
All politicians are rich
No student is politician

No student is rich



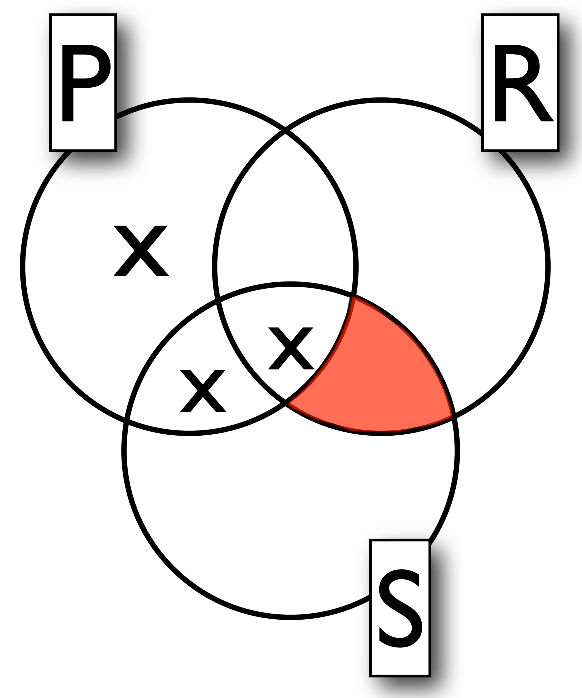
All politicians are rich
No student is rich

No student is politician



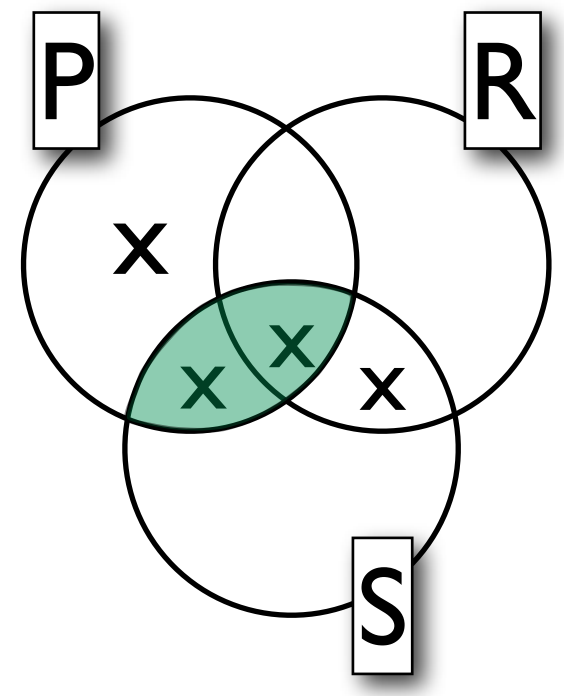
All politicians are rich
No student is politician

No student is rich



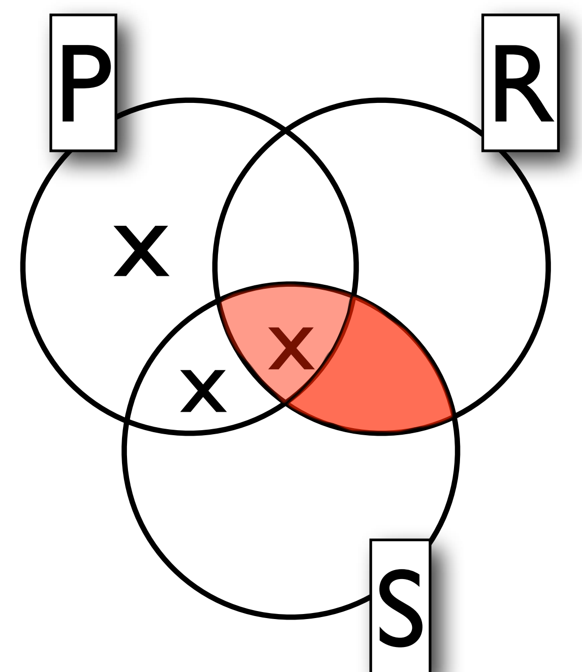
All politicians are rich
No student is rich

No student is politician



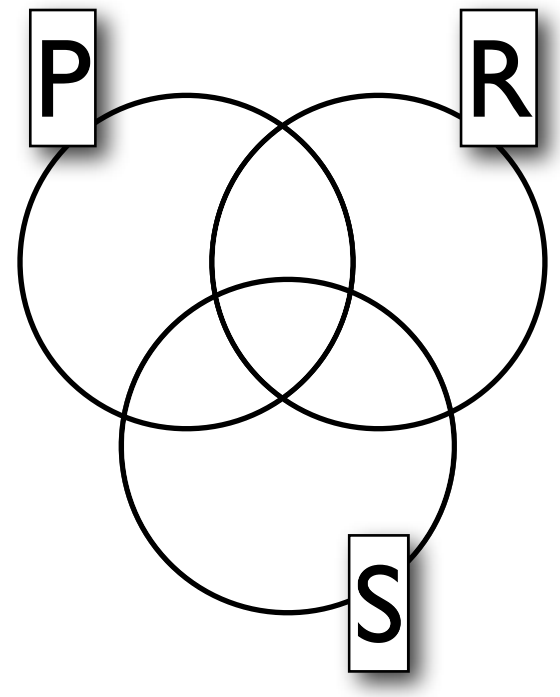
All politicians are rich
No student is politician

No student is rich



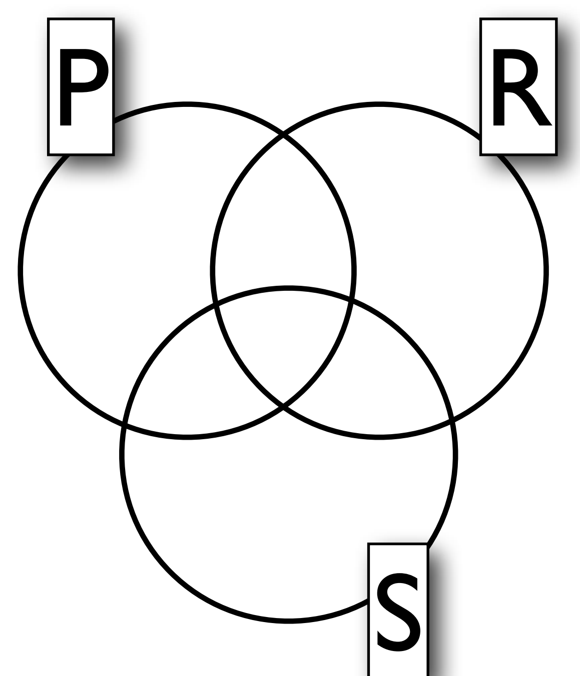
All politicians are rich
Some students are politician

Some students are rich



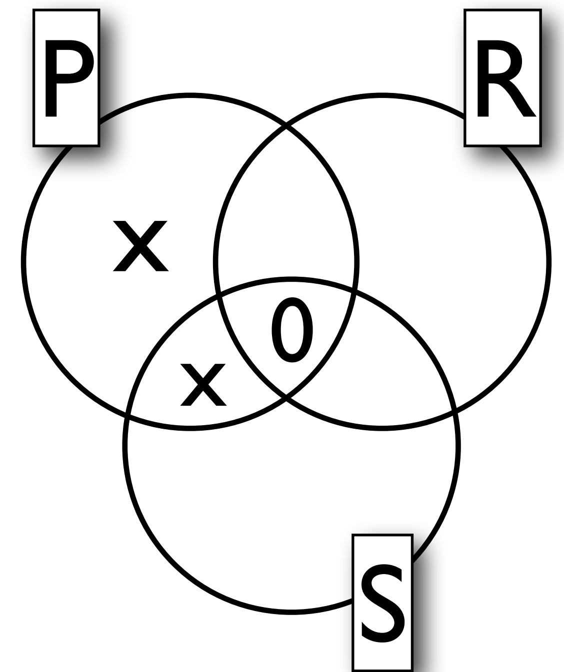
All politicians are rich
Some students are rich

Some students are politician



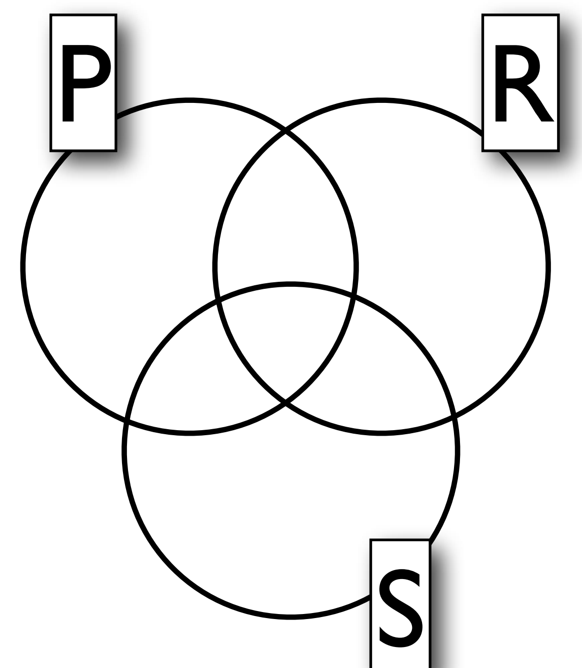
All politicians are rich
Some students are politician

Some students are rich



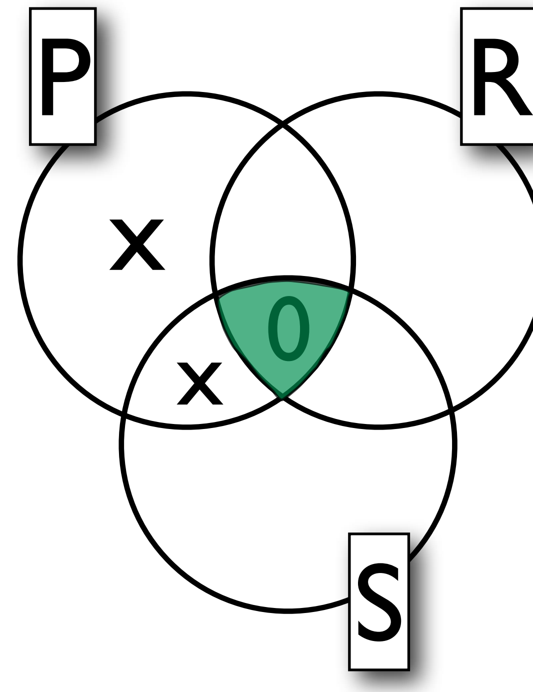
All politicians are rich
Some students are rich

Some students are politician



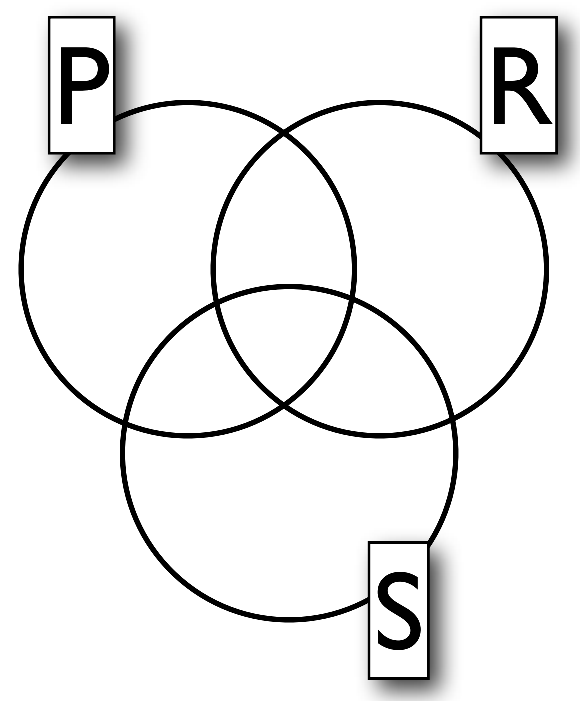
All politicians are rich
Some students are politician

Some students are rich



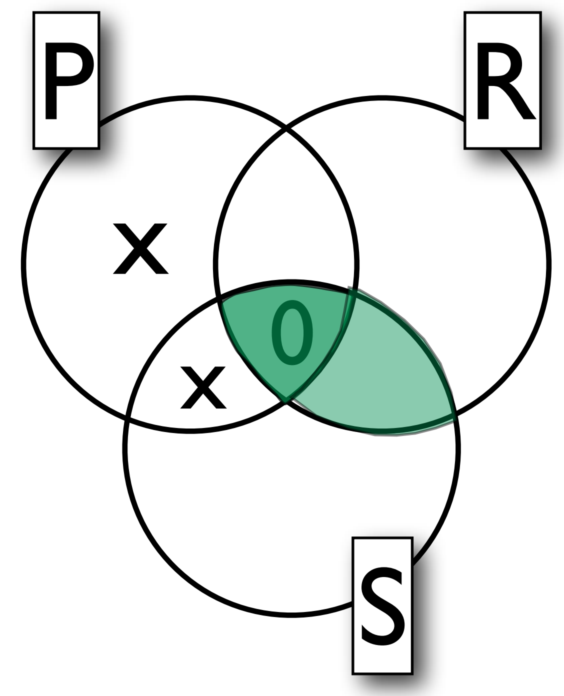
All politicians are rich
Some students are rich

Some students are politician



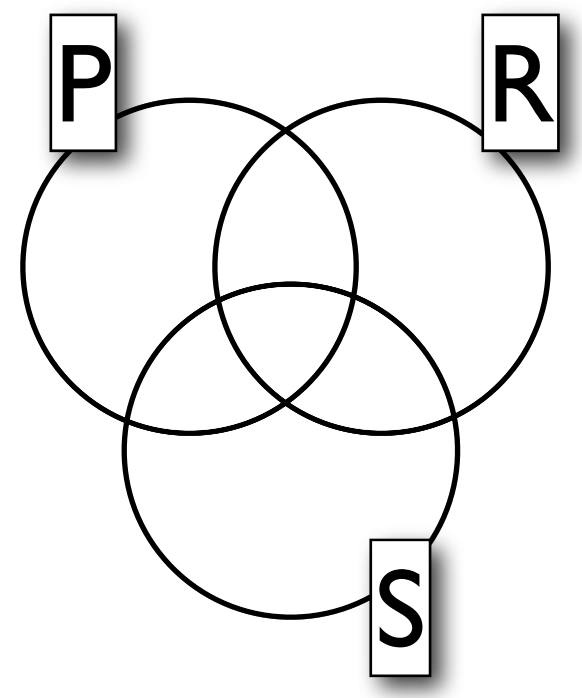
All politicians are rich
Some students are politician

Some students are rich



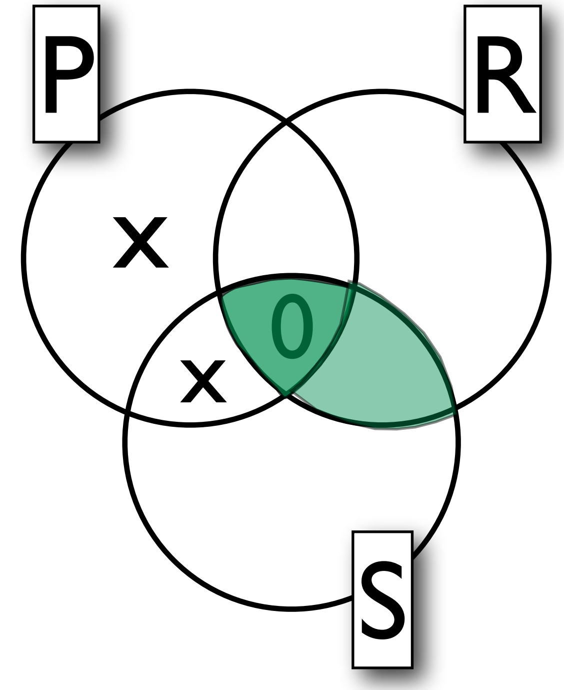
All politicians are rich
Some students are rich

Some students are politician



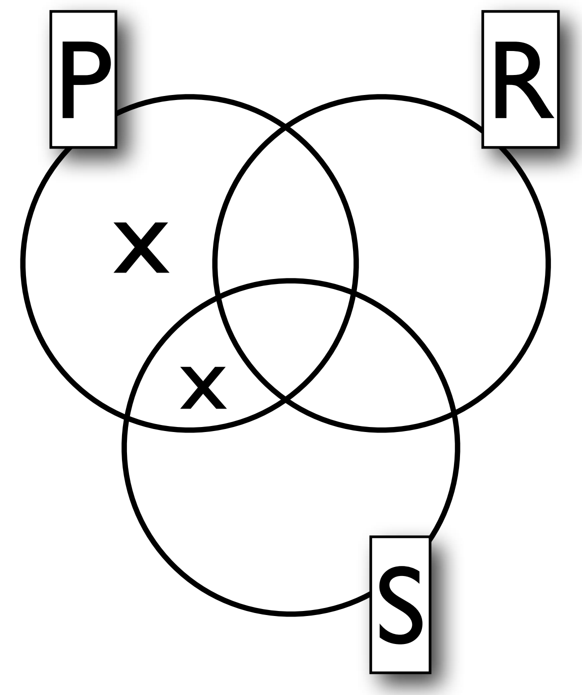
All politicians are rich
Some students are politician

Some students are rich



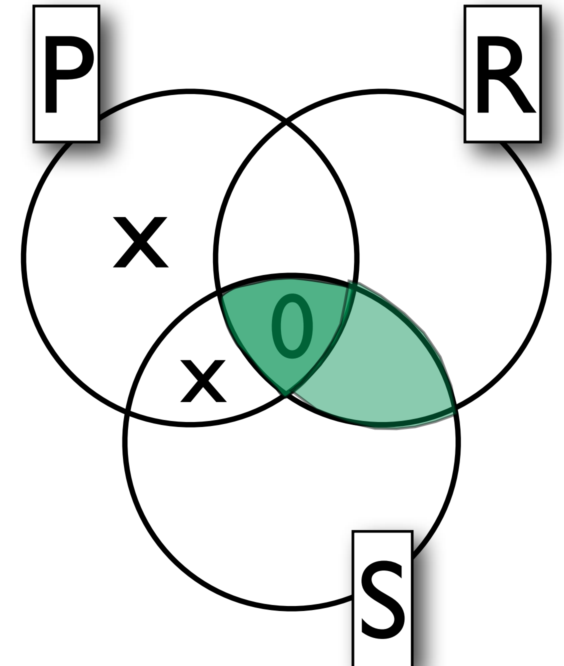
All politicians are rich
Some students are rich

Some students are politician



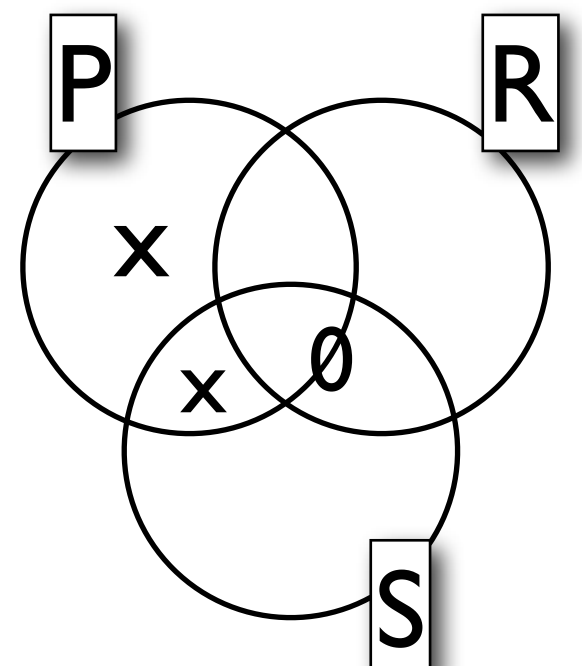
All politicians are rich
Some students are politician

Some students are rich



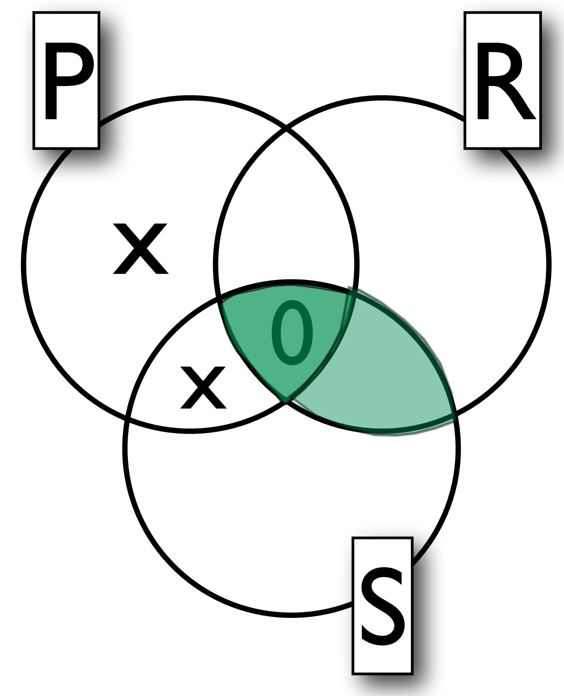
All politicians are rich
Some students are rich

Some students are politician



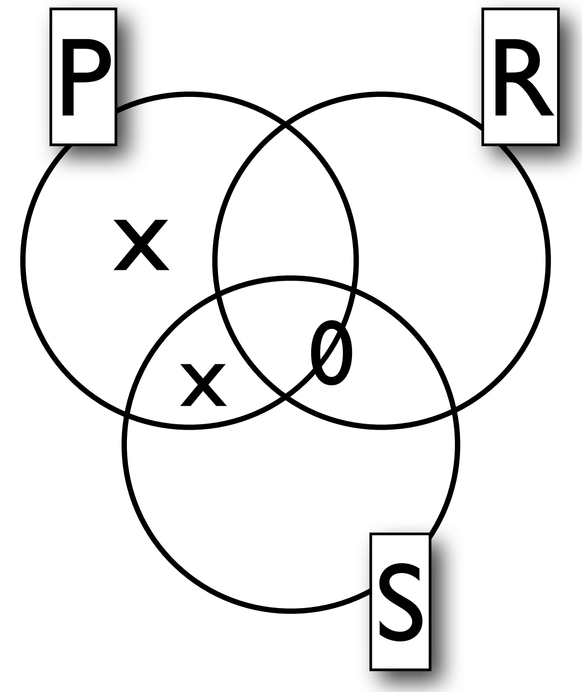
All politicians are rich
Some students are politician

Some students are rich



All politicians are rich
Some students are rich

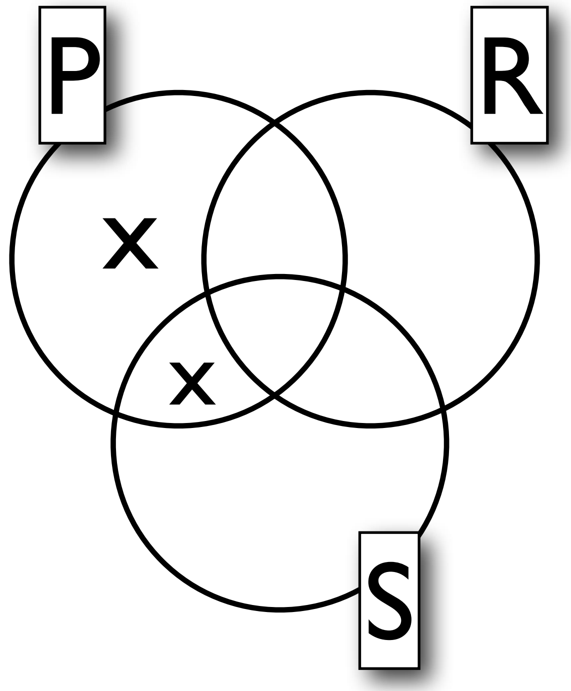
Some students are politician

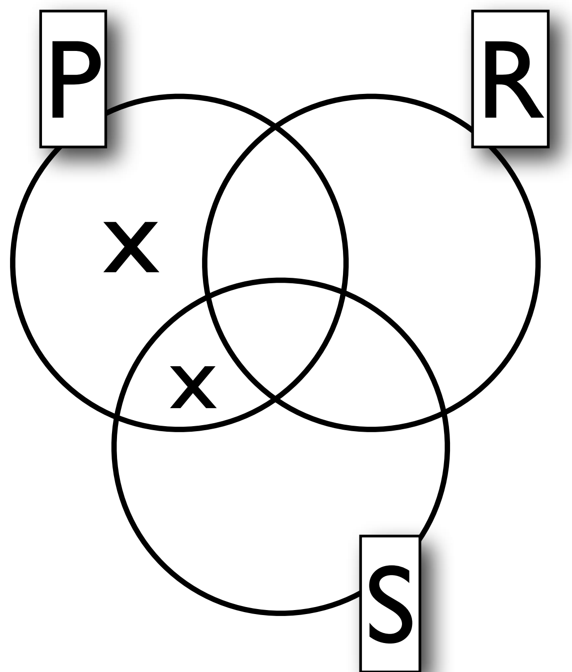


?

All politicians are rich
Some students are rich

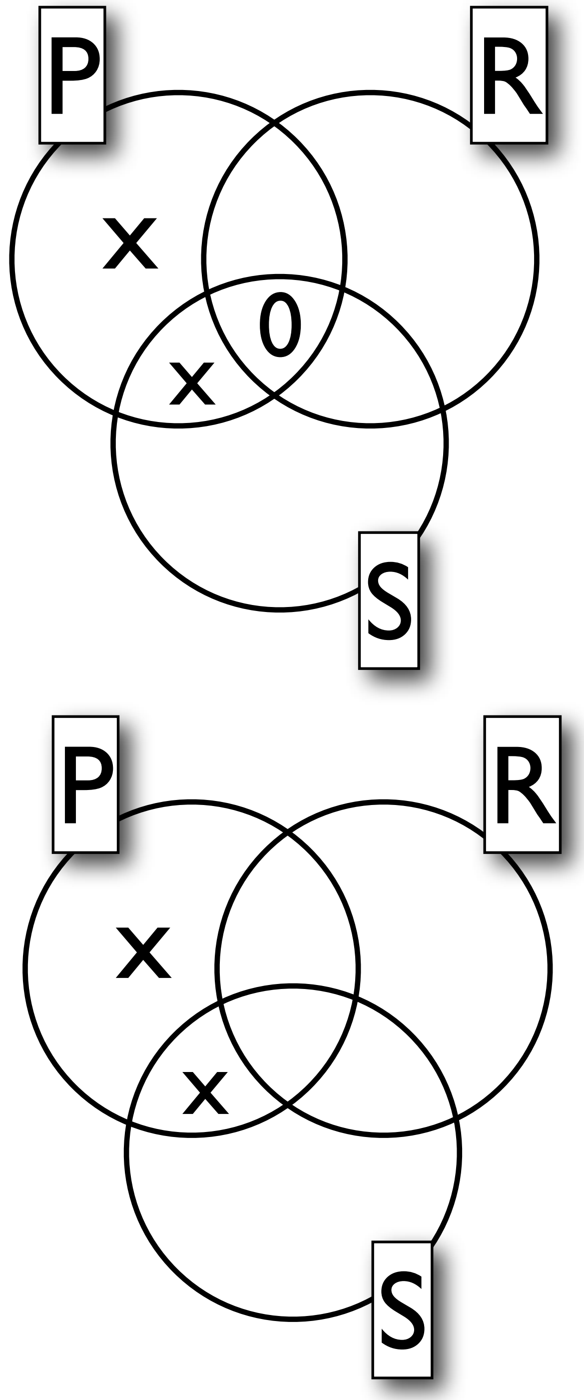
Some students are politician





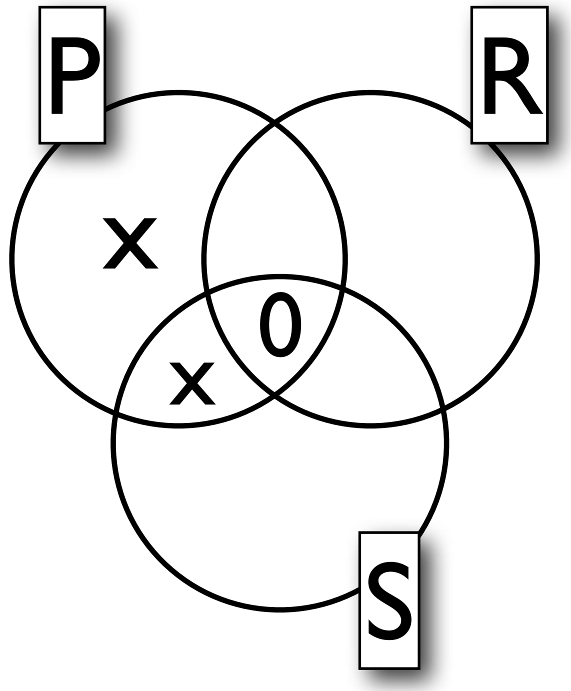
All politicians are rich
Some students are rich

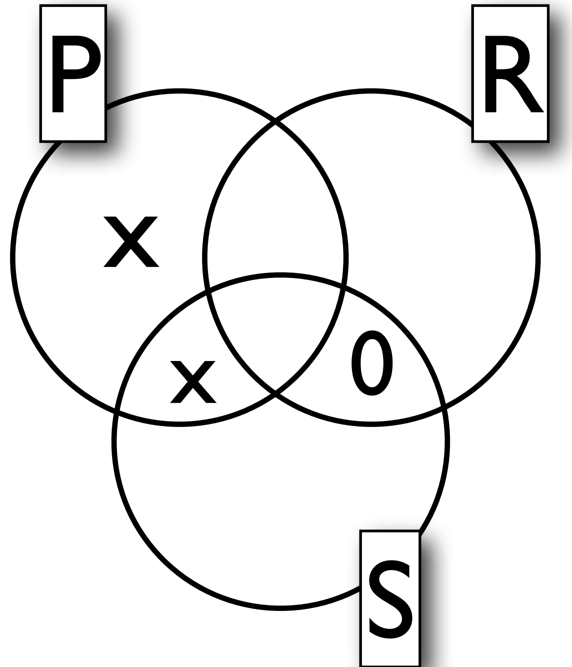
Some students are politician



All politicians are rich
Some students are rich

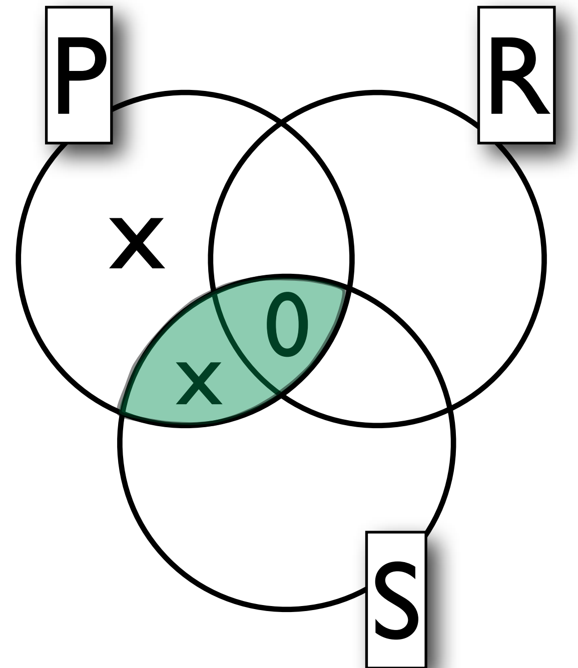
Some students are politician

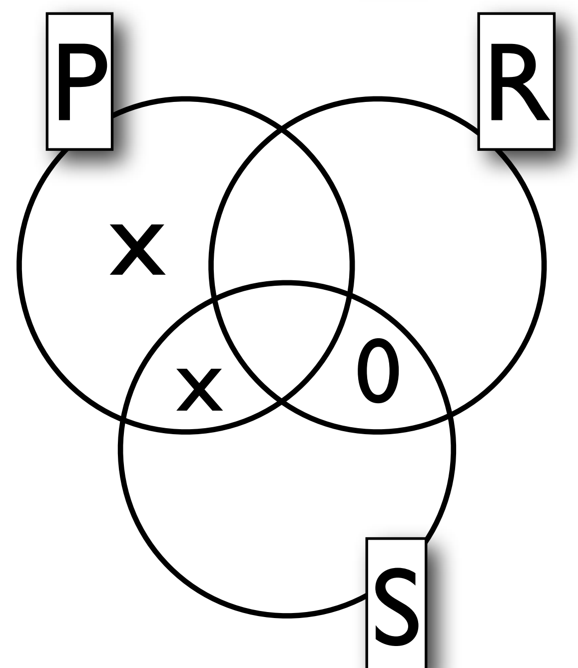




All politicians are rich
Some students are rich

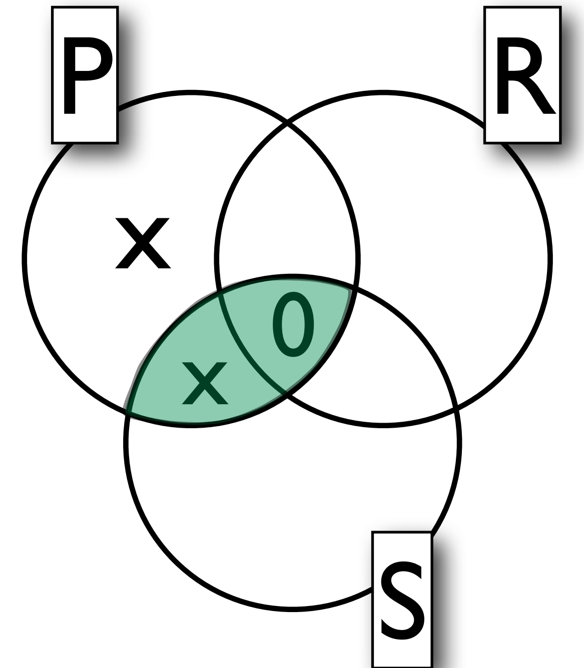
Some students are politician

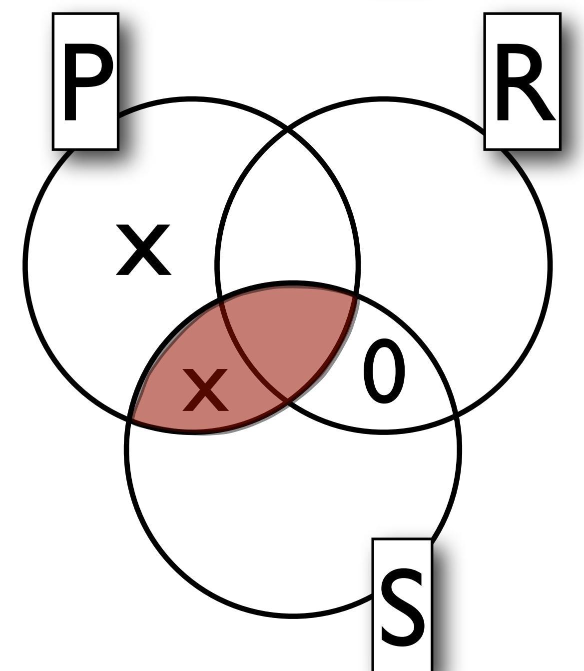




All politicians are rich
Some students are rich

Some students are politician





$$\phi_1,\ldots,\phi_n \vDash \psi$$

if & only if

$$\phi_1,\ldots,\phi_n \circ \psi$$

can be rewritten as
a closing tableau
(no counter-examples)

This property holds for propositional logic, *and also for predicate logic*.

$$\phi_1,\ldots,\phi_n \not\models \psi$$

if & only if

$$\phi_1,\ldots,\phi_n \circ \psi$$

can be rewritten as
a tableau with an open branch
(repr. a counter-example)

This property holds for propositional logic, *but not for predicate logic*.

$$x^2 = x+1$$

$$\Leftrightarrow$$

$$x^2 - (x+1) = 0$$

Christine Ladd

A syllog. inference is valid if & only if its anti-logism not sattisfiable.

Christine Ladd

A syllog. inference is ~valid if & only if its ***anti****-logism* ***not*** *sattisfiable.*

All politicians are rich
No student is rich

No student is politician

is valid

if and only if

All politicians are rich
No student is rich
Some student is politician

is not satisfiable

Sattisfiability tests for sets of
Aristotelean forms
=
Set up diagram/table
of all combinations ...
update with all the universal information
(All, No = remove objects) ...
then check whether you still can
add objects to support all the existential
forms (Some, Not-all).

$\Downarrow$

... symbolic version can be done in
propositional logic with low complexity.

Language of predicate logic

- Predicates P,Q,R,A,B,...
- Names (constants) a,b,c,...
- Variables x,y,z,....
- Functional Symbols

Lexicon

- Connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \bot$
- Quantifiers $\forall, \exists$
- Equality =

Logical symbols

Basic (atomic) formulas

Mj
Vm
Hjm
Gjxm

Basic (atomic) formulas

Predicate $term_1$ ... $term_n$

For now, term is either constant or variable

Basic (atomic) formulas (n=1)

Predicate term_1

Predicate stands for a property
(*unary* or *monadic* predicate)

Basic (atomic) formulas (n=2)

Predicate $term_1$ $term_2$

Predicate stands for a relation
(*binary* predicate)

Basic (atomic) formulas (n=3)

Predicate $term_1$ $term_2$ $term_3$

Ternary predicate

Basic (atomic) formulas

Predicate $term_1$ ... $term_n$

n-ary Predicate

Basic (atomic) formulas (n=0)

Predicate

0-ary Predicate = propositional variable as in propositional logic

Basic (atomic) formulas

Mj	John is a man
Wm	Mary is a woman
Ljm	John loves Mary
Gjxm	John gives ‘something unknown’ (x) to Mary

Basic (atomic) formulas

Mj	1 is *odd*
Wm	2 is *even*
Ljm	1 is *smaller than* 2
Gjxm	There *is* ‘something unknown’ (x) *between* 1 and 2

Connectives

- $Mj \wedge Wm$
- $Mj \rightarrow \neg Mm$
- $(Mj \wedge Wm) \rightarrow Ljm$
- $Hmj \rightarrow (Mj \vee Ljj)$
- $(Lmj \wedge \neg Ljj) \rightarrow Mj$

Don't!

BFx

x is a blue (B) bike (F)

Sj∧pf

John (j) and Peter (p) are students of philosophy (f)

Write ...

$Bx \wedge Fx$

x is a blue (B) bike (F)

$Sjf \wedge Spf$

John (j) and Peter (p) are students of philosophy (f)

Don't

Sj∧pf∨w

John and Peter are students of philosophy or mathematics

(Sj∧pf) ∨ (Sj∧pm) ? (Sjf∨m) ∧ (Spf∨m)

Write ...

$$(Sjf \land Spf) \lor (Sjw \land Spw)$$

or

$$(Sjf \lor Sjw) \land (Spf \lor Spw)$$

Quantifiers

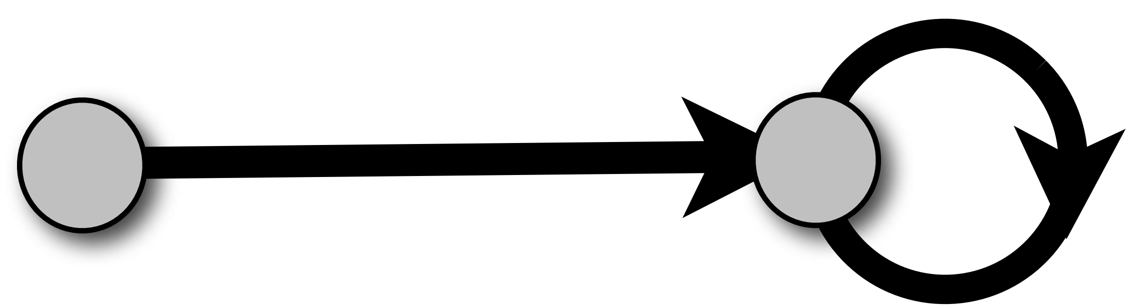
- $\exists x\ Mx$
- $\exists y\ \neg Lyy$
- $\forall x\ (Mx \leftrightarrow \neg Vx)$
- $\forall x \exists y\ Hxy$
- $\forall x(Hxx \rightarrow \exists y\ \neg Hyx)$

Quantifiers

- $\forall$**x**$\exists$**y Hxy** ~ "*everything* **R**-*s something*".
- $\exists$**y**$\forall$**x Hxy** ~ "*there's something* **R**-*ed by everything*".
- $\forall$**x**$\exists$**y Hyx** ~ "*everything is* **R**-*ed by something*".
- $\exists$**y**$\forall$**x Hyx** ~ "*there's something which* **R**-*s everything*".

• $\forall x \exists y\ Rxy$	Everybody *loves* somebody
• $\forall x \exists y\ Ryx$	Everybody *is loved by* somebody
• $\exists y \forall x\ Rxy$	There is at least one person who *is loved by* everybody
• $\exists y \forall x\ Ryx$	There is at least one person who *loves* everybody

- $\forall x \exists y\ Rxy$
- $\forall x \exists y\ Ryx$
- $\exists y \forall x\ Rxy$
- $\exists y \forall x\ Ryx$

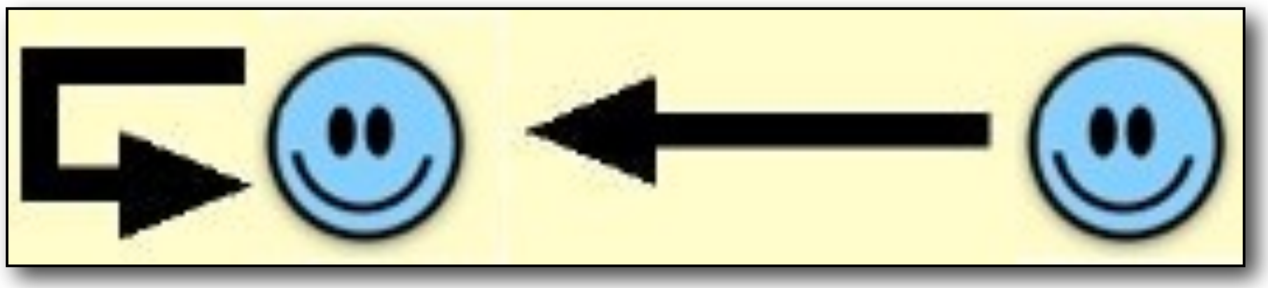


- $\forall x \exists y\ Rxy$
- $\forall x \exists y\ Ryx$
- $\exists y \forall x\ Rxy$
- $\exists y \forall x\ Ryx$

- ∀x∃y Rxy
- ∀x∃y Ryx
- ∃y∀x Rxy
- ∃y∀x Ryx

Graphs

∀x∃yRxy



Graphs

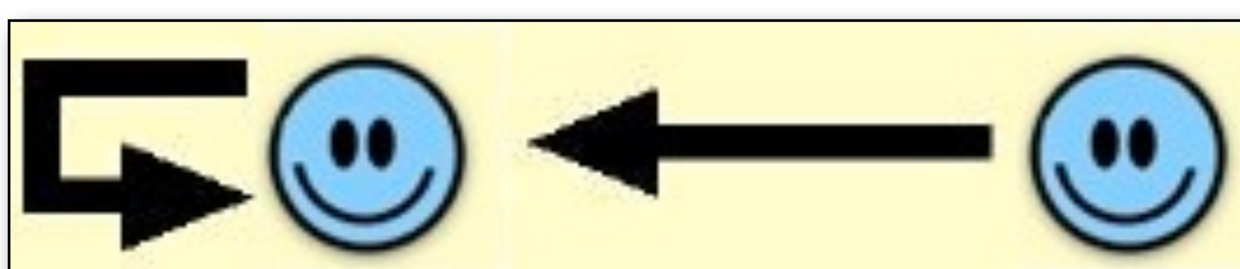
$\forall x \exists y Rxy$

↓

$\exists y Rxy$

↓

Rxy



Graphs

$\forall x \exists y Rxy$

↓

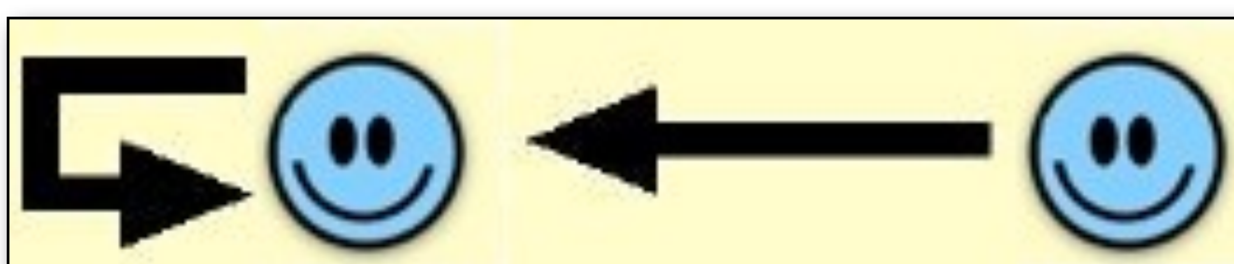
$\exists y Rxy$

↓

Rxy

x → left, y → left

x → right, y → left



Graphs

$\forall x \exists y Rxy$

↓

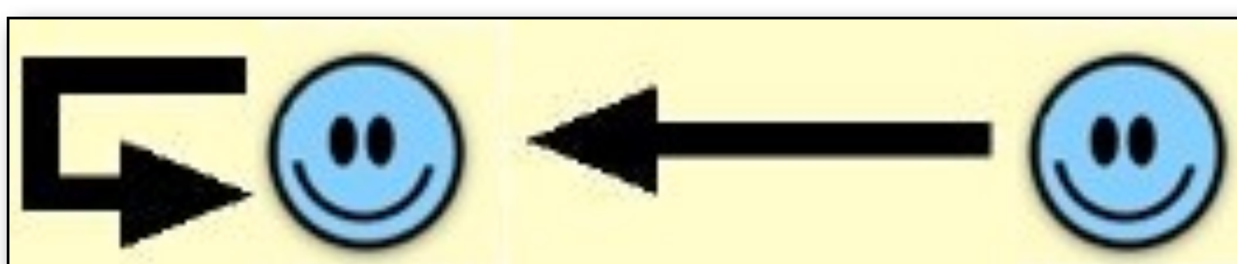
$\exists y Rxy$

↓

Rxy

x→left, y → left
x→right, y → left

x→left, y → right
x→right, y → right



Graphs

$\forall x \exists y Rxy$

↓

x→left
x→right

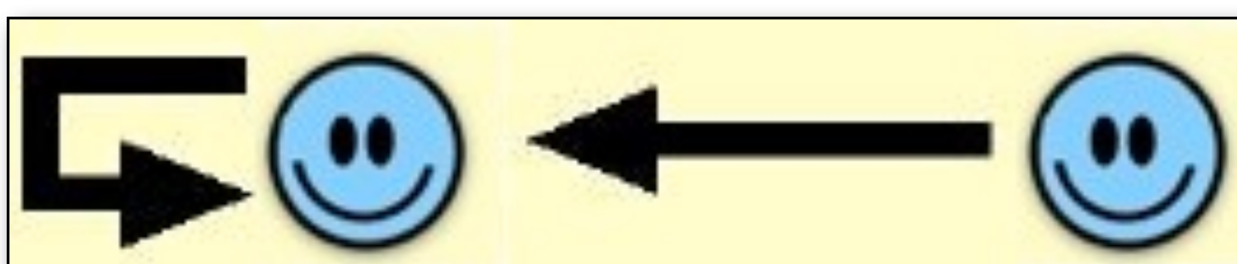
$\exists y Rxy$

↓

x→left, y → left
x→right, y → left

Rxy

x→left, y → right
x→right, y → right



Graphs

$\forall x \exists y Rxy$

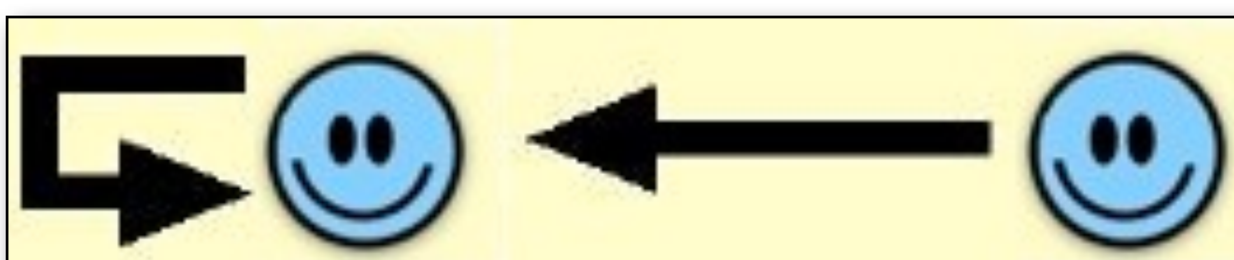
$\exists y Rxy$

x → left
x → right

Rxy

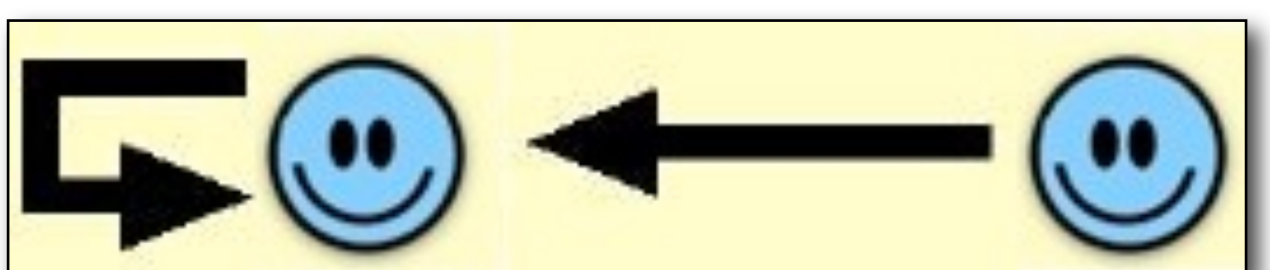
x → left, y → left
x → right, y → left

x → left, y → right
x → right, y → right



Graphs

$\forall x \exists y R\mathbf{yx}$



Graphs

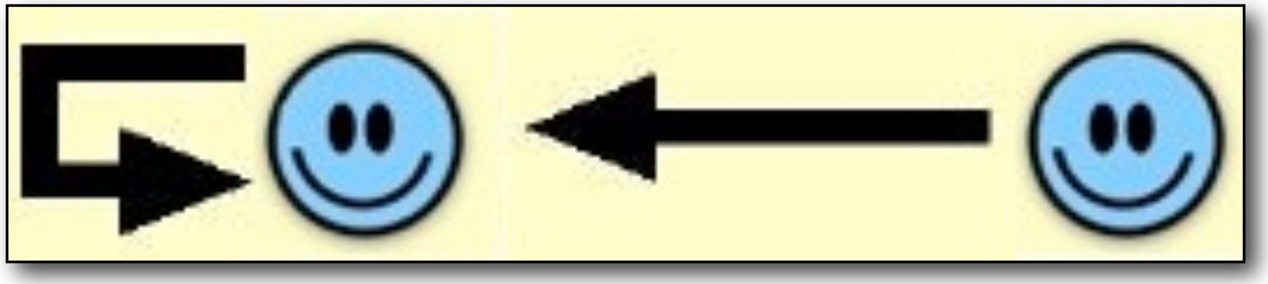
$\forall x \exists y R\mathbf{yx}$

↓

$\exists y R\mathbf{yx}$

↓

$R\mathbf{yx}$



Graphs

$\forall x \exists y R\mathbf{yx}$

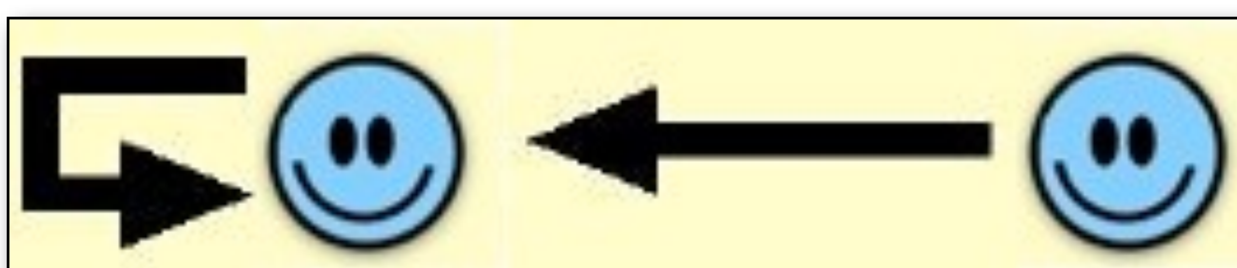
↓

$\exists y R\mathbf{yx}$

↓

$R\mathbf{yx}$

x→left, y → left
x→left, y → right



Graphs

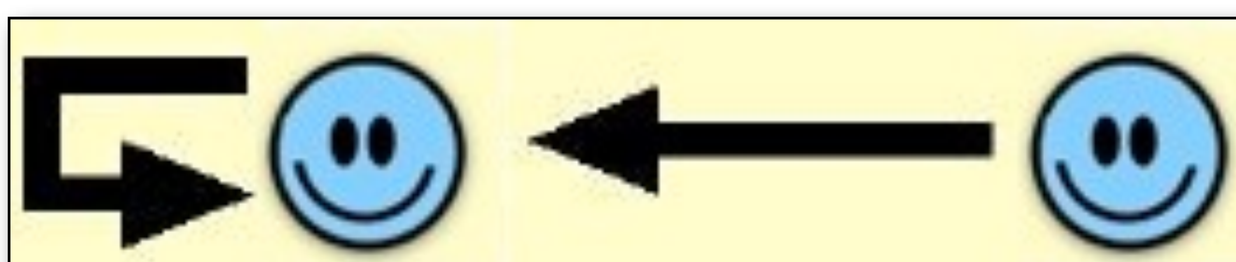
$\forall x \exists y R\mathbf{yx}$

$\exists y R\mathbf{yx}$

$R\mathbf{yx}$

x → left, y → left
x → left, y → right

x → right, y → left
x → right, y → right



Graphs

$\forall x \exists y R\mathbf{yx}$

↓

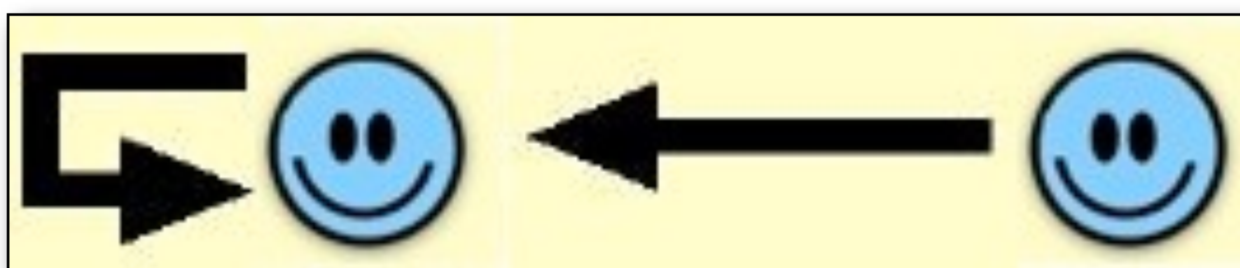
x→left — $\exists y R\mathbf{yx}$

↓

x→left, y → left
x→left, y → right

$R\mathbf{yx}$

x→right, y → left
x→right, y → right



Graphs

$\forall x \exists y R\mathbf{yx}$

↓

x→left — $\exists y R\mathbf{yx}$ — x→right

↓

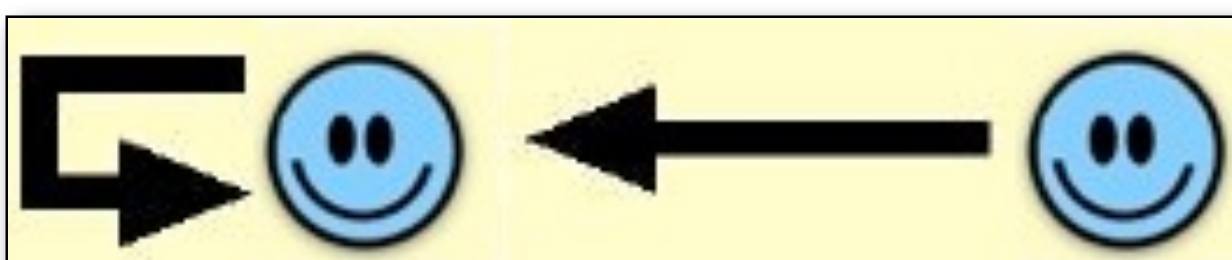
x→left, y → left

x→left, y → right

$R\mathbf{yx}$

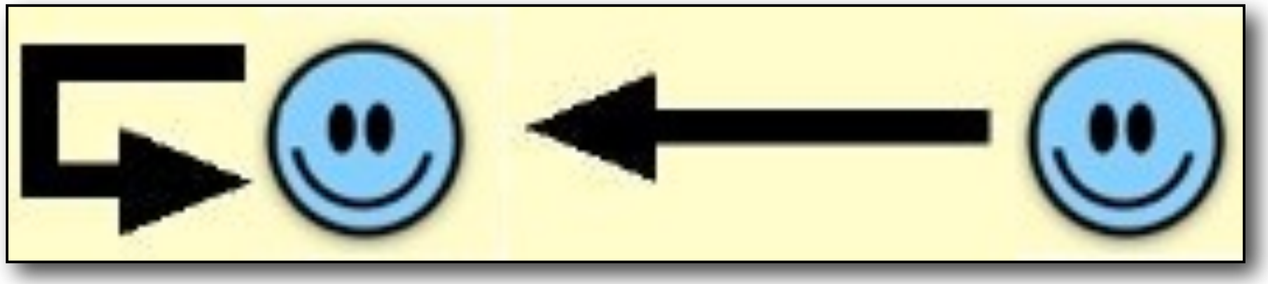
x→right, y → left

x→right, y → right



Graphs

∃y∀xRxy



Graphs

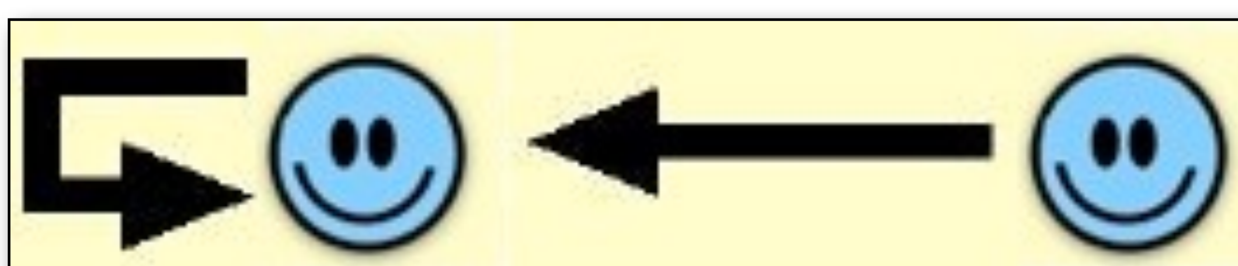
$\exists y \forall x Rxy$

↓

$\forall x Rxy$

↓

Rxy



Graphs

∃y∀xRxy

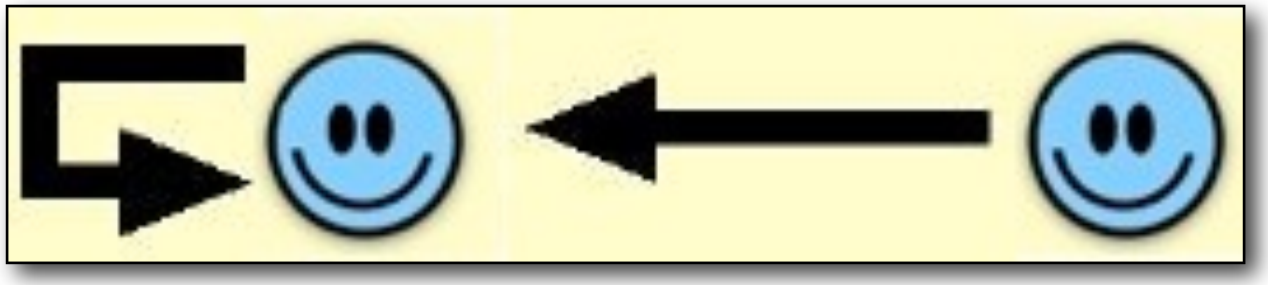
↓

∀xRxy

↓

Rxy

x→left, y → left
x→right, y → left



Graphs

$\exists y \forall x Rxy$

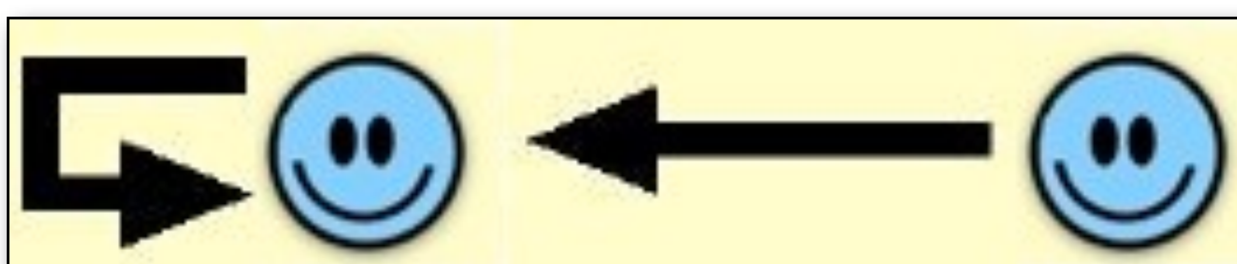
$\forall x Rxy$

y → left

Rxy

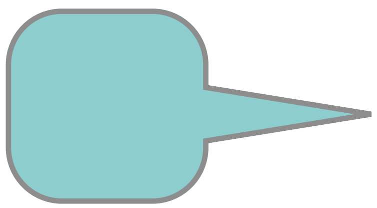
x→left, y → left

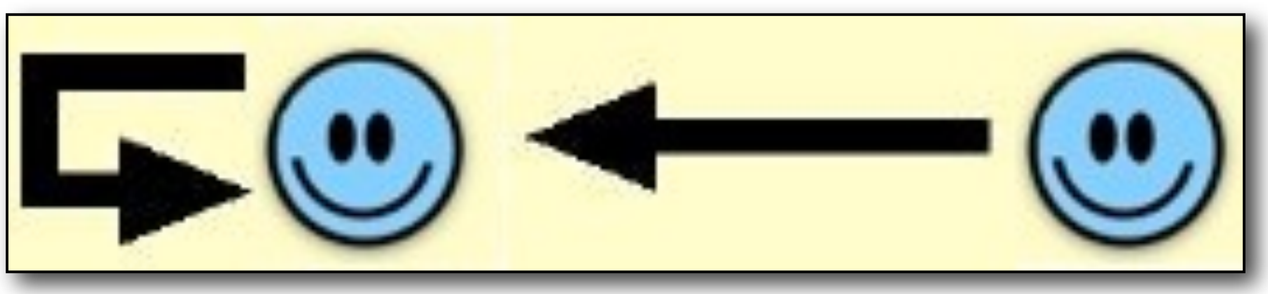
x→right, y → left



Graphs

$\exists y \forall x R\mathbf{yx}$





Graphs

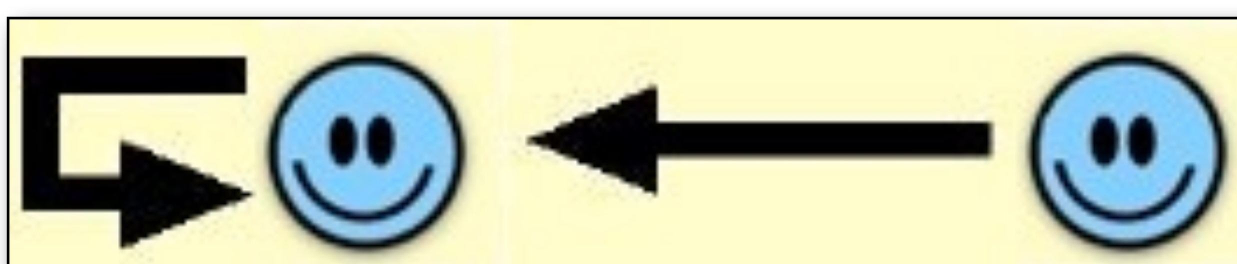
$\exists y \forall x R\mathbf{yx}$

↓

$\forall x R\mathbf{yx}$

↓

$R\mathbf{yx}$



Graphs

$\exists y \forall x R\mathbf{yx}$

↓

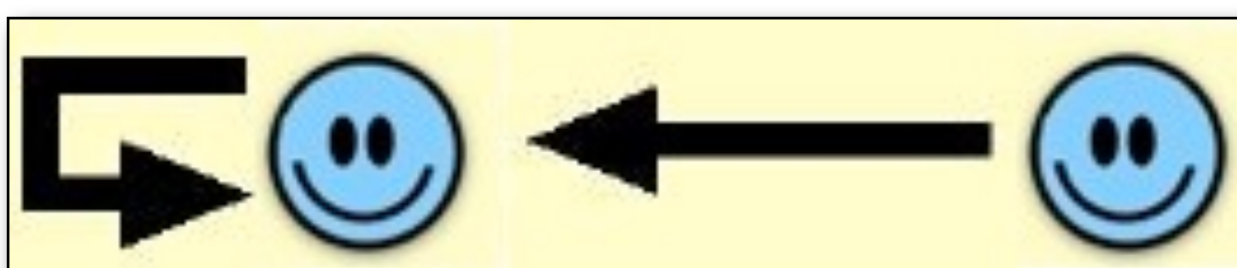
$\forall x R\mathbf{yx}$

↓

$x \rightarrow$ left, $y \rightarrow$ left

$x \rightarrow$ left, $y \rightarrow$ right

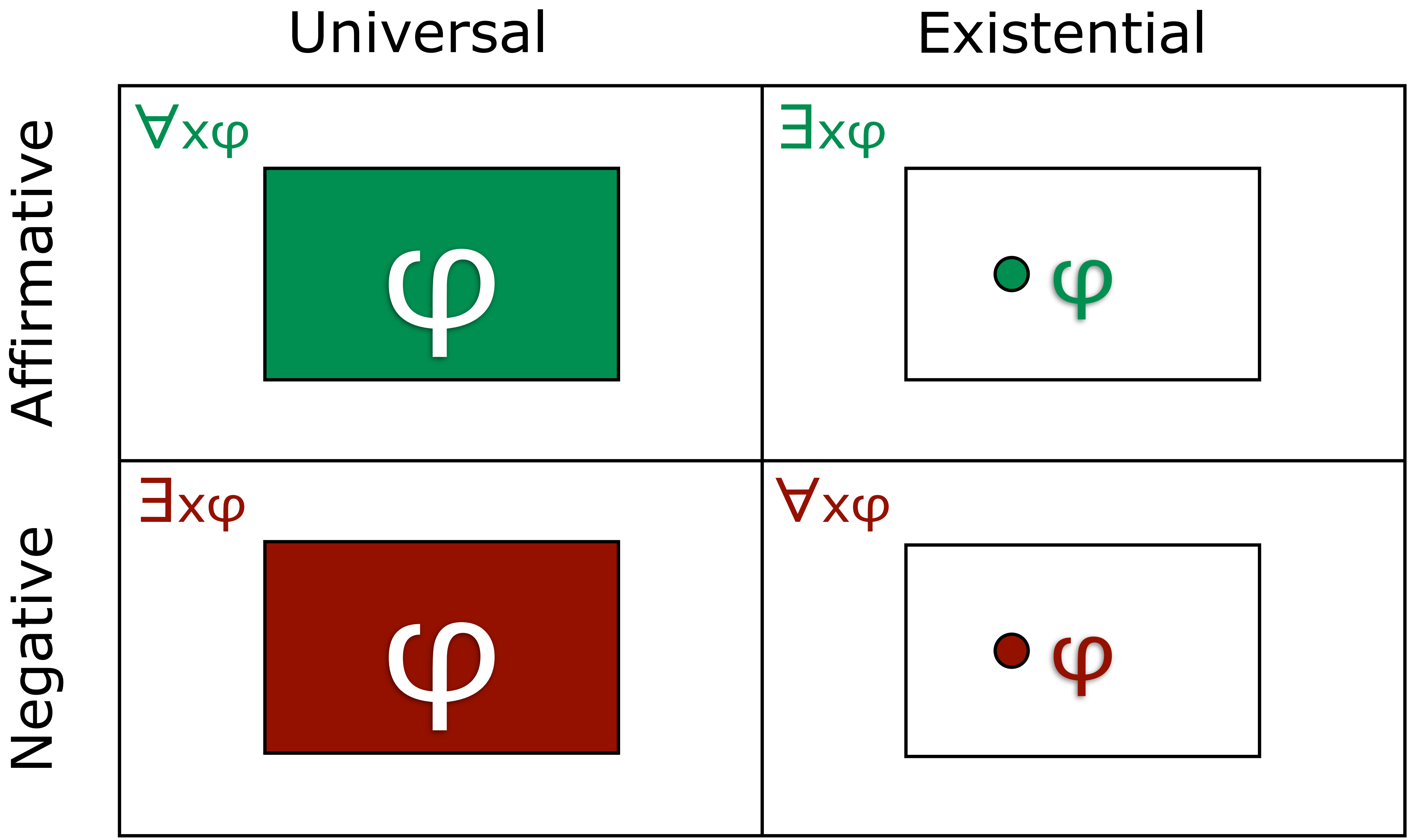
$R\mathbf{yx}$



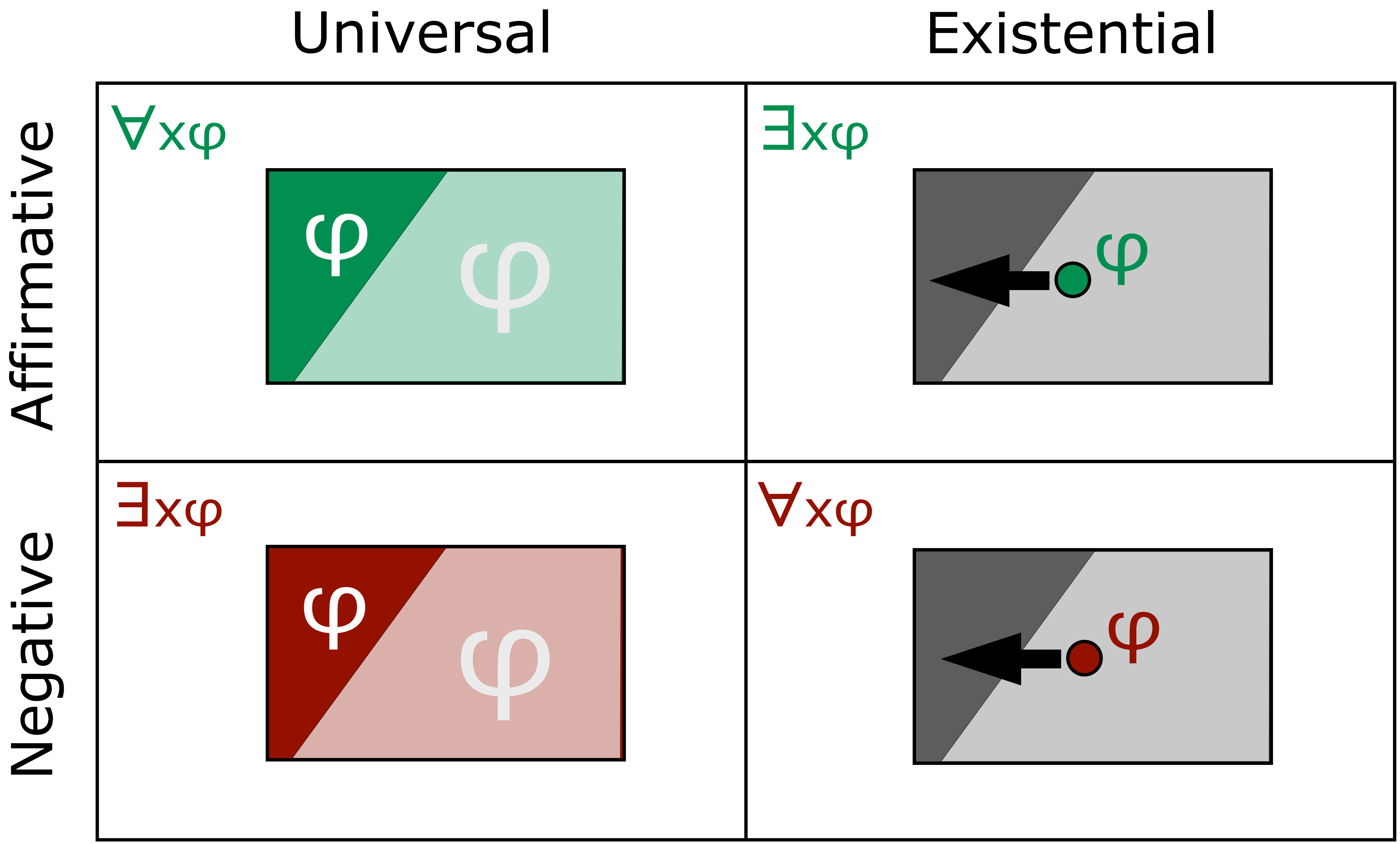
“Arist. Tableau” diagram (lang.)

	Universal	Existential
Affirmative	ALL .. ∀xφ .. ∘	SOME .. ∃xφ .. ∘ ....
Negative	NO .... ∘ .. ∃xφ ..	NOT ALL .... ∘ .. ∀xφ ..

“Arist. Tableau” diagram (1)



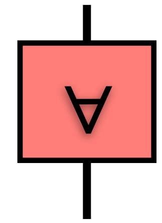
“Arist. tableau” diagram (2)



$$\forall x\, Px \rightarrow \forall x\, Qx \;\not\models\; \forall x\, (Px \rightarrow Qx)$$

$$\forall x\, Px \to \forall x\, Qx \circ \forall x\, (Px \to Qx)$$

$\forall x\ Px \to \forall x\ Qx \circ \forall x\ (Px \to Qx)$



$$\forall x\, Px \to \forall x\, Qx \circ \forall x\, (Px \to Qx)$$

∀

$$\forall x\, Px \to \forall x\, Qx \circ P\blacksquare \to Q\blacksquare$$

$\forall x\, Px \to \forall x\, Qx \;\circ\; \forall x\, (Px \to Qx)$

∀

$\forall x\, Px \to \forall x\, Qx \;\circ\;$ P▮ → Q▮

→

$$\forall x\, Px \rightarrow \forall x\, Qx \circ \forall x\, (Px \rightarrow Qx)$$

∀

$$\forall x\, Px \rightarrow \forall x\, Qx \circ P\mathbf{I} \rightarrow Q\mathbf{I}$$

→

$$\forall x\, Px \rightarrow \forall x\, Qx,\, P\mathbf{I} \circ Q\mathbf{I}$$

$\forall x\, Px \rightarrow \forall x\, Qx \circ \forall x\, (Px \rightarrow Qx)$

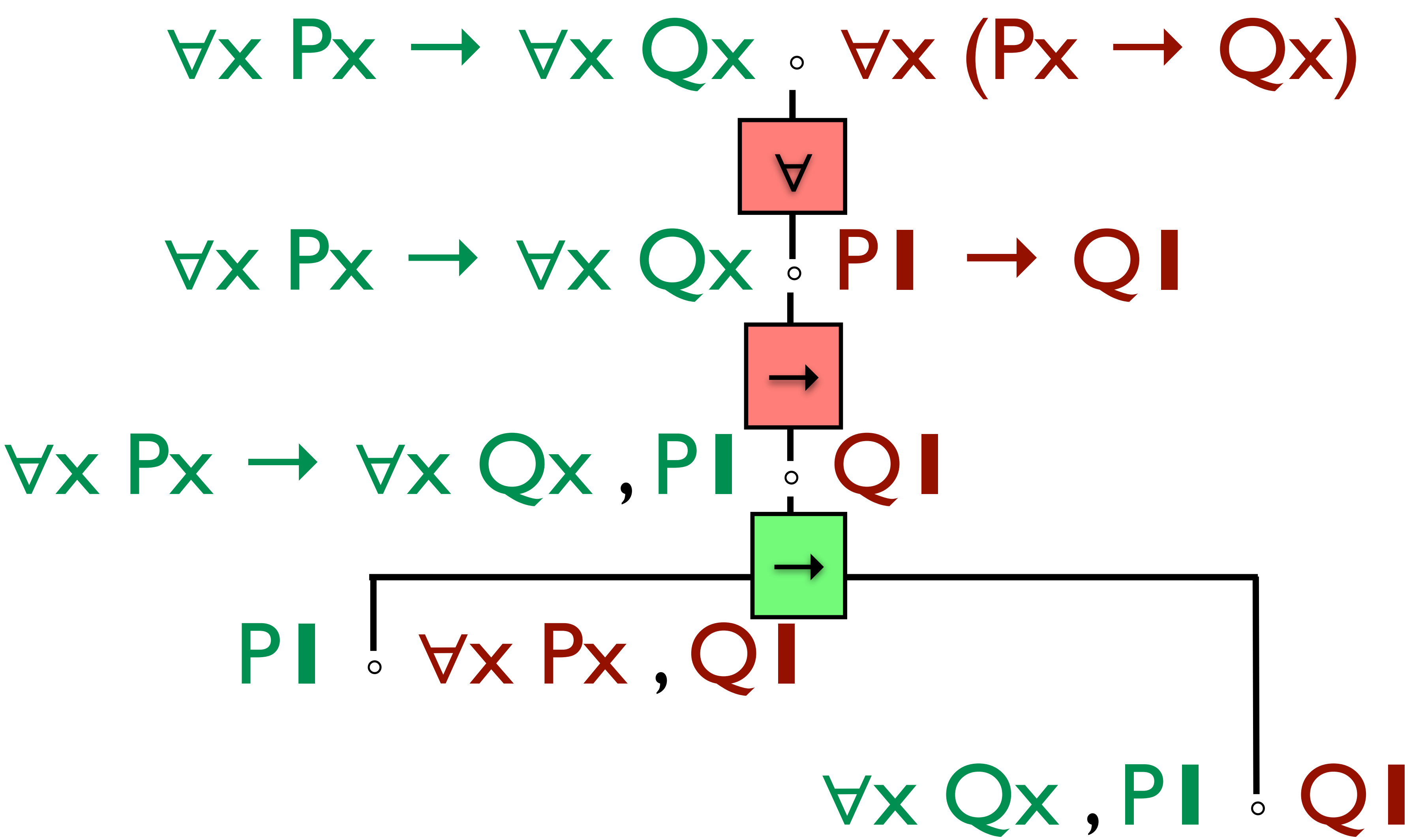
$\forall$

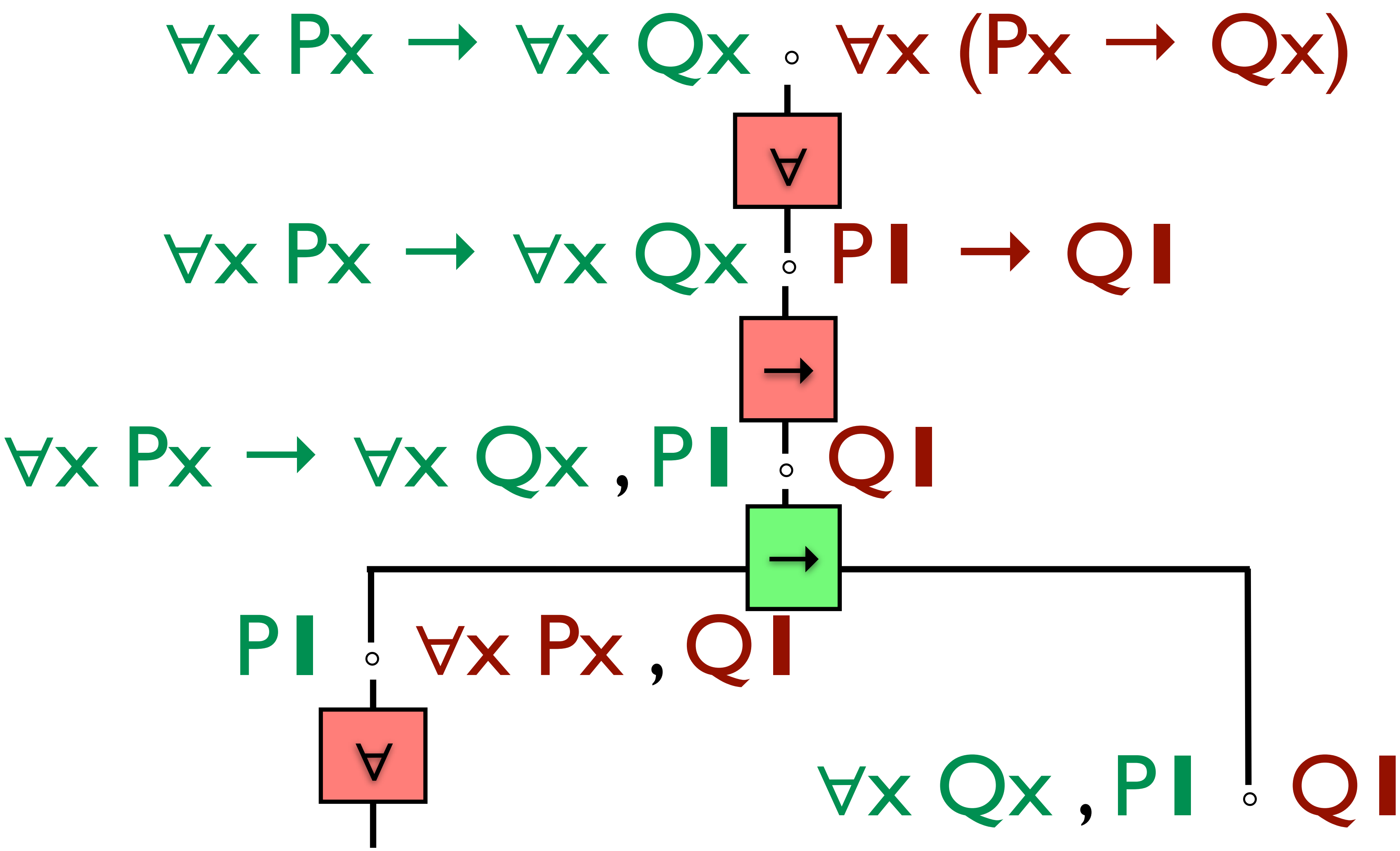
$\forall x\, Px \rightarrow \forall x\, Qx \circ P\mathbf{I} \rightarrow Q\mathbf{I}$

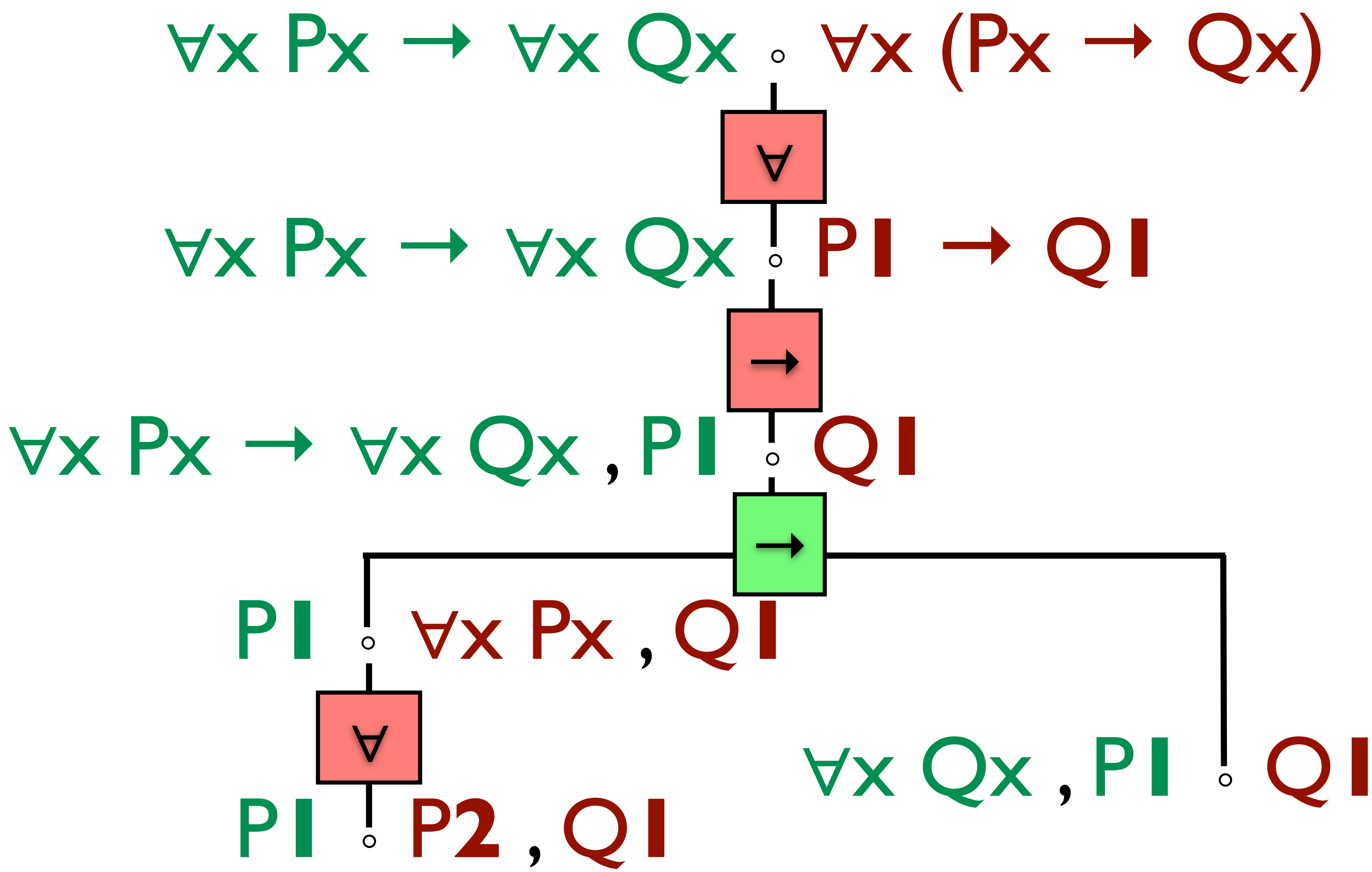
$\rightarrow$

$\forall x\, Px \rightarrow \forall x\, Qx\, , P\mathbf{I} \circ Q\mathbf{I}$

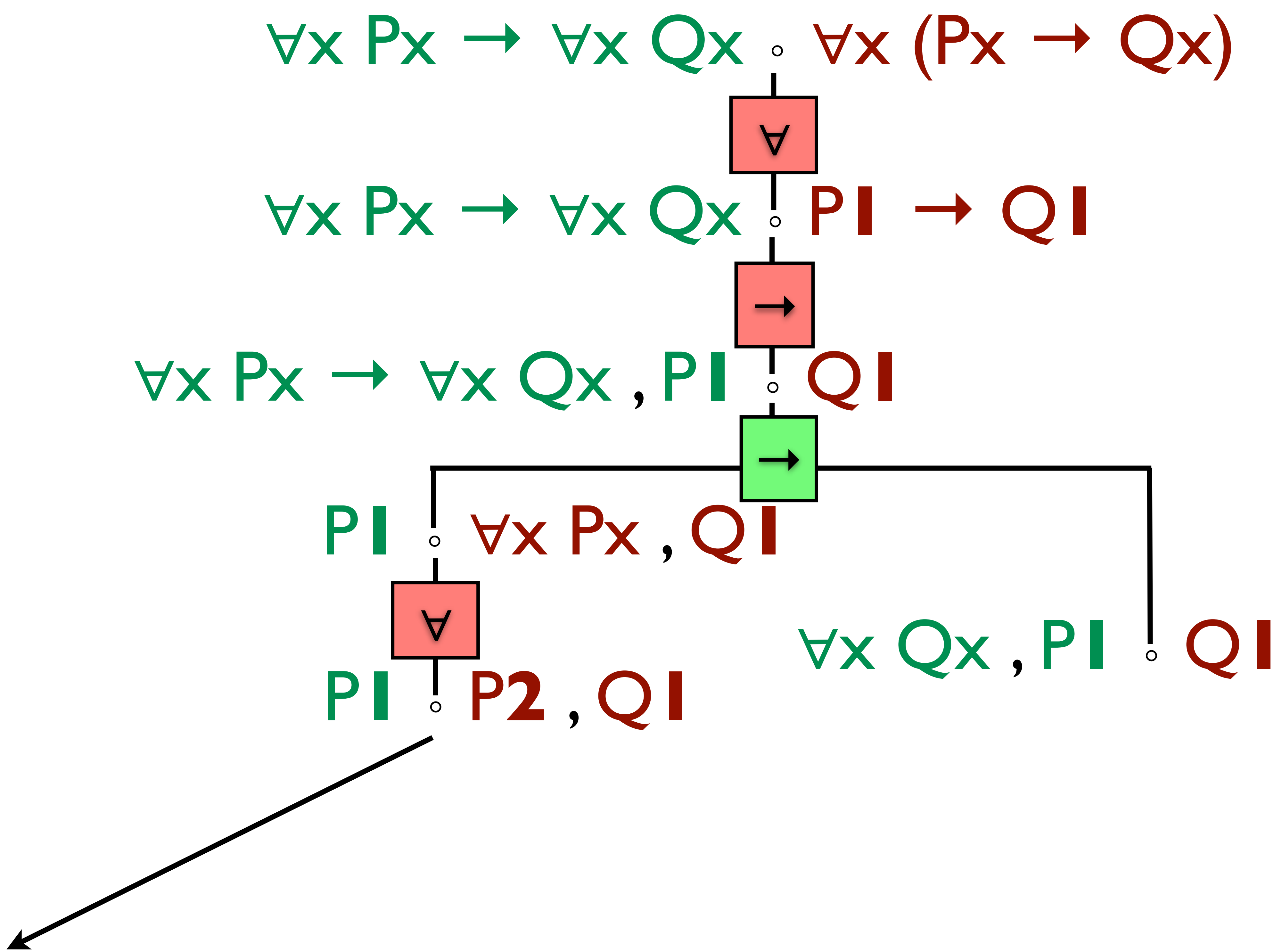
$\rightarrow$

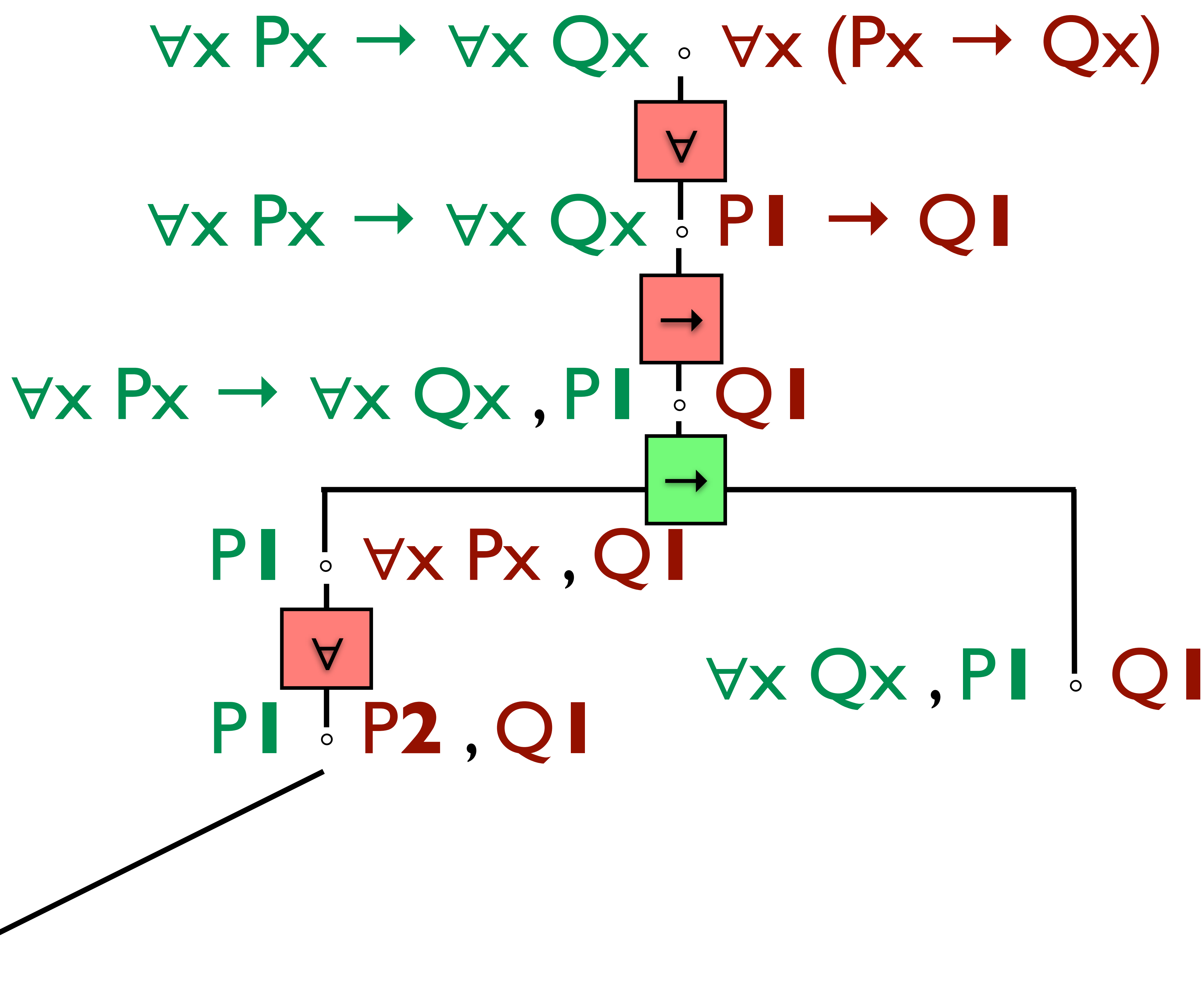

∀x Px → ∀x Qx ∘ ∀x (Px → Qx)
∀
∀x Px → ∀x Qx ∘ PI → QI
→
∀x Px → ∀x Qx , PI ∘ QI
→
PI ∘ ∀x Px , QI
∀x Qx , PI ∘ QI


∀x Px → ∀x Qx ∘ ∀x (Px → Qx)
∀
∀x Px → ∀x Qx ∘ PI → QI
→
∀x Px → ∀x Qx , PI ∘ QI
→
PI ∘ ∀x Px , QI
∀
∀x Qx , PI ∘ QI



∀x Px → ∀x Qx ∘ ∀x (Px → Qx)
∀
∀x Px → ∀x Qx ∘ P1 → Q1
→
∀x Px → ∀x Qx , P1 ∘ Q1
→
P1 ∘ ∀x Px , Q1
∀
P1 ∘ P2 , Q1
∀x Qx , P1 ∘ Q1


∀x Px → ∀x Qx ∘ ∀x (Px → Qx)
∀
∀x Px → ∀x Qx ∘ P1 → Q1
→
∀x Px → ∀x Qx , P1 ∘ Q1
→
P1 ∘ ∀x Px , Q1
∀
P1 ∘ P2 , Q1
∀x Qx , P1 ∘ Q1

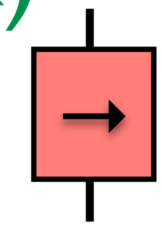


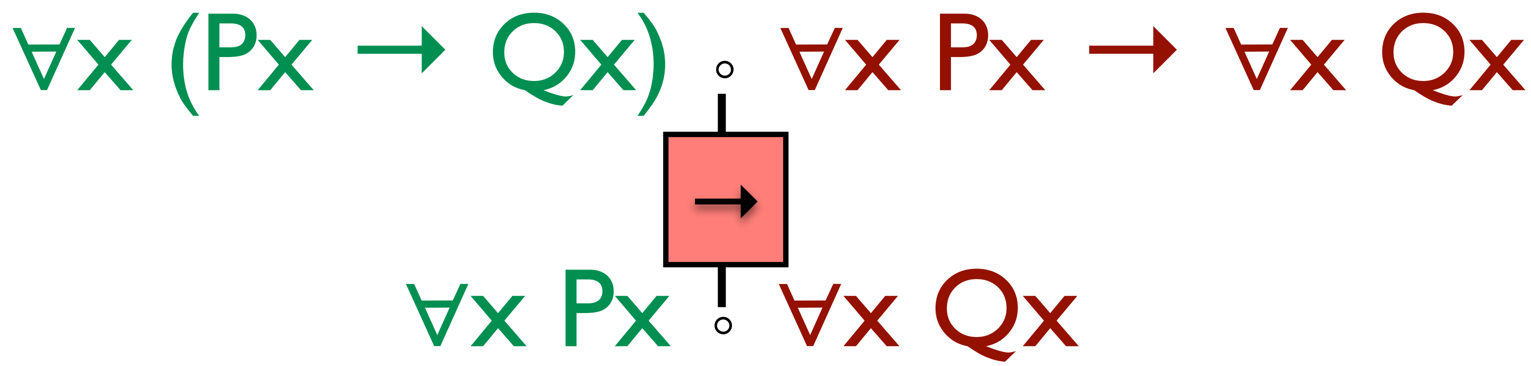
Open branch. Counter-example with two individuals 1 and 2, with 1 being a P and not a Q, and 2 a non-P.

$$\forall x\ (Px \rightarrow Qx) \models \forall x\ Px \rightarrow \forall x\ Qx$$

$$\forall x\,(Px \rightarrow Qx) \circ \forall x\, Px \rightarrow \forall x\, Qx$$

$\forall x\,(Px \to Qx) \circ \forall x\,Px \to \forall x\,Qx$



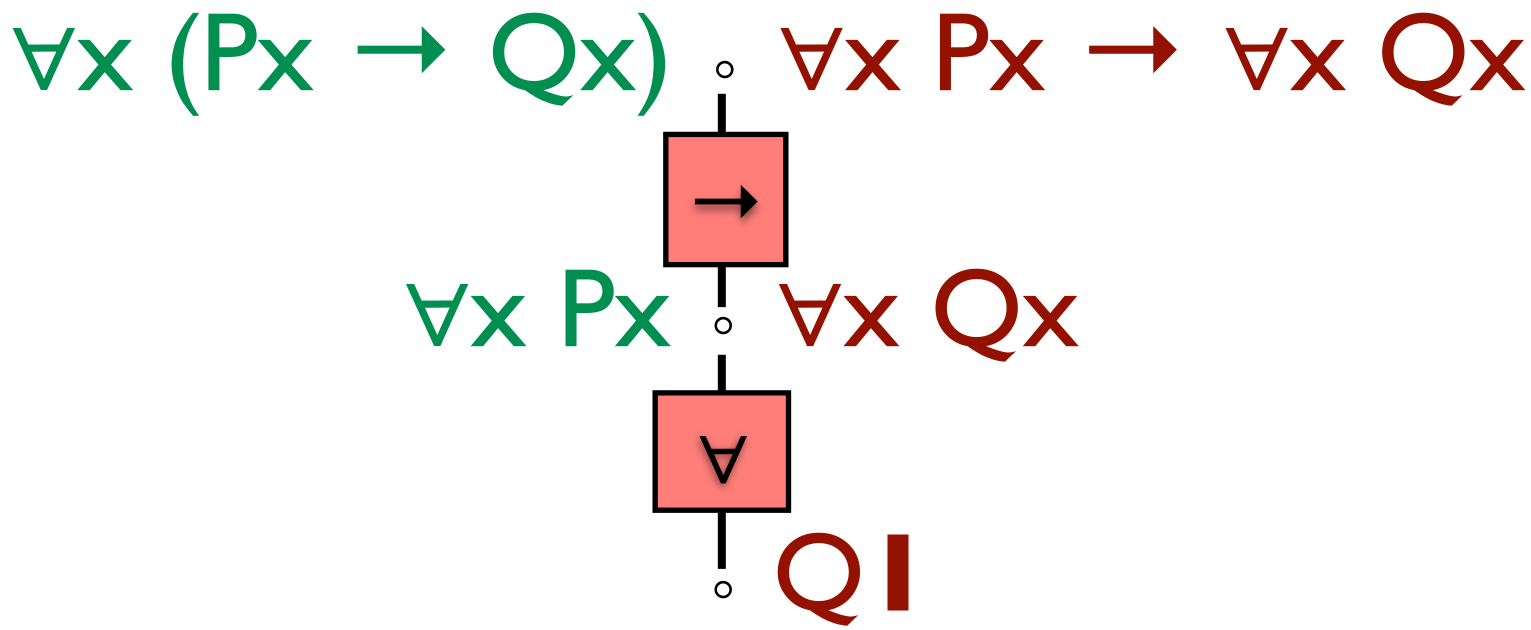

∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx

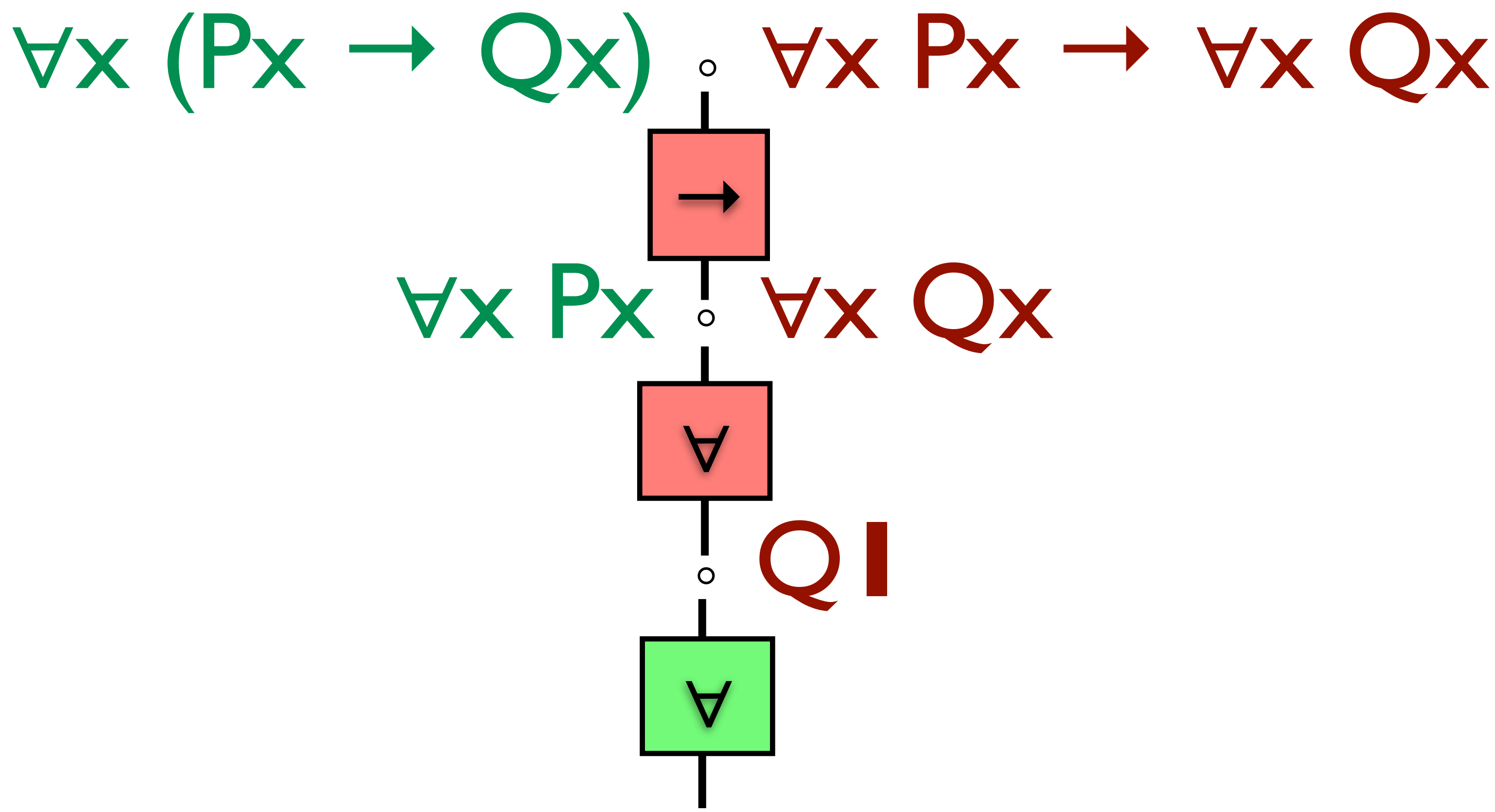
$\forall x\,(Px \rightarrow Qx) \circ \forall x\, Px \rightarrow \forall x\, Qx$

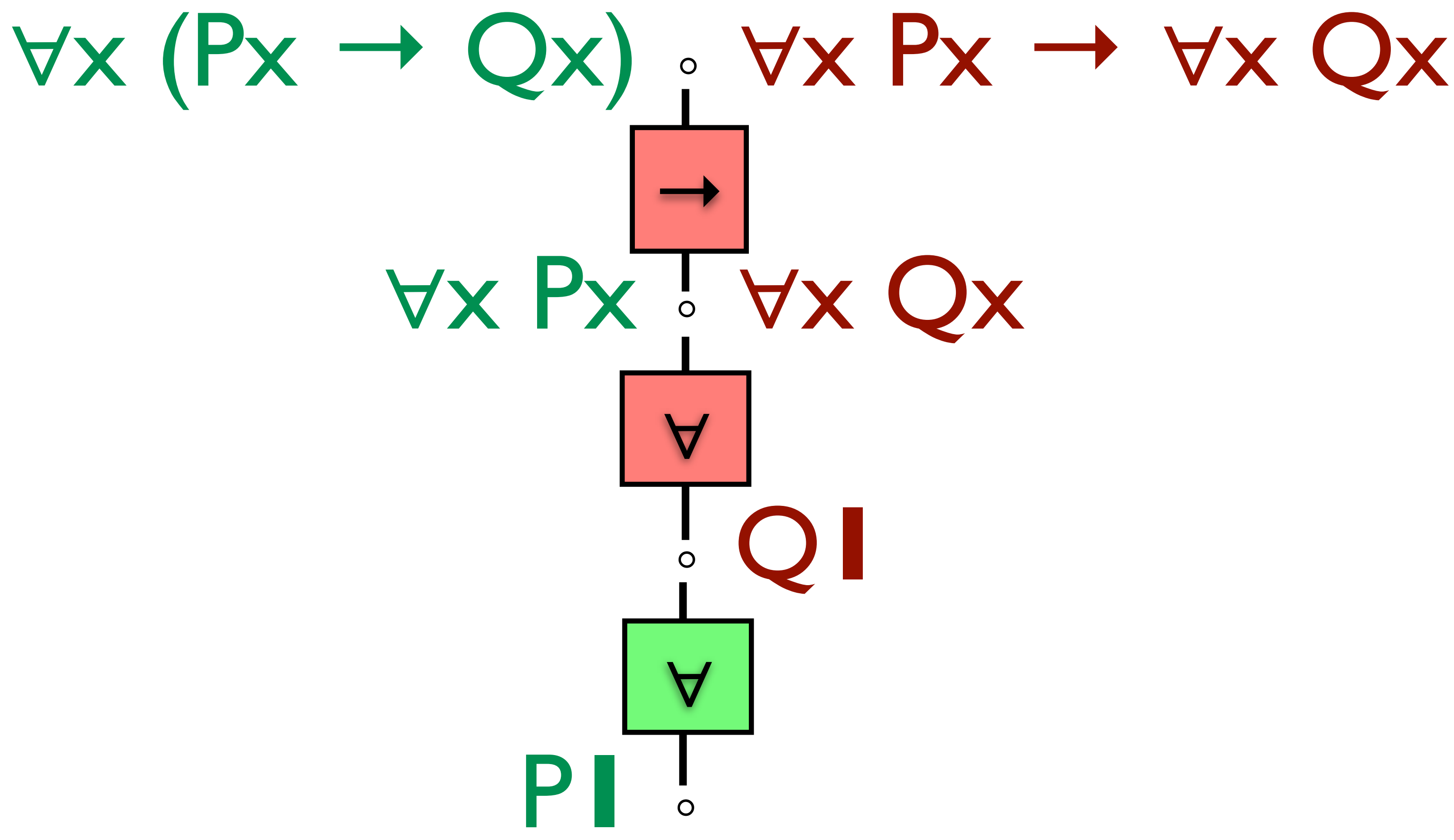
→

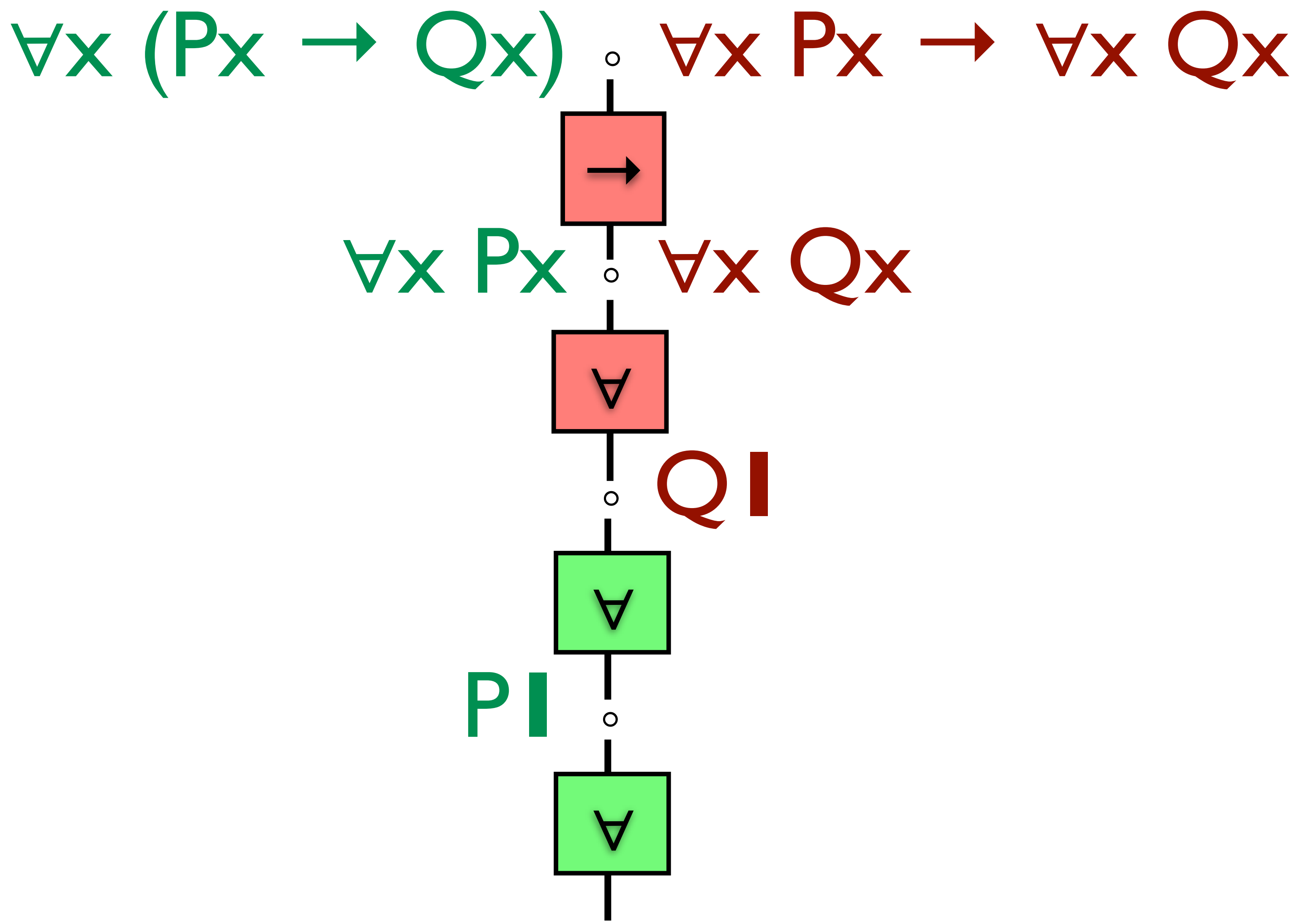
$\forall x\, Px \circ \forall x\, Qx$

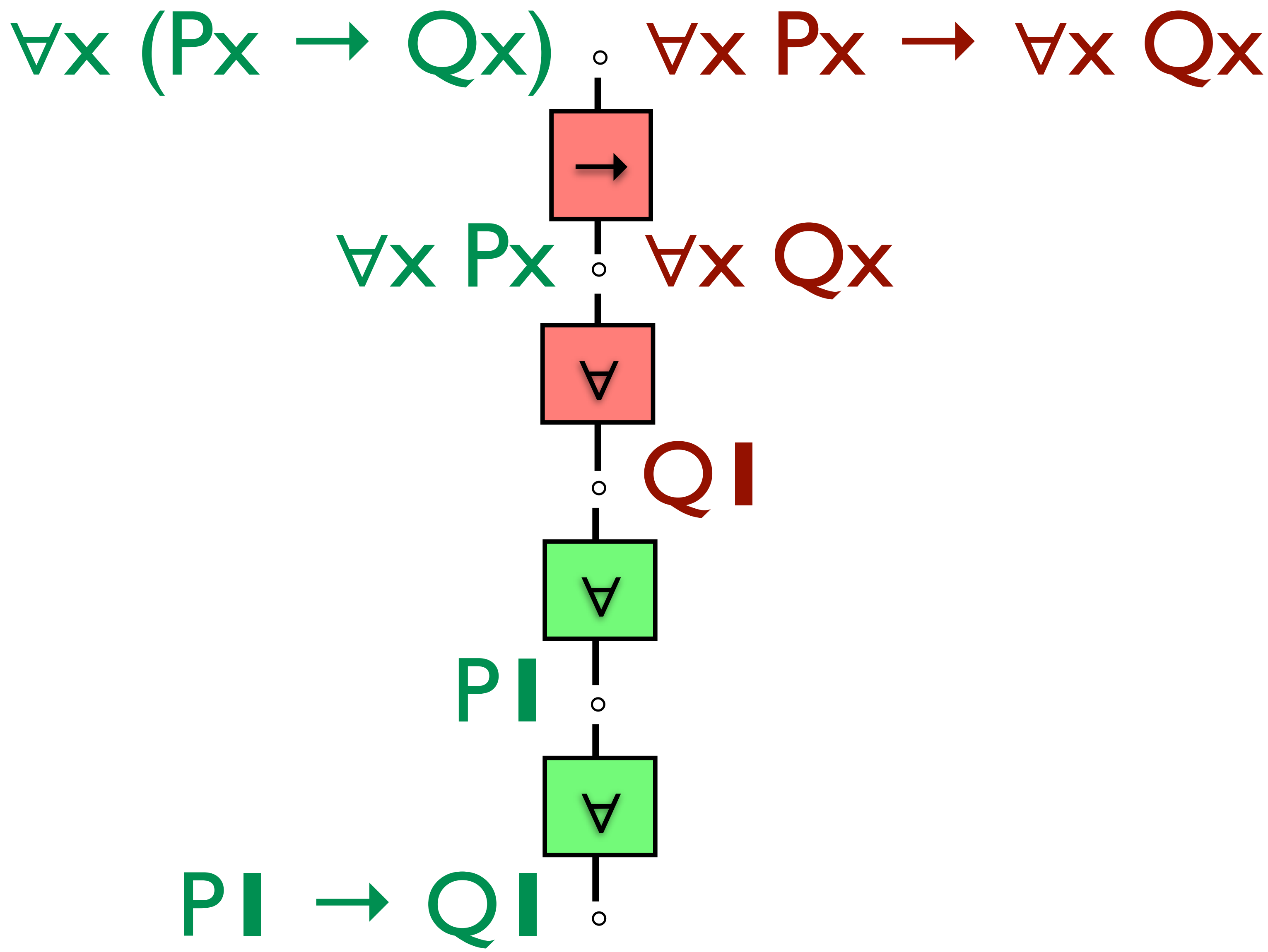
∀

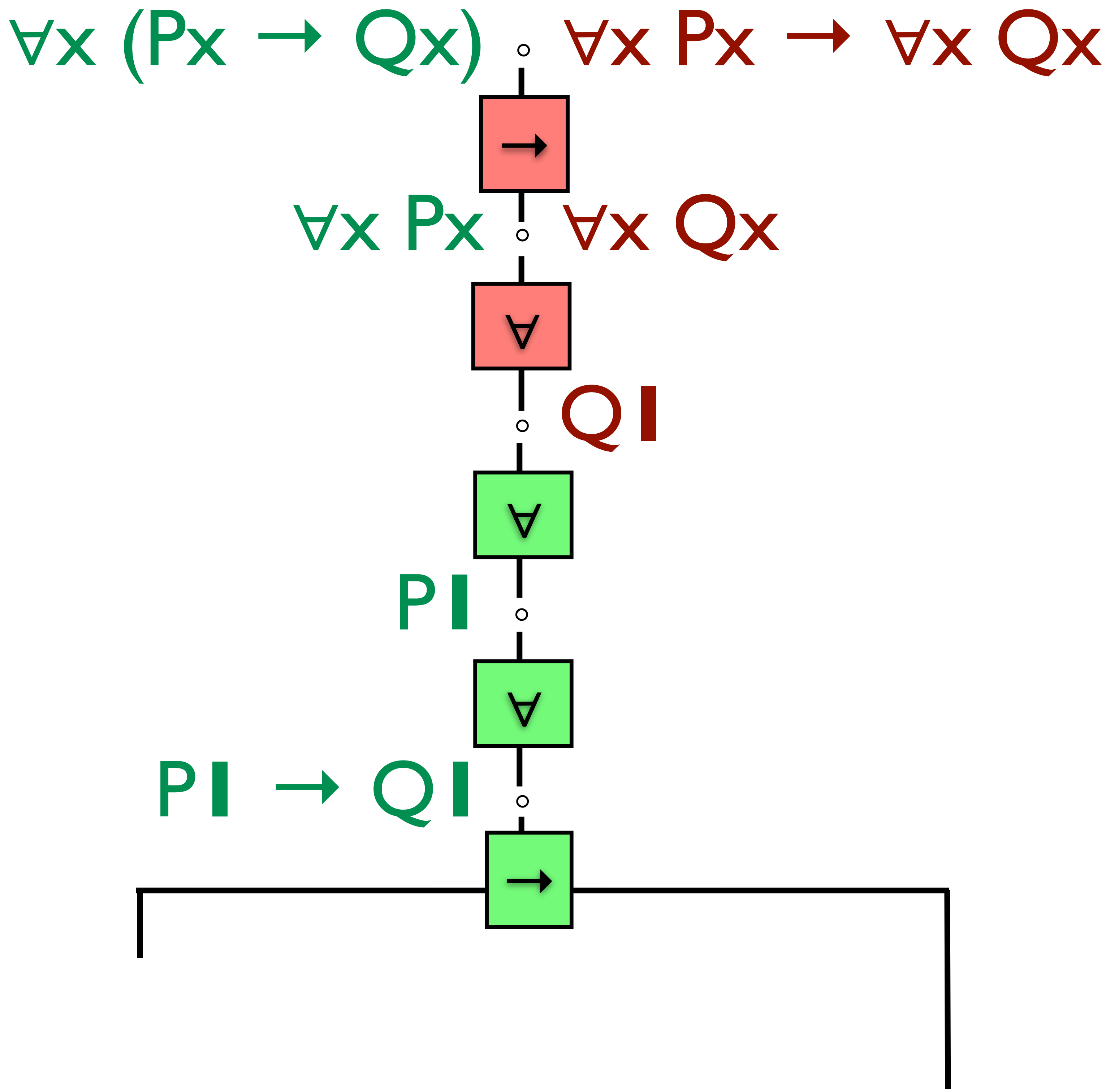

∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx
∀
Q1


∀x (Px → Qx) ∘ ∀x Px → ∀x Qx
→
∀x Px ∘ ∀x Qx
∀
∘ Q1
∀

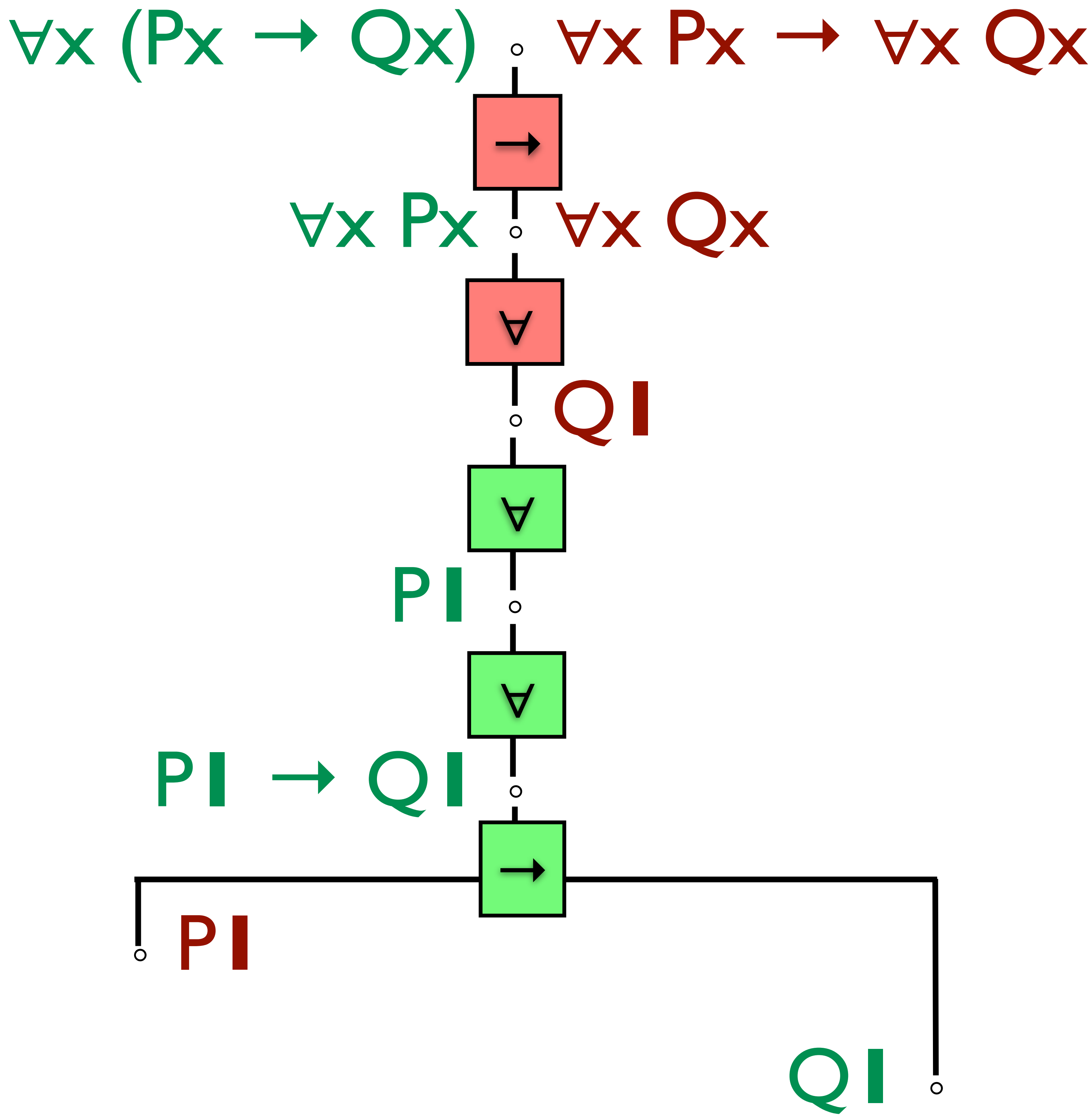

∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx
∀
Q
∀
P

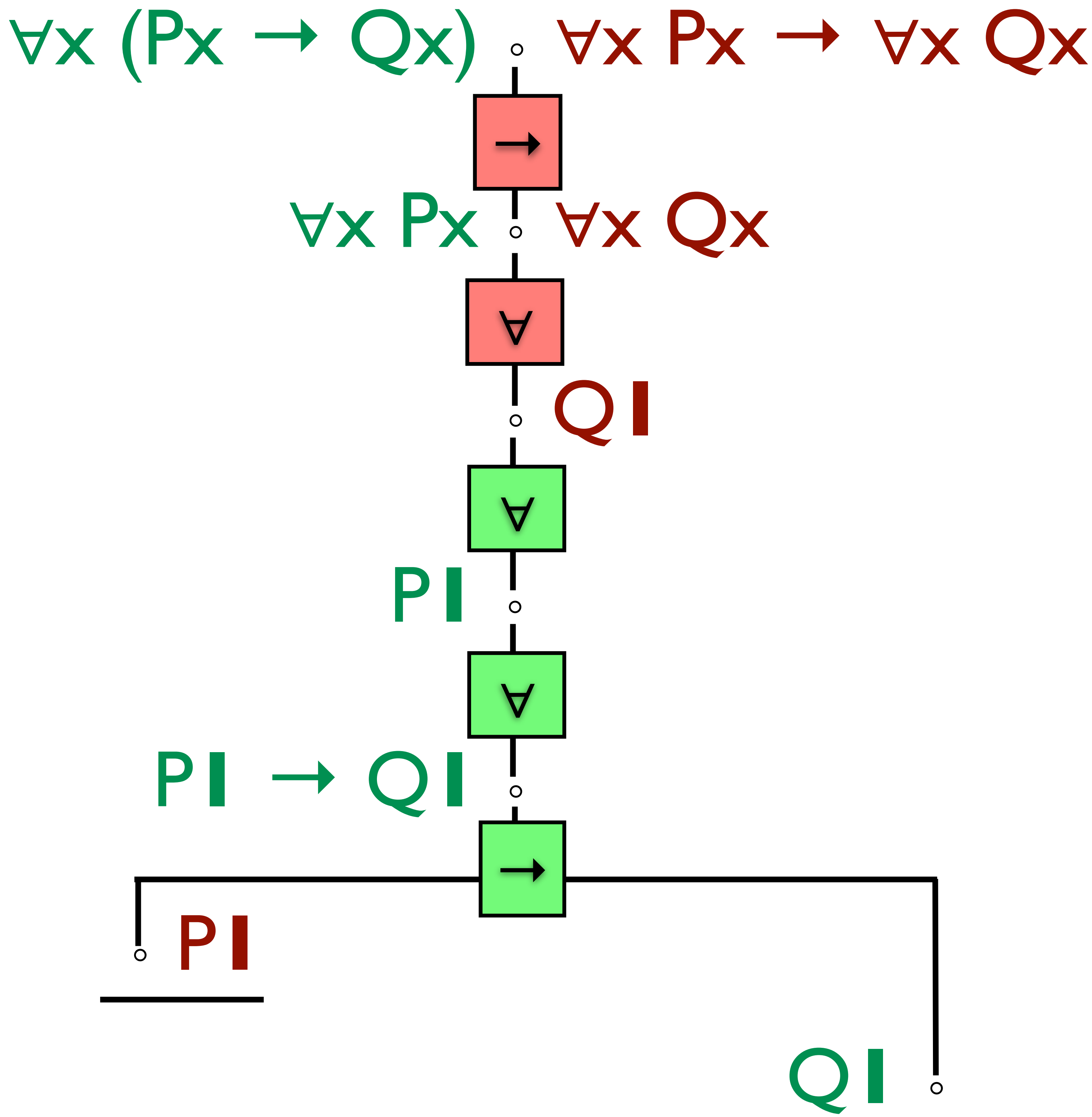

∀x (Px → Qx) ∘ ∀x Px → ∀x Qx
→
∀x Px ∘ ∀x Qx
∀
Q I
∀
P I
∀

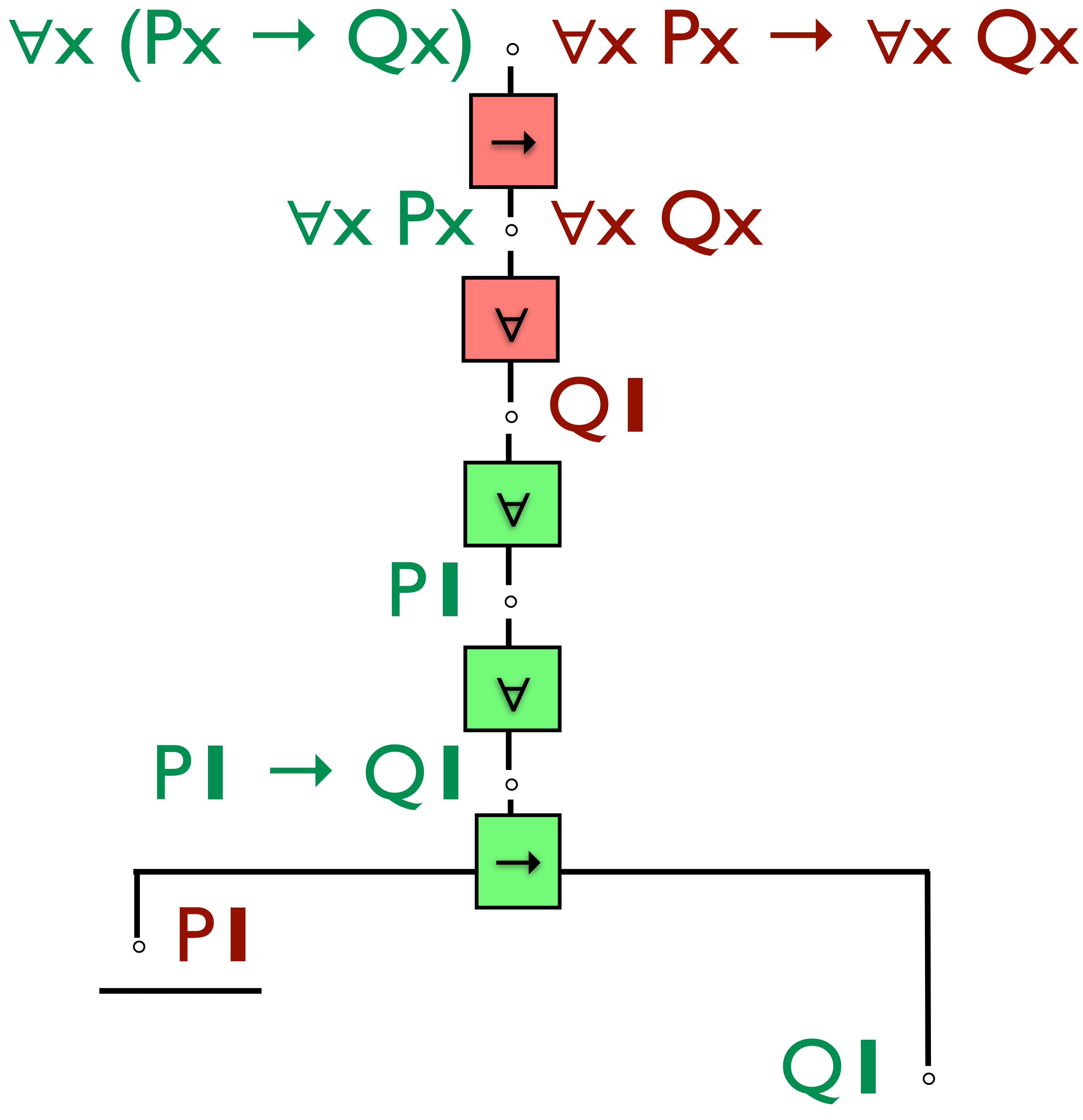

∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx
∀
QI
∀
PI
∀
PI → QI



∀x (Px → Qx) ∀x Px → ∀x Qx
→
∀x Px ∀x Qx
∀
Q I
∀
P I
∀
P I → Q I
→


∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx
∀
Q1
∀
P1
∀
P1 → Q1
→
P1
Q1


∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx
∀
Q1
∀
P1
∀
P1 → Q1
→
P1
Q1


∀x (Px → Qx)
∀x Px → ∀x Qx
→
∀x Px
∀x Qx
∀
Q I
∀
P I
∀
P I → Q I
→
P I
Q I

Quantifier rules

	Universal	Existential
Affirmative	∀xφ ***make*** everything φ all	∃xφ ***add*** a φ-er some
Negative	∃xφ ***make*** everything non-φ no	∀xφ ***add*** a non-φ-er not all

Problem 1

0 $\forall x\ Px \circ \exists x\ Px$ 0

???

Solution 1

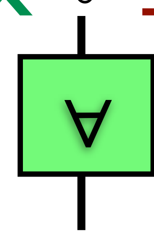
0 = 1   0 = 1

$\forall x\, Px \circ \exists x\, Px$

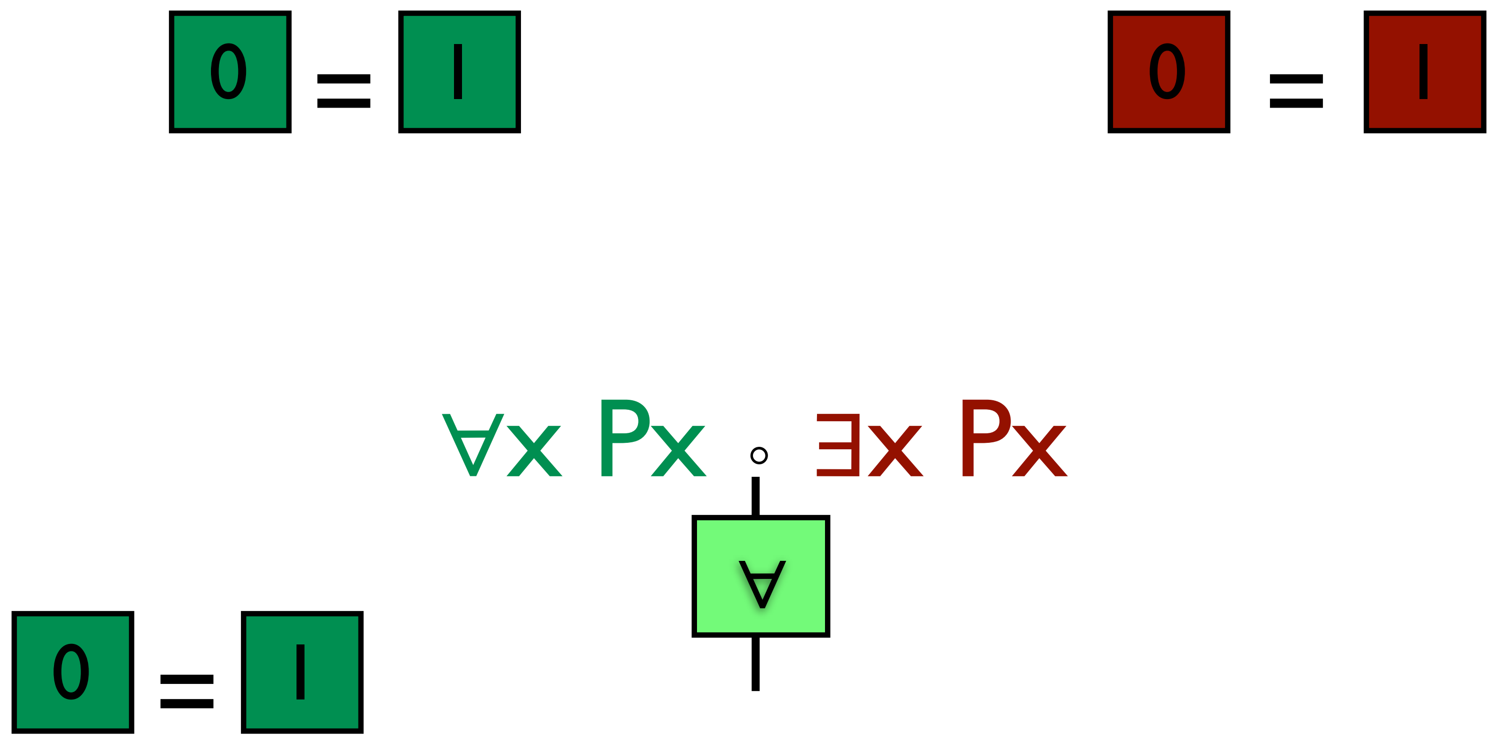
Solution 1

0 = 1 0 = 1

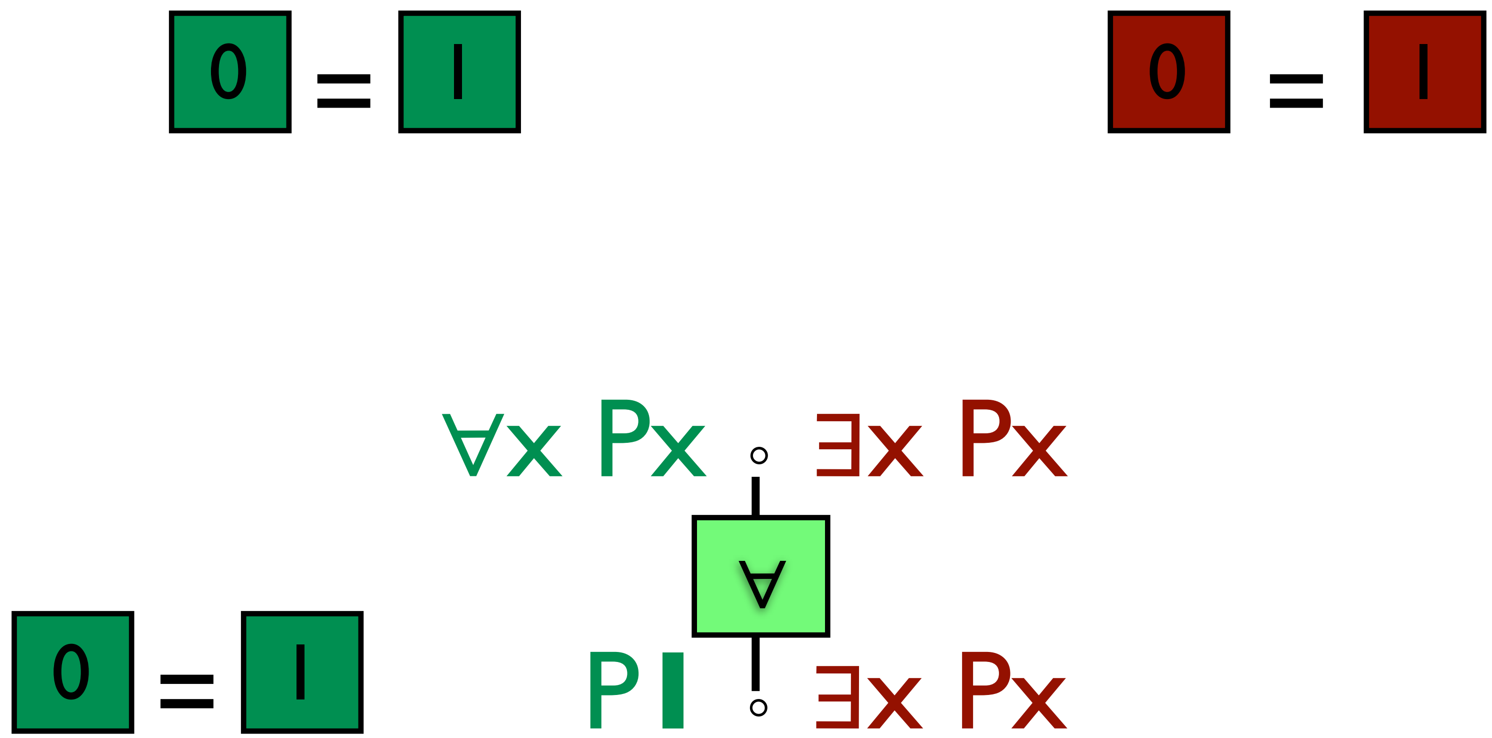
$\forall x\ Px \circ \exists x\ Px$



Solution 1



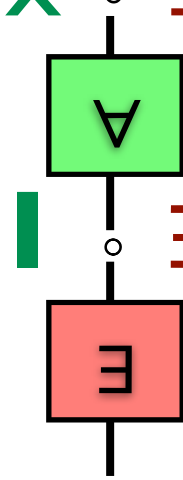
Solution 1



Solution 1

0 = 1　　　　0 = 1

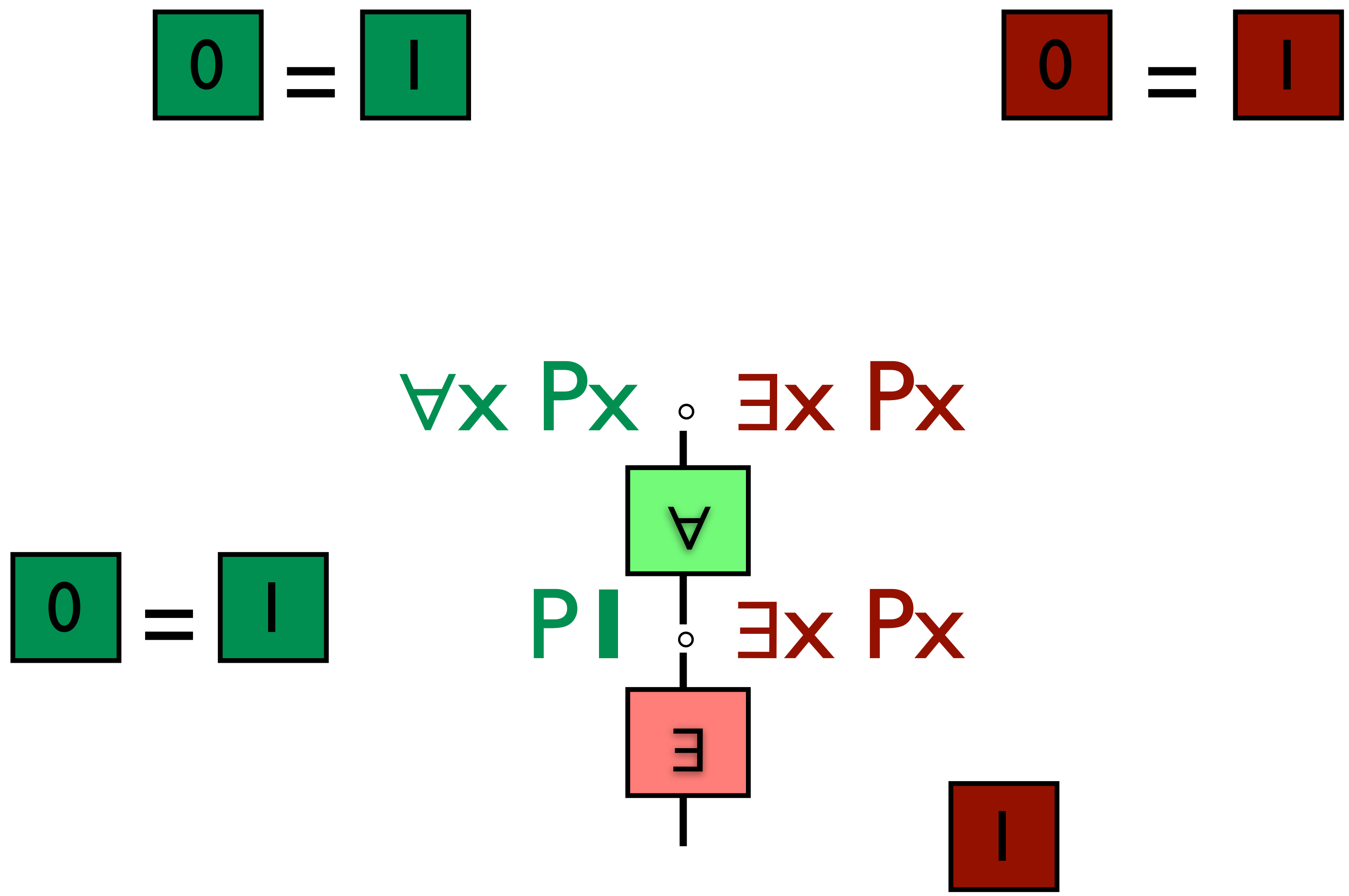
∀x Px ∘ ∃x Px



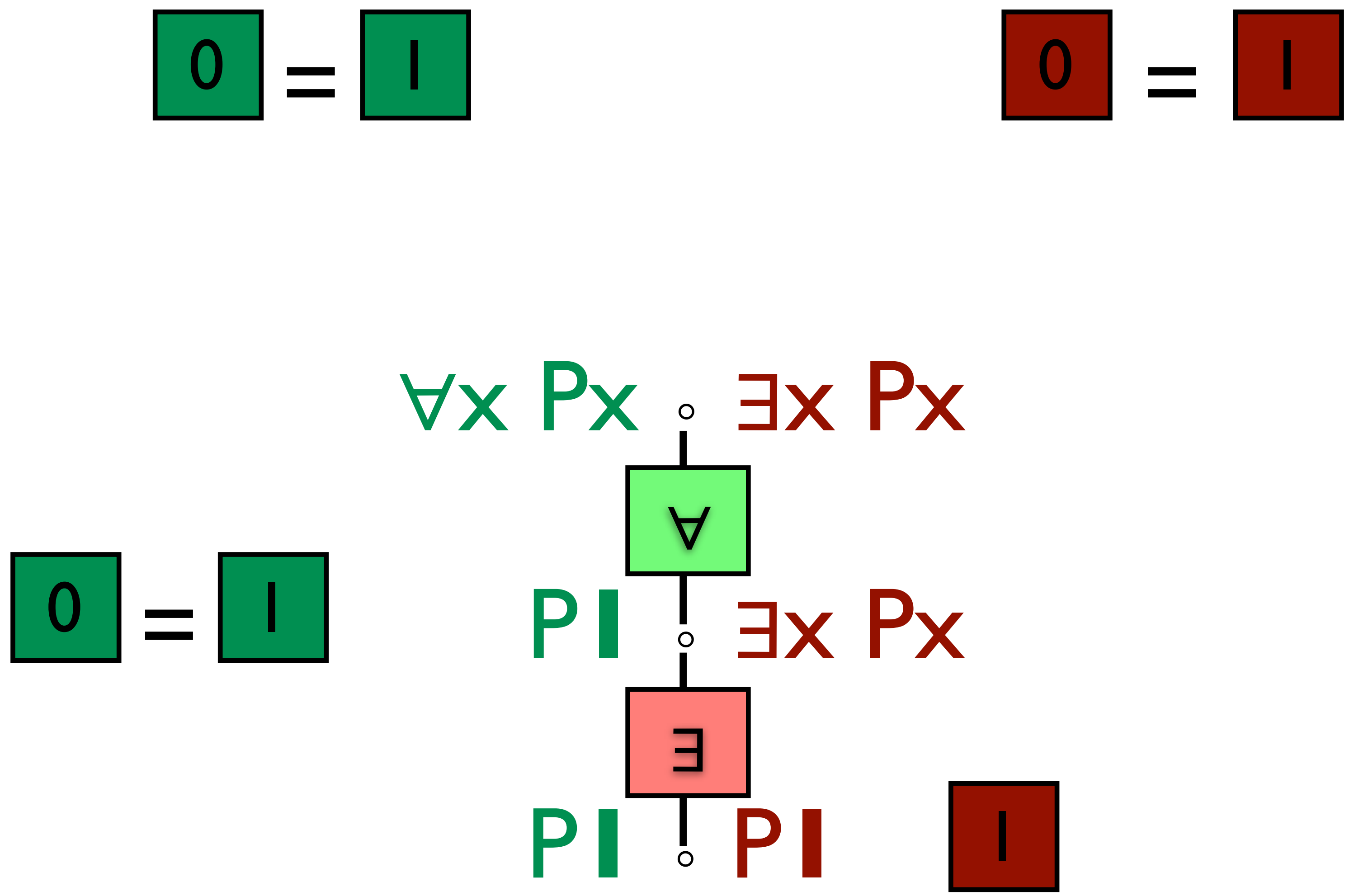
0 = 1

P 1 ∘ ∃x Px

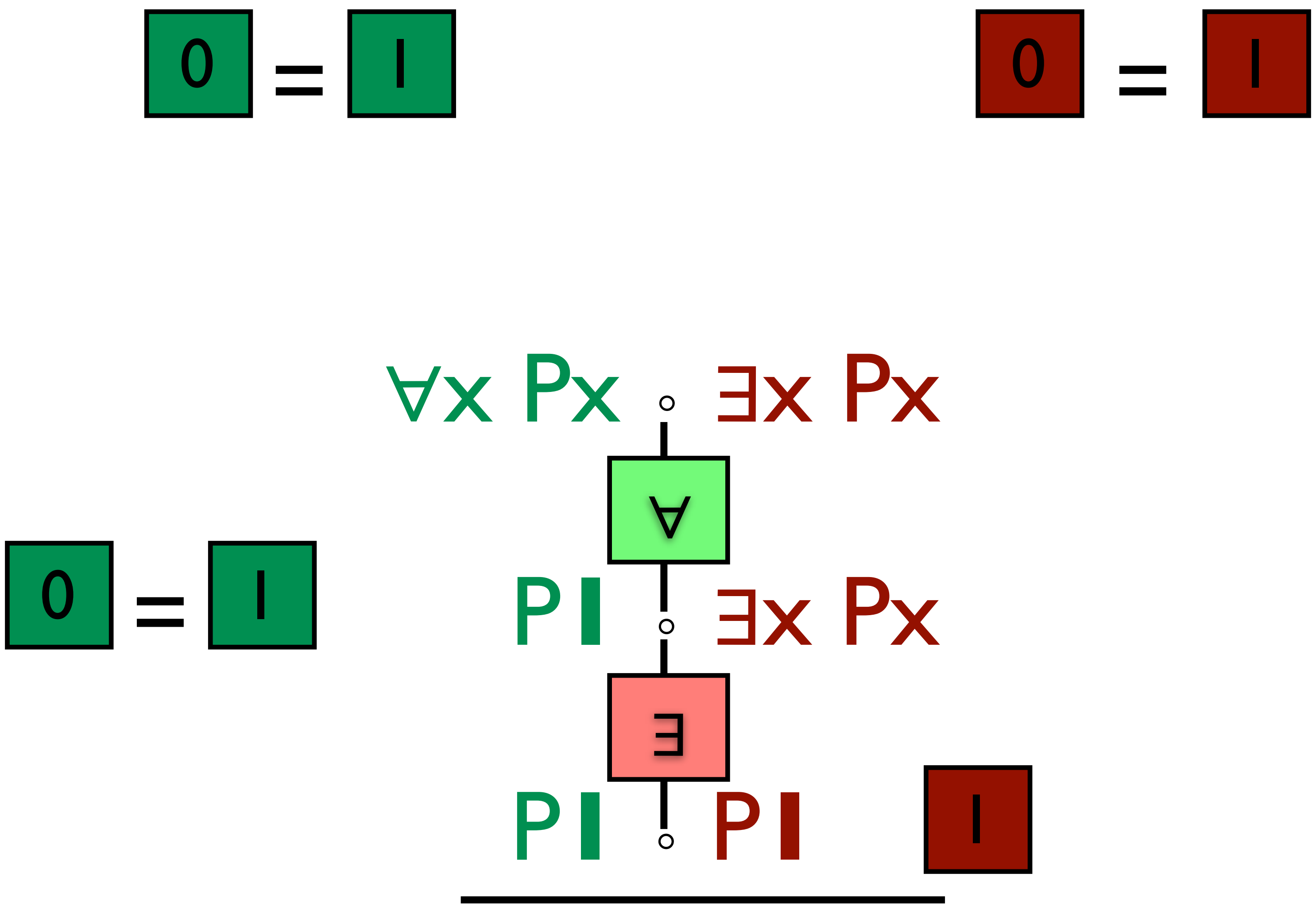
Solution 1



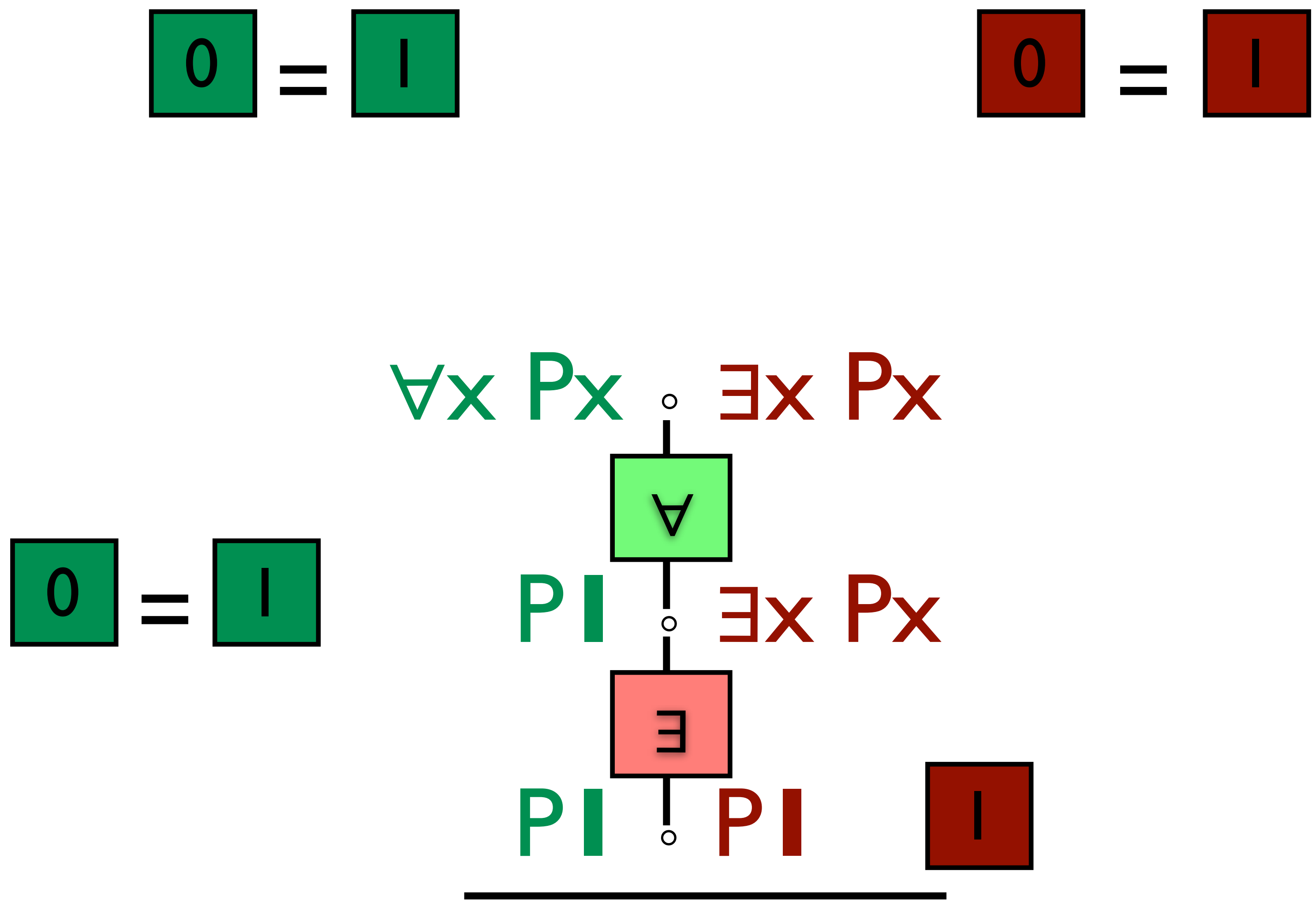
Solution 1



Solution 1



Solution 1



In every situation there is always at least one object!

Problem 2

- $\exists x\ (Tx \rightarrow \forall y Ty)$

Problem 2

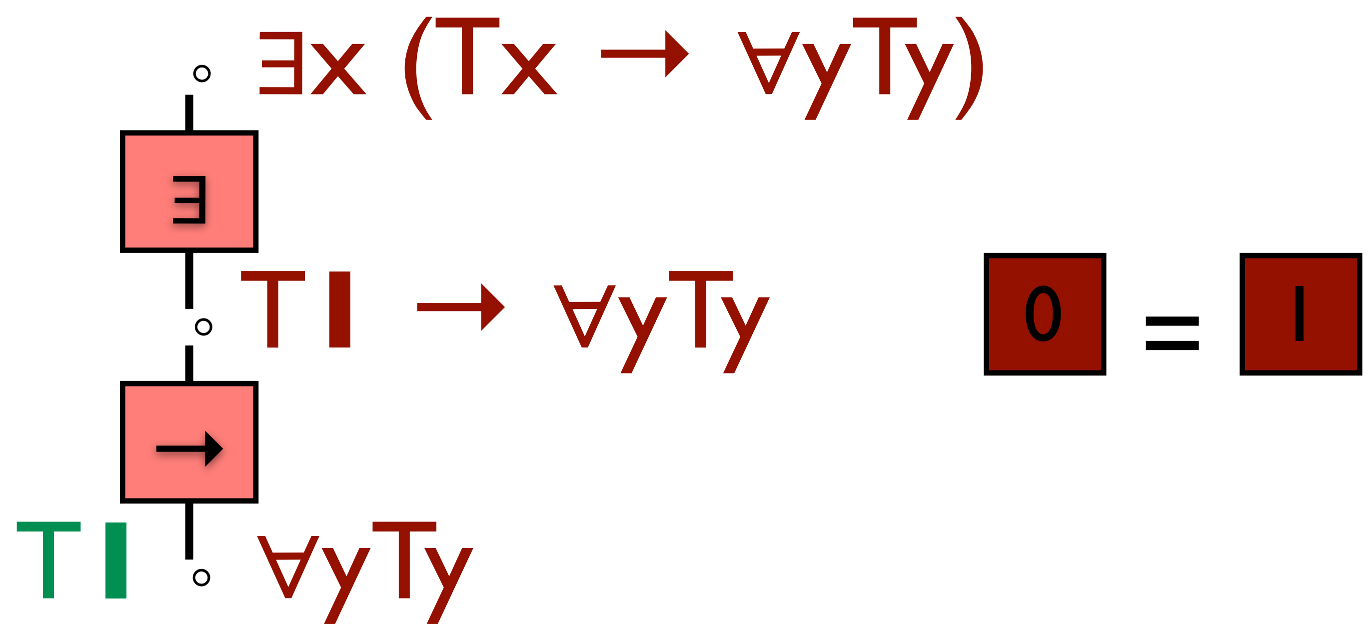
$\exists x\,(Tx \rightarrow \forall y Ty)$

∃

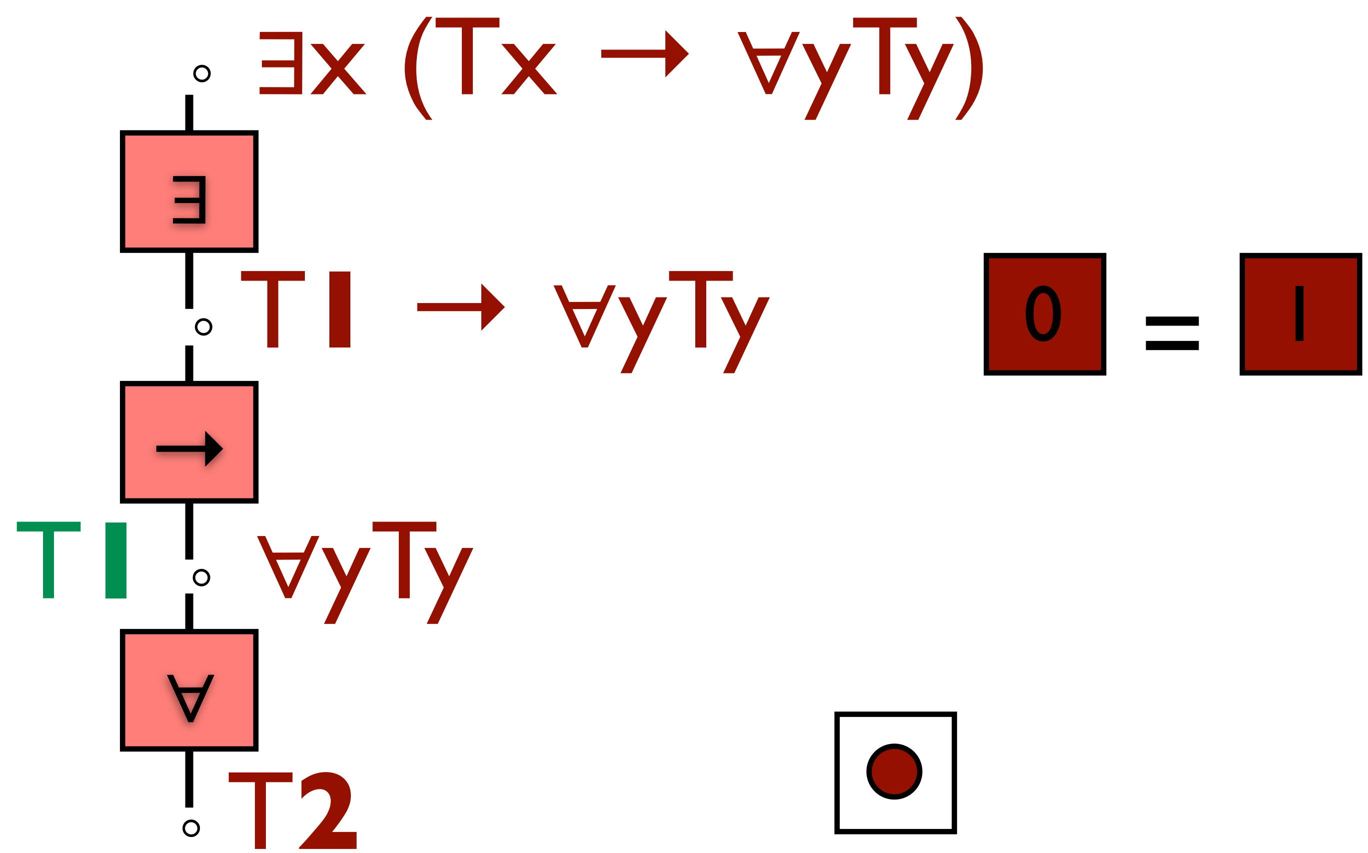
$T\blacksquare \rightarrow \forall y Ty$

0 = 1

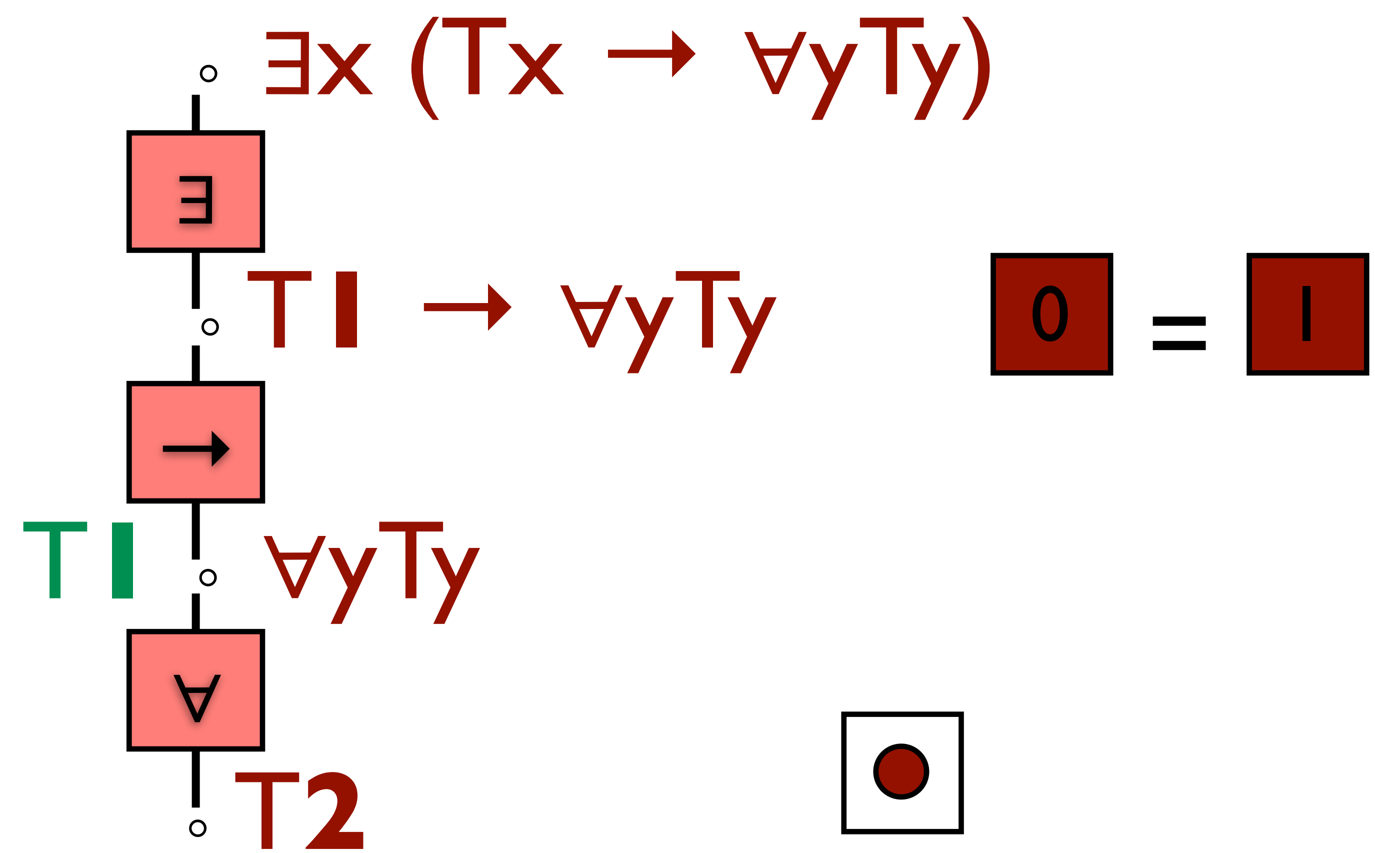
Problem 2



Problem 2

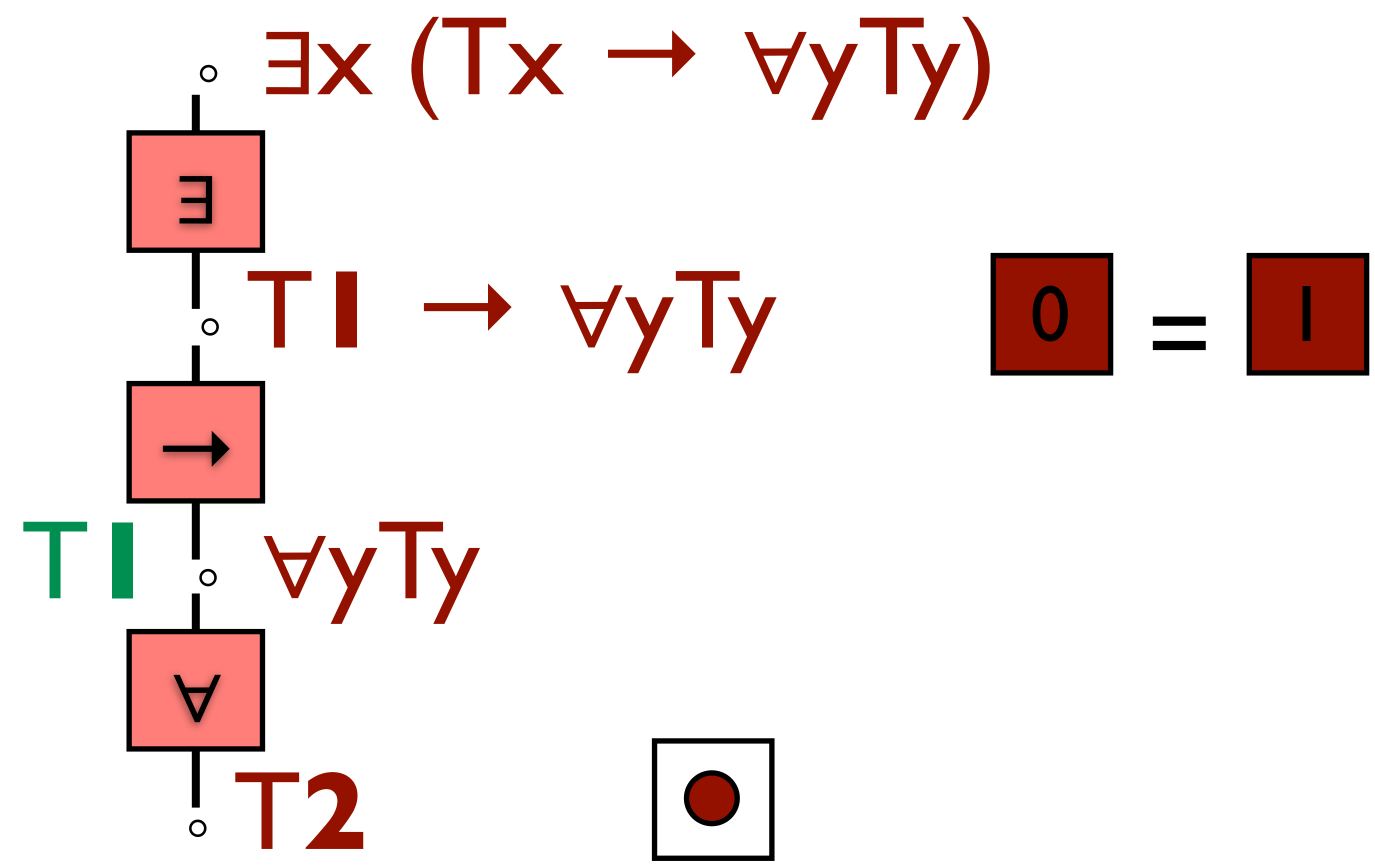


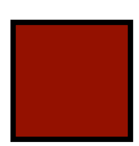
Problem 2



????? ∃x (Tx → ∀yTy) is always true (valid)!

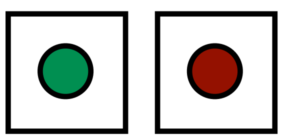
Problem 2





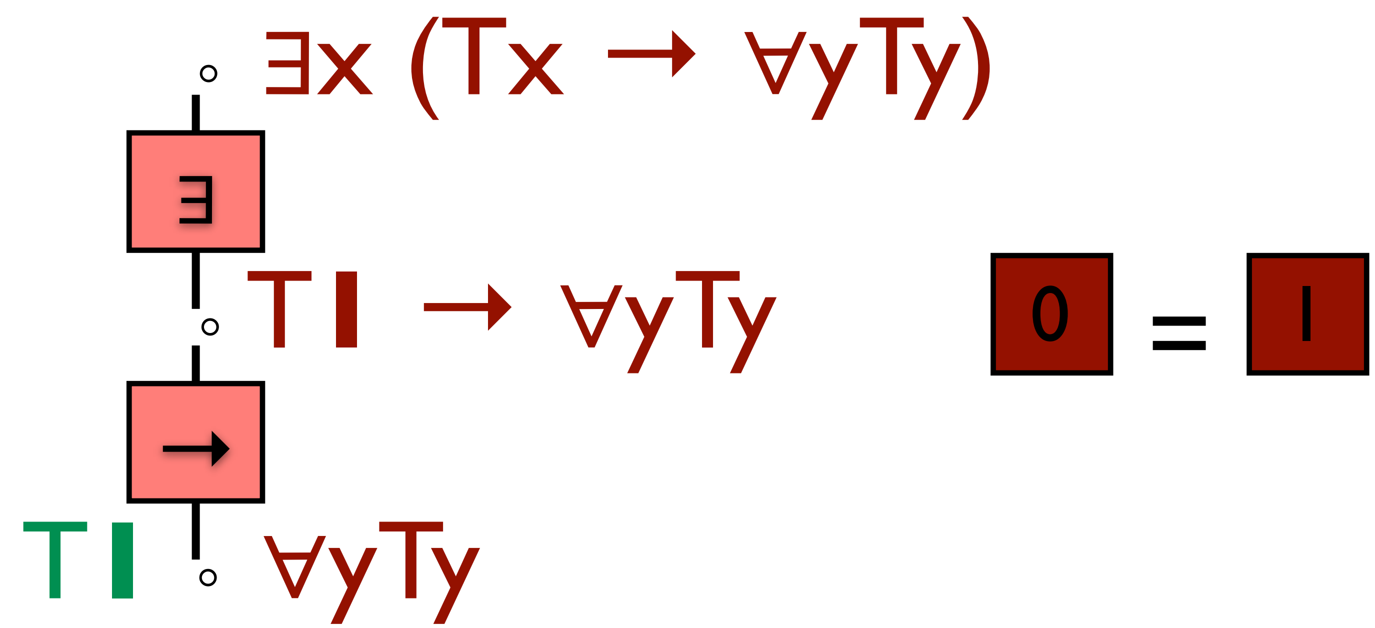
has not been applied to all objects (2)!

Solution

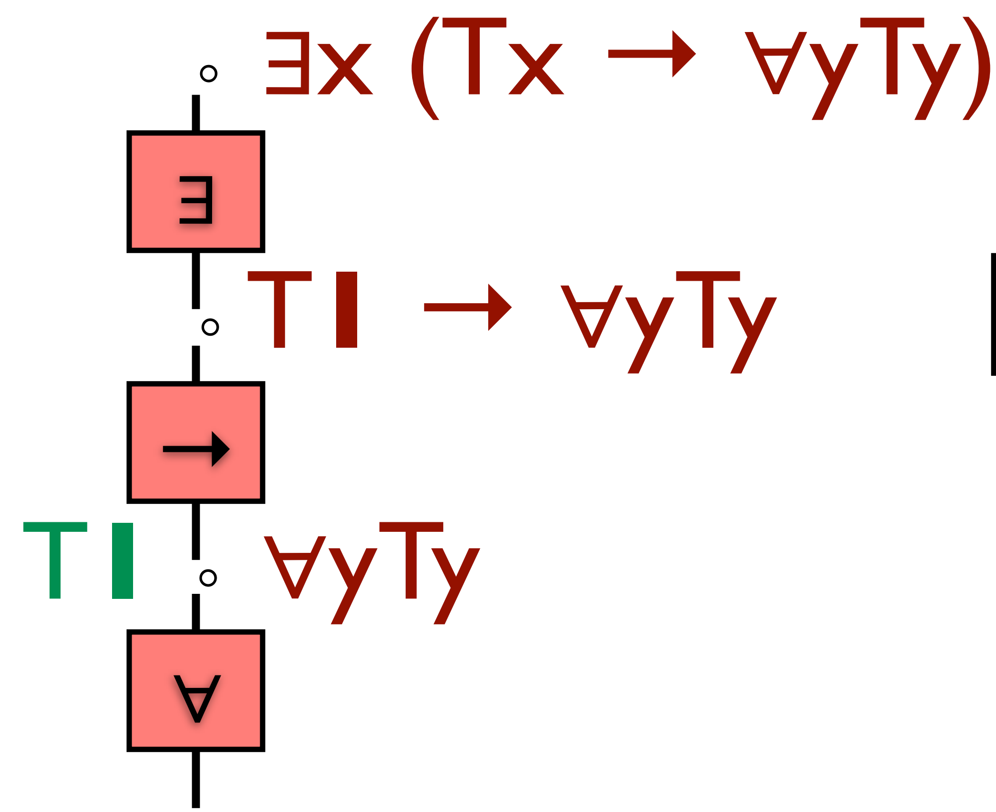
 =

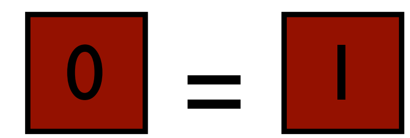
- Add a φ-er / non-φ-er, *ánd*
- *re-activate all universals* (all = $\forall$, no = $\exists$)

Solution 2

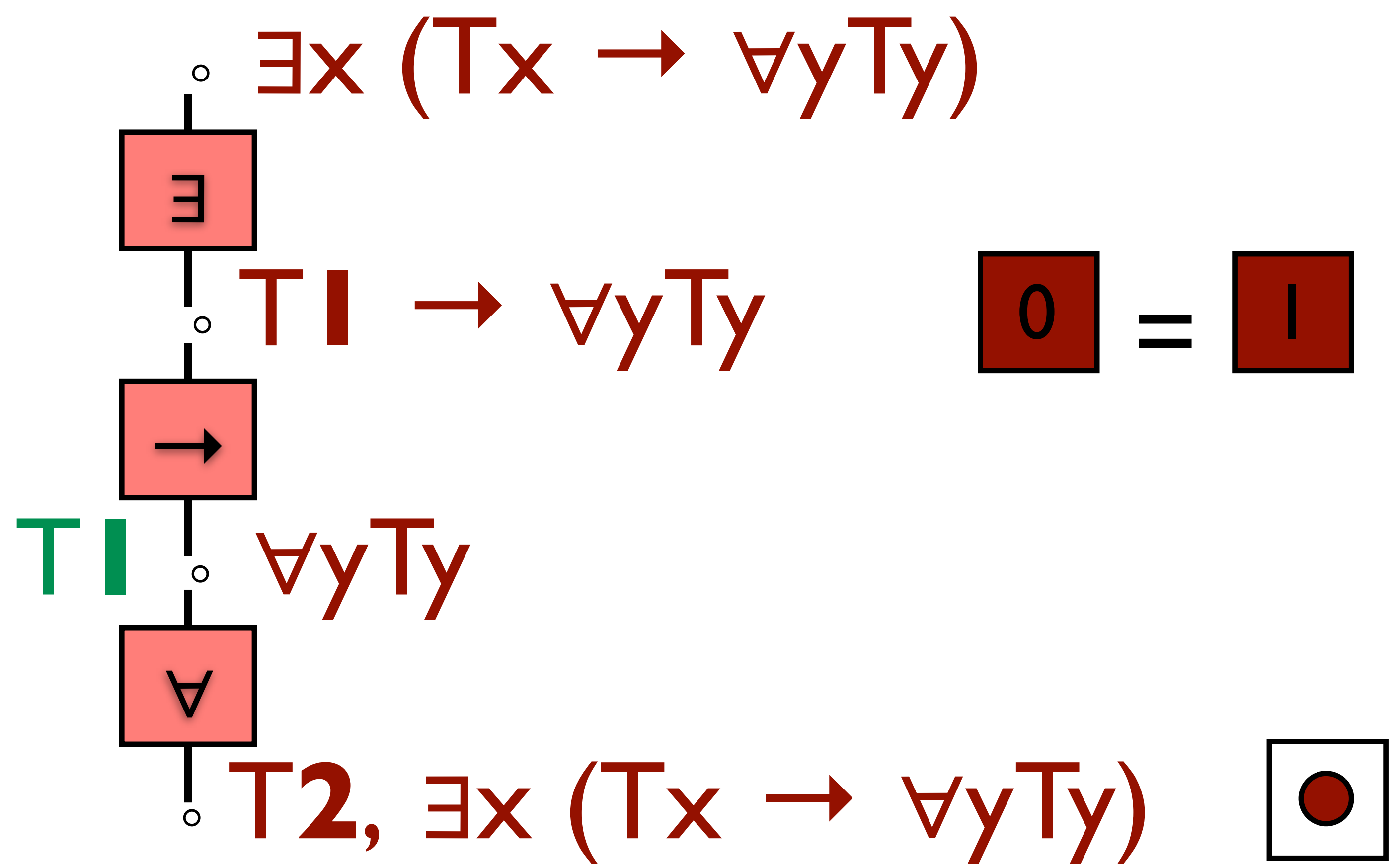


Solution 2

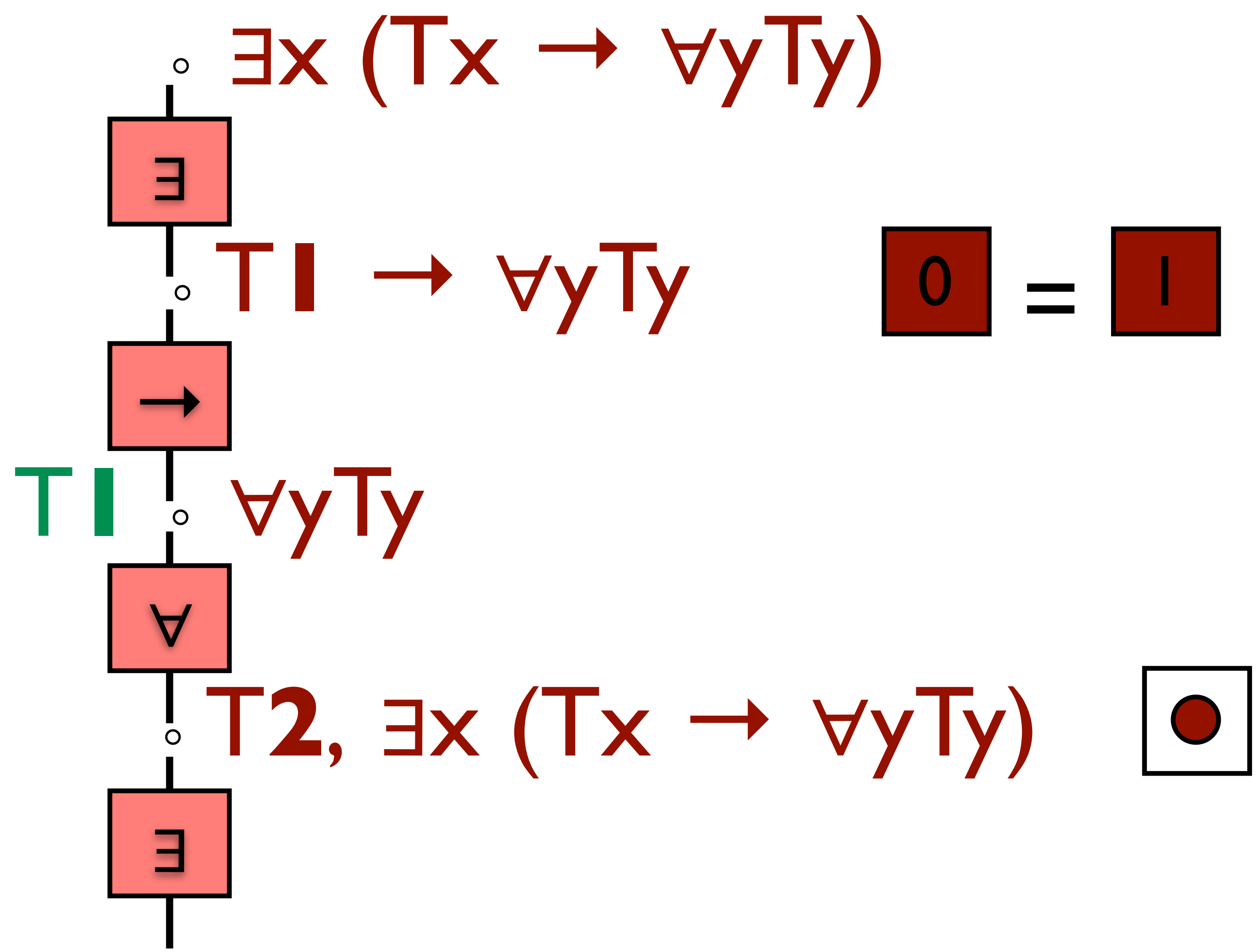




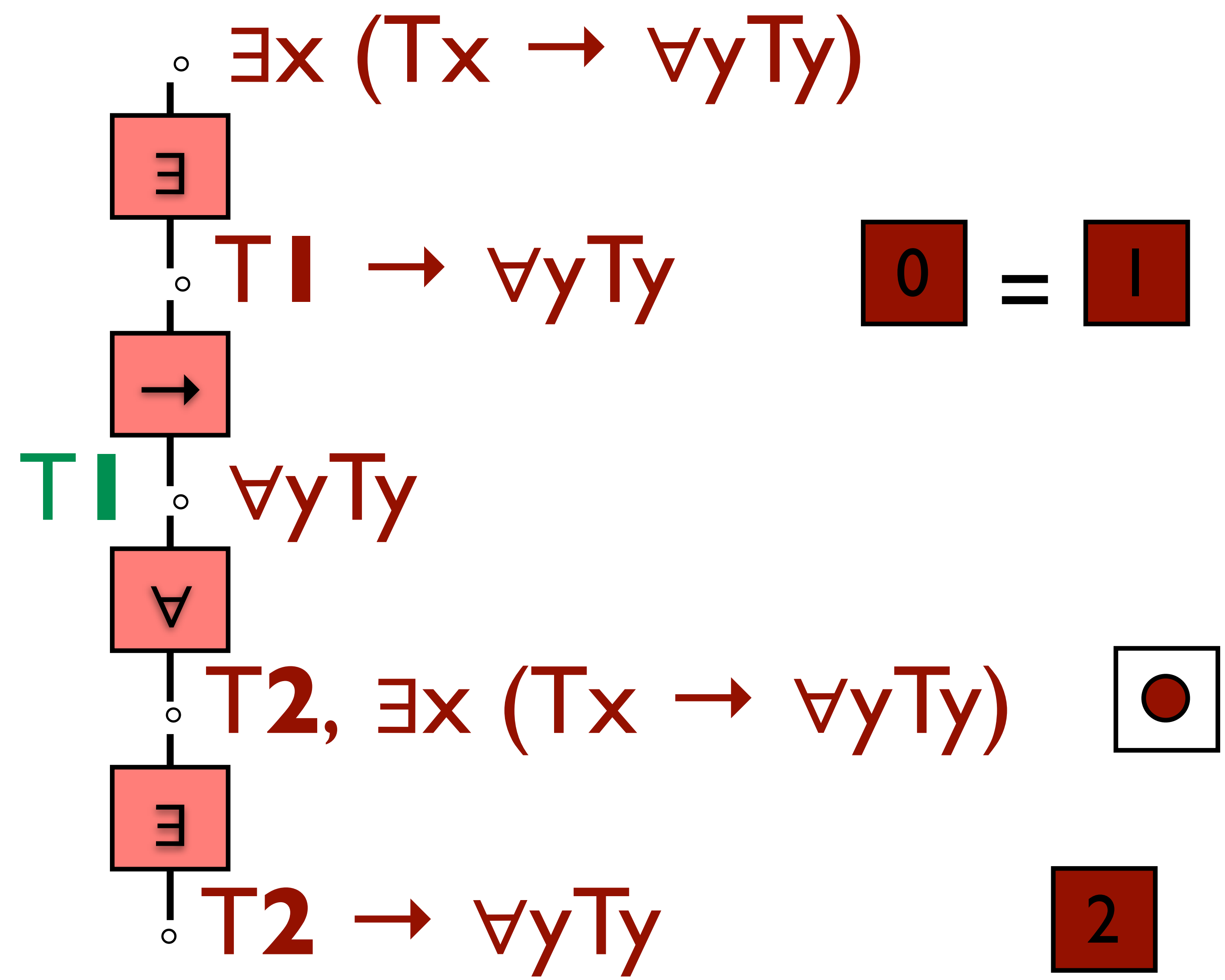
Solution 2



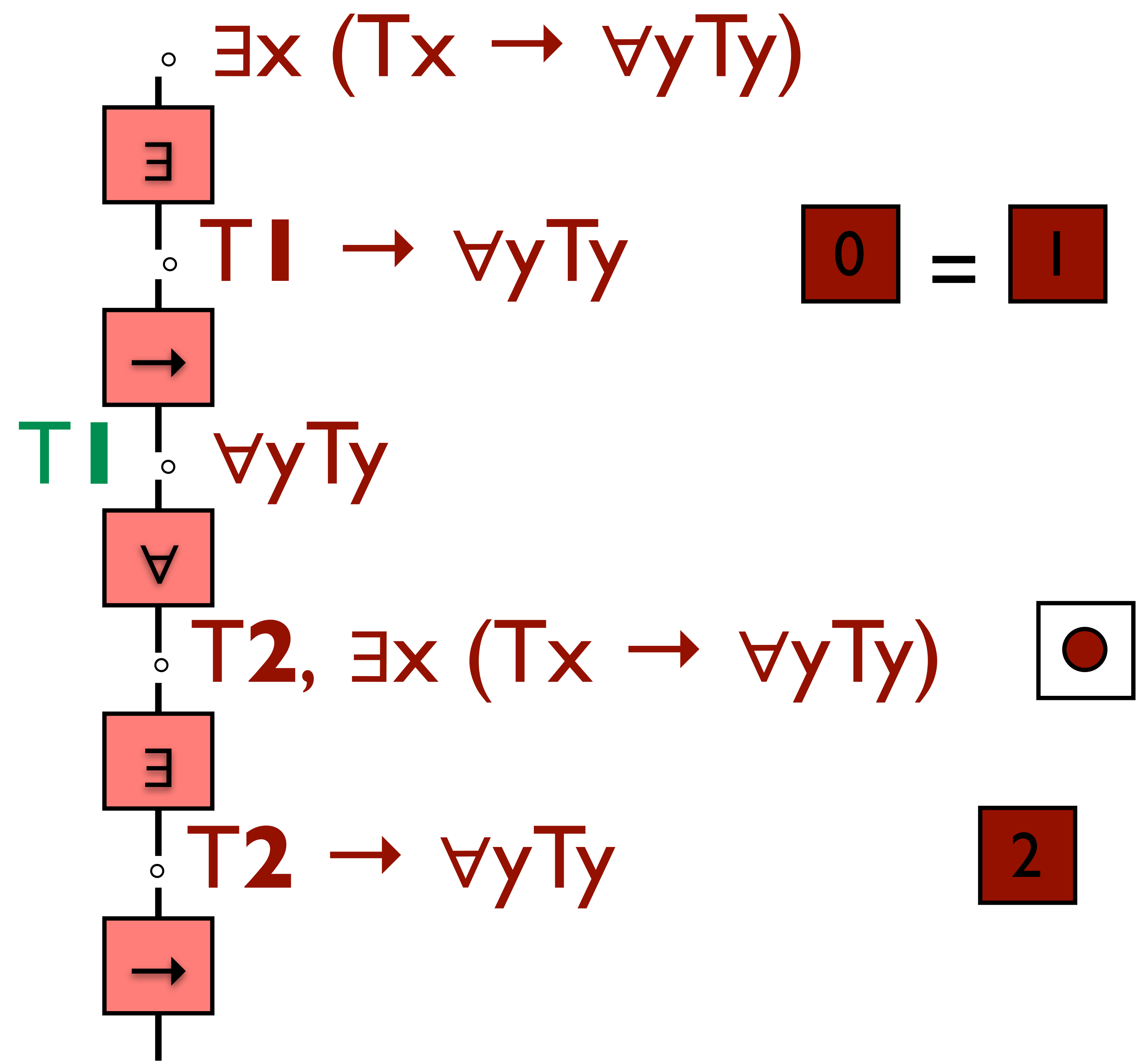
Solution 2



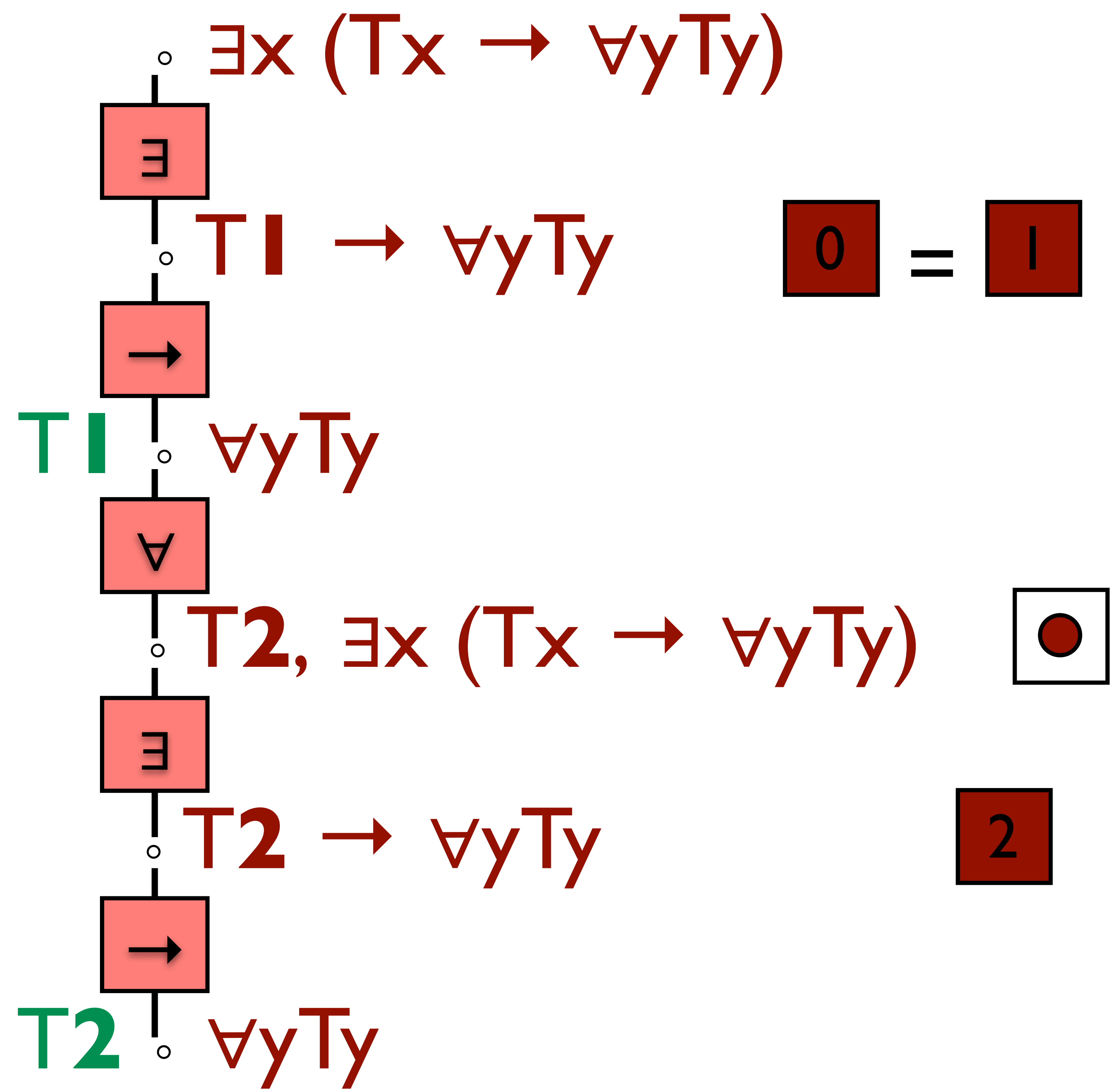
Solution 2



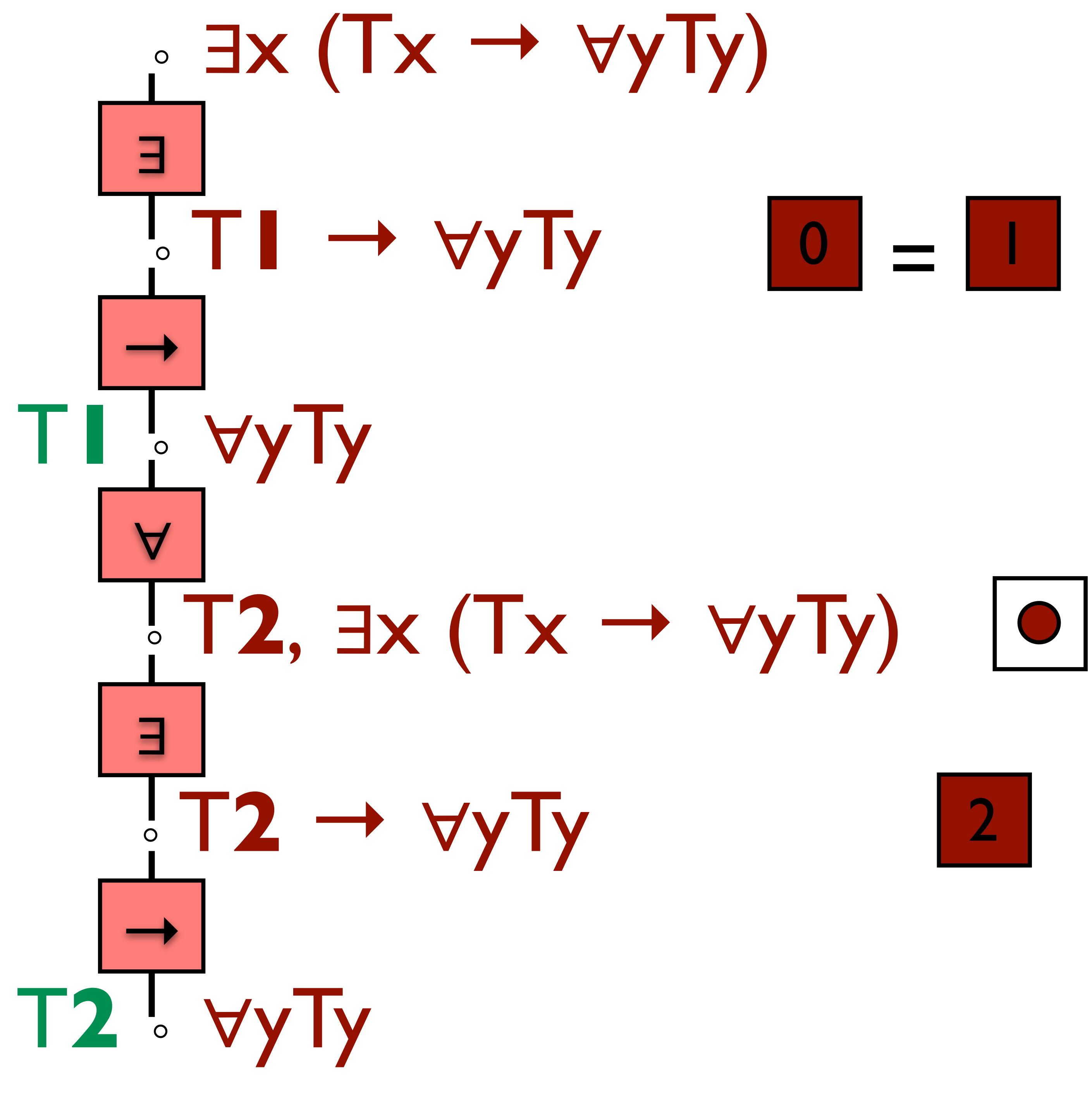
Solution 2



Solution 2



Solution 2

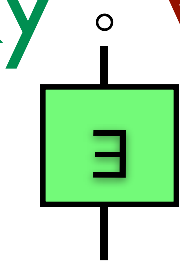


Rules for quantifiers (fin)

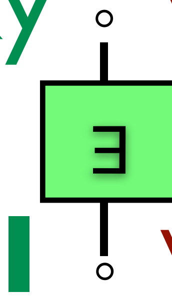
	Universal	Existential
Affirmative	$\forall x\varphi$ ***make*** everything φ all	$\exists x\varphi$ ***add*** a φ-er + ***re-activate*** $\forall$ & $\exists$ some
Negative	$\exists x\varphi$ ***make*** everything non-φ no	$\forall x\varphi$ ***add*** a non-φ-er + ***re-activate*** $\forall$ & $\exists$ not all

$\exists y\ \forall x\ Rxy \circ \forall x\ \exists y\ Rxy$

$\exists y\ \forall x\ Rxy$ $\quad$ $\forall x\ \exists y\ Rxy$



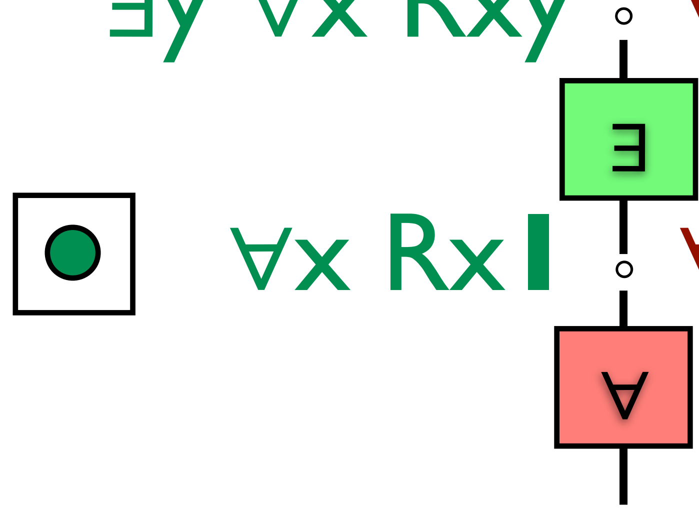
$\exists y\ \forall x\ Rxy \quad \forall x\ \exists y\ Rxy$

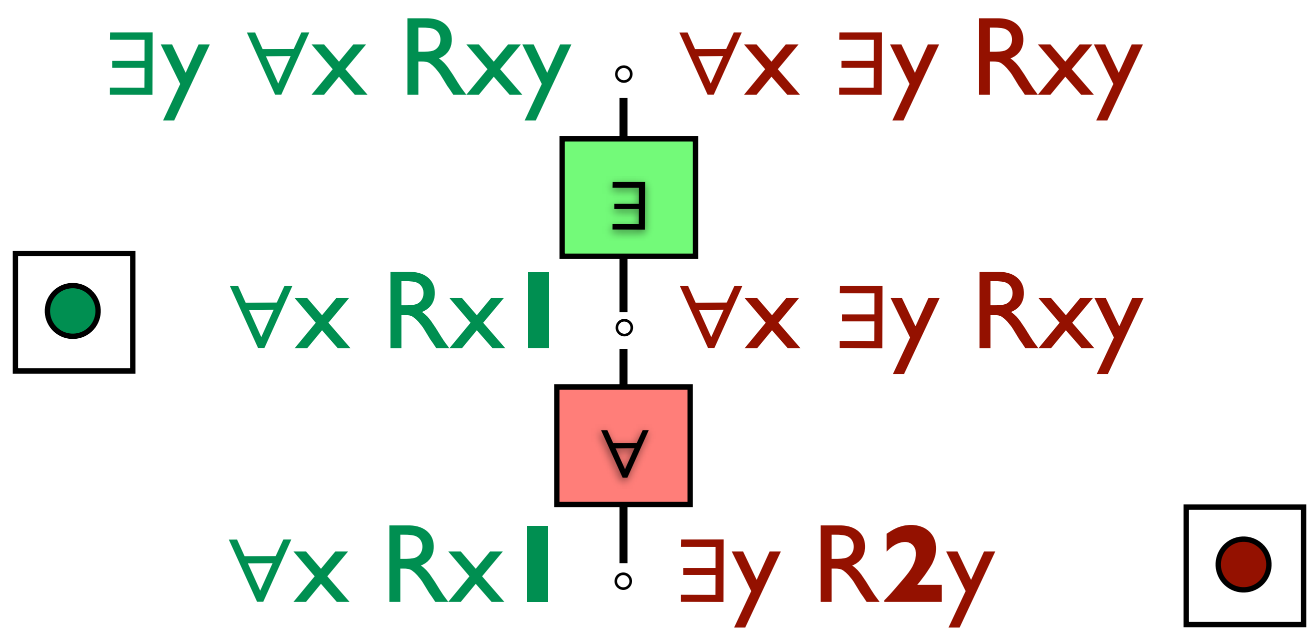


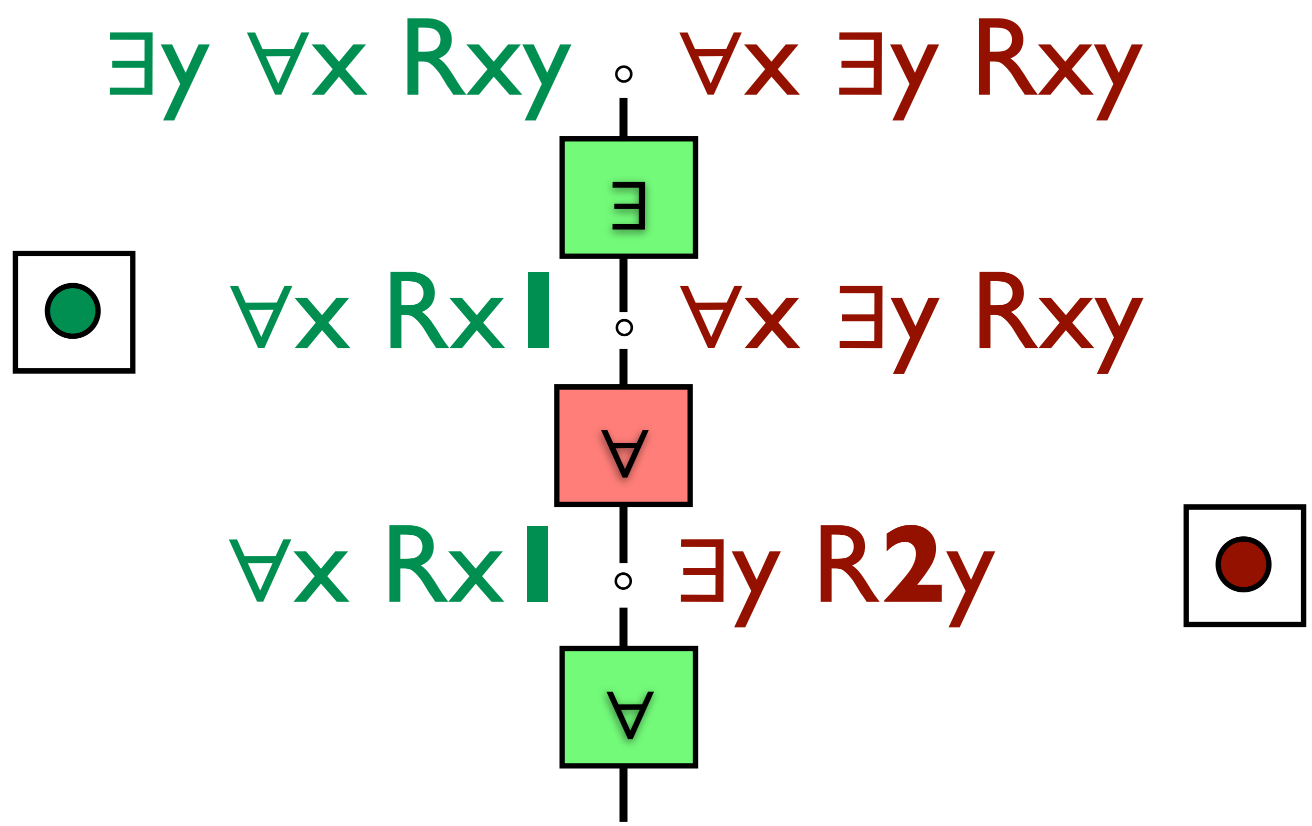
$\forall x\ Rx\blacksquare \quad \forall x\ \exists y\ Rxy$

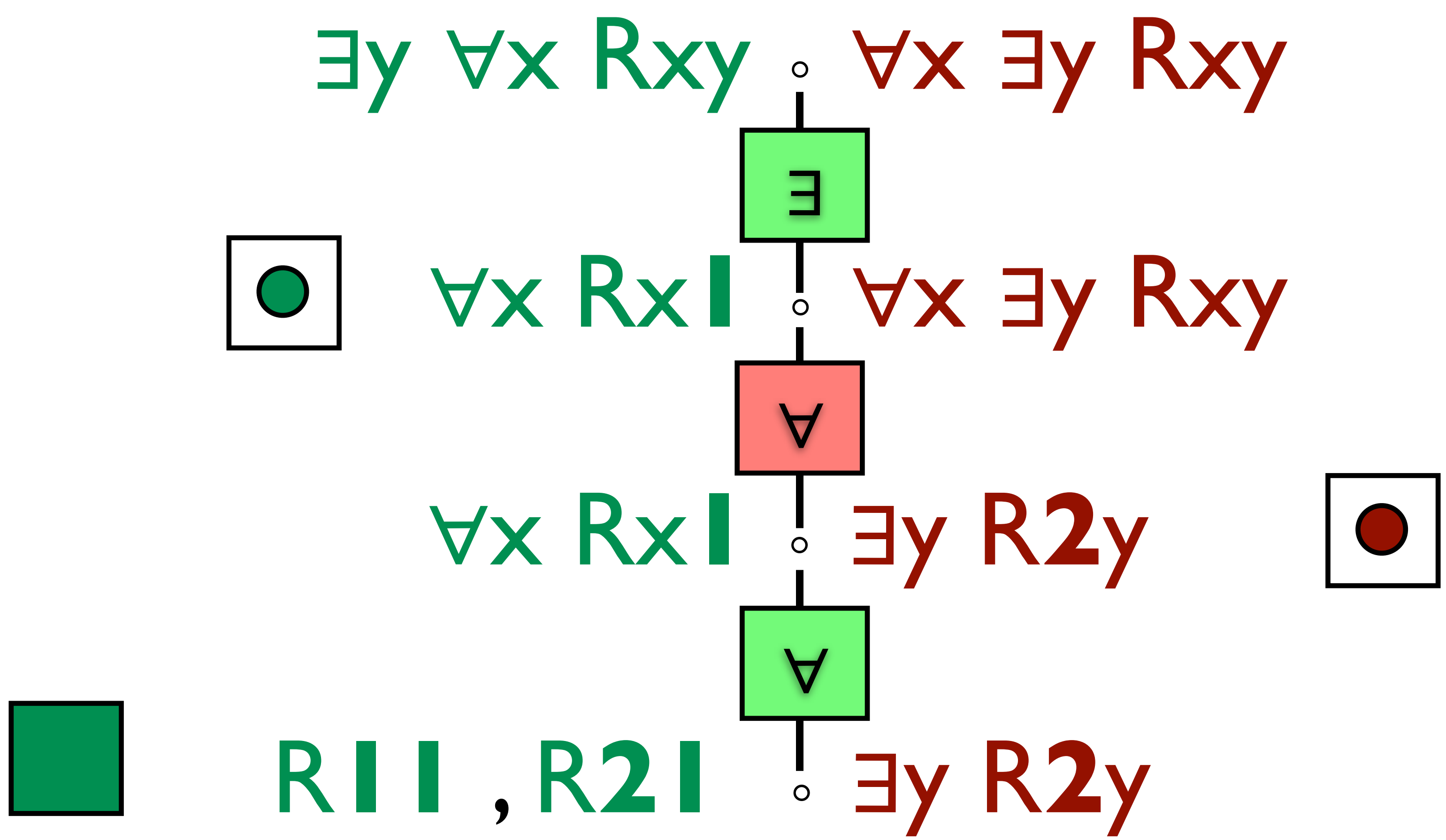
$\exists y\ \forall x\ Rxy \quad \forall x\ \exists y\ Rxy$

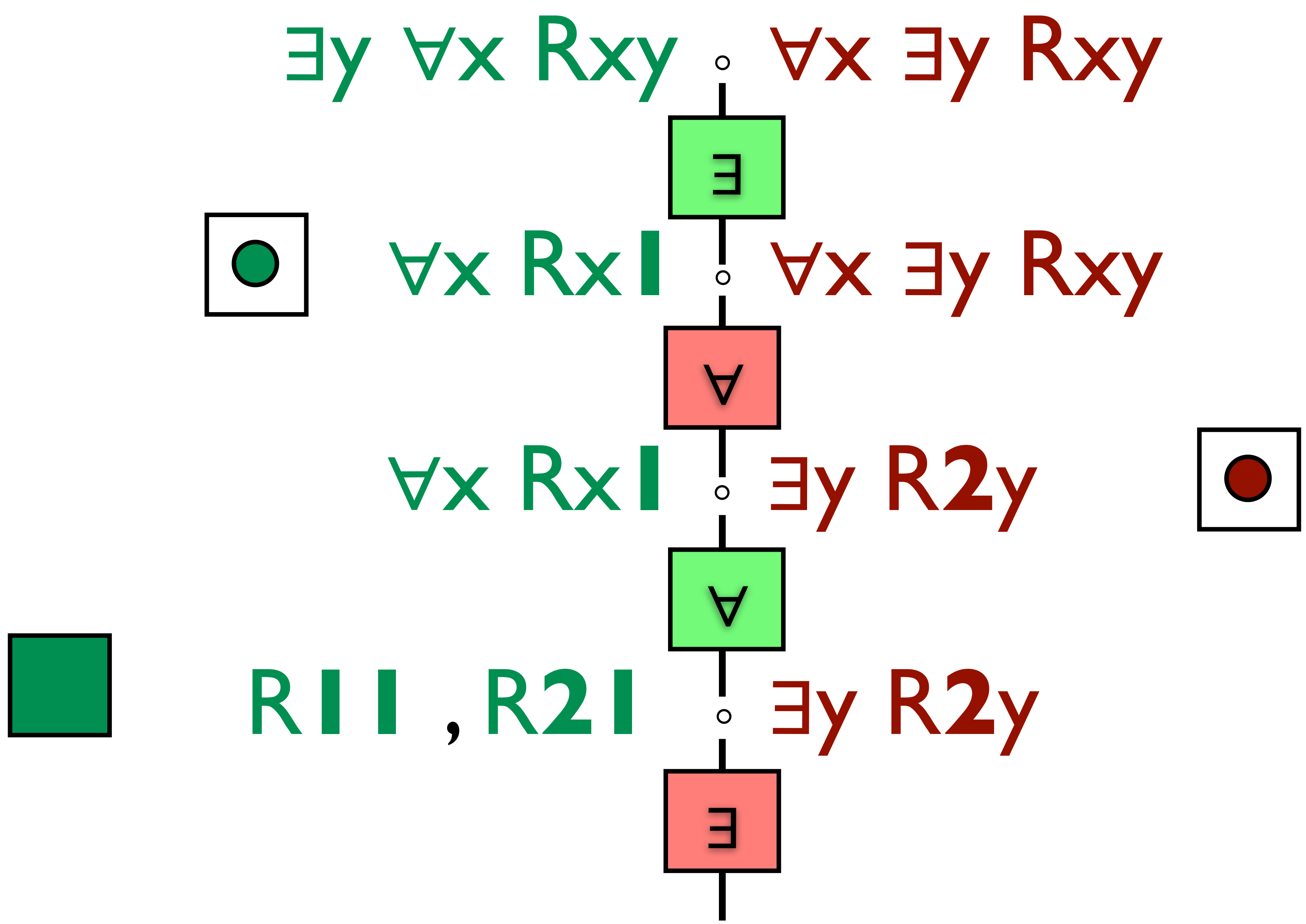
$\forall x\ Rx\blacksquare \quad \forall x\ \exists y\ Rxy$

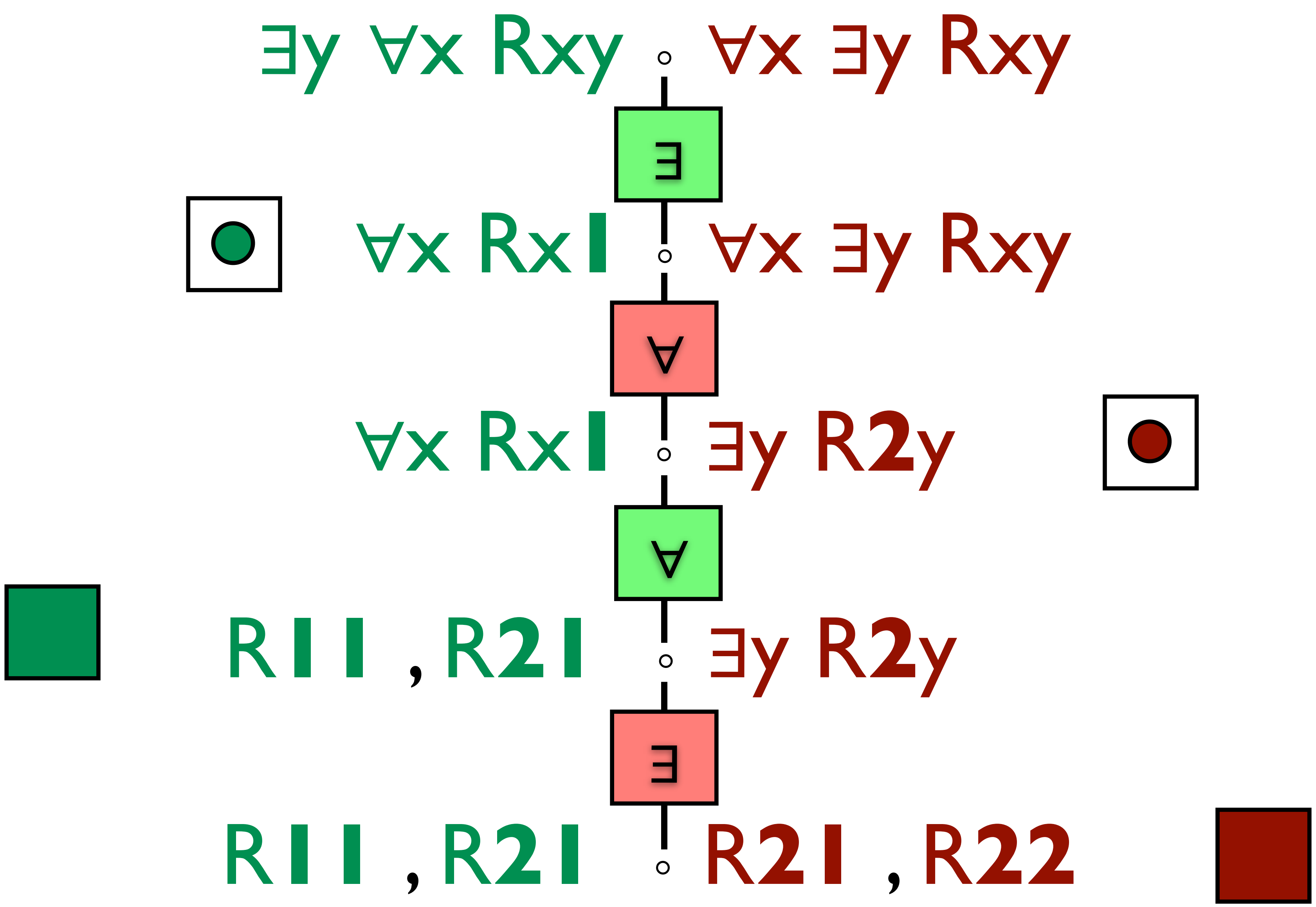


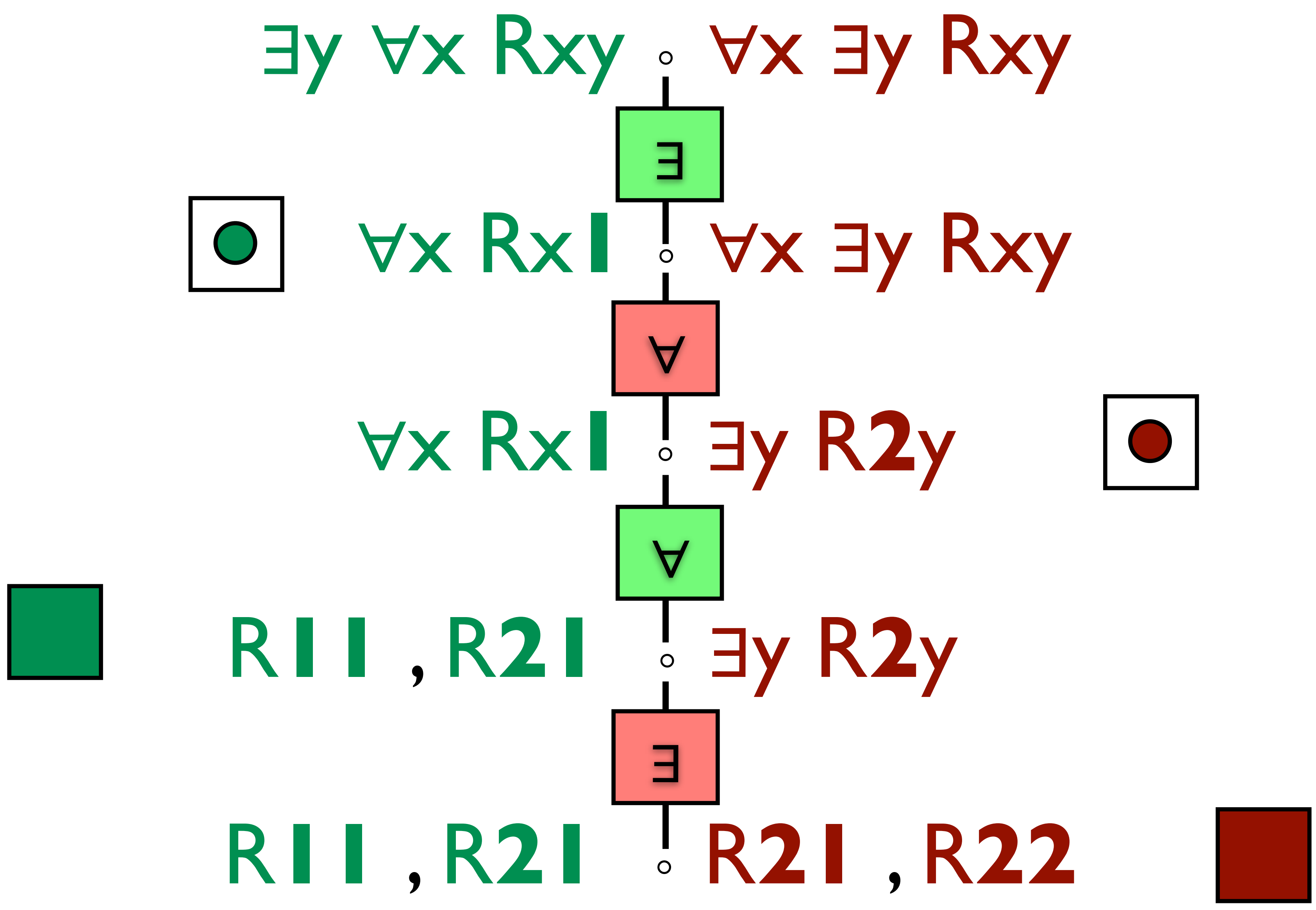

∃y ∀x Rxy
∀x ∃y Rxy
∃
∀x Rx1
∀x ∃y Rxy
∀
∀x Rx1
∃y R2y


∃y ∀x Rxy ∘ ∀x ∃y Rxy
∃
∀x Rx1 ∘ ∀x ∃y Rxy
∀
∀x Rx1 ∘ ∃y R2y
∀


∃y ∀x Rxy
∀x ∃y Rxy
∃
∀x Rx1
∀x ∃y Rxy
∀
∀x Rx1
∃y R2y
∀
R11 , R21
∃y R2y

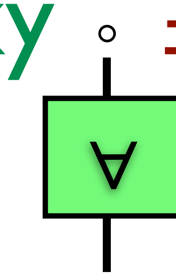

∃y ∀x Rxy
∀x ∃y Rxy
∃
∀x Rx1
∀x ∃y Rxy
∀
∀x Rx1
∃y R2y
∀
R11 , R21
∃y R2y
∃


∃y ∀x Rxy
∀x ∃y Rxy
∃
∀x Rx1
∀x ∃y Rxy
∀
∀x Rx1
∃y R2y
∀
R11 , R21
∃y R2y
∃
R11 , R21
R21 , R22


∃y ∀x Rxy ∘ ∀x ∃y Rxy
∃
∀x Rx1 ∘ ∀x ∃y Rxy
∀
∀x Rx1 ∘ ∃y R2y
∀
R11 , R21 ∘ ∃y R2y
∃
R11 , R21 ∘ R21 , R22

$\forall x \, \exists y \, Rxy \circ \exists y \, \forall x \, Rxy$

$\forall x\ \exists y\ Rxy$ $\exists y\ \forall x\ Rxy$



$\forall x\ \exists y\ Rxy \circ \exists y\ \forall x\ Rxy$

$\forall x\ \exists y\ Rxy$ $\exists y\ \forall x\ Rxy$

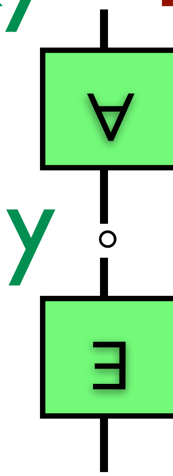
$\forall x\ \exists y\ Rxy$ $\exists y\ \forall x\ Rxy$

$\forall$

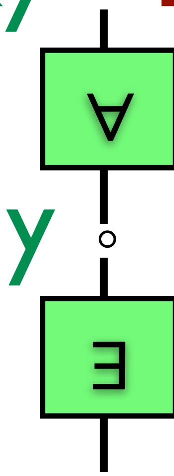
1 $\exists y\ R1y$

$\forall x\ \exists y\ Rxy \circ \exists y\ \forall x\ Rxy$

1 $\exists y\ R1y$

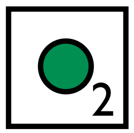


$\forall x\ \exists y\ Rxy \circ \exists y\ \forall x\ Rxy$



I

$\exists y\ R\,\mathbf{I}\,y \circ$



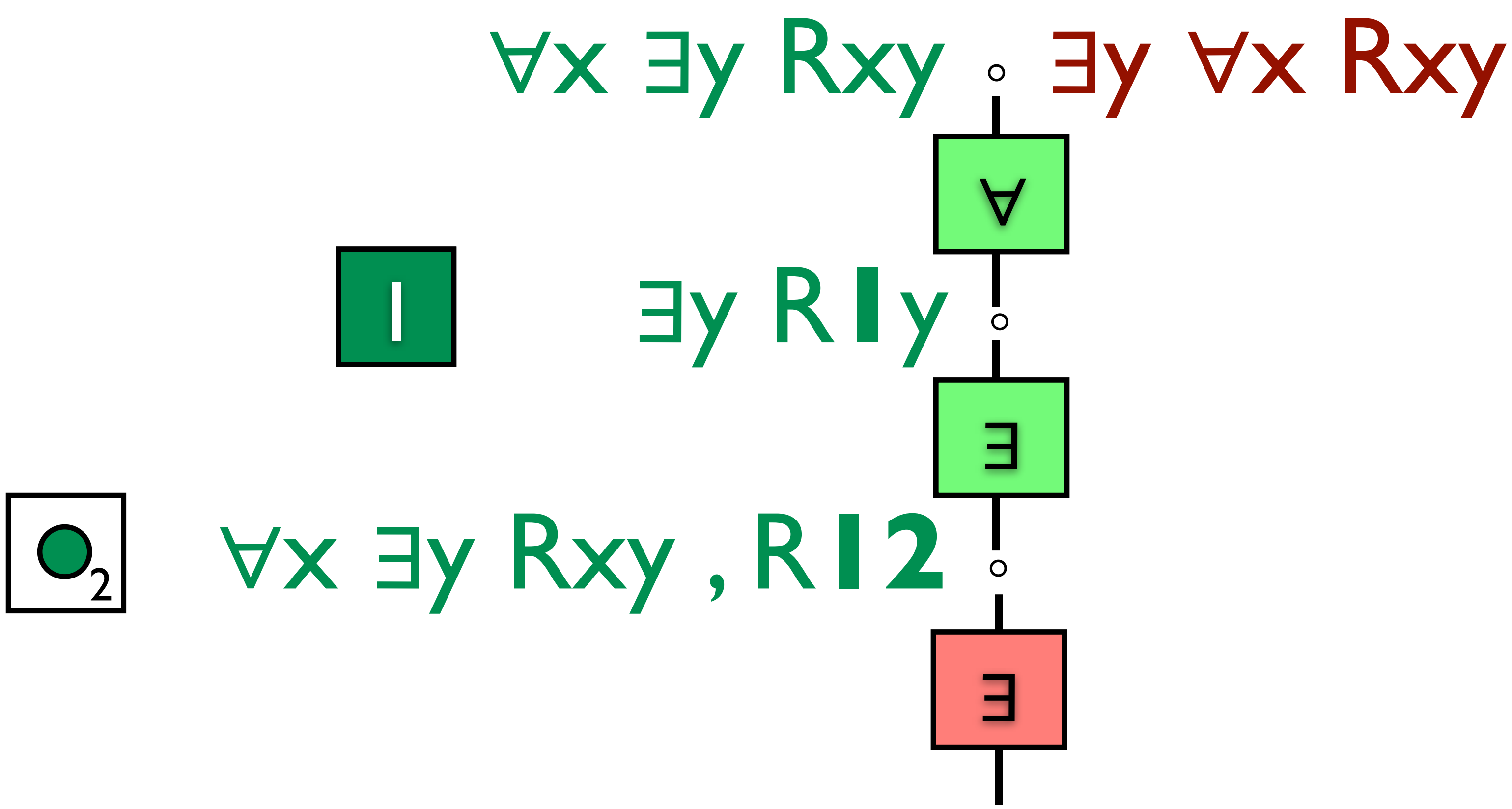
$\forall x\ \exists y\ Rxy\ \circ\ \exists y\ \forall x\ Rxy$

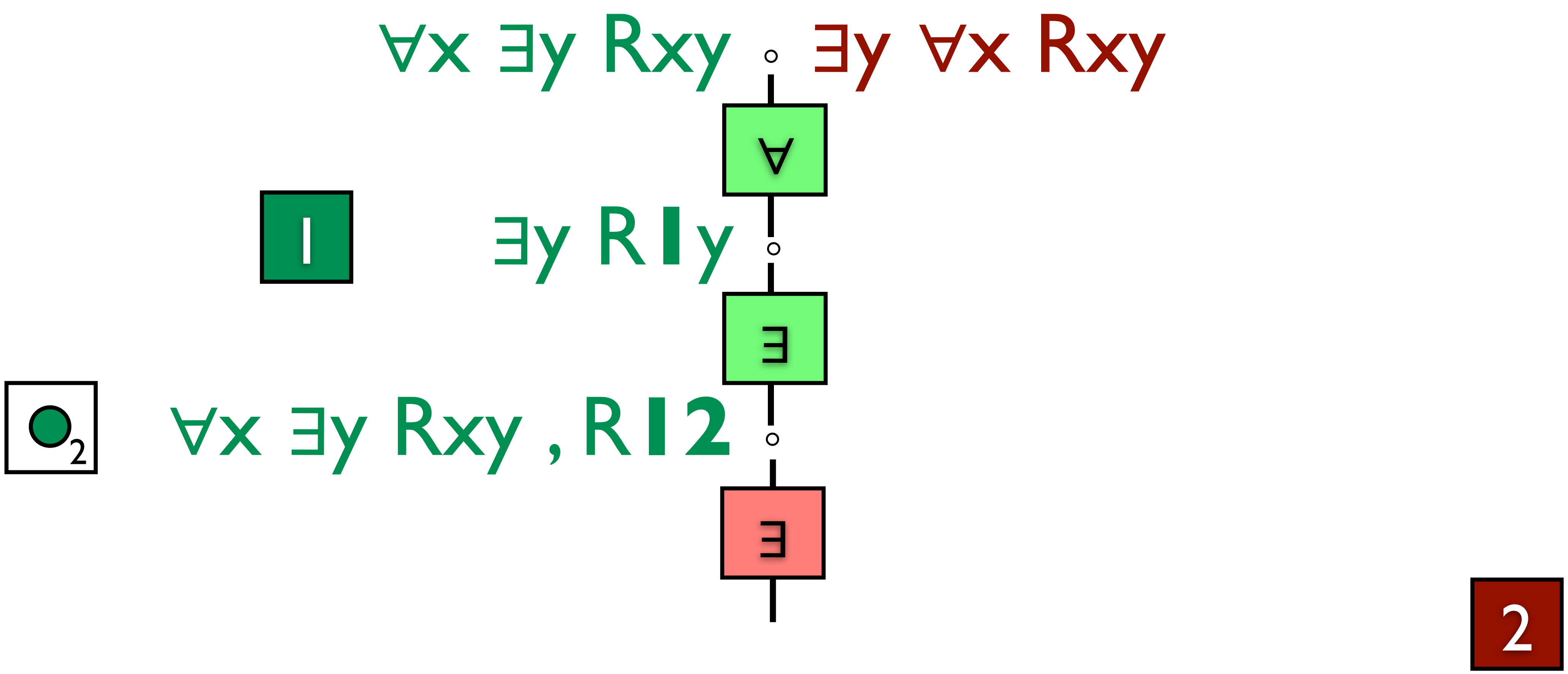
$\forall$

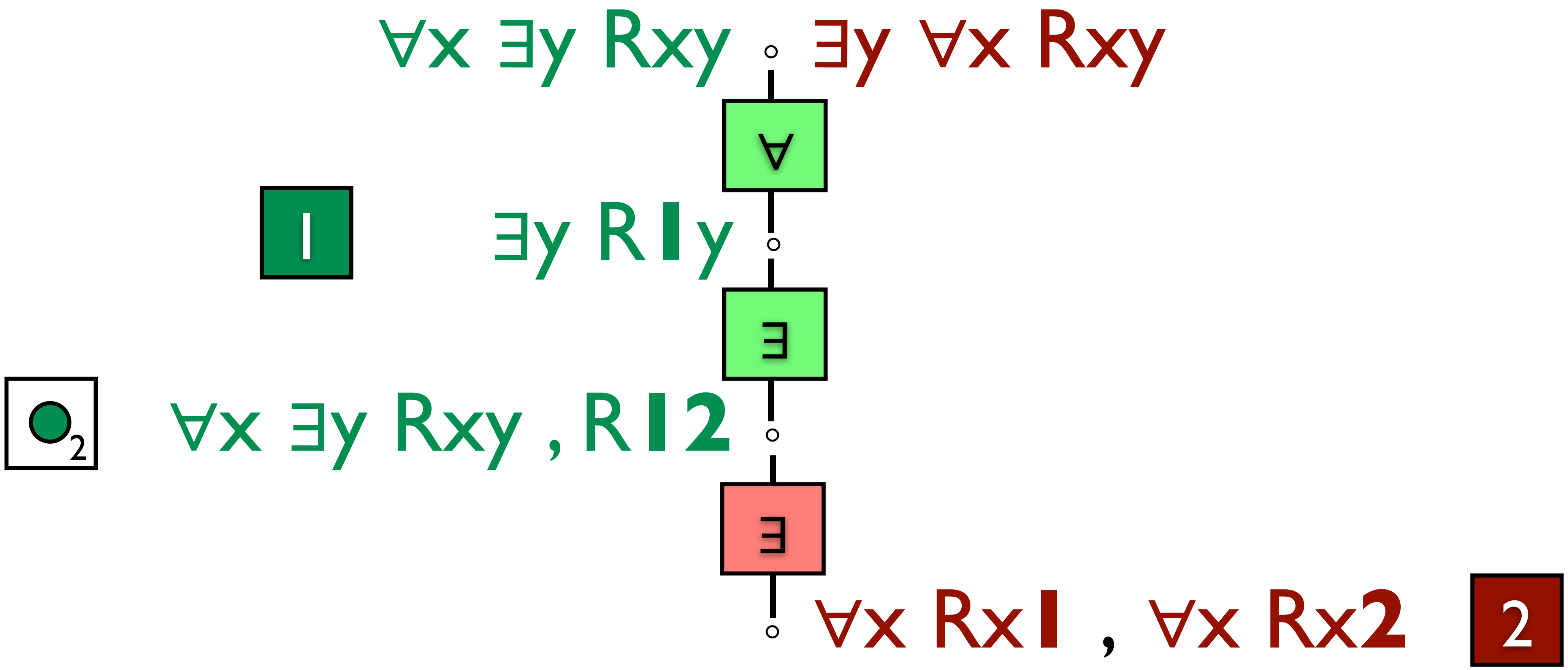
1 $\quad \exists y\ R\mathbf{1}y\ \circ$

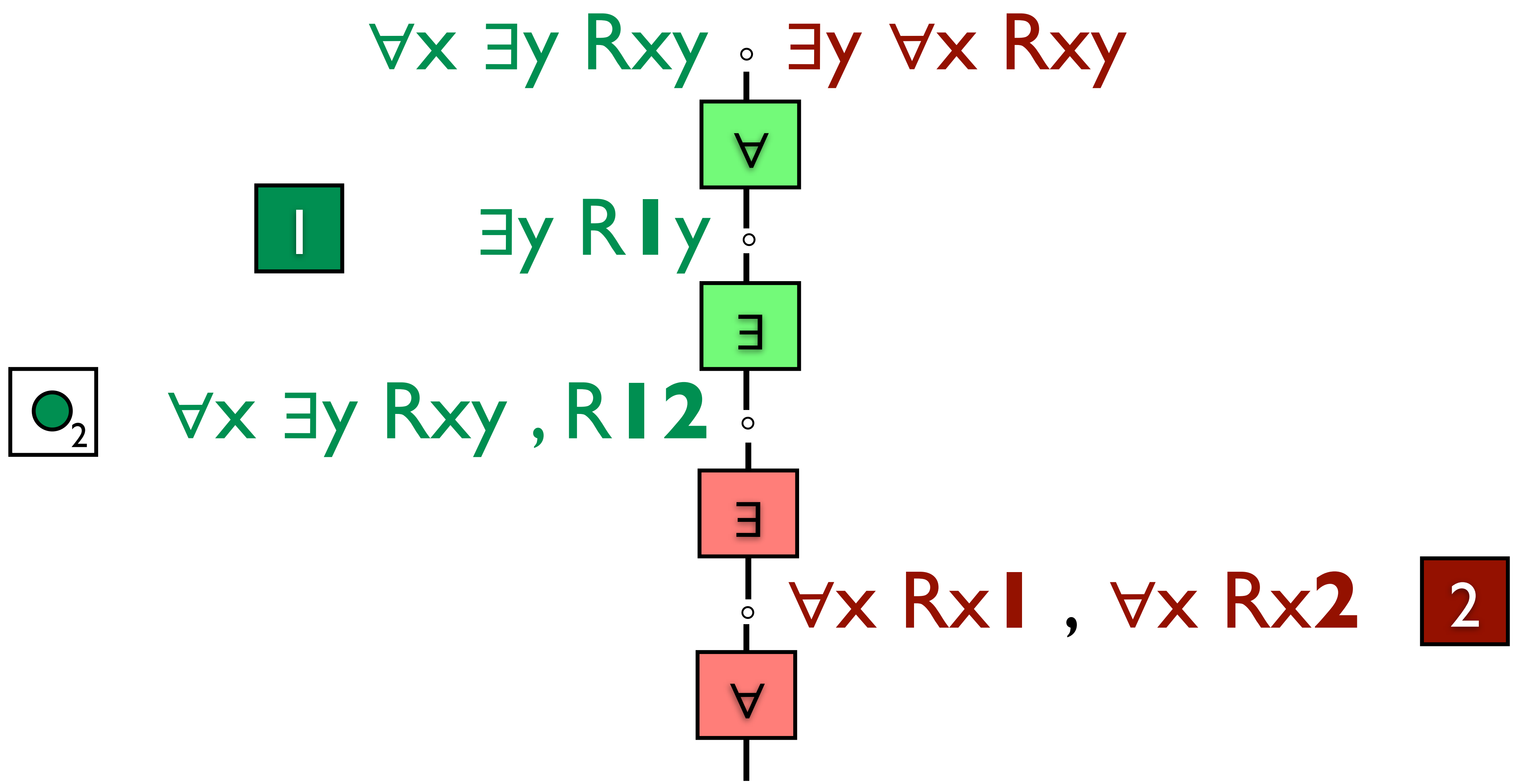
$\exists$

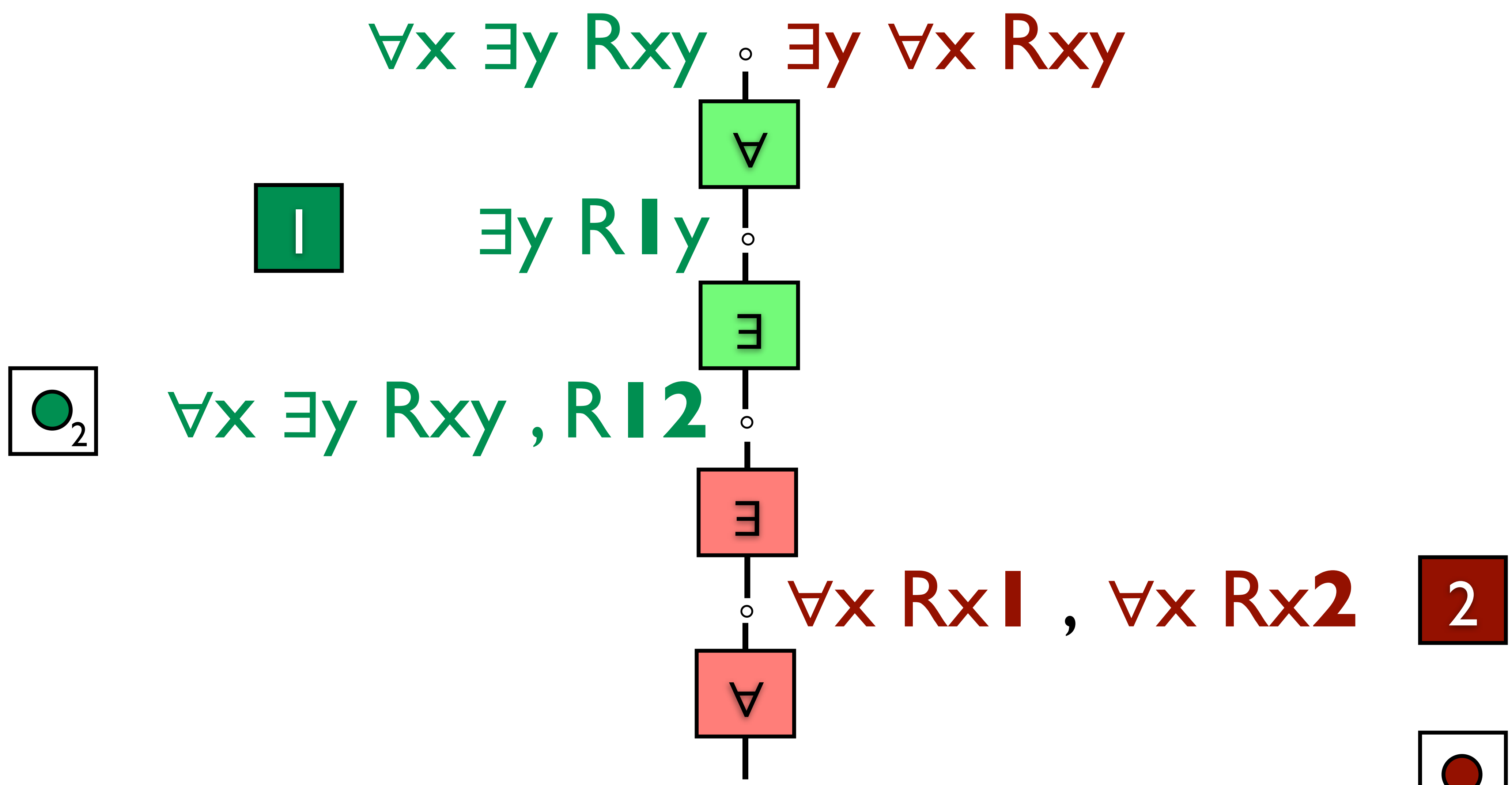
$\bullet_2 \quad \forall x\ \exists y\ Rxy\ ,\ R\mathbf{1}\mathbf{2}\ \circ$

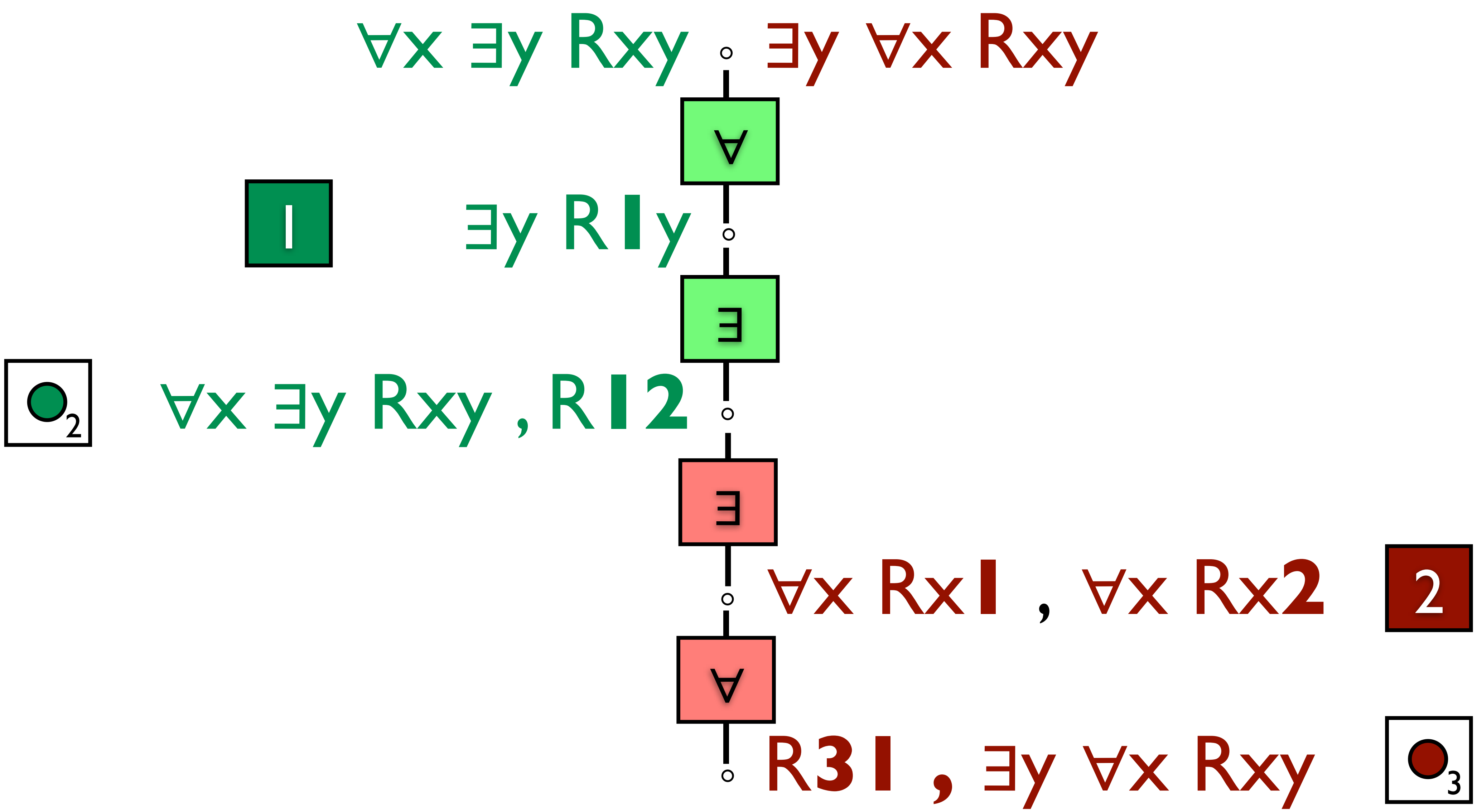

∀x ∃y Rxy ∘ ∃y ∀x Rxy
∀
1
∃y R1y
∃
2
∀x ∃y Rxy , R12
∃

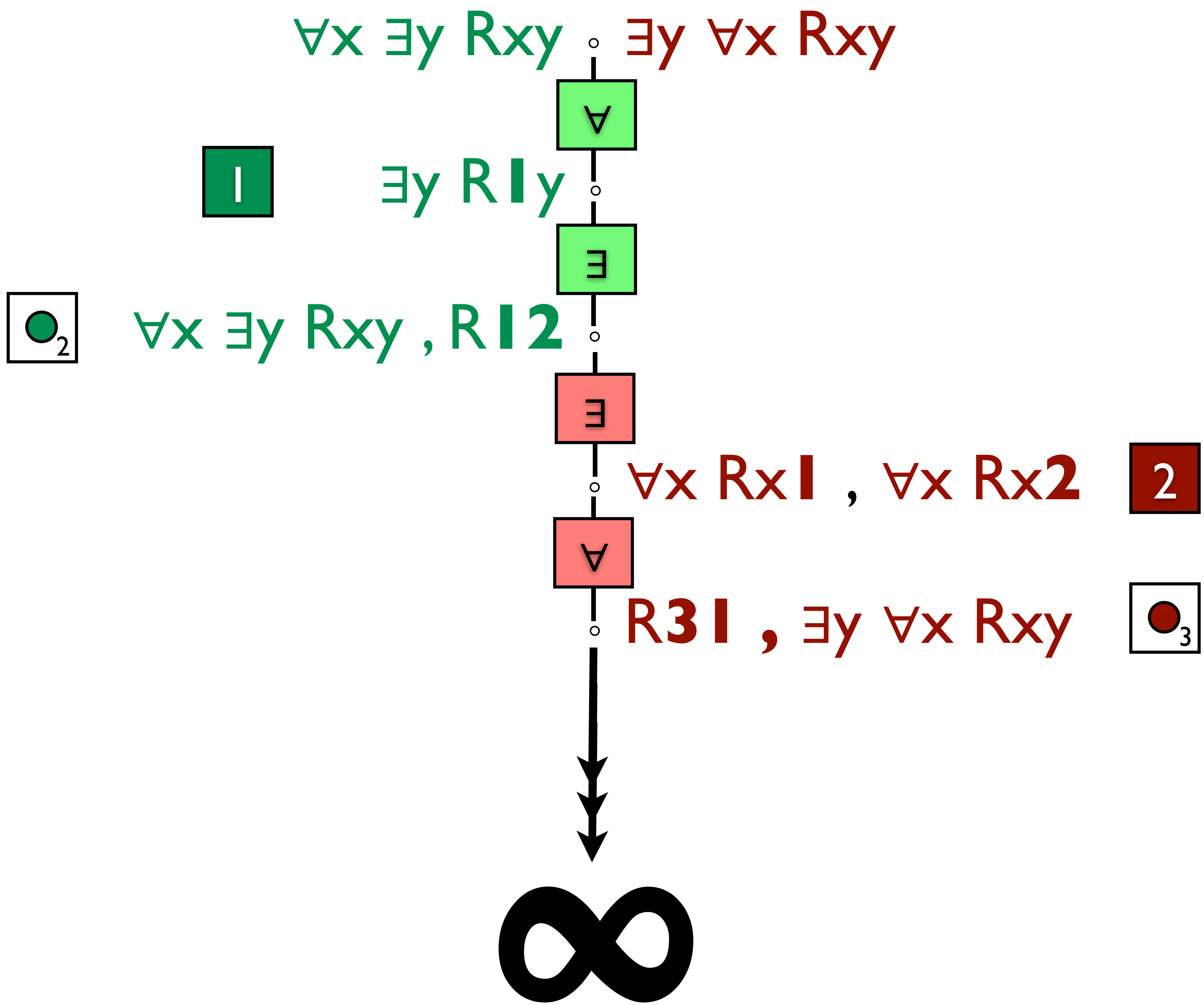

∀x ∃y Rxy
∃y ∀x Rxy
∀
1
∃y R1y
∃
∀x ∃y Rxy , R12
∃
2


∀x ∃y Rxy : ∃y ∀x Rxy
∀
1
∃y R1y
∃
2
∀x ∃y Rxy , R12
∃
∀x Rx1 , ∀x Rx2
2

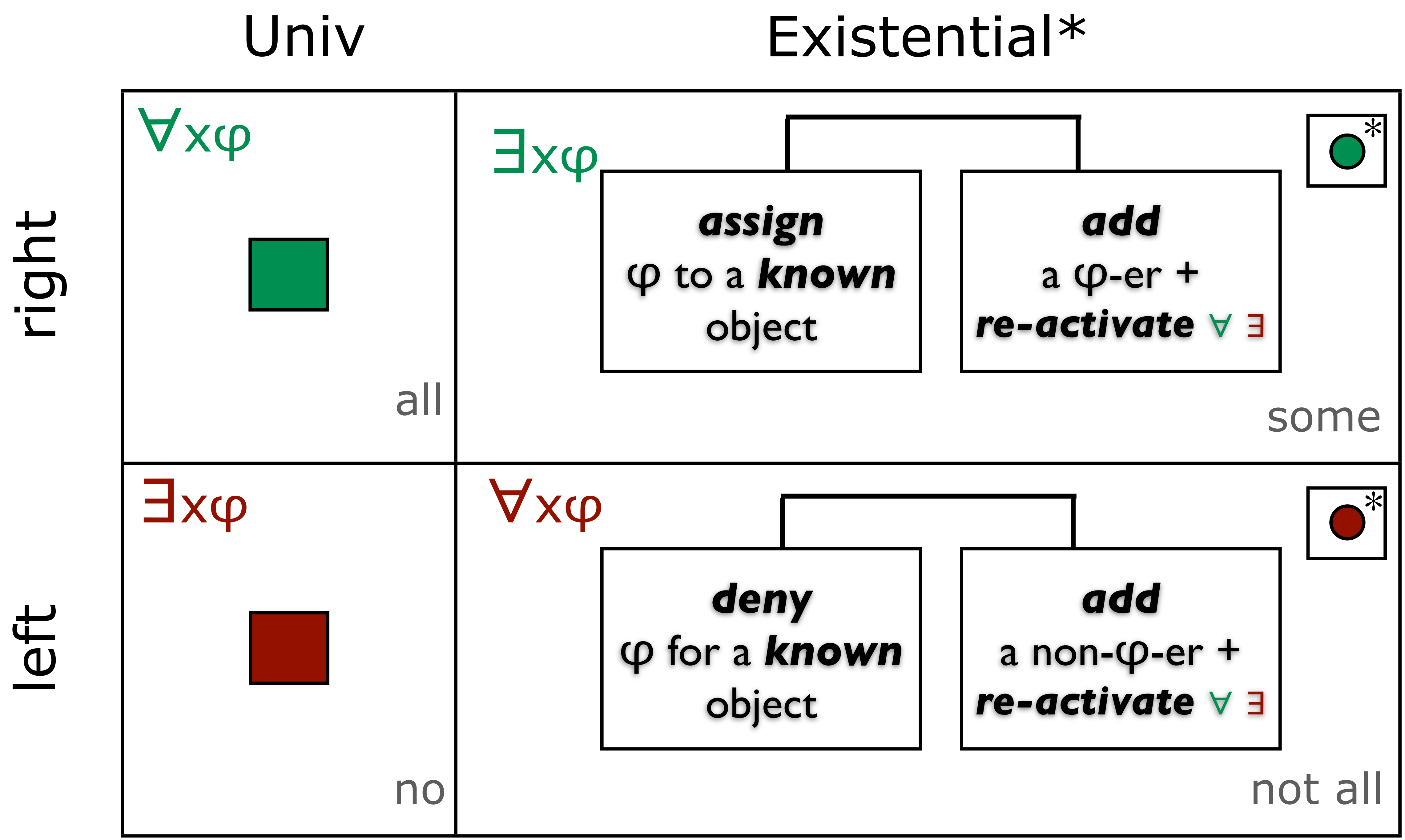

∀x ∃y Rxy
∃y ∀x Rxy
∀
1
∃y R1y
∃
2
∀x ∃y Rxy , R12
∃
∀x Rx1 , ∀x Rx2
2
∀


∀x ∃y Rxy
∃y ∀x Rxy
∀
1
∃y R1y
∃
2
∀x ∃y Rxy , R12
∃
∀x Rx1 , ∀x Rx2
2
∀
3

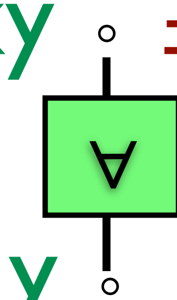

∀x ∃y Rxy
∃y ∀x Rxy
∀
1
∃y R1y
∃
2
∀x ∃y Rxy , R12
∃
∀x Rx1 , ∀x Rx2
2
∀
R31 , ∃y ∀x Rxy
3


∀x ∃y Rxy
∃y ∀x Rxy
∀
1
∃y R1y
∃
2
∀x ∃y Rxy , R12
∃
∀x Rx1 , ∀x Rx2
2
∀
R31 , ∃y ∀x Rxy
3
∞

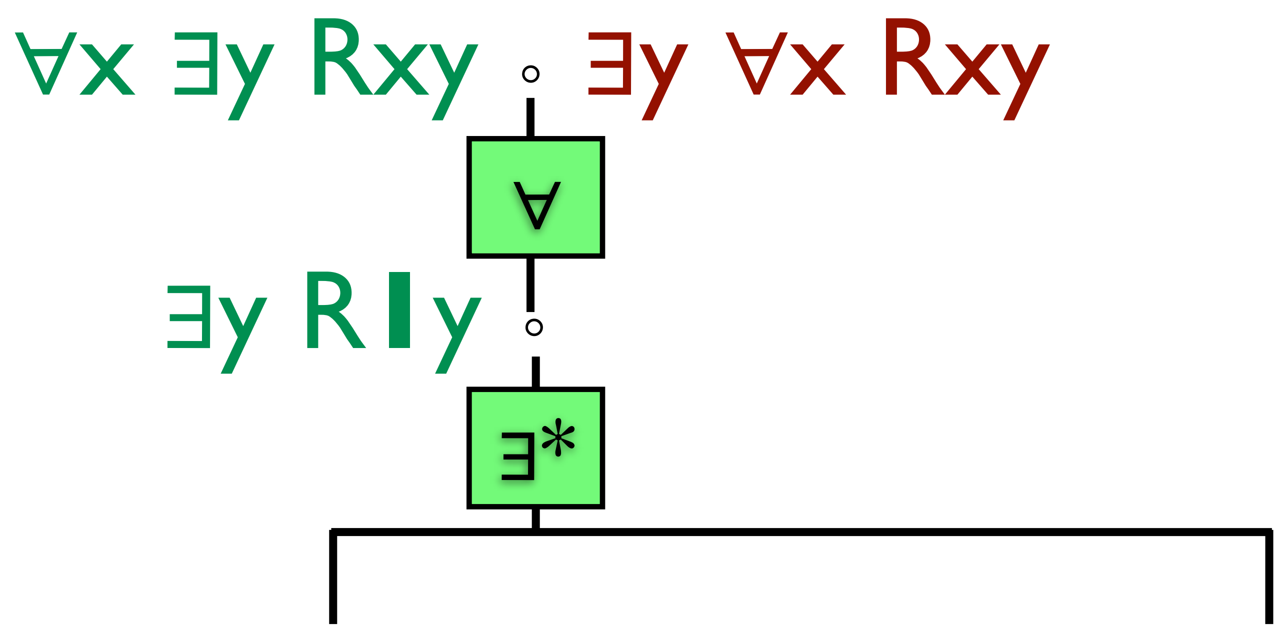
Extended Rules for existentials

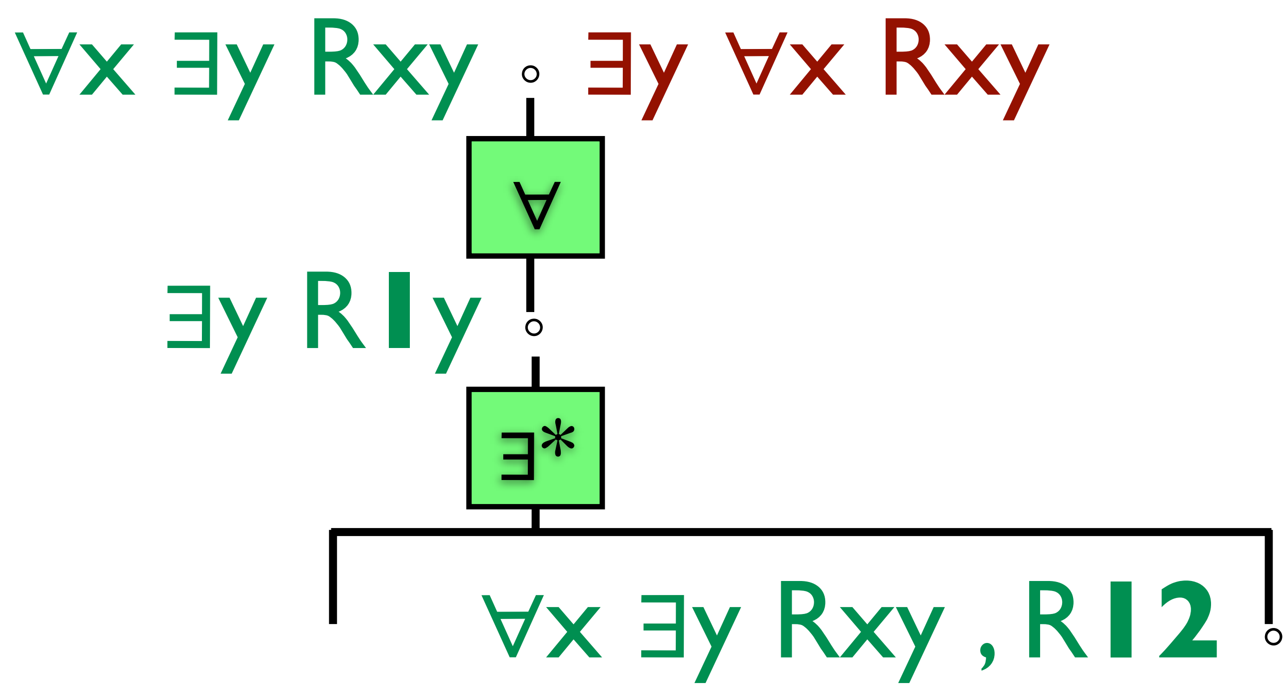


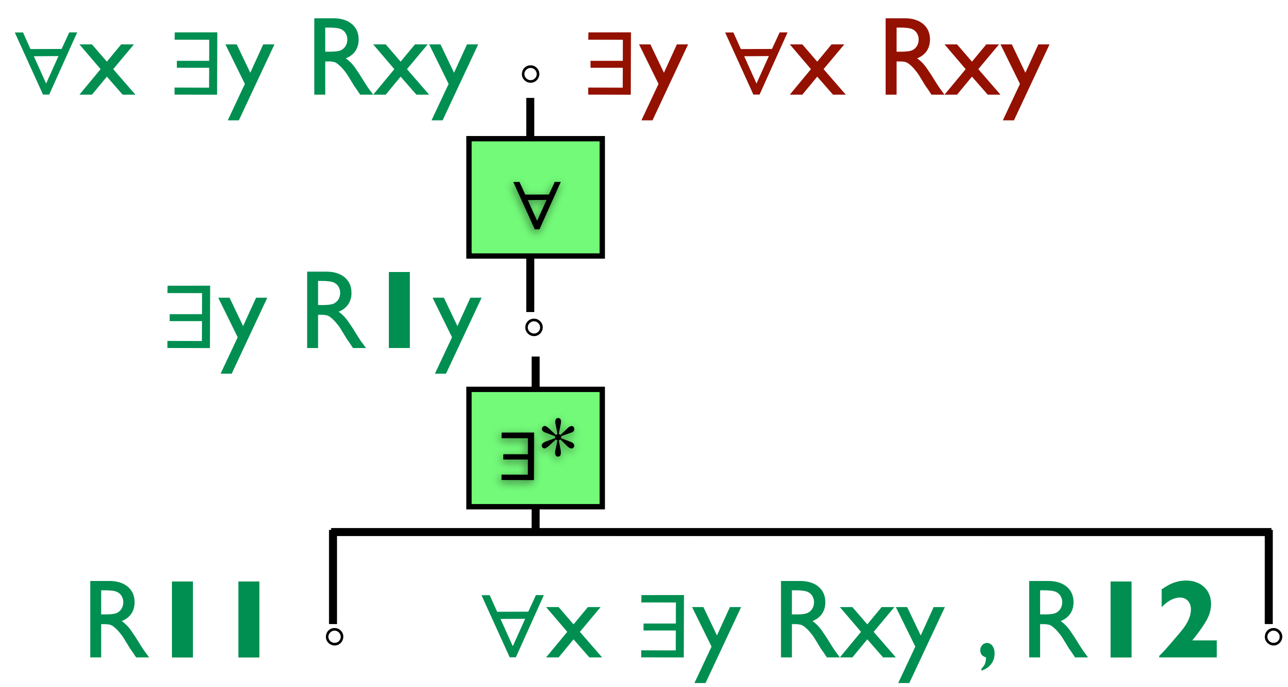
$\forall x\, \exists y\, Rxy$ $\exists y\, \forall x\, Rxy$

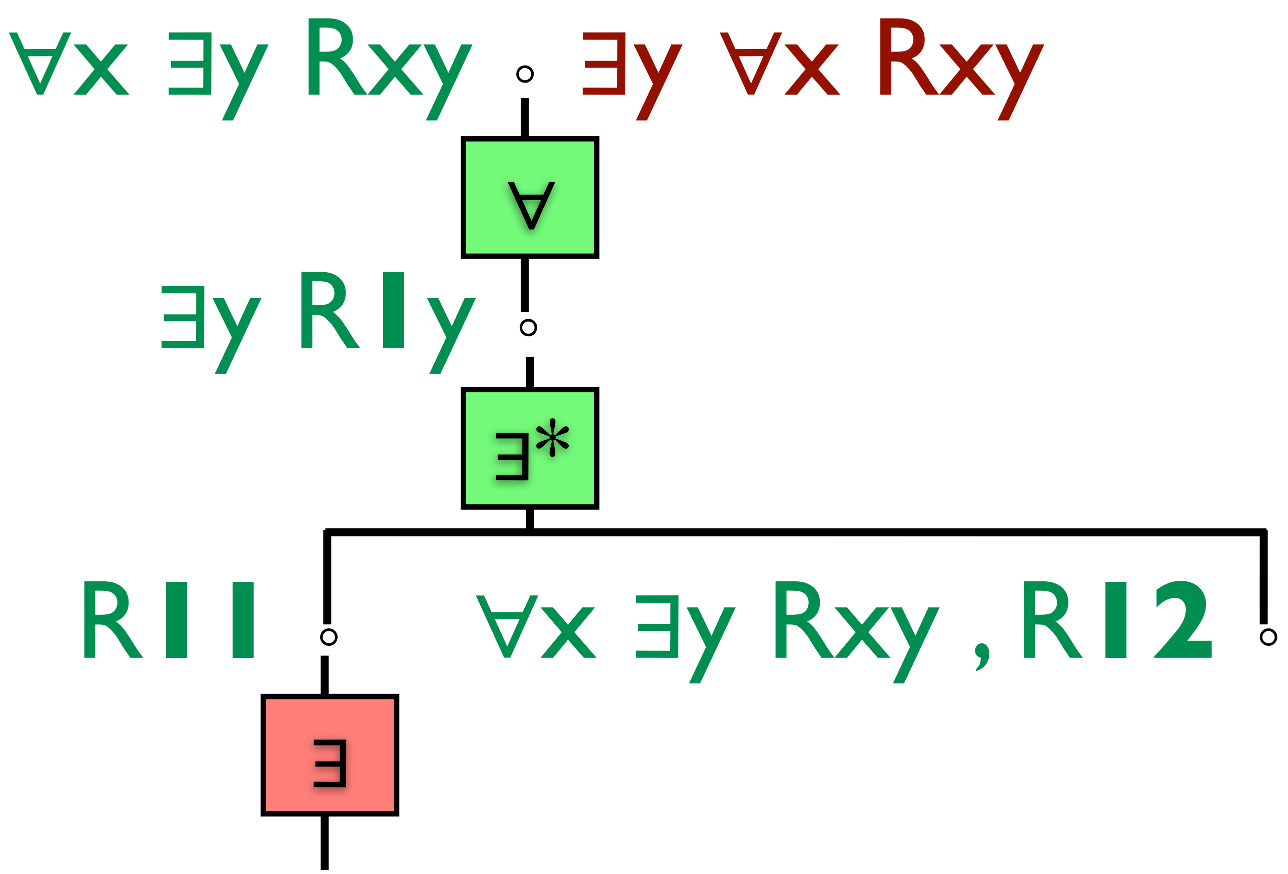


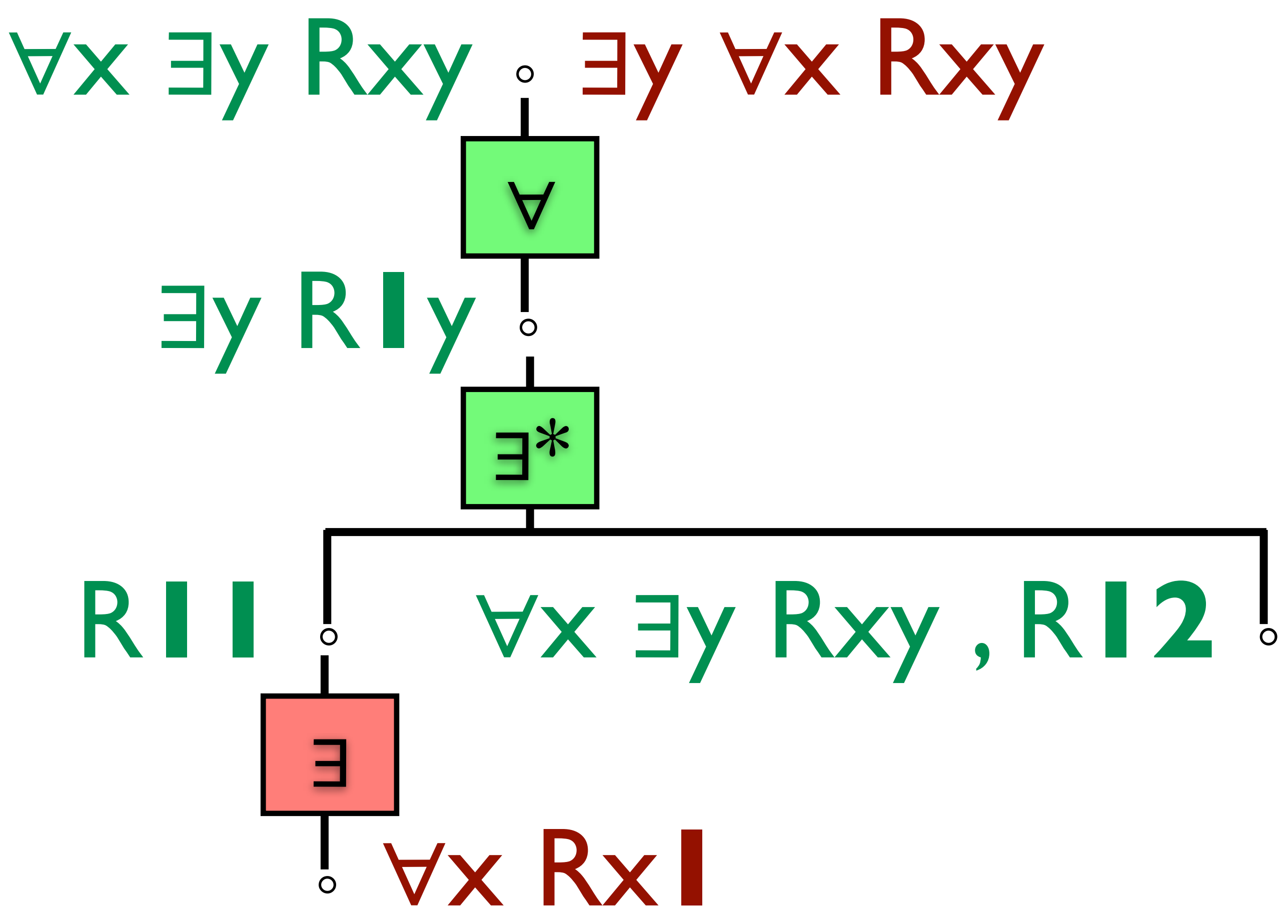
$\exists y\, R$ ▮ y

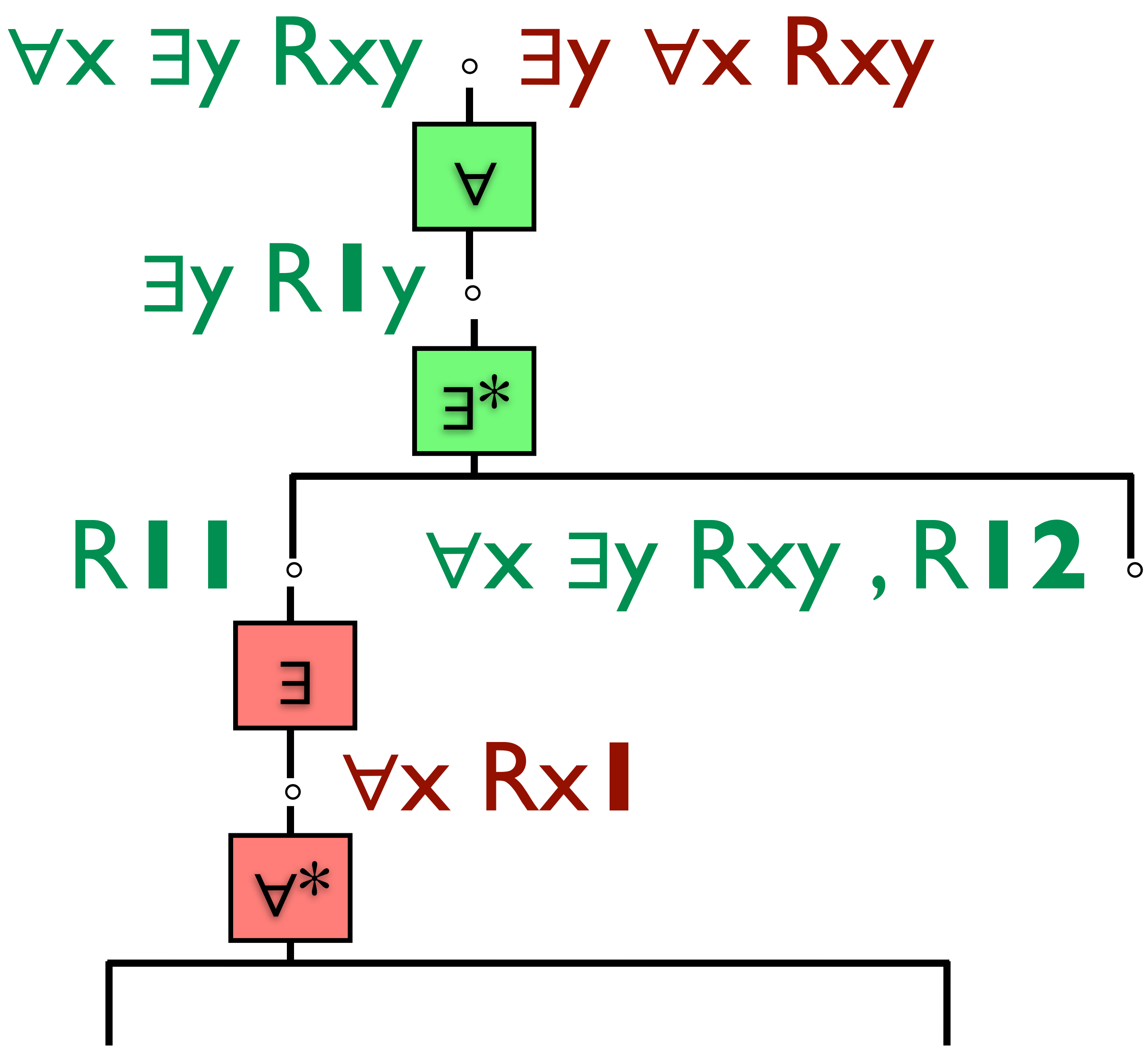

∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R▮y
∃*

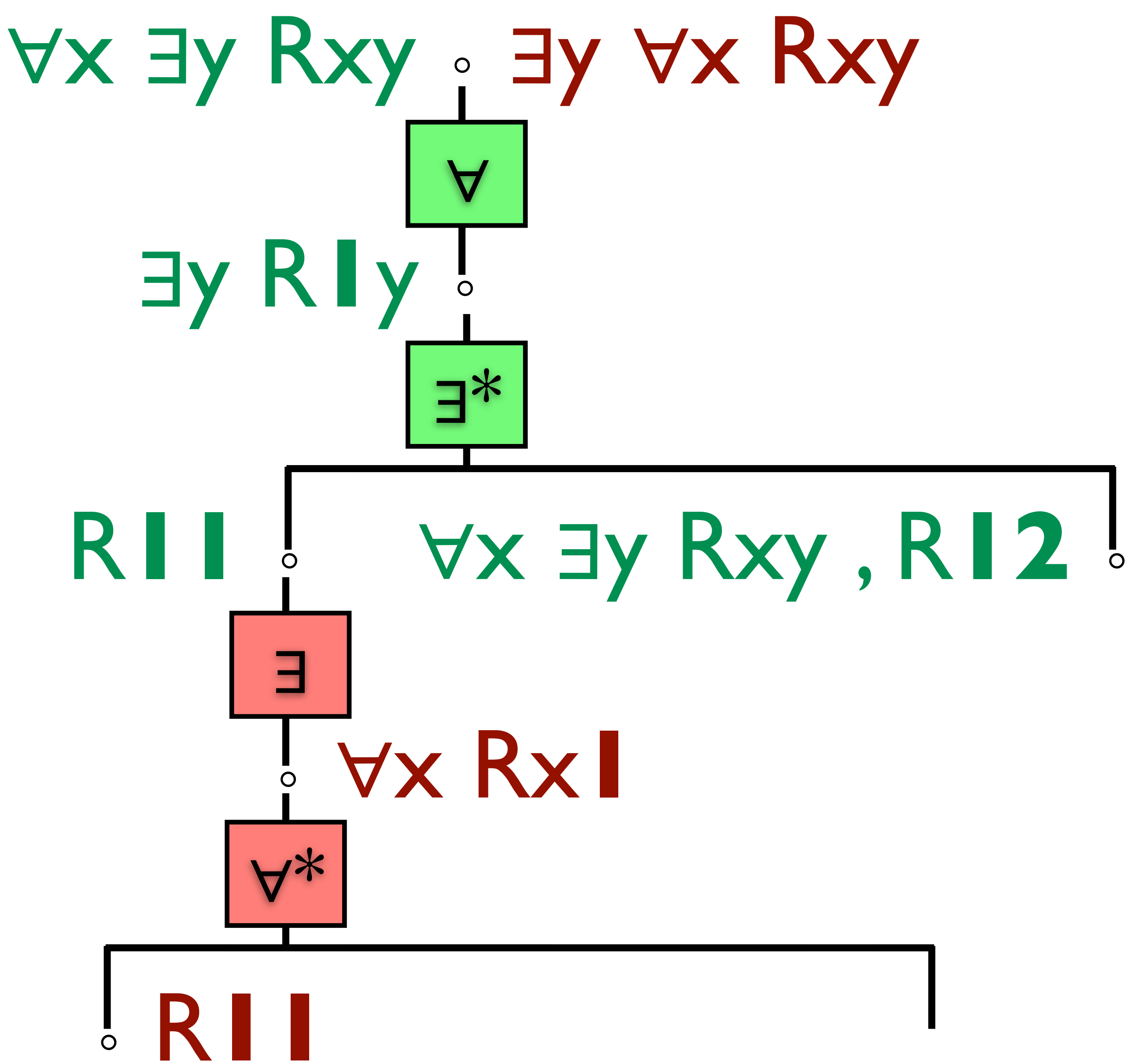

∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
∀x ∃y Rxy , R12

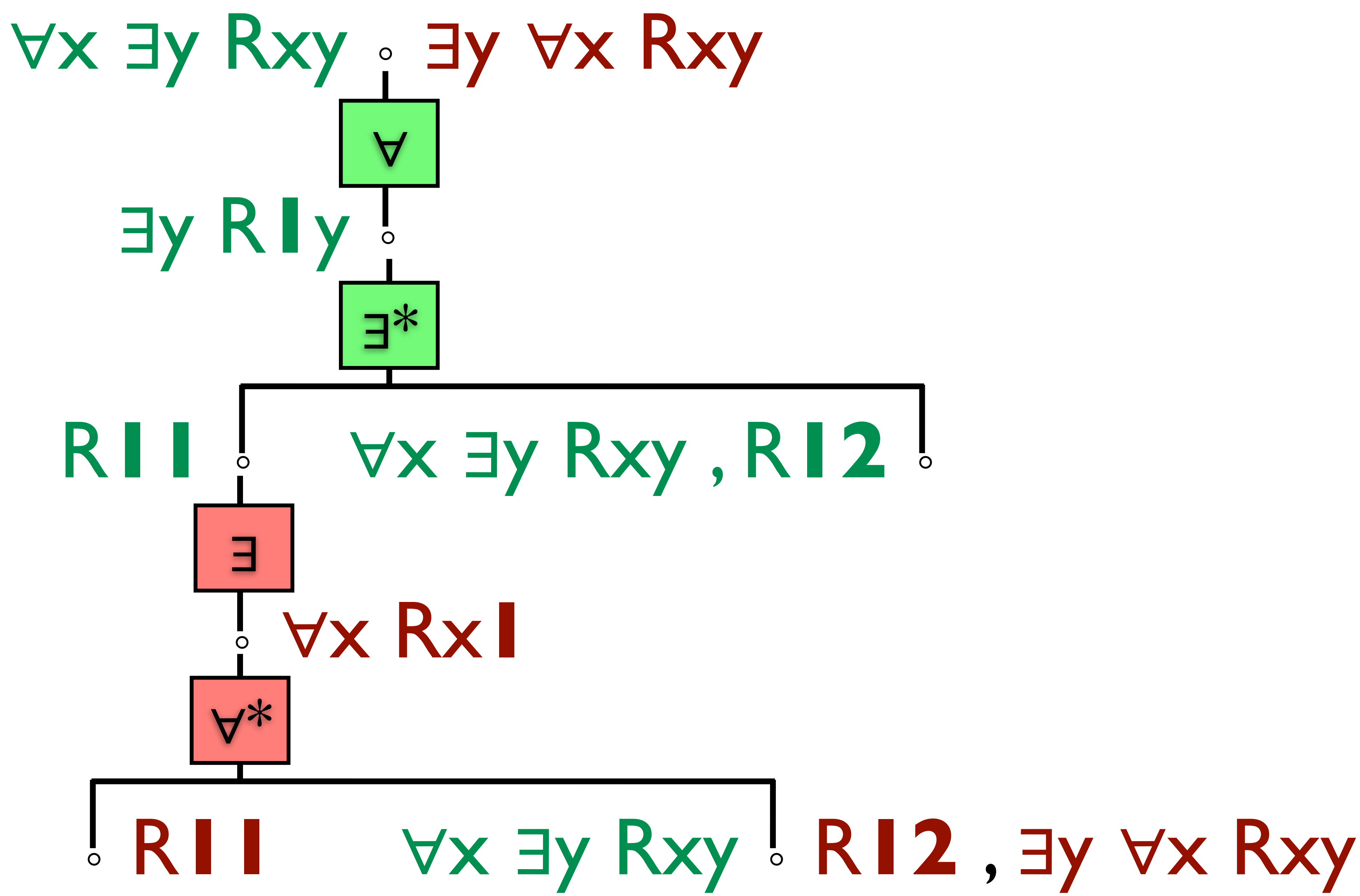

∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12

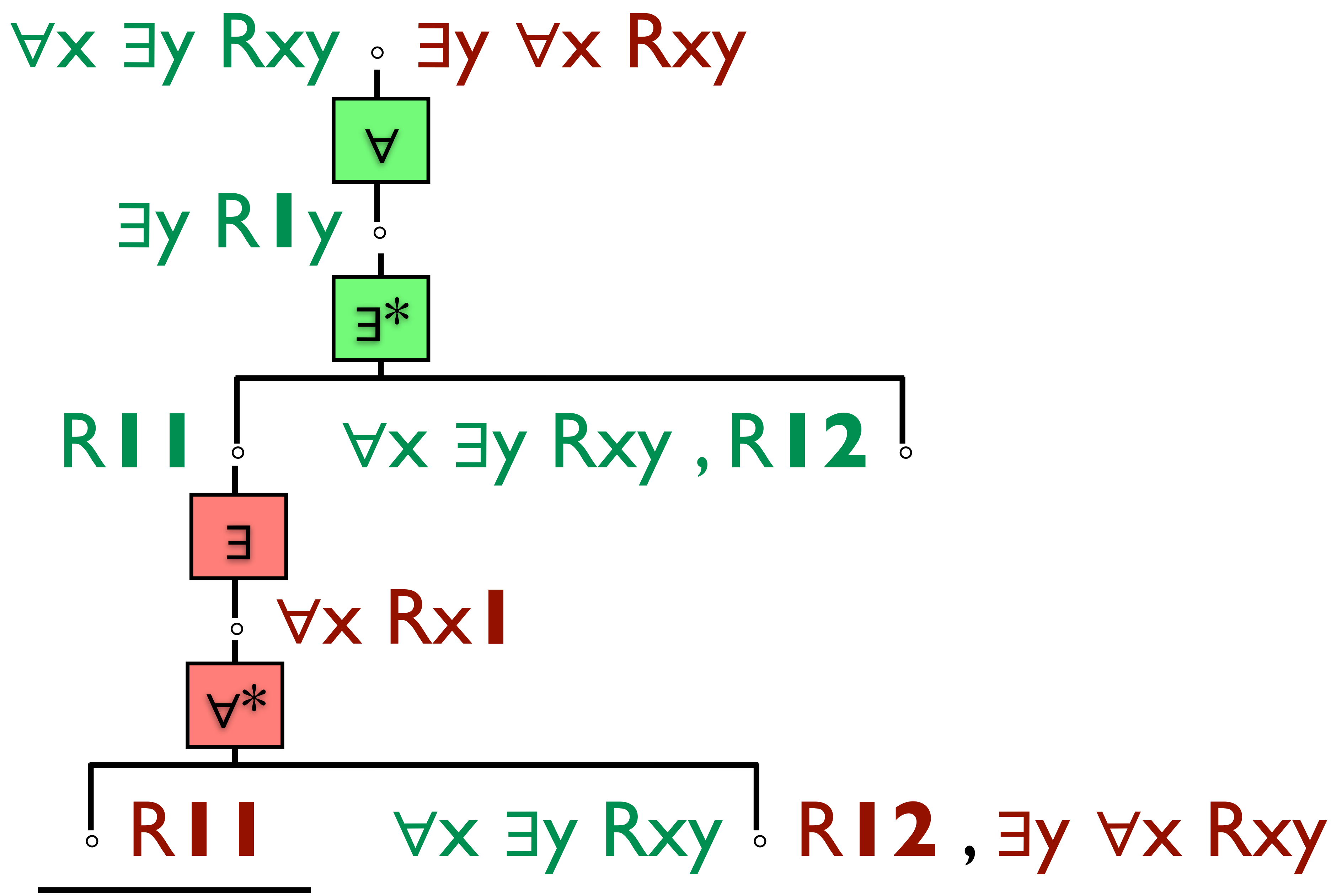

∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃

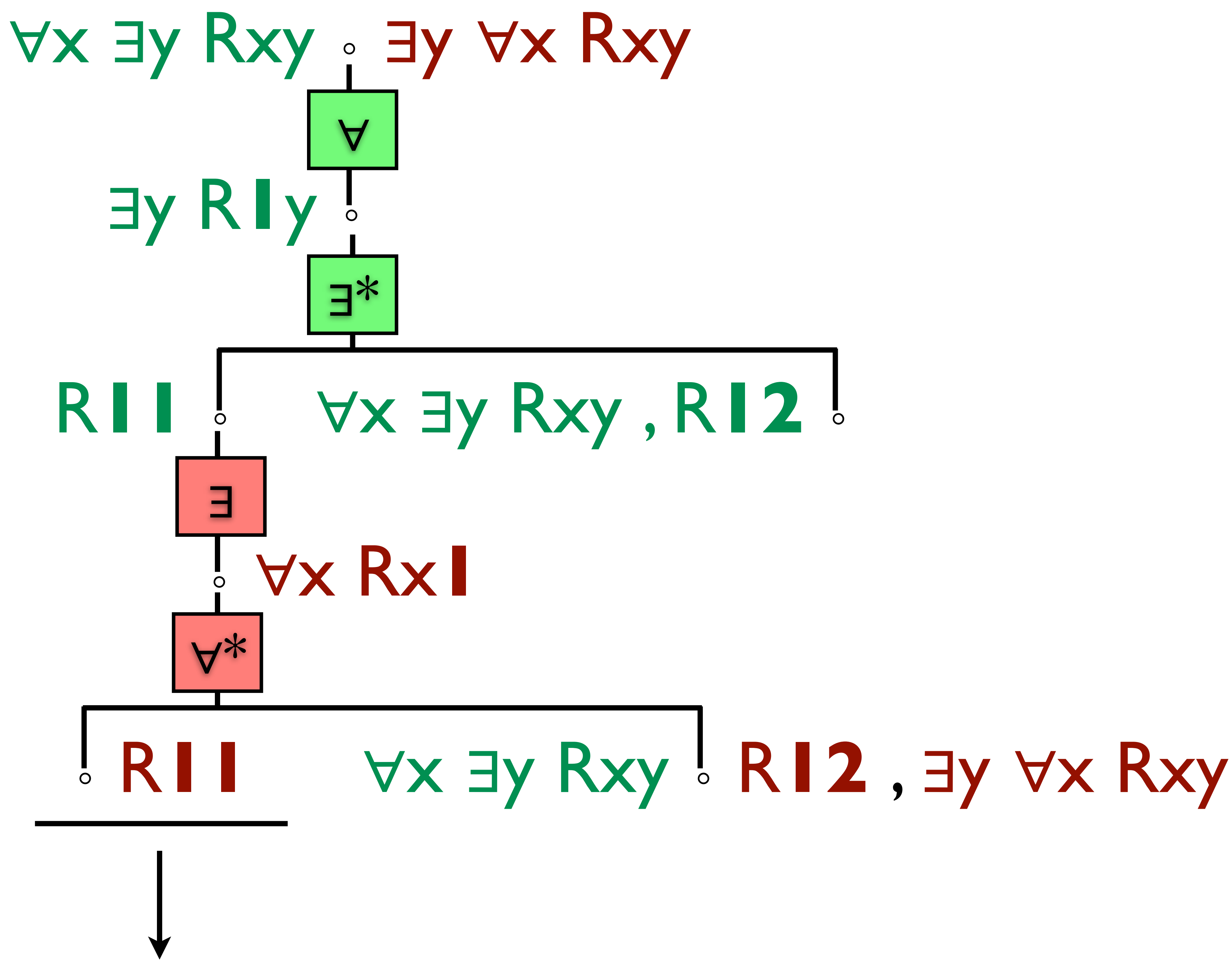

∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃
∀x Rx1

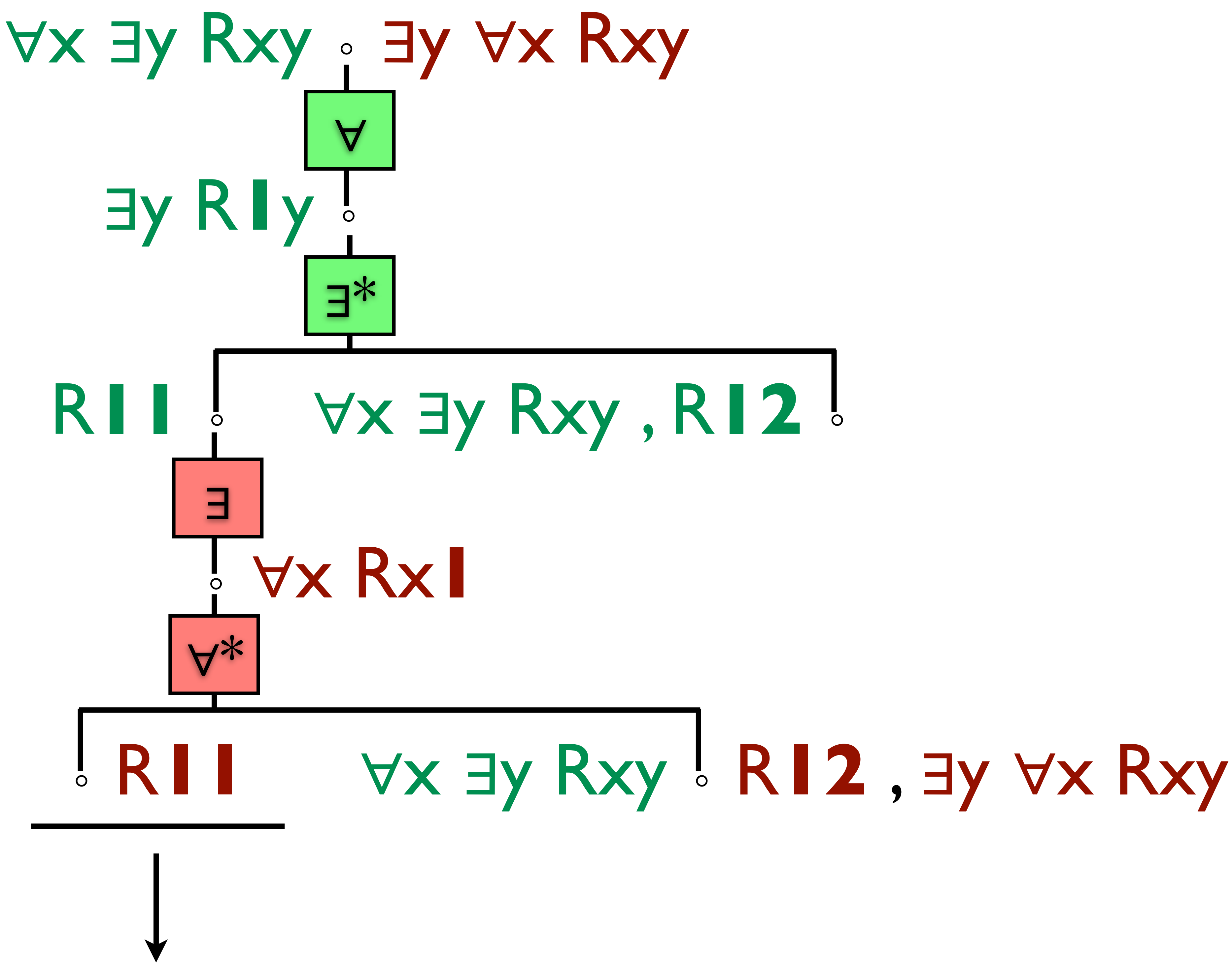

∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃
∀x Rx1
∀*


∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃
∀x Rx1
∀*
R11


∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃
∀x Rx1
∀*
R11
∀x ∃y Rxy
R12 , ∃y ∀x Rxy


∀x ∃y Rxy ∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃
∀x Rx1
∀*
R11
∀x ∃y Rxy
R12 , ∃y ∀x Rxy


∀x ∃y Rxy
∃y ∀x Rxy
∀
∃y R1y
∃*
R11
∀x ∃y Rxy , R12
∃
∀x Rx1
∀*
R11
∀x ∃y Rxy
R12 , ∃y ∀x Rxy

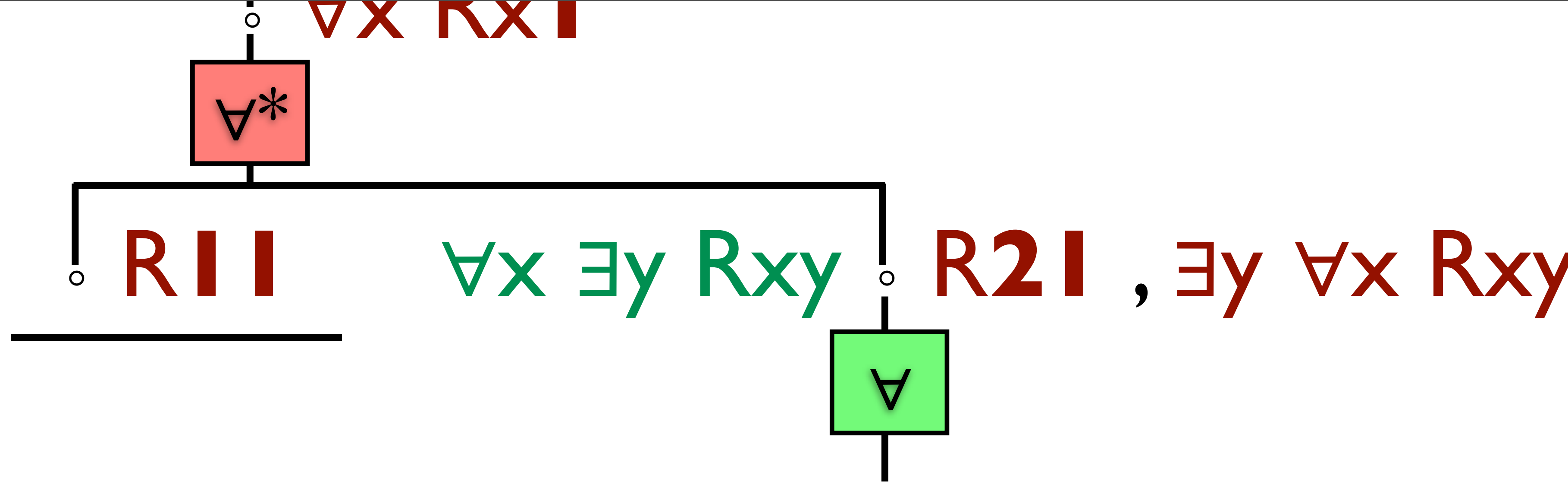


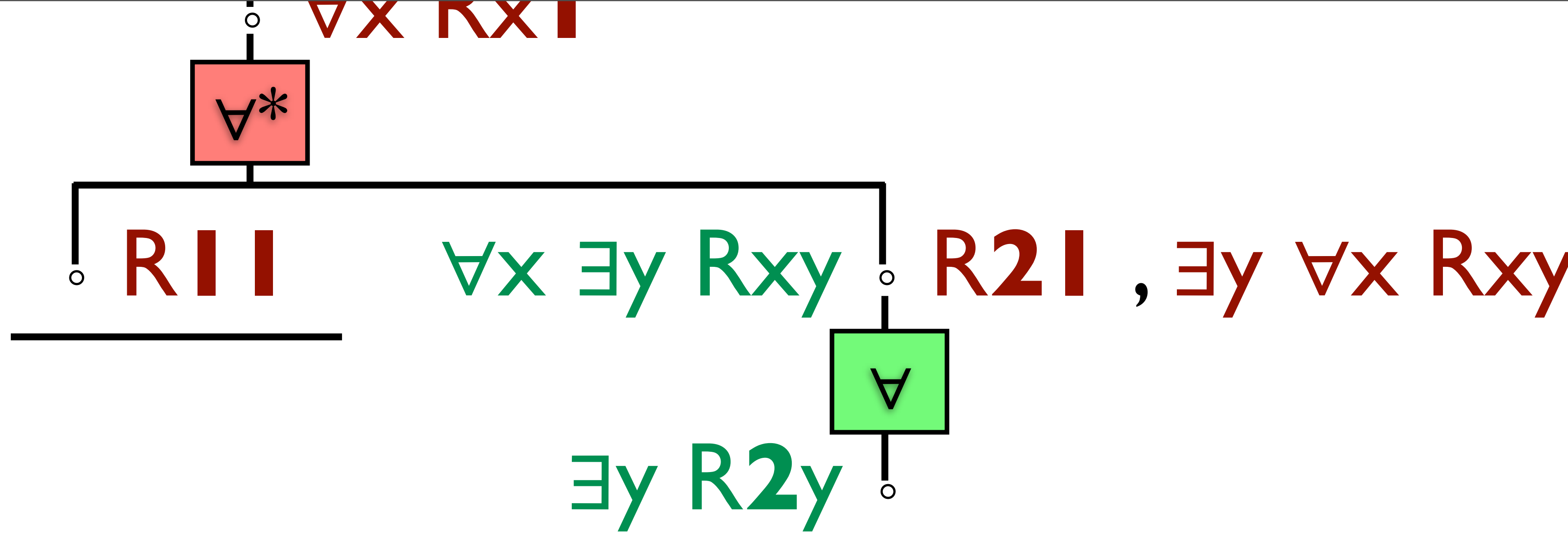
There is no counter-example
with only one object!

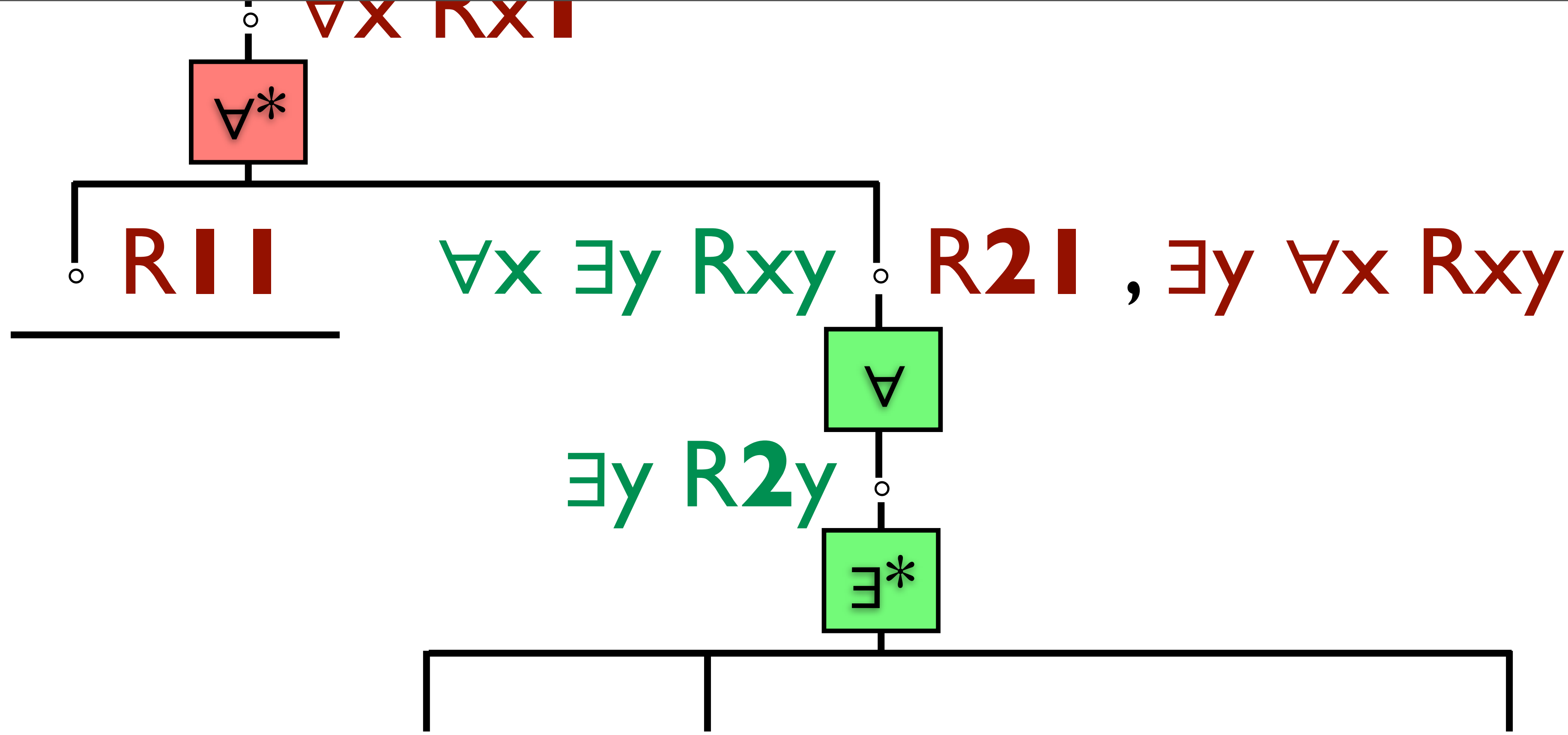
∀x Rx1

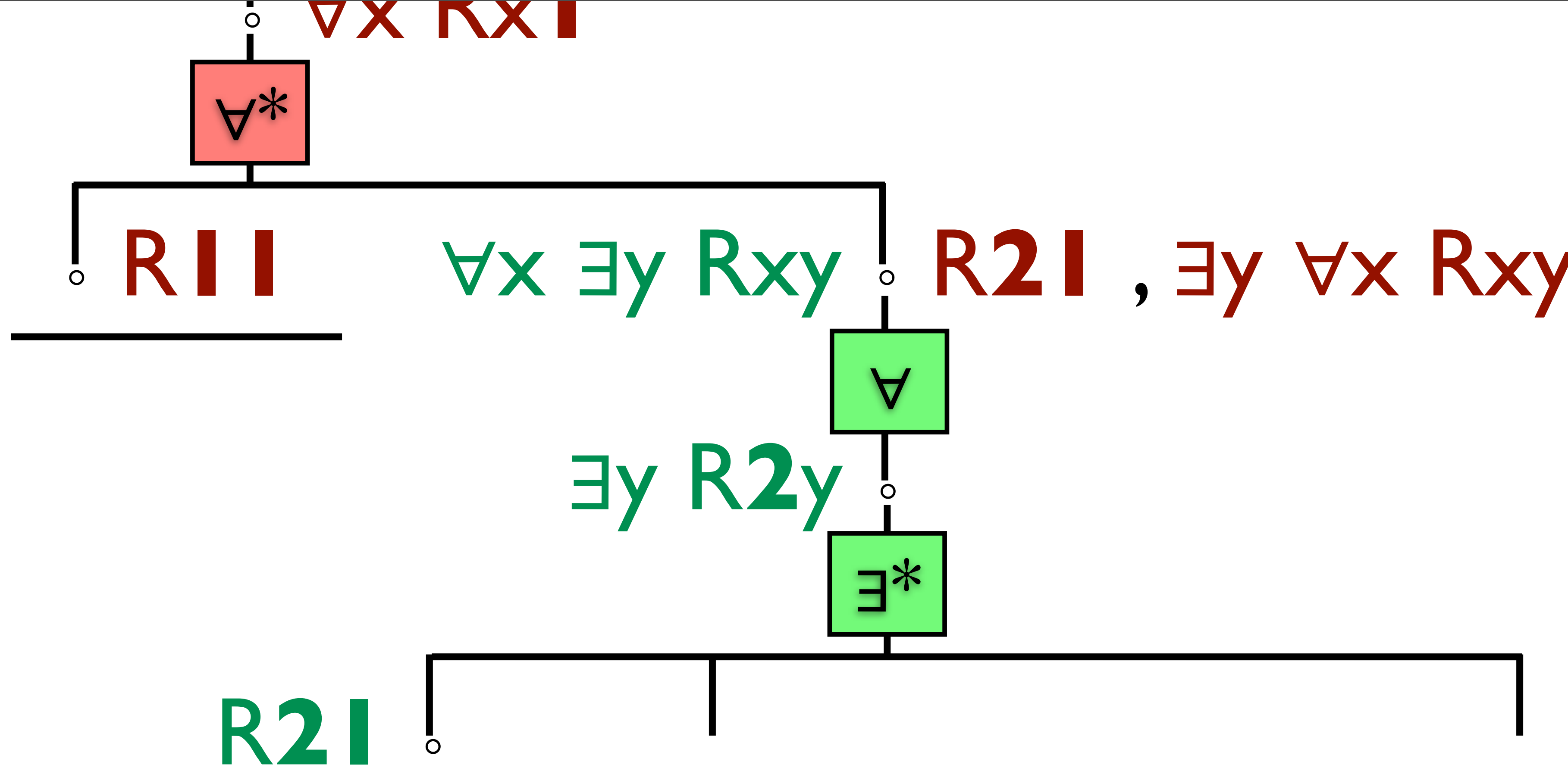
∀*

R11   ∀x ∃y Rxy   R21 , ∃y ∀x Rxy

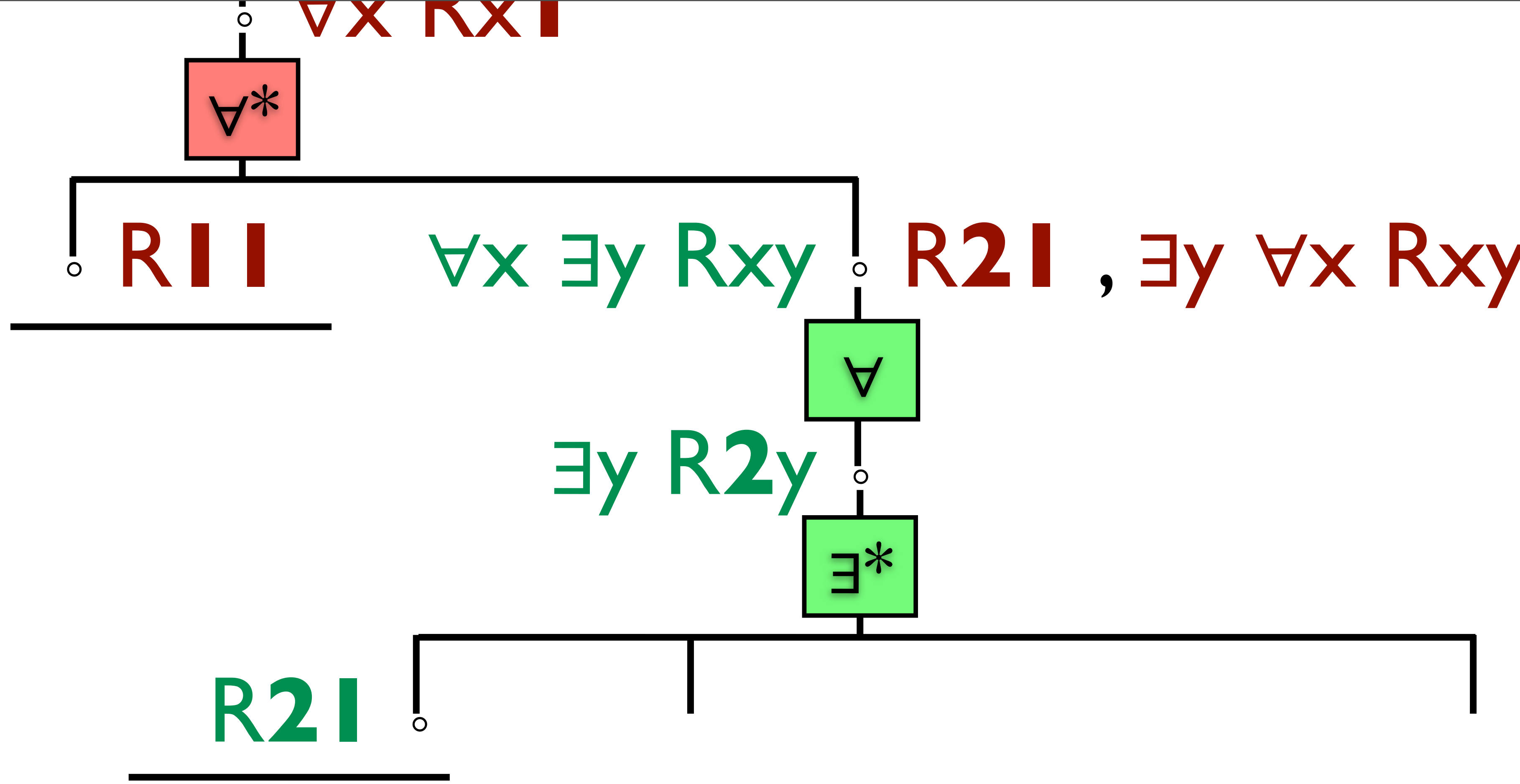

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀

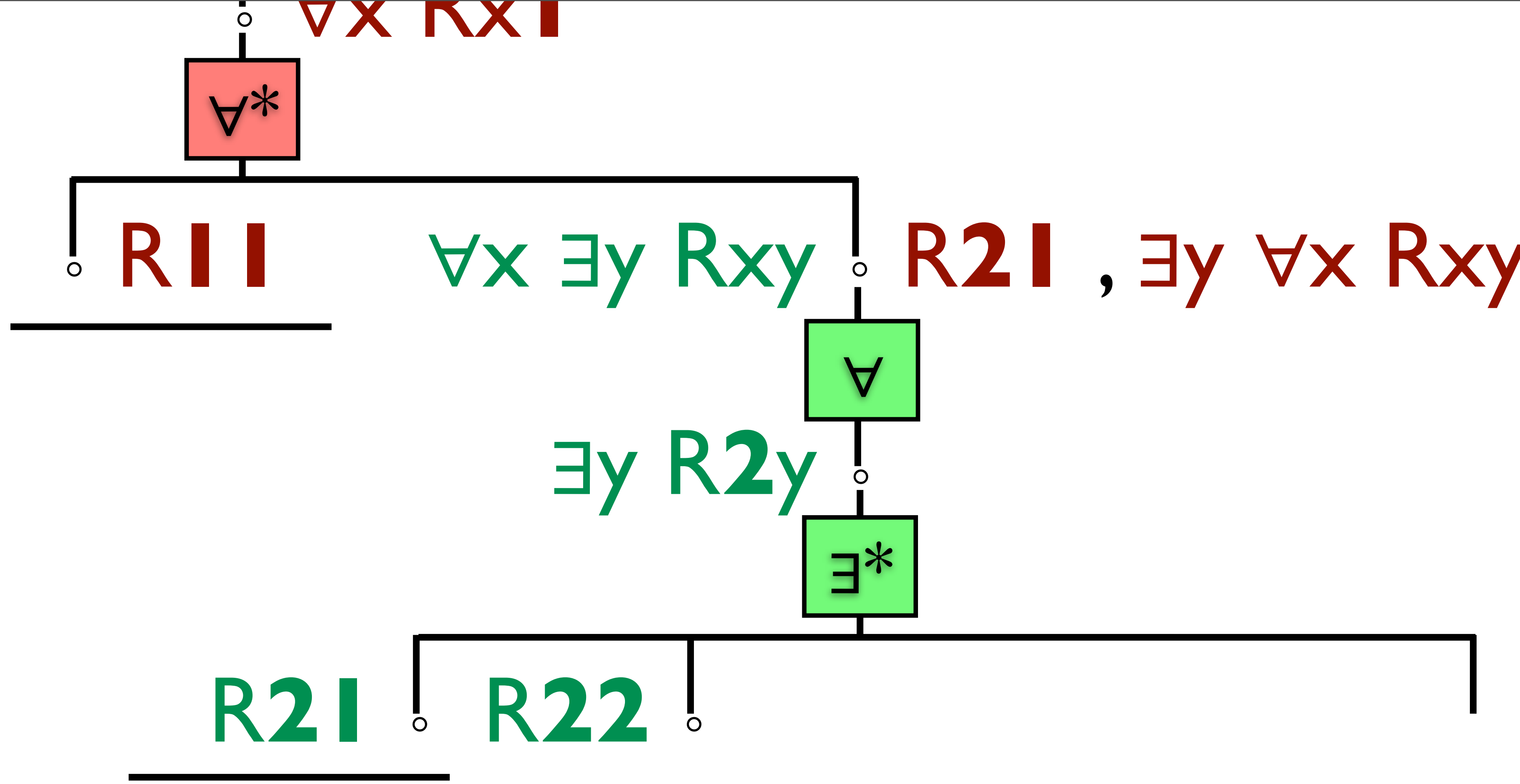

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y

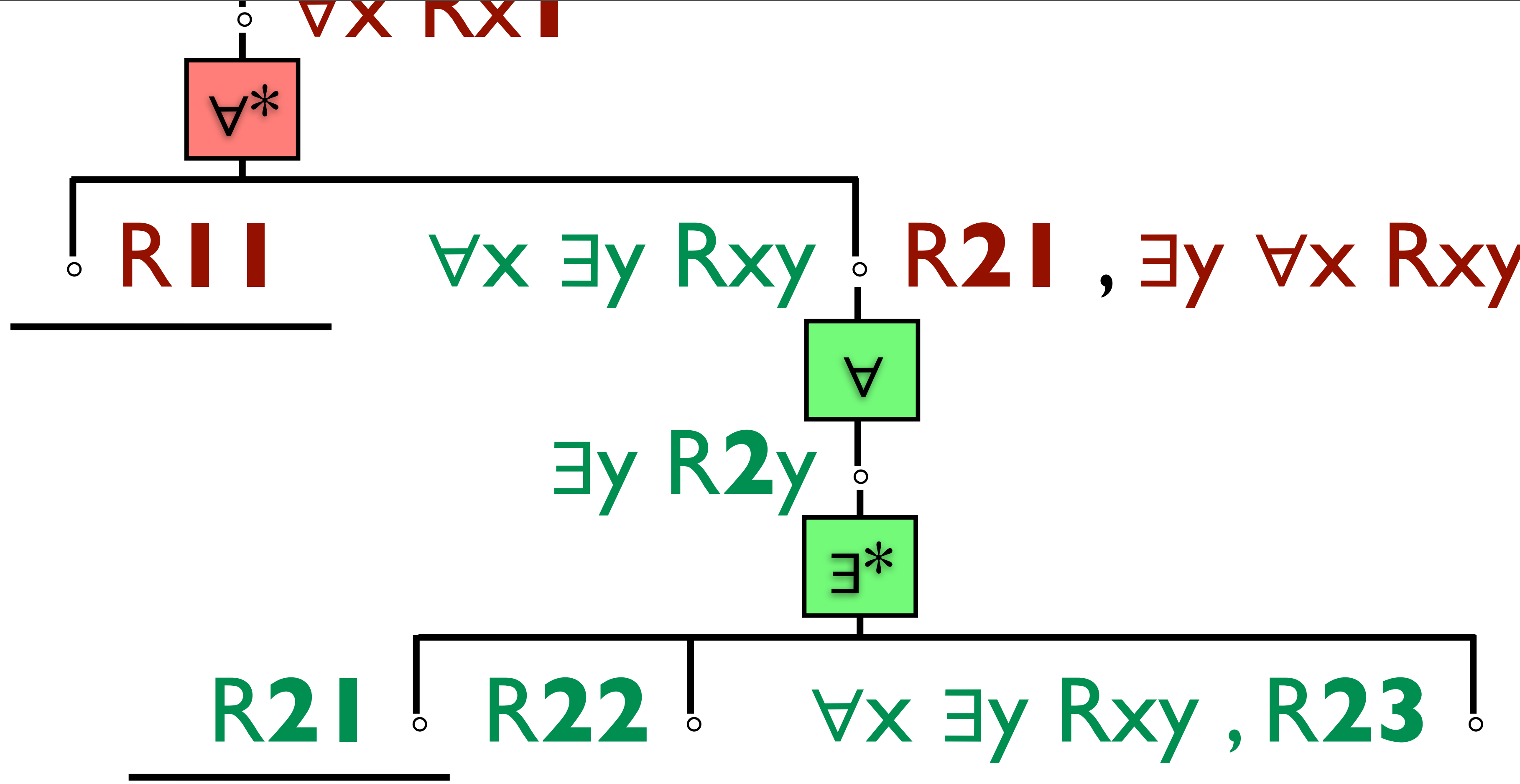

∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*

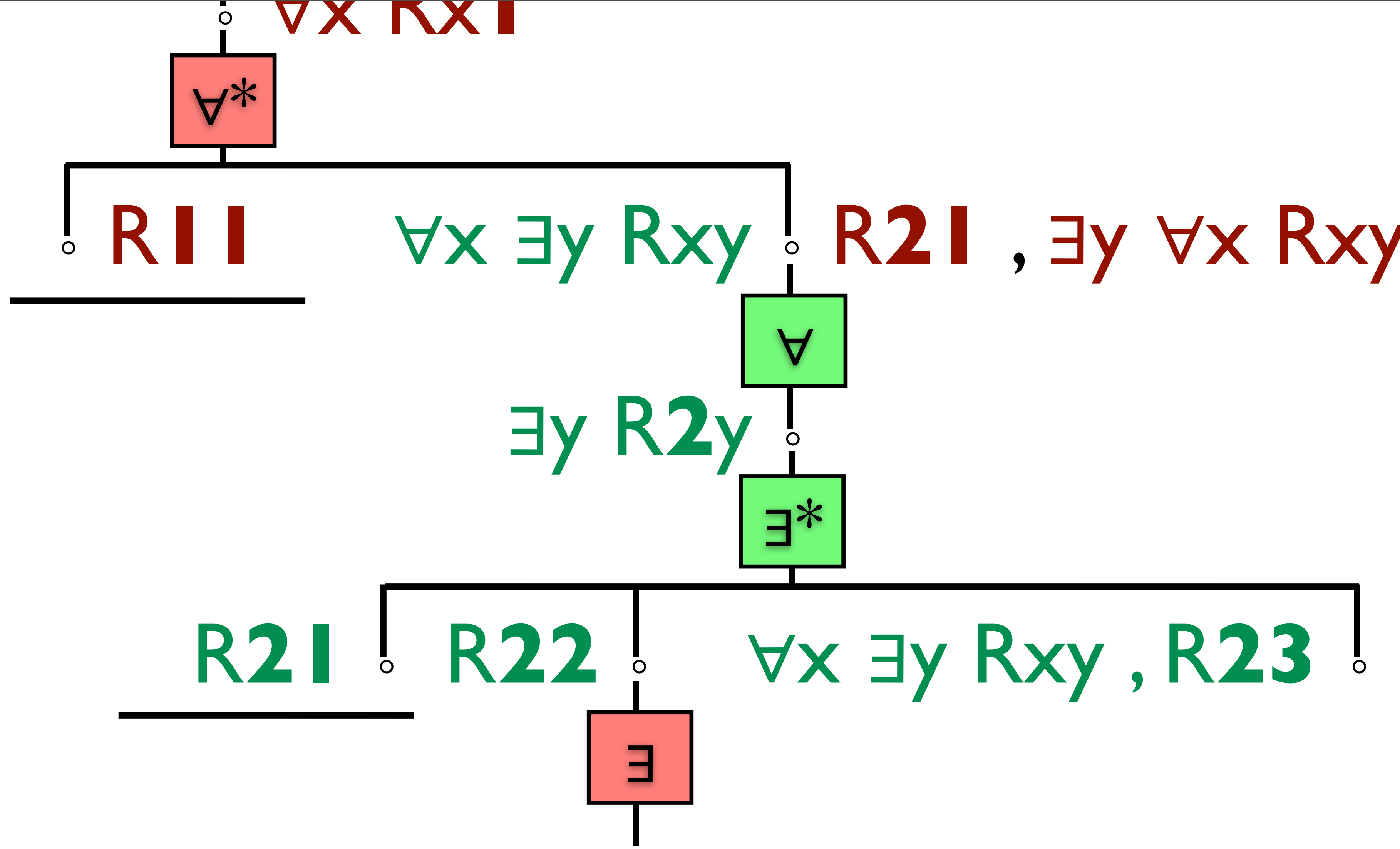


∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21

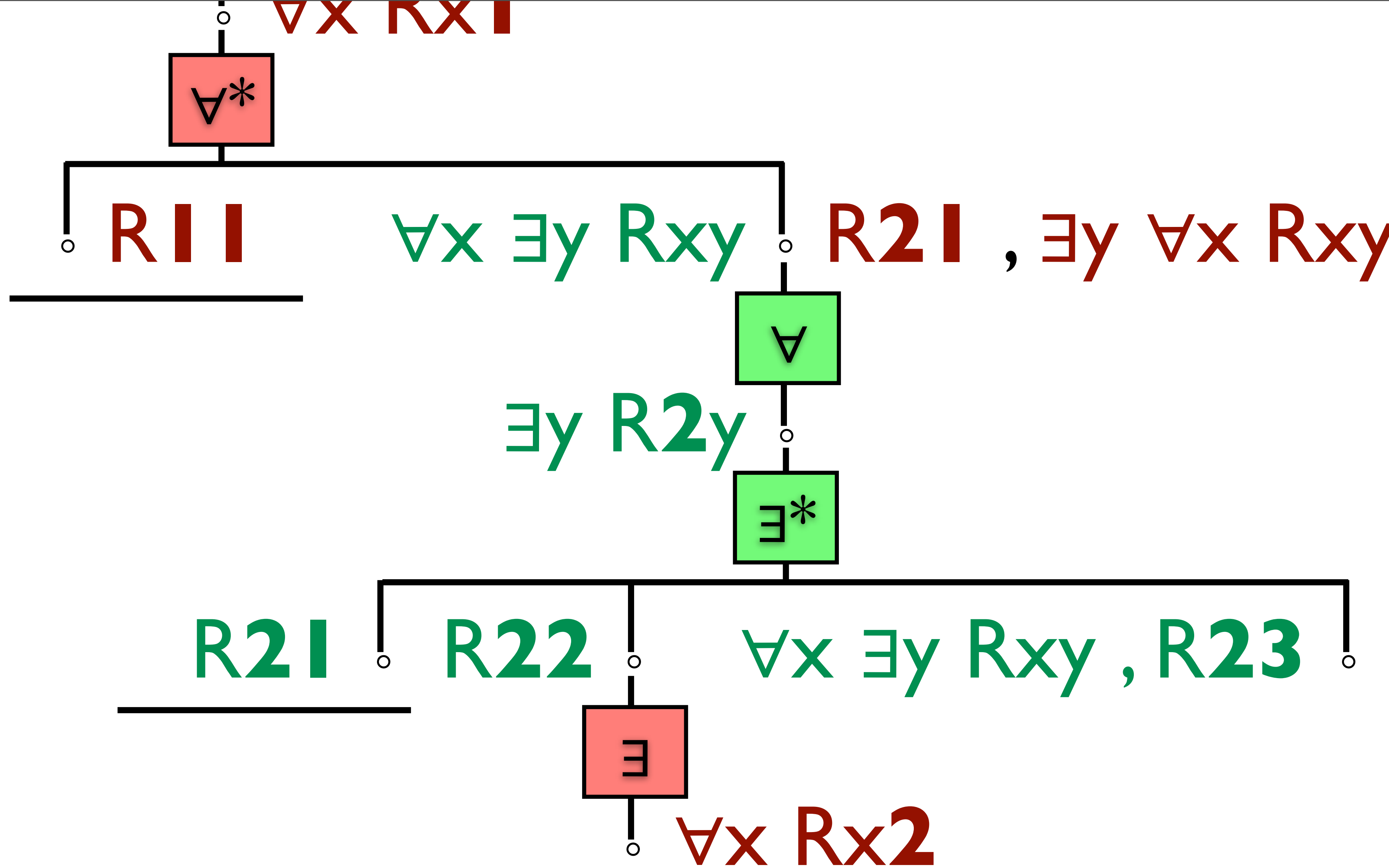

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21

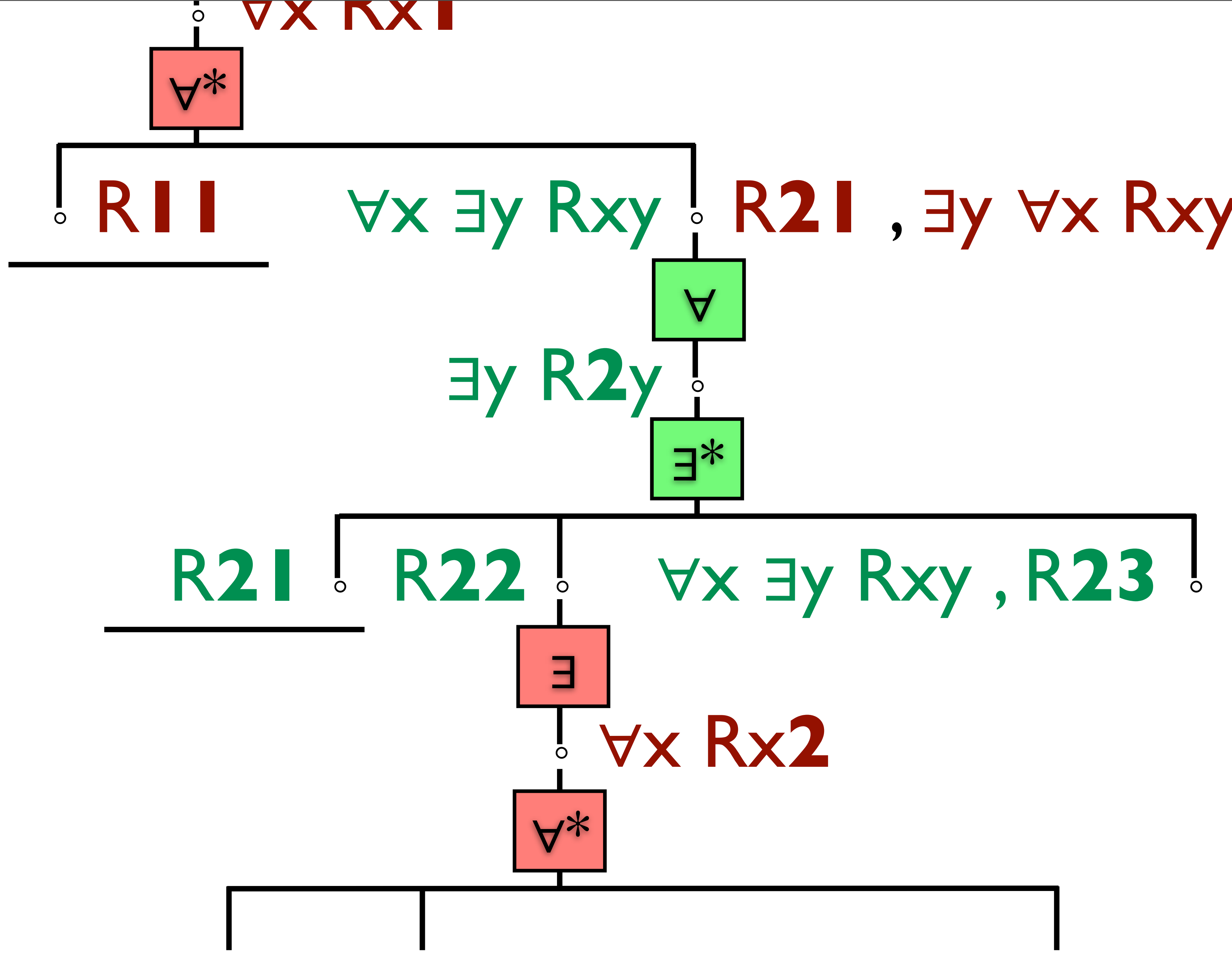

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22

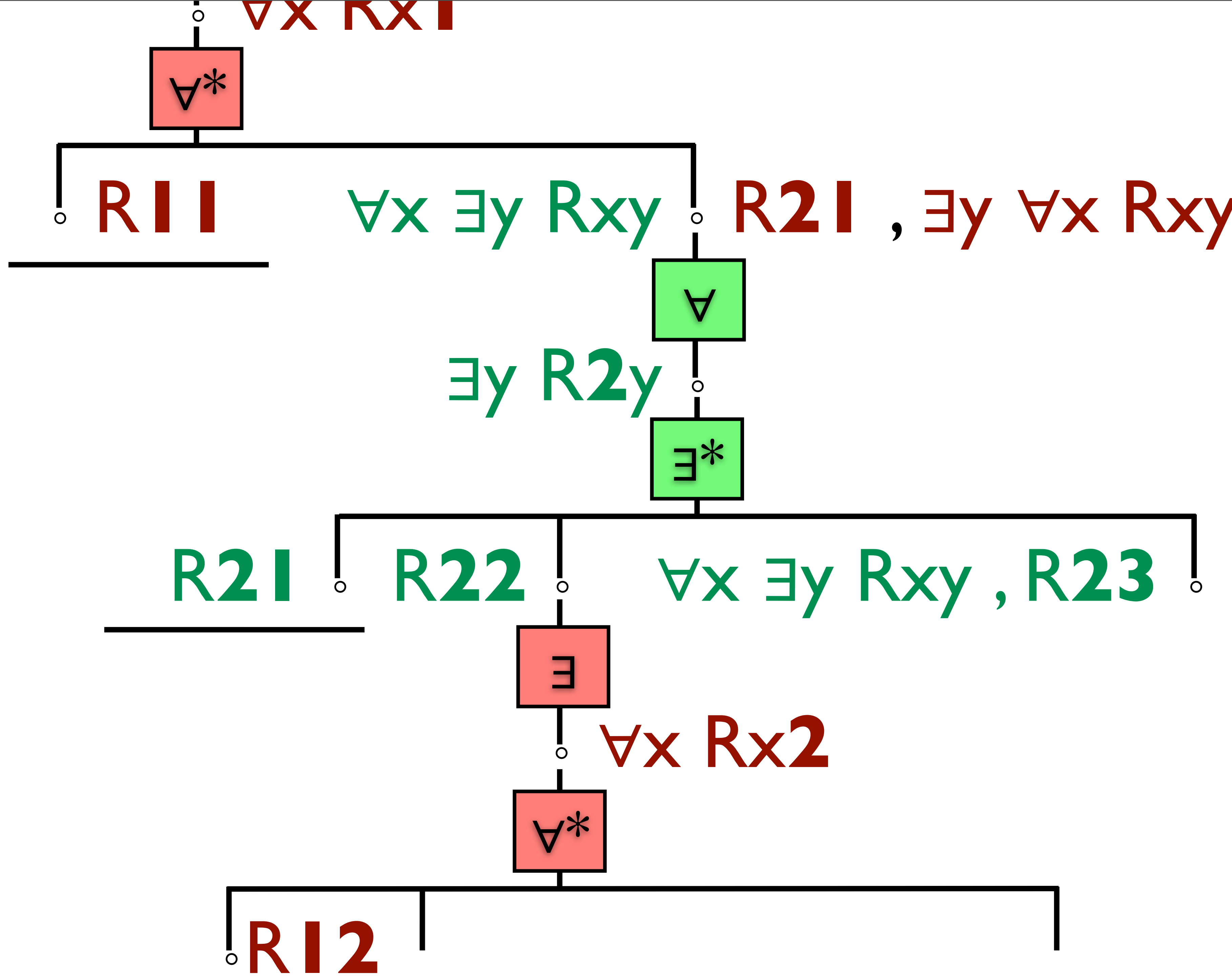

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23

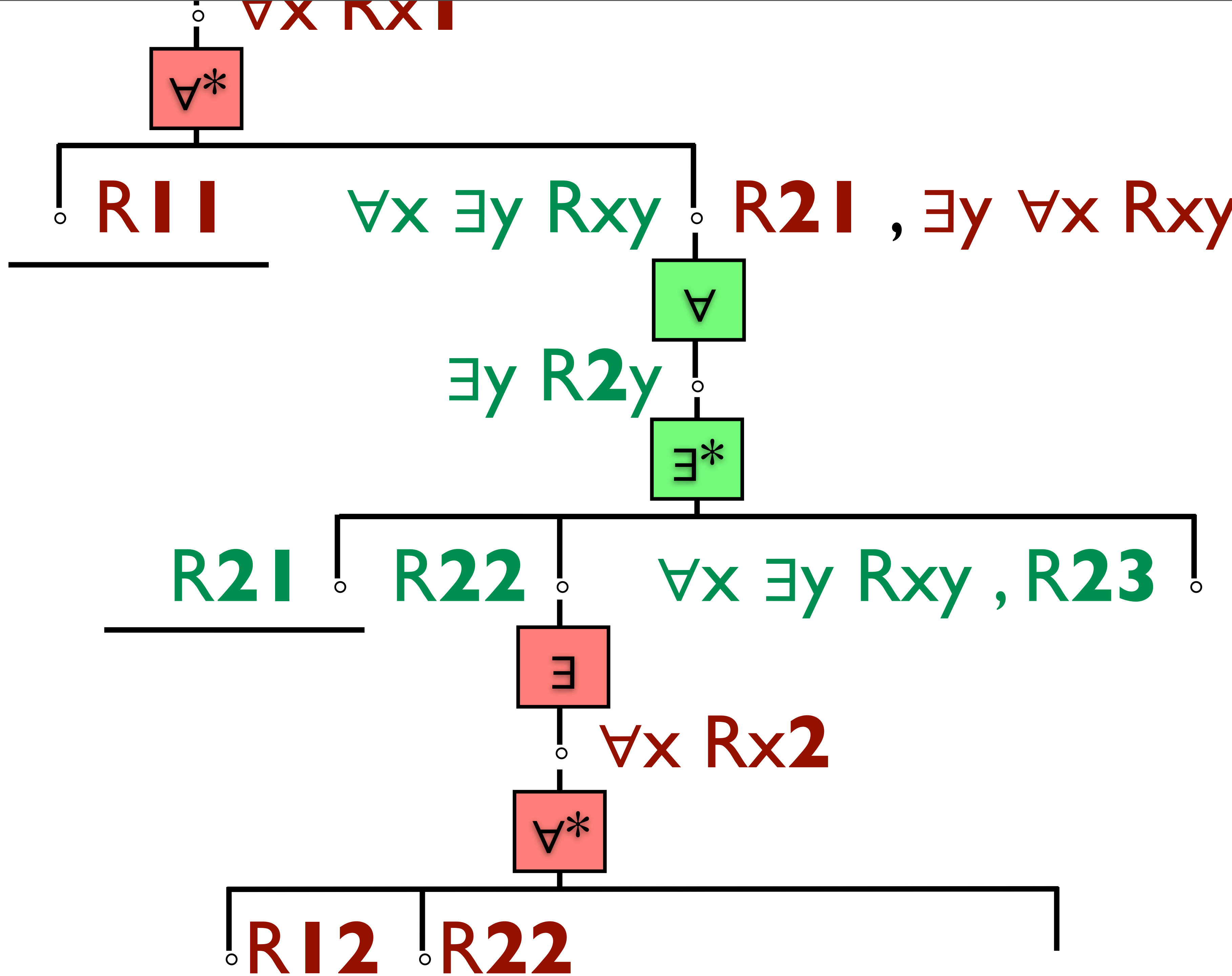


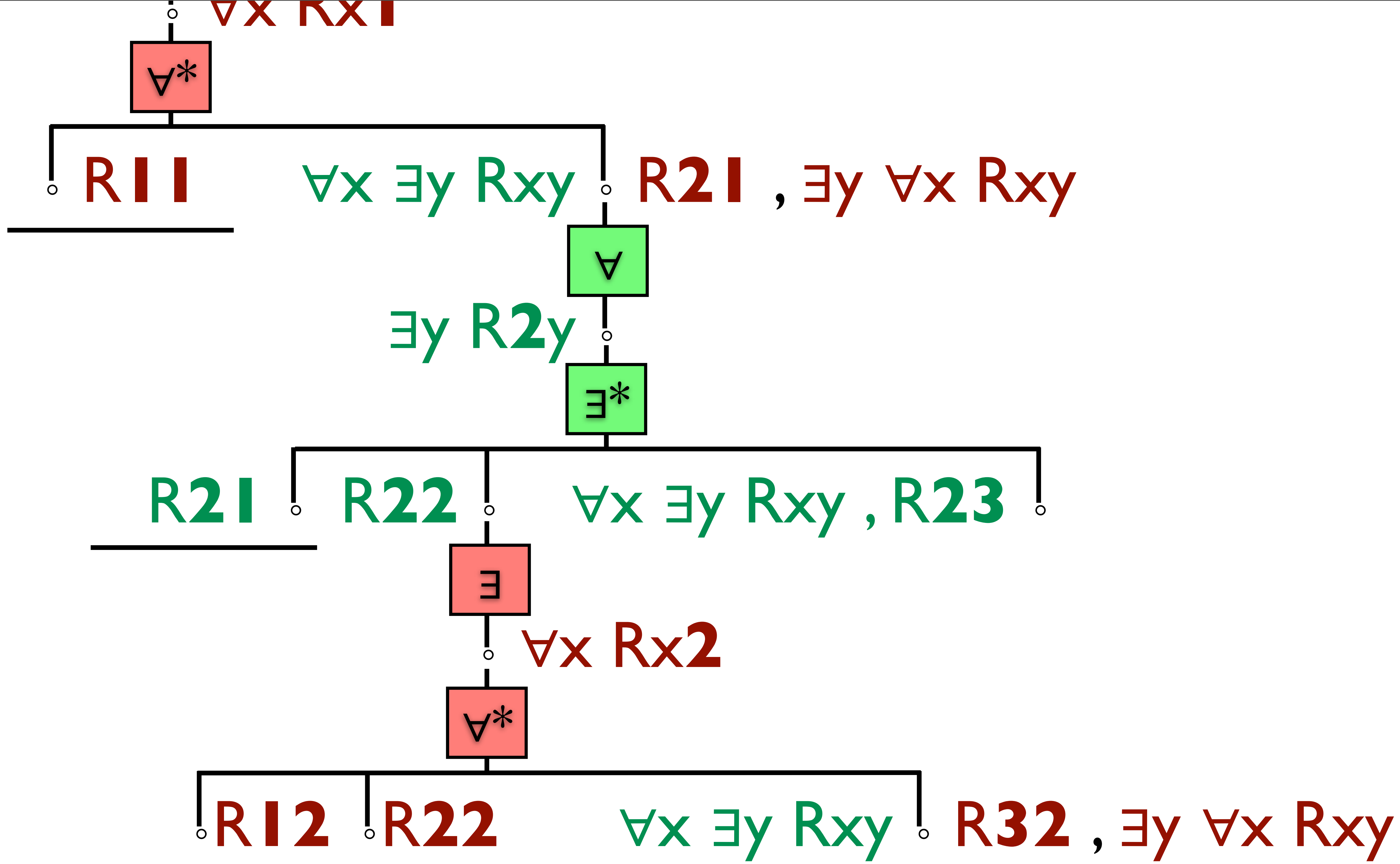
∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃

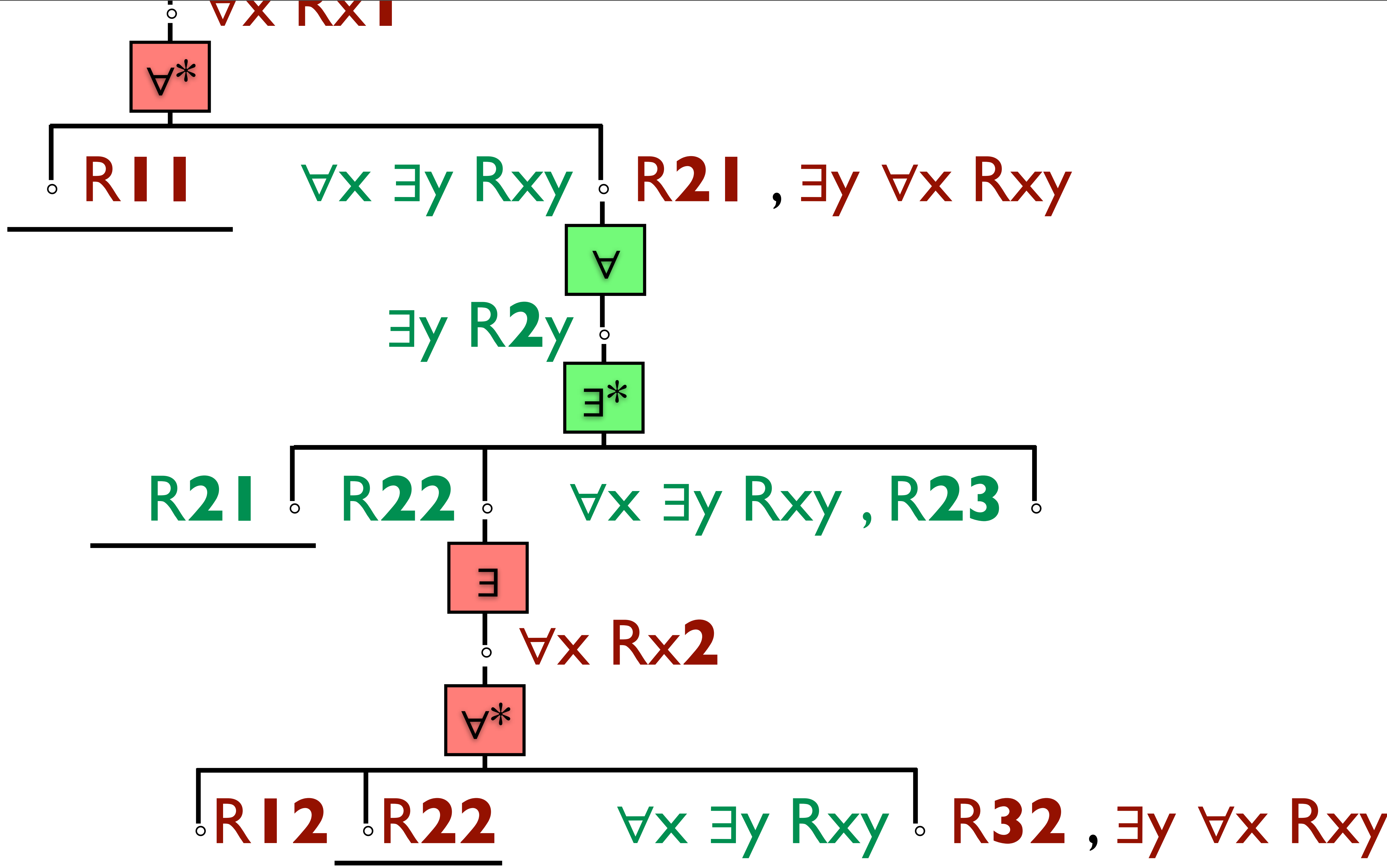

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2

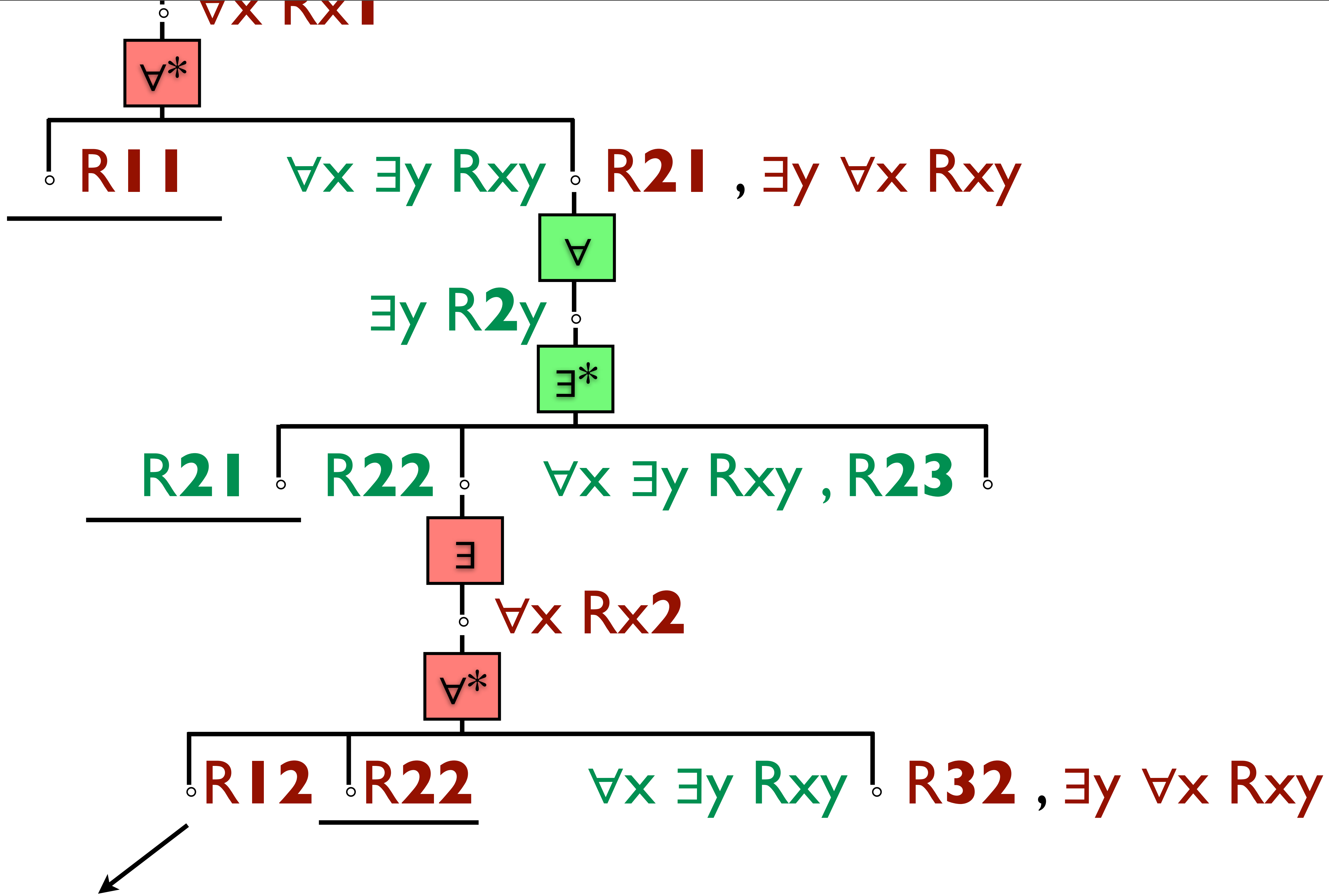

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2
∀*

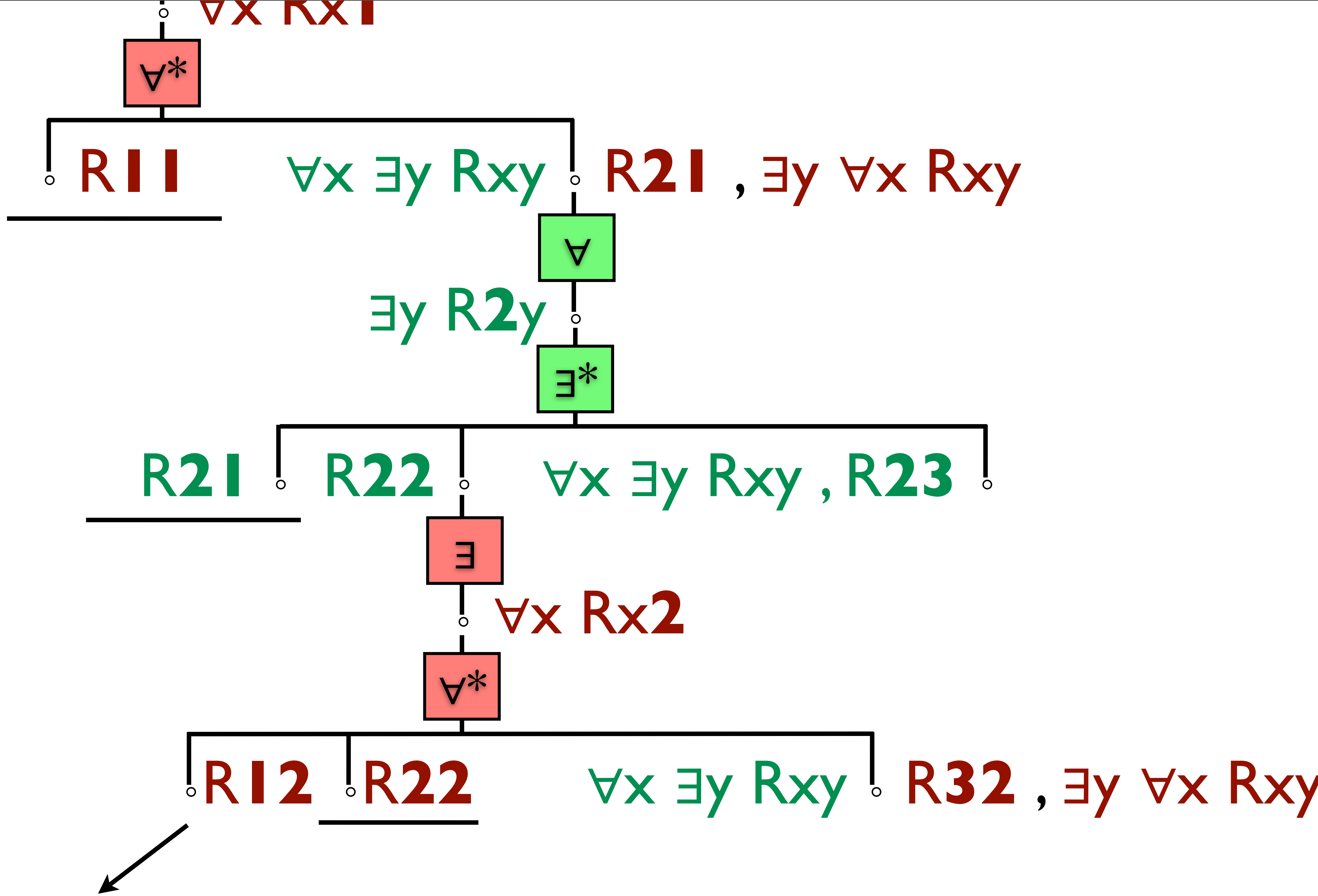

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2
∀*
R12


∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2
∀*
R12
R22

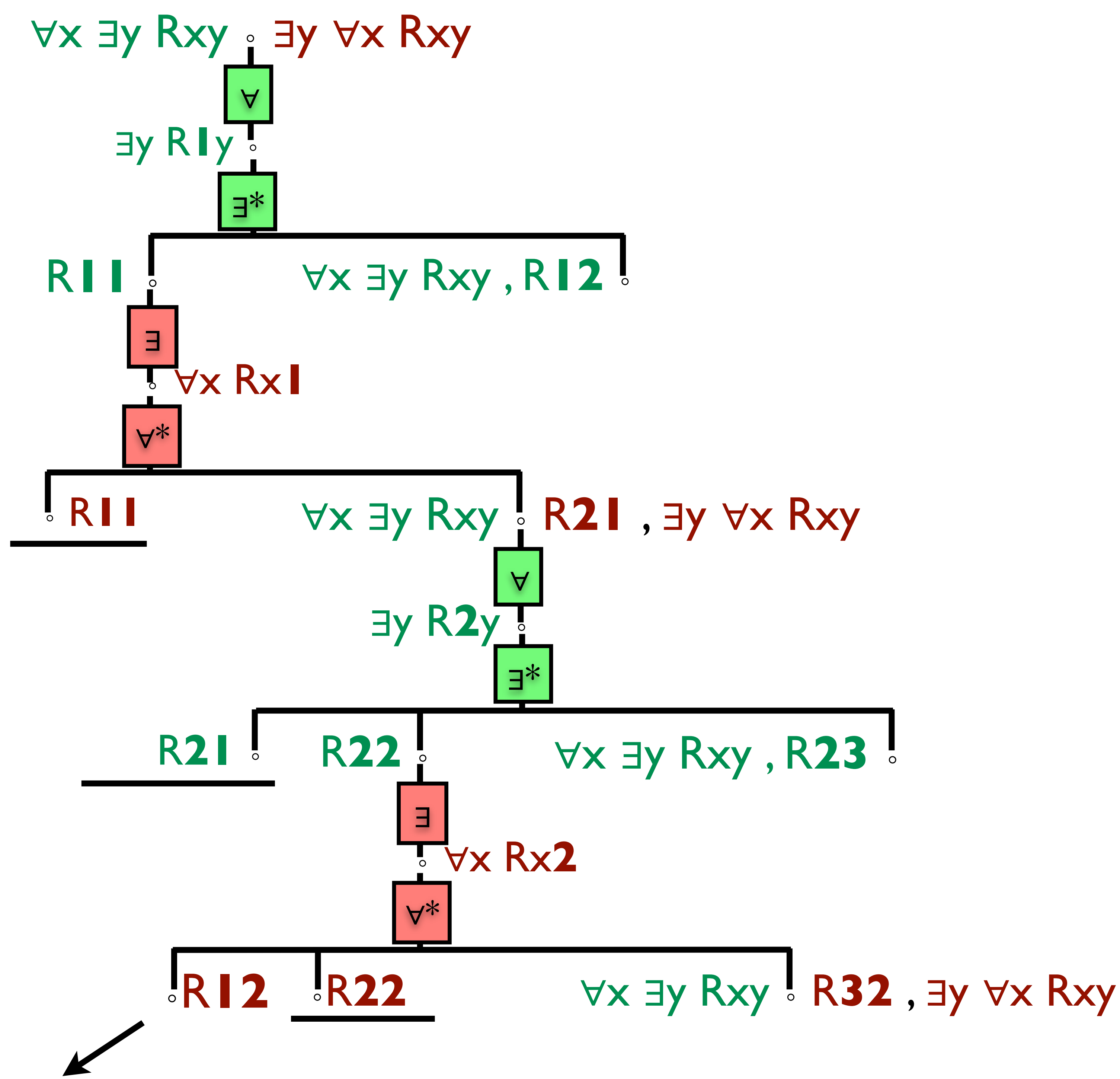

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2
∀*
R12
R22
∀x ∃y Rxy
R32 , ∃y ∀x Rxy


∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2
∀*
R12
R22
∀x ∃y Rxy
R32 , ∃y ∀x Rxy

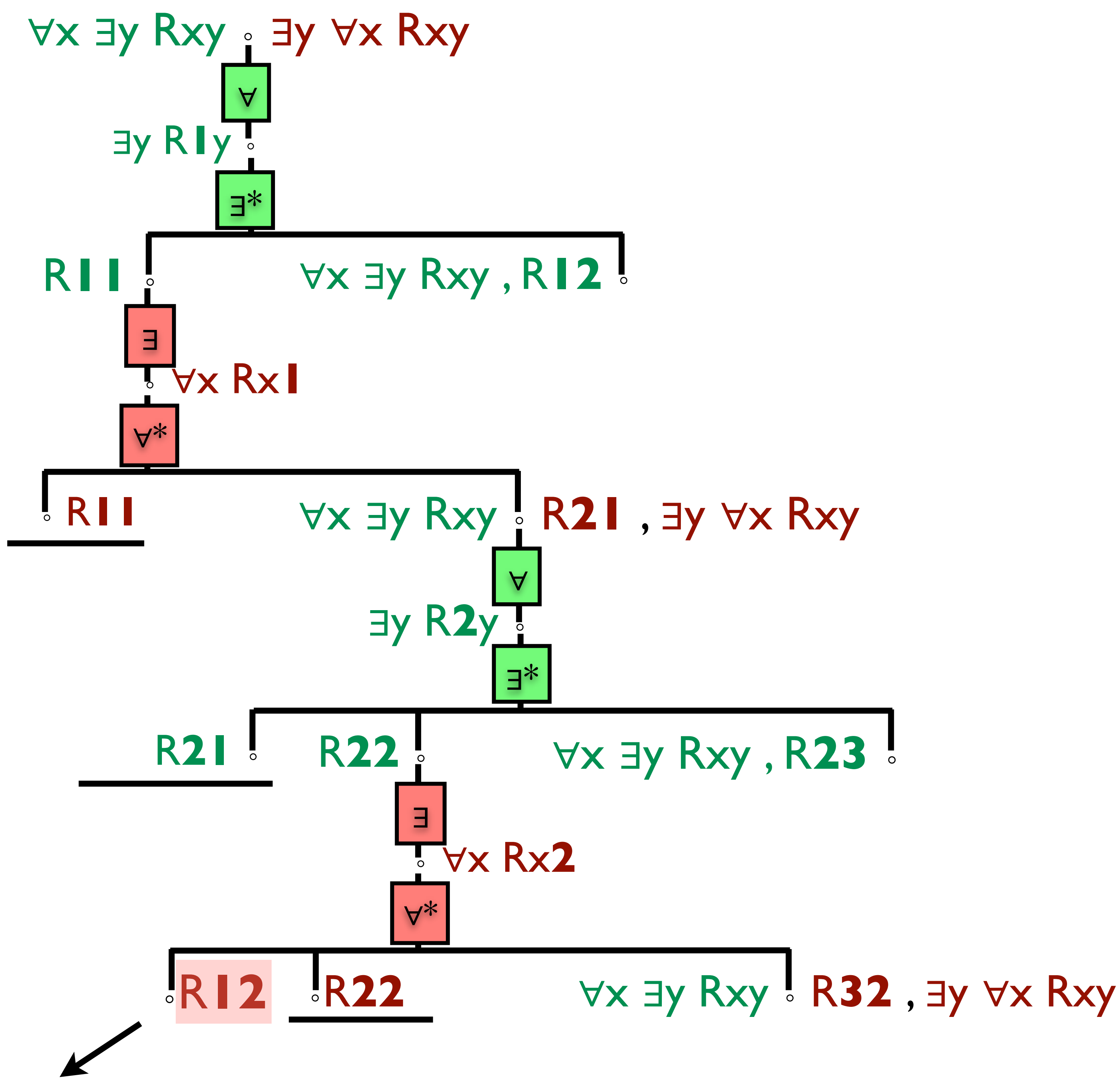

∀x Rx1
∀*
R11
∀x ∃y Rxy
R21 , ∃y ∀x Rxy
∀
∃y R2y
∃*
R21
R22
∀x ∃y Rxy , R23
∃
∀x Rx2
∀*
R12
R22
∀x ∃y Rxy
R32 , ∃y ∀x Rxy



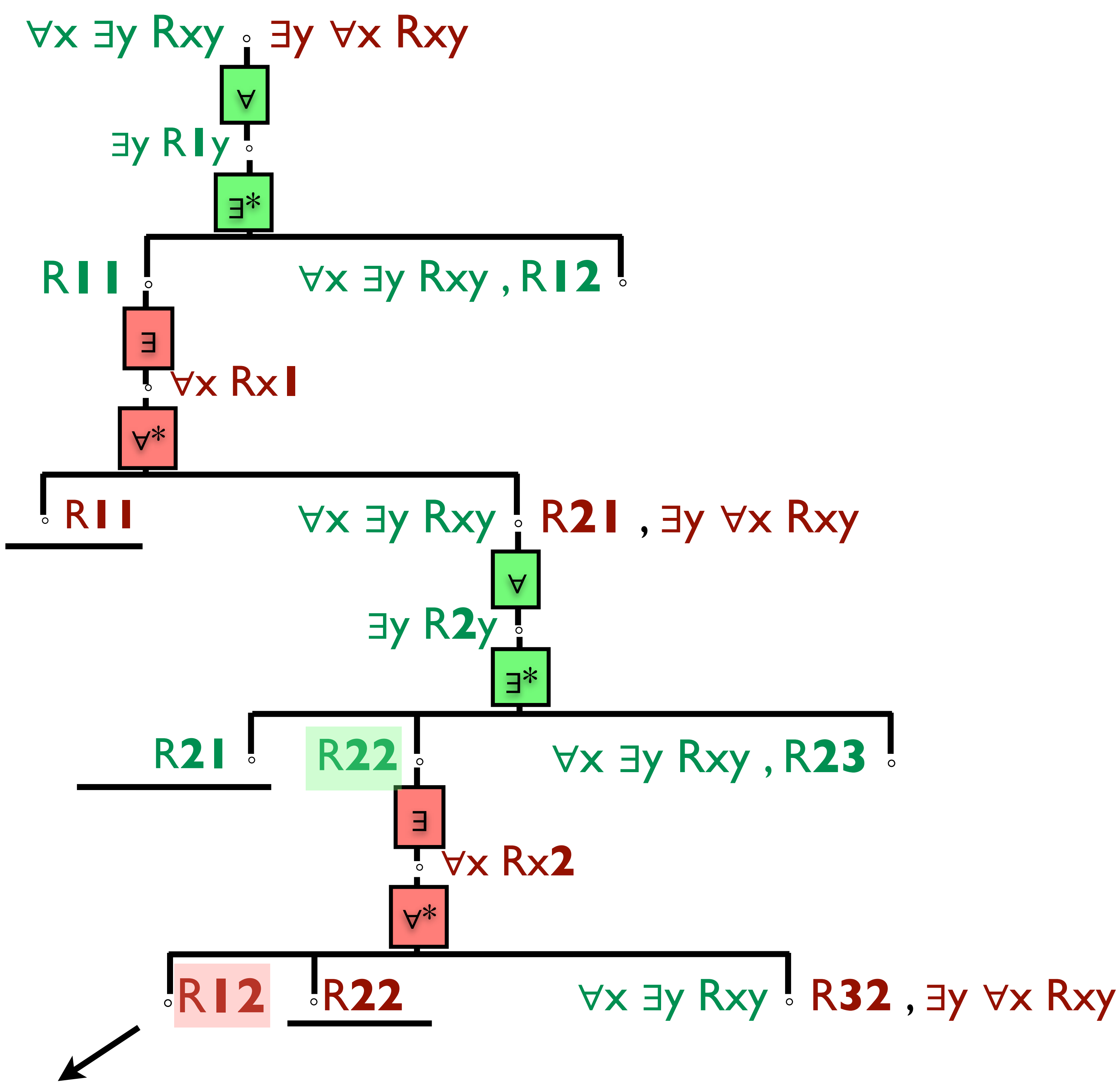
Open branch: counter-example R = {<1,1>,<2,2>}



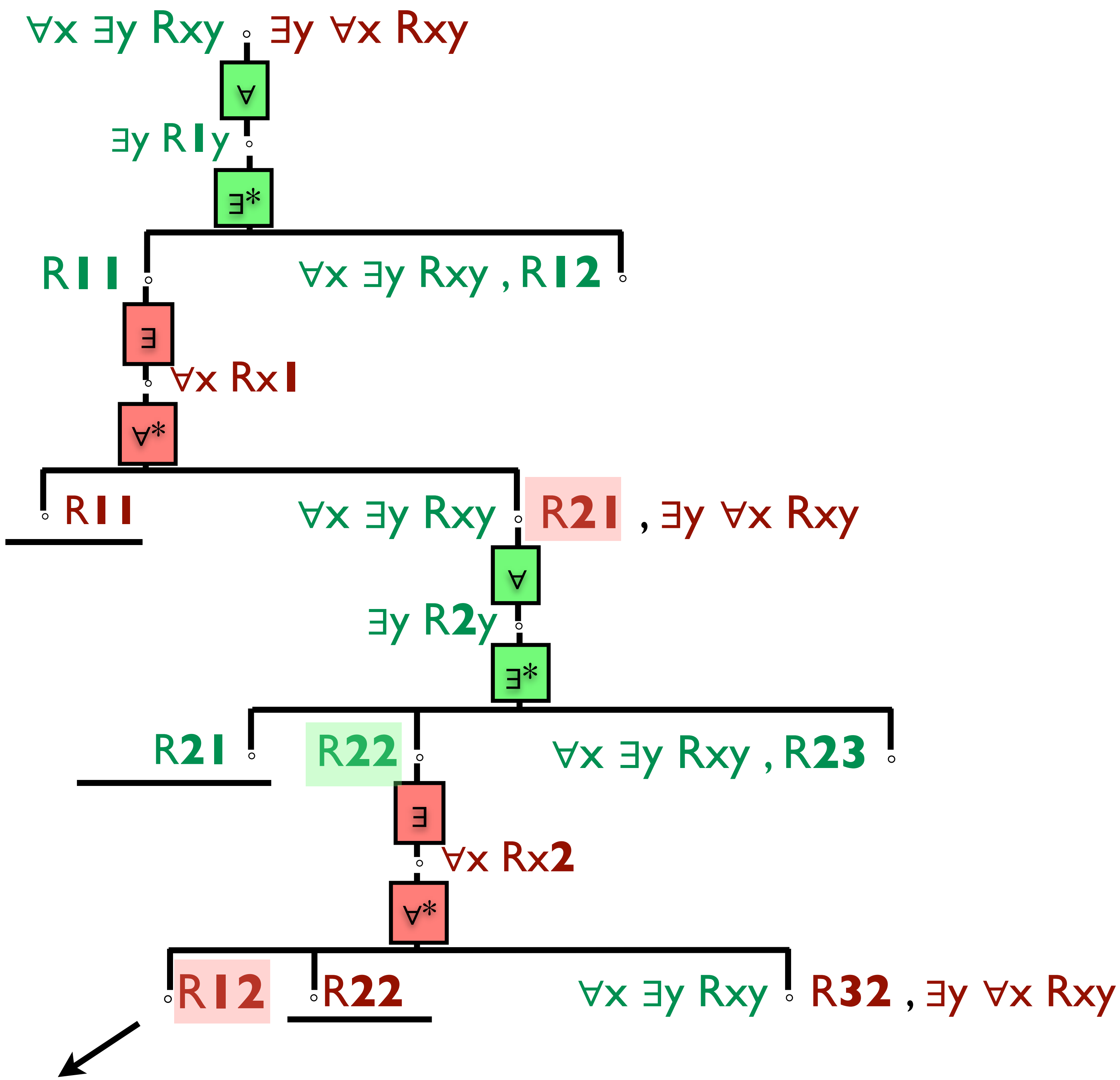
Open branch: counter-example R = {<0,0>,<1,1>}



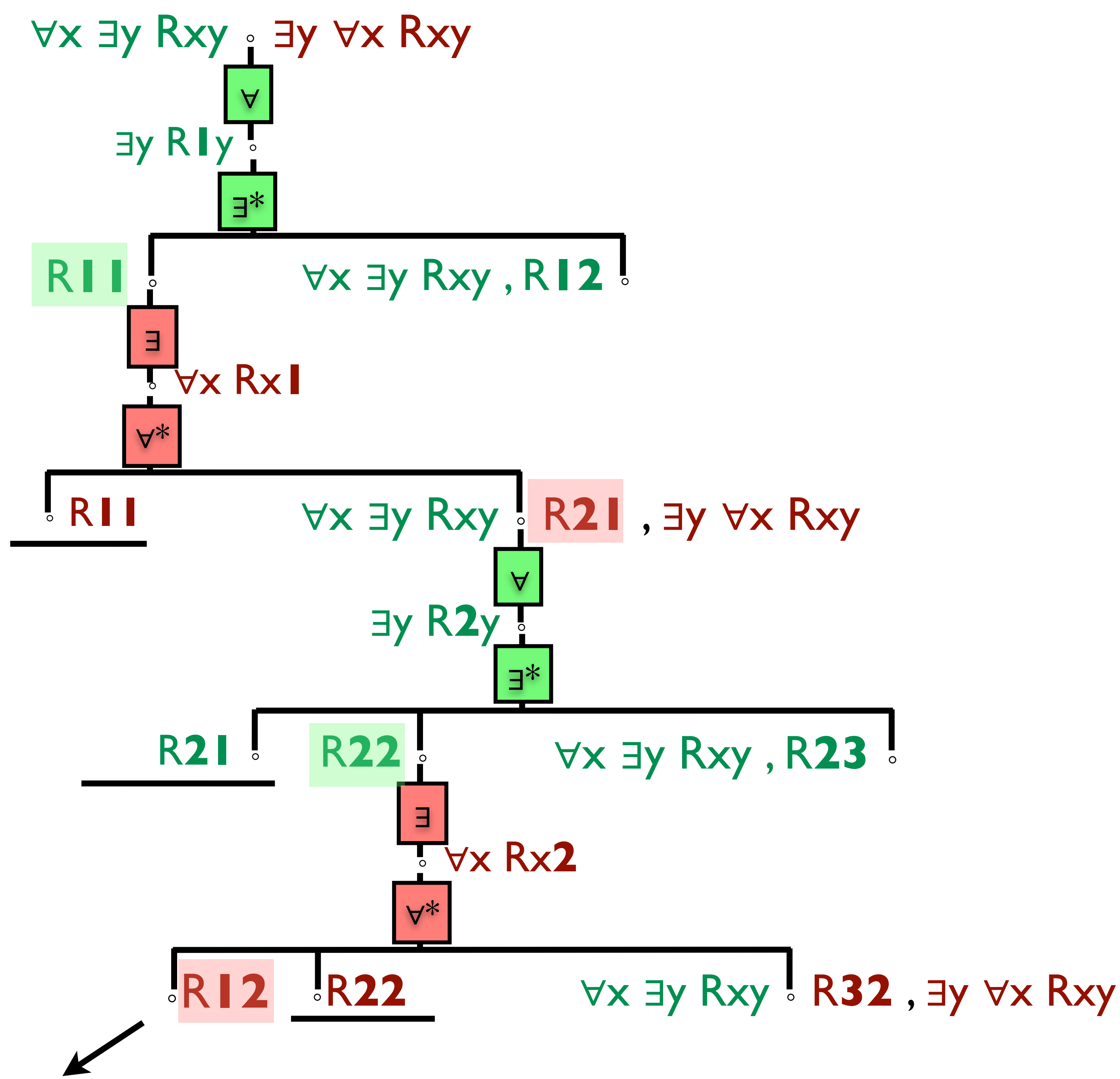
Open branch: counter-example R = {<0,0>,<1,1>}



Open branch: counter-example R = {<0,0>,<1,1>}



Open branch: counter-example R = {<0,0>,<1,1>}



Open branch: counter-example R = {<0,0>,<1,1>}

$$\forall x \exists y\, Rxy,\ \forall x \forall y \forall z ((Rxy \wedge Ryz) \rightarrow Rxz)\ \not\models\ \exists x \exists y (Rxy \wedge Ryx)$$

....but there exist ***only*** counter-examples with ***infinitely many*** objects!

$\forall x \exists y\, Rxy,\ \forall x \forall y \forall z((Rxy \wedge Ryz) \rightarrow Rxz)\ \circ\ \exists x \exists y(Rxy \wedge Ryx)$

$\forall x \exists y\, Rxy,\ \forall x \forall y \forall z((Rxy \wedge Ryz) \rightarrow Rxz)\ \circ\ \exists x \exists y(Rxy \wedge Ryx)$