Variables & quantifiers

Relational Logic

object constants

relational constants

variables

---

relational constants

- arity ⇒ the number of arguments with which the constant can be combined

a relational constant that can be combined with 1 arg.

unary → 1 argument
binary → 2 arguments
ternary → 3 arguments
n-ary → n arguments

Vocabulary ≡

1) set of object constants
2) set of relation constants
3) an assignment of arities for each relational const

term ≡ variable or object constant

i.e. joe or someone or person_a

Three Types of Sentences

1. Relational Sentences (these are like propositions in prop logic)
2. Logical Sentences (these are like logical sentences in prop logic)
3. Quantified Sentences

Relational Sentence: formed from an n-ary relational constant and n arguments.

So, for example if q is a binary relational constant, then

$$\underbrace{q(\underset{\text{term}}{a}, \underset{\text{term}}{y})}_{\text{Sometimes called an atom}}$$

is a legal relational constant.

Logical Sentences: Defined as in propositional Logic

Negation $\neg p(a)$

conjunctions $p(a) \wedge q(b,c)$

disjunction $p(a) \vee q(b,c)$

implication $p(a) \rightarrow q(b,c)$

biconditional $p(a) \Longleftrightarrow q(b,c)$

Syntax is the same as propositional logic but relational sentences replace propositional constants

$p \rightarrow q \vee r$ $\qquad\qquad$ $p(a,b) \rightarrow q(c) \vee r(d,e)$

propositional constants $\qquad\qquad$ relational sentences

Quantified Sentences are formed from a quantifier a variable and an ~~quantified~~ embedded sentence, called the scope of the quantifier.

1. Universally Quantified Sentences
2. Existentially Quantified Sentences.

Universally Quantified Sentences: used to assert that all objects have a certain property

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

for all x it is true that if p holds for an object x then q also holds for that object and itself

Existentially Quantified Sentences: ④

- asserts that some object has a certain property

$$(\exists x.(p(x) \wedge q(x,x)))$$

↗
there exists some object x that satisfies p and also satisfies q when paired with itself.

Quantified Sentences can be Nested

$$\cancel{\forall p x}\;\underbrace{\Big(\underbrace{(\underbrace{\forall x.}_{\text{quantifier}}\underbrace{(p(x))}_{\text{scope}})}_{\text{Quantified Sentence}} \vee \underbrace{(\underbrace{\exists x.}_{\text{quantifier}}\underbrace{(q\underbrace{(x,x)}_{\text{relational sentence}})}_{\text{scope}})}_{\text{Quantified Sentence}}\Big)}_{\text{logical Sentence.}}$$

precedence

$\forall \;\; \exists$ — Quantifiers take precedence over logical operators

$\neg$

$\wedge$

$\vee$

$\rightarrow \;\; \Leftrightarrow$

Applying precedence

$\forall x. p(x) \Rightarrow q(x)$ $\qquad$ $(\forall x. p(x)) \Rightarrow q(x)$

For all x, p(x) implies q(x)

The quantifier stays with the relational sentence unless specified otherwise with parenth

$\exists x. p(x) \land q(x)$ $\qquad$ $(\exists x. p(x)) \land q(x)$

(There exists some x such that p(x) is true) and q(x) is true

<u>Note:</u> $\forall x. \underbrace{(p(x) \rightarrow q(x))}$ has different meaning

this entire logical sentence is now quantified.

$\exists x. \underbrace{(p(x) \land q(x))}$

this entire logical sentence is now quantified

Ground: an expression in relational logic is ground IFF it contains no variables.

i.e. $p(a)$ is ground

$\forall x. p(x)$ is <u>NOT</u> ground

Free An occurance of a variable is "free" (6)

IFF the variable is <u>not</u> in scope of quantifier

Free Variable: <u>Not</u> in Scope of Variable

Bound Variable: In Scope of Variable

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

bound variable   bound variable

$\exists x.\, p(x) \rightarrow q(x)$

bound variable   free variable

$\exists x.\, p(x,y)$

bound variable   free variable

Open Sentence: IFF it has free variables

Closed Sentence: IFF all variables are bound.

$p(y) \Rightarrow \exists x.\, q(x,y)$ open sentence

free   bound   free

x is bound but y is free

$\forall y.(p(y) \Rightarrow \exists x.\, q(x,y))$ Closed sentence

bound   bound   bound

y is bound

x is bound

do problem 6.1

Semantics

Herbrand base ≡ the set of all ground relational sentences that can be formed from the constants of the language.

So, all sentences of the form

$r(t_1, \ldots, t_n)$ where $r \equiv$ n-ary relation constant

$t_1, \ldots, t_n \equiv$ object constants

Consider a vocabulary with {object constants a, b}

Object constants a, b

relation constants p, q

where p is unary arity 1 $p(x)$

q is binary arity 2 $q(x,y)$

The Herbrand Base is

$\{p(a), p(b), q(a,a), q(a,b), q(b,a), q(b,b)\}$

This is the set of all ground relational sentences that can be formed from the relational constants and object constants

Herbrand Base is finite.

So, if the language includes

b ~~relational~~ object constants that have n-arity

then there are b^n ground relational sentences

for each relational constant that has n-arity

in our example $\underbrace{p, q}_{\text{relational constants}}$ $\underbrace{a, b}_{\text{object constants}}$

$$2^1 + 2^2$$

$$\underbrace{p(a), p(b)}_{b^n} \quad \underbrace{q(a,a), q(a,b), q(b,a), q(b,b)}_{b^n}$$

b^n ← 1-arity, b = 2 object constants

b^n ← 2-arity, b = 2 object constants

Truth Assignment: map each ground relational sentence to a truth value

so, we could have

$p(a) \to 1$ $\quad$ $q(a,a) \to 1$

$p(b) \to 0$ $\quad$ $q(ab) \to 0$

$q(ba) \to 1$

$q(b,b) \to 0$

Now, the semantics of relational logic are the same as the semantics of propositional logic.

Instance: An instance of an expression is an expression in which all free variables have been consistently replaced by ground terms.

A universeally quantified sentence is true for a truth assignment [in which the free variables have been replaced by ground terms] iff every instance in the scope of the quantified sentence is true

An existentially quantified sentence is true for a truth assignment iff (at least 1) some instance ~~of~~ in the scope of the qualified sentence is true

(9a)

Consider a vocabulary with the following

object constants	relation constants
deer (d)	h(x) unary relational constant herbivor
rabbit (r)	c(x) unary relational constant carnivore
wolf (w)	p(x,y) binary relational constant x preys upon y.
fox (f)	

The Herbrand Base is

h(d)	c(d)	p(d,d)	p(r,d)	p(w,d)	p(f,d)
h(r)	c(r)	p(d,r)	p(r,r)	p(w,r)	p(f,r)
h(w)	c(w)	p(d,w)	p(r,w)	p(w,w)	p(f,w)
h(f)	c(f)	p(d,f)	p(r,f)	p(w,f)	p(f,f)

This is the set of all ground relational sentences that can be formed from the object constants and the relational constants in the vocabulary

Truth Assignments:

Map each ground relational sentence to a truth value.

(9b)

$h(d) = 1$ $c(d) = 0$ $\exists x.\, h(x) \wedge \exists x.\, c(x)$

$h(r) = 1$ $c(r) = 0$

$h(w) = 0$ $c(w) = 1$ $\forall x.(\neg h(x) \rightarrow c(x))$

$h(f) = 0$ $c(f) = 1$ $\forall x.(\neg c(x) \rightarrow h(x))$

$\neg \exists x.(h(x) \wedge c(x))$

$p(d,d) = 0$

$p(d,r) = 0$ $\exists x.\, p(d,x)$ This quantified sentence is F

$p(d,w) = 0$ $\forall x.(\neg p(d,x))$ this quantified sentence is T

$p(d,f) = 0$

$p(r,d) = 0$ $\exists x.(p(d,x) \vee p(r,x))$ This quantified sentence =

$p(r,r) = 0$ $\forall x.(\neg p(d,x) \wedge \neg p(r,x))$ This quantified sentence is T

$p(r,w) = 0$

$p(r,f) = 0$

9c

$p(w,d) = 1$

$p(w,r) = 1$

$p(w,w) = 1$

$p(w,f) = 1$

$p(f,d) = 0$

$p(f,r) = 1$

$p(f,w) = 0$

$p(f,f) = 1$

$\exists x.\ p(x,x)$ This quantified sentence is true $= T$

$\forall x.\ p(x,x)$ this quantified sentence is false. $= F$

$\forall x.(C(x) \to p(x,g))$ This quantified sentence is T

~~Then~~

$\forall x.\ (C(x) \to p(x,d))$ This quantified Sentence is F

$\exists x.(\ h(x) \to p(x,r))$ This quantified sentence is true! T

$\exists x.\ (\ h(x) \wedge p(x,r))$ This quantified Sentence is False.

$\forall x.(h(x) \to \neg p(x,y))$ This quantified sentence is True

$\forall x.\ (h(x) \vee c(x))$ This quantified sentence is true

$\forall y.\forall x.\ (p(x,y) \vee p(y,x))$ this quantified sentence is False

$\forall x.\exists y.(c(x) \to p(x,y))$ This quantified sentence is true T

$\forall x.\forall y.\ (c(x) \to p(x,y))$ this quantified sentence is False

Consider nested quantifiers 9d

$\forall x. \forall y. (c(x) \to p(x,y))$

$c(d) \to$ $p(d,d)$, $p(d,r)$, $p(d,w)$, $p(d,f)$ — all T

$c(w) \to$ $p(w,d)$, $p(w,r)$, $p(w,w)$, $p(w,f)$ — All T

$c(r) \to$ $p(r,d)$, $p(r,r)$, $p(r,w)$, $p(r,f)$ — all T

$c(f) \to$ $p(f,d)$ F, $p(f,r)$ T, $p(f,w)$ F, $p(f,f)$ T

So this quantified sentence is False

$\forall x. \exists y. (c(x) \to p(x,y))$ Now this quantified sentence is True

$\forall x. \exists y. (c(x) \land p(x,y))$

(9e)

Consider

$$\neg \forall x. \neg \phi \models\!\dashv \exists x. \phi$$

(1) $\exists x. p(x,x) \iff$ (2) $\neg \forall x. \neg p(x,x)$ (if this is a valid sentence)

(3) $\forall x. (\neg c(x) \to h(x)) \iff$ (4) $\forall x. (c(x) \vee h(x))$

This is $(\phi \to \psi) \iff (\neg \phi \vee \psi)$

(5) $\exists x. (h(x) \wedge c(x))$

If this is a valid sentence

6 $\exists x. h(x) \wedge \exists x. c(x)$

~~6~~ 7 $\forall x. \neg (p(d,x) \wedge p(r,x))$

8. $\forall x. (\neg p(d,x) \vee \neg p(r,x)$

9f

Axiomitization

For example in carnivore world

$$\forall x. \left(C(x) \rightarrow \exists y. P(x,y) \right)$$

This might be an axiom in our world

this is an axiom in carnivore world

for all animals, if the animal is a carnivore then there is at least one other animal that the original animal preys upon

A truth assignment satisfies a sentence with free variables iff it satisfies every instance of that sentence.

A truth assignment satisfies a set of sentences iff it satisfies every sentence in that set.

Evaluation:

1) To evaluate the universally quantified sentence, we check every instance within the scope
   the conjunction of all Instances within the scope of the sentence must be true for the sentence to be true.

2. To evaluate the existentially quantified sentence we only need one instance in the scope to be true
   the disjunction of all instances within the scope of the sentence

Assume we have

$p(a) = 1$ $\quad$ $q(a,a) = 1$

$p(b) = 0$ $\quad$ $q(a,b) = 0$

$q(b,a) = 1$

$q(b,b) = 0$

what is the truth value of

$\forall x.(p(x) \rightarrow q(x,x))$ under this assignment

we only have two instances of this sentence within our assignment namely

both return a true value

$$\left\{ \begin{array}{lll} p(a) \rightarrow q(a,a) & \text{which has values} & 1 \rightarrow 1 \quad \text{True} \\ p(b) \rightarrow q(b,b) & \text{which has values} & 0 \rightarrow 0 \quad \text{True} \end{array} \right.$$

So we say $\forall x.(p(x) \rightarrow q(x,x))$ is True

or $\forall x.(p(x) \rightarrow q(x,x)) \Rightarrow 1$

Do problems 6.2, 6.3, 6.4

Now consider nested quantifiers

φ $\forall x . \exists y . q(x,y)$ is True or False

For ϕ to be true, then every instance of the sentence must be true. So we have

True $\exists y . q(a,y) \rightarrow$ $q(a,a) \rightarrow 1$, $q(a,b) \rightarrow 0$

True $\exists y . q(b,y) \rightarrow$ $q(b,a) \rightarrow 1$, $q(b,b) \rightarrow 1$

Since each ~~intstance~~ instance is true then we say

$\phi = T$ where $\phi : \forall x . \exists y . q(x,y)$

Another way to state this is to say for both instances of the scope of the original universal sentence are true then the sentence is true

~~$\forall x . \exists y .$~~ β

$\forall x . \exists y . q(x,y) \rightarrow 1$

Is a true sentence

Assume we have a language with 2 unary relation contants p, q and two object constants a, b consider the logical sentences 13

Δ set of sentences

$$
\begin{cases}
p(a) \vee p(b) \\
\forall x.\,(p(x) \to q(x)) \\
\exists x.\,q(x)
\end{cases}
$$

also could b $q(a) \vee q(b)$

Build a truth table

	$p(a)$	$p(b)$	$q(a)$	$q(b)$	$p(a) \vee p(b)$	$\forall x.(p(x) \to q(x))$	$\exists x.\,q(x)$
	T	T	T	T	T	T	T
	T	T	T	F	T	F	T
	T	T	F	T	T	F	T
	T	T	F	F	T	F	F
	T	F	T	T	T	T	T
	T	F	T	F	T	T	T
7	T	F	F	T	T	F	T
8	T	F	F	F	T	F	F
	F	T	T	T	T	T	T
0	F	T	T	F	T	F	T
	F	T	F	T	T	T	T
2	F	T	F	F	T	F	F
3	F	F	T	T	F	T	T
4	F	F	T	F	F	T	T
5	F	F	F	T	F	T	T
6	F	F	F	F	F	T	F

This set of sentences is satisfiable since cases 1, 5, 6, 9, 11 all satisfy the set Δ of sentences

Example Sorority World

we have 4 object ~~[illegible]~~ constants in our world

abby = a   cody = c   and

bess = b   dana = d

we have 1 relational constant which is binary.

likes = $L(x,y)$

one possible state

	a	b	c	d
a	$\neg L(a,a)$	$\neg(L(a,b))$	✓ $L(a,c)$	$\neg L(a,d)$
b			✓	
c	✓	✓		✓
d			✓	

$\neg L(a,a)$, $\neg L(a,b)$ $L(a,c)$ $\neg L(a,d)$

$\neg L(b,a)$ $\neg L(b,b)$ $L(b,c)$

etc.

We can completely represent the world if we have complete information about all the states in the world.

But, this is not always possible

Sorority World (14a)

abby = a   cody = c   4 object constants
bess = b   dana = d

$l(x,y)$ relational constant

x likes y

write a relational logic sentence for

(1) bess likes cody or dana $\boxed{l(b,c) \vee l(b,d)}$ (1)

Abby likes everyone bess likes (2)

$l(b,y) \to l(a,y)$

$\boxed{\forall y.(l(b,y) \to l(a,y))}$ (2)

(3) Cody likes everyone who likes her

$\cancel{l(c,y)}$ $l(y,c) \to l(c,y)$

(3) $\forall y.(l(y,c) \to l(c,y))$

(4) Bess likes somebody who likes her.

$\exists x.(l(b,x) \wedge l(x,b))$ (4) This is correct

$\boxed{\exists x.(l(x,b) \to l(b,x))}$ this is wrong because the implication can be true even when $l(x,b) = F$ that is if there is never a person x : $l(x,b) = T$ then the statement is still true. So this existentially quantified implication does not guarantee the existence of someone who likes bess

⑤ Nobody likes herself (14b)

$\neg \exists x.\ l(x,x)$ or $\forall x.\ \neg l(x,x)$ ⑤

---

consider $\neg \forall x.\ \phi \models\!\dashv \exists x.\ \neg \phi$ Consider these logical Equivalences

it is not true that for all x, $\phi = T$ is the same as sayin there is at least one x such that $\phi = F$

$\forall x.\ \phi \models\!\dashv \neg \exists x.\ \neg \phi$

For all x, $\phi = T$ — There does not exist an x such that $\phi = F$

$\forall x.\ \neg \phi \models\!\dashv \neg \exists x.\ \phi$

For all x, $\phi = F$ — there does not exist an x for which $\phi = T$

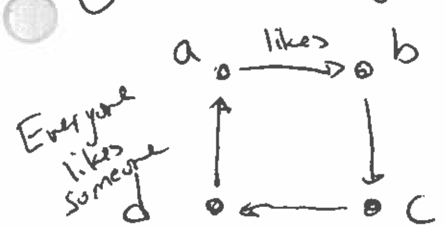
$\neg \forall x.\ \neg \phi \models\!\dashv \exists x.\ \phi$

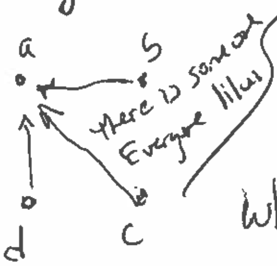
It is not true that for all x $\phi = F$ — there exists at least one x for which $\phi = T$

---

⑥ Everyone likes someone.

⑥ $\forall x.\ \exists y.\ l(x,y)$





which is not the same as

$\cancel{\forall x. \exists y. l(x,y)}$

$\exists x.\ \forall y.\ l(x,y)$

which says there is someone who likes everyone

14c

7. there is someone who everyone likes

7 $\exists x. \forall y. \; l(y,x)$

~~Start with a disjunction~~ Encoding Sentences

Example Bess likes Cody or Danai

$$L(b,c) \vee L(b,d)$$

Abby likes everyone Bess likes

Example ~~B~~ $L(b,y) \to L(a,y)$

if Bess likes y then Abby Likes y

$$\forall y. (L(b,y) \to L(a,y))$$

For all y, if bess likes y then abby likes y

Example Cody likes everyone who likes her.

$$L(x,c) \to L(c,x)$$

if someone Likes cody then cody likes her back

$$\forall x. (L(x,c) \to L(c,x))$$

Example Bess likes somebody who likes her.

NO If someone likes Bess then Bess likes ~~her back~~

$$L(y,b) \to L(b,y)$$

$$\exists y. (L(y,b) \to L(b,y))$$

$$\exists y. (L(b,y) \wedge L(y,b))$$

Example Nobody Likes herself

$\neg \exists x . L(x,x)$

Example Everybody likes somebody.

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

contrast this with

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

there is someone who Likes everyone.

Example there is someone who everyone likes

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

Relational Logic Properties

Sentences can still be

$$\underbrace{\text{Valid, contingent}}_{\text{satisfiable}}\ \text{or}\ \underbrace{\text{unsatisfiable}}_{\text{falsifiable}}$$

ground relational sentences that are valid

$$p(a) \lor \neg p(a)$$

$$p(a) \Longleftrightarrow \neg\neg p(a)$$

$$\neg(p(a) \land q(a,b)) \Longleftrightarrow (\neg p(a) \lor \neg q(a,b))$$

$$\neg(p(a) \lor q(a,b)) \Longleftrightarrow (\neg p(a) \land \neg q(a,b))$$

Not all relational sentences are ground. There are valid sentences in Relational Logic without analogue in Propositional Logic

Common Quantifier Reversal

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

$$\exists x. \exists y. q(x,y) \Longleftrightarrow \exists y. \exists x. q(x,y)$$

Existential Distribution

$$\exists y . \forall x . q(x,y) \Rightarrow \forall x . \exists y . q(x,y)$$

NOTE NOT Reverse

Negation Distribution

$$\neg \forall x . p(x) \Leftrightarrow \exists x . \neg p(x)$$

$$\neg \exists x . p(x) \Leftrightarrow \forall x . \neg p(x)$$

Logical Entailment

$\Delta \models \phi$ Iff every truth assignment that satisfies Δ also satisfies ϕ

$$p(a) \models p(a) \vee p(b)$$

$$p(a) \not\models p(a) \wedge p(b)$$

$$\exists y. \forall x. q(x,y) \models \forall x. \exists y. q(x,y)$$

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

Does $q(x,y)$ entail $q(y,x)$?

when every instance of x,y entails then a sentence with free variables entails.

So. we say $q(x,y) \models q(y,x)$

Iff $\forall x. \forall y. q(x,y) \models \forall x. \forall y. q(y,x)$

usually $\forall x . (p(x) \to q(x))$ but not $\exists x . (p(x) \to q(x))$

usually $\exists x . (p(x) \wedge q(x))$ but not $\boxed{\forall x . p(x) \wedge q(x)}$

very rare

m

p

case 1 $\forall x . (m(x) \wedge p(x))$ All objects are mushrooms and are poisonous

case 2 $\forall x . (m(x) \to p(x))$ For all x: if x is a mushroom then it is poisonous.

much more common

case 3 $\exists x . (m(x) \wedge p(x))$ There exists at least one x such that an object is both a mushroom and is poisonous

case 4 $\exists x . (m(x) \Rightarrow p(x))$

very rare

case 1 says all objects are both mushrooms and poisonous. this implies the entire universe is made up of mushrooms which are poisonous.

case 2 says all mushrooms are poisonous, but does not guarantee that any mushrooms exist.

case 3 says there is at least one object which is a poisonous mushroom but does not guarantee that all mushrooms are poisonous, just that at least one of them is

case 4 says there is at least one x such that if it is a mushroom then it is poisonous but does not guarantee there are any mushrooms

Exercise 8.8

Goal $\neg\exists x.\neg p(x)$

suggests

$$\frac{\exists x.\neg p(x) \Rightarrow \psi \qquad \exists x.\neg p(x) \Rightarrow \neg\psi}{\neg\exists x.\neg p(x)}\ NI$$

where $\psi = \forall x.p(x)$ since we are given that

1. $\forall x.p(x)$ premise
2. $\quad \exists x.\neg p(x)$ Assume
3. $\quad \forall x.p(x)$ Reiterate(1)
4. $\exists x.\neg p(x) \Rightarrow \forall x.p(x)$ II(2,3)
5. $\quad \exists x.\neg p(x)$ Assume
6. $\quad\quad \neg p(x)$ Assume
7. $\quad\quad\quad \forall x.p(x)$ ~~Reiterate~~(1)
8. $\quad\quad\quad p(x)$ UE(7
9. $\quad\quad \forall x.p(x) \Rightarrow p(x)$ II(7,8)
10. $\quad\quad\quad \forall x.p(x)$ Assume
11. $\quad\quad\quad \neg p(x)$ reit.(6)
12. $\quad\quad \forall x.p(x) \Rightarrow \neg p(x)$ II(10,11)
13. $\quad\quad \neg\forall x.p(x)$ NI(9,12)
14. $\quad \neg p(x) \Rightarrow \neg\forall x.p(x)$ II(6,13)
15. $\quad \forall x.(\neg p(x) \Rightarrow \neg\forall x.p(x))$ UI(14)
16. $\quad \neg\forall x.p(x)$ EE(5,15)
17. $\exists x.\neg p(x) \Rightarrow \neg\forall x.p(x)$ II(5,16)
18. $\neg\exists x.\neg p(x)$ NI(4,17)

This was the easy part now we need to show

$\exists x.\neg p(x) \Rightarrow \neg\forall x.p(x)$

$\neg p(x) \Rightarrow \neg\forall x.p(x)$

$\forall x.(\neg p(x) \Rightarrow \neg\forall x.p(x))$

Exercise 8.7

1. $\exists x.\neg p(x)$ premise goal $\neg\forall x.p(x)$

2. $\forall x.p(x)$ Assume
3. $\exists x.\neg p(x)$ Reiteration(1)
4. $\forall x.p(x) \rightarrow \exists x.\neg p(x)$ II(2,3)
5. $\forall x.p(x)$ Assume
6. $p(x)$ UE.(5)
7. $\exists x.\neg p(x)$ reiteration (1)

2. $\neg p(x)$ assume
3. $\forall x.p(x)$ assume
4. $\neg p(x)$ reiteration(2)
5. $\forall x.p(x) \rightarrow \neg p(x)$ II(3,4)
6. $\forall x.p(x)$ Assume
7. $p(x)$ UE(6)
8. $\forall x.p(x) \rightarrow p(x)$ II(6,7)
9. $\neg\forall x.p(x)$ NI(5,8)
10. $\neg p(x) \rightarrow \neg\forall x.p(x)$ II(2,9)
11. $\forall x.(\neg p(x) \rightarrow \neg\forall x.p(x))$ UI(10)
12. $\neg\forall x.p(x)$ EE(1,11)

Consider

$$\frac{\exists x.\phi \qquad \forall x.\phi \rightarrow \psi}{\psi}$$

Consider trying

$$\frac{\phi \rightarrow \psi \qquad \phi \rightarrow \neg\psi}{\neg\phi}$$

$\phi = \forall x.p(x)$

$\psi = \exists x.\neg p(x)$

Consider

$\phi = \neg p(x)$

$\psi = \neg\forall x.p(x)$

now we have

$$\frac{\exists x.\phi \qquad \forall x.(\phi \rightarrow \psi)}{\psi}$$

CHAPTER 7

Relational Analysis

7.1 Introduction

In Relational Logic, it is possible to analyze the properties of sentences in much the same way as in Propositional Logic. Given a sentence, we can determine its validity, satisfiability, and so forth by looking at possible truth assignments. And we can confirm logical entailment or logical equivalence of sentences by comparing the truth assignments that satisfy them and those that don't.

The main problem in doing this sort of analysis for Relational Logic is that the number of possibilities is even larger than in Propositional Logic. For a language with n object constants and m relation constants of arity k, the Herbrand base has $m*n^k$ elements; and consequently, there are 2^{m*nk} possible truth assignments to consider. If we have 10 objects and 5 relation constants of arity 2, this means 2^{500} possibilities.

Fortunately, as with Propositional Logic, there are some shortcuts that allow us to analyze sentences in Relational Logic without examining all of these possibilities. In this chapter, we start with the truth table method and then look at some of these more efficient methods.

7.2 Truth Tables

As in Propositional Logic, it is in principle possible to build a truth table for any set of sentences in Relational Logic. This truth table can then be used to determine validity, satisfiability, and so forth or to determine logical entailment and logical equivalence.

As an example, let us assume we have a language with just two object constants a and b and two relation constants p and q. Now consider the sentences shown below, and assume we want to know whether these sentences logically entail $\exists x.q(x)$.

$$p(a) \vee p(b)$$
$$\forall x.(p(x) \Rightarrow q(x))$$

A truth table for this problem is shown below. Each of the first four columns represents one of the elements of the Herbrand base for this language. The two middle columns represent our premises, and the final column represents the conclusion.

$p(a)$	$p(b)$	$q(a)$	$q(b)$	$p(a) \lor p(b)$	$\forall x.(p(x) \Rightarrow q(x))$	$\exists x.q(x)$
1	1	1	1	1	1	1
1	1	1	0	1	0	1
1	1	0	1	1	0	1
1	1	0	0	1	0	0
1	0	1	1	1	1	1
1	0	1	0	1	1	1
1	0	0	1	1	0	1
1	0	0	0	1	0	0
0	1	1	1	1	1	1
0	1	1	0	1	0	1
0	1	0	1	1	1	1
0	1	0	0	1	0	0
0	0	1	1	0	1	1
0	0	1	0	0	1	1
0	0	0	1	0	1	1
0	0	0	0	0	1	0

Looking at the table, we see that there are 12 truth assignments that make the first premise true and nine that make the second premise true and five that make them both true (rows 1, 5, 6, 9, and 11). Note that every truth assignment that makes both premises true also makes the conclusion true. Hence, the premises logically entail the conclusion.

7.3. Semantic Trees

While the Truth Table method works in principle, it is impractical when the tables get very large. As with Propositional Logic, we can sometimes avoid generating such tables by incrementally constructing the corresponding "semantic trees". By interleaving unit propagation and simplification with tree generation, we can often prune away unrewarding subtrees before they are generated and thereby reduce the size of the trees.

7.4. Boolean Models

Truth tables and semantic trees are good ways of explicitly representing multiple models for a set of sentences. In some cases, there is just one model.

In this approach, we write out an empty table for each relation and then fill in values based on the constraints of the problem. For example, for any unit constraint, we can immediately enter the corresponding truth value in the appropriate box. Given these partial assignments, we then simplify the constraints (as in the semantic trees method), possibly leading to new unit constraints. We continue until there are no more unit constraints.

As an example, consider the Sorority problem introduced in Chapter 1. We are given the constraints shown below, and we want to know whether Dana likes everyone that Bess likes. In other words, we want to confirm that, in every model that satisfies these sentences, Dana likes everyone that Bess likes.

Dana likes Cody.
Abby does not like Dana.
Dana does not like Abby.
Abby likes everyone that Bess likes.
Bess likes Cody or Dana.
Abby and Dana both dislike Bess.
Cody likes everyone who likes her.
Nobody likes herself.

In this particular case, it turns out that there is just one model that satisfies all of these sentences. The first step in creating this model is to create an empty table for the *likes* relation.

	Abby	**Bess**	**Cody**	**Dana**
Abby				
Bess				
Cody				
Dana				

The data we are given has three units - the fact that Dana likes Cody and the facts that Abby does not like Dana and Dana does not like Abby. Using this information we can refine our model by putting a one into the third box in the fourth row and putting zeros in the fourth box of the first row and the first box of the fourth row.

	Abby	Bess	Cody	Dana
Abby				0
Bess				
Cody				
Dana	0		1	

Now, we know that Abby likes everyone that Bess likes. If Bess likes Dana, then we could conclude that Abby likes Dana as well. We already know that Abby does not like Dana, so Bess must not like Dana either.

	Abby	Bess	Cody	Dana
Abby				0
Bess				0
Cody				
Dana	0		1	

At the same time, we know that Bess likes Cody or Dana. Since Bess does not like Dana, she must like Cody. Once again using the fact that Abby likes everyone whom Bess likes, we know that Abby also likes Cody.

	Abby	Bess	Cody	Dana
Abby			1	0
Bess			1	0
Cody				
Dana	0		1	

Abby and Dana both dislike Bess. Using this fact we can add 0s to the first and last cells of the second column.

	Abby	Bess	Cody	Dana
Abby		0	1	0
Bess			1	0
Cody				
Dana	0	0	1	

On the other hand, Cody likes everyone who likes her. This allows us to put a 1 in every column of the third row where there is a 1 in the corresponding rows of the third column.

	Abby	Bess	Cody	Dana
Abby		0	1	0
Bess			1	0
Cody	1	1		1
Dana	0	0	1	

Since nobody likes herself, we can put a 0 in each cell on the diagonal.

	Abby	Bess	Cody	Dana
Abby	0	0	1	0
Bess		0	1	0
Cody	1	1	0	1
Dana	0	0	1	0

Finally, using the fact that Abby likes everyone that Bess likes, we conclude that Bess does not like Abby. (If she did then Abby would like herself, and we know that that is false.)

	Abby	Bess	Cody	Dana
Abby	0	0	1	0
Bess	0	0	1	0
Cody	1	1	0	1
Dana	0	0	1	0

At this point, we have a complete model, and we can check our conclusion to see that this model satisfies the desired conclusion. In this case, it is easy to see that Dana indeed does like everyone that Bess likes.

We motivated this method by talking about cases where the given sentences have a unique model, as in this case. However, the method can also be of value even when there are multiple possible models. For example, if we had left out the belief that Cody likes everyone who likes her, we would still have eight models (corresponding to the eight possible combinations of feelings Cody has for Abby, Bess, and Dana). Yet, even with this ambiguity, it would be possible to determine whether Dana likes everyone Bess likes using just the portion of the table already filled in.

7.5 Non-Boolean Models

As defined in Chapter 6, a model in Relational Logic is an assignment of truth values to the ground atoms of our language. We treat each ground atom in our language as a variable and assign it a single truth value (1 or 0). In general, this is a good way to proceed. However, we can sometimes do better.

Consider, for example, a theory with four object constants and two unary relation constants. In this case, there would be eight elements in the Herbrand base and 2^8 (256) possible truth assignments. Now, suppose we had the constraint that each relation is true of at most a single object. Most of these assignments would not satisfy the single value constraint and thus considering them is wasteful.

Luckily, in cases like this, there is a representation for truth assignments that allows us to eliminate such possibilities and thereby save work. Rather than treating each *ground atom* as a separate variable with its own Boolean value, we can think of each *relation* as a variable with 4 possible values. In order to analyze sentences in such a theory, we would need to consider only 4^2 (16) possibilities.

Even if we search the entire space of assignments, this a significant saving over the pure truth table method. Moreover, we can combine this representation with the techniques described earlier to find assignments for these non-Boolean variables in an even more efficient manner.

The game of Sukoshi illustrates this technique and its benefits. (Sukoshi is similar to Sudoku, but it is smaller and simpler.) A typical Sukoshi puzzle is played on a 4x4 board. In a typical instance of Sukoshi, several of the squares are already filled, as in the example below. The goal of the game is to place the numerals 1 through 4 in the remaining squares of the board in such a way that no numeral is repeated in any row or column.

	4		1
2			
			3
		4	

We can formalize the rules of this puzzle in the language of Logic. Once we have done that, we can use the techniques described here to find a solution.

In our formalization, we use the expression *cell*(1,2,3) to express the fact that the cell in the first row and the second column contains the numeral 3. For example, we can describe the initial board shown above with the following sentences.

cell(1,2,4)
cell(1,4,1)
cell(2,1,2)
cell(3,4,3)
cell(4,3,4)

We use the expression *same*(*x*,*y*) to say that *x* is the same as *y*. We can axiomatize *same* by simply stating when it is true and where it is false. An abbreviated axiomatization is shown below.

same(1,1)	¬*same*(2,1)	¬*same*(3,1)	¬*same*(4,1)
¬*same*(1,2)	*same*(2,2)	¬*same*(3,2)	¬*same*(4,2)

$$\neg same(1,3) \quad \neg same(2,3) \quad same(3,3) \quad \neg same(4,3)$$
$$\neg same(1,4) \quad \neg same(2,4) \quad \neg same(3,4) \quad same(3,4)$$

Using this vocabulary, we can write the rules defining Sukoshi as shown below. The first sentence expresses the constraint that two cells in that same row can contain the same value. The second sentence expresses the constraint that two cells in that same column can contain the same value. The third constraint expresses the fact that every cell must contain at least one value.

$$\forall x.\forall y.\forall z.\forall w.(cell(x,y,w) \wedge cell(x,z,w) \Rightarrow same(y,z))$$
$$\forall x.\forall y.\forall z.\forall w.(cell(x,z,w) \wedge cell(y,z,w) \Rightarrow same(x,y))$$
$$\forall x.\forall y.\exists w.cell(x,z,w)$$

typo

As a first step in solving this problem, we start by focussing on the fourth column, since two of the cells in that column are already filled. We know that there must be a 4 in one of the cells. It cannot be the first, since that cell contains a 1, and it cannot be the third since that cell contains a 3. It also cannot be the fourth, since there is already a 4 in the third cell of the fourth row. By process of elimination, the 4 must go in the fourth cell of the second row, leading to the board shown below.

	4		1
2			4
			3
		4	

At this point, there is a four in every row and every column except for the first column in the third row. So, we can safely place a four in that cell.

	4		1
2			4
4			3
		4	

Since there is just one empty cell in the fourth column, we know it must be filled with the single remaining value, viz. 2. After adding this value, we have the following board.

	4		1
2			4
4			3
		4	2

Now, let's turn our attention to the first column. We know that there must be a 1 in one of the cells. It cannot be the first, since there is already a 1 in that row, and it cannot be the second or third since those cell already contain values. Consequently, the 1 must go in the first cell of the fourth row.

	4		1
2			4
4			3
1		4	2

Once again, we have a column with all but one cell filled. Column 1 has a 2 in the second cell, a 4 in the third, and a 1 in the fourth. So, we can place a 3 in the first cell of that column.

3	4		1
2			4
4			3
1		4	2

At this point, we can fill in the single empty cell in the first row, leading to the following board.

3	4	2	1

2			4
4			3
1		4	2

And we can fill in the single empty cell in the fourth row as well.

3	4	2	1
2			4
4			3
1	3	4	2

Now, let's consider the second column. We cannot put a 2 in the second cell, since there is already a 2 in that row. Since the first and last cells are already full, the only option is to put the 2 into the third cell.

3	4	2	1
2			4
4	2		3
1	3	4	2

Finishing off the third row leads to the board below.

3	4	2	1
2			4
4	2	1	3
1	3	4	2

Finishing off the third column leads to the following board.

3	4	2	1
2		3	4
4	2	1	3
1	3	4	2

Finally, we can place a 1 in the second cell of the second row. And, with that, the board is full. We have a distinct numeral in every row and every column, as required by the rules.

3	4	2	1
2	1	3	4
4	2	1	3
1	3	4	2

Given the initial assignment in this case, it is fairly easy to find a complete assignment that satisfies the Sukoshi constraints. For other initial assignments, solving the problem is more difficult. However, the techniques described here still work to cut down on the amount of work necessary. In fact, virtually all Sukoshi puzzles can be solved using these techniques without any form of trial and error.

Exercises

Exercise 7.1: Mr. Red, Mr. White, and Mr. Blue meet for lunch. Each is wearing a red shirt, a white shirt, or a blue shirt. No one is wearing more than one color, and no two are wearing the same color. Mr. Blue tells one of his companions, "Did you notice we are all wearing shirts with different color from our names?", and the other man, who is wearing a white shirt, says, "Wow, that's right!" Use the Boolean model technique to figure out who is wearing what color shirt.

Exercise 7.2: Amy, Bob, Coe, and Dan are traveling to different places. One goes by train, one by car, one by plane, and one by ship. Amy hates flying. Bob rented his vehicle. Coe tends to get seasick. And Dan loves trains. Use the Boolean models method to figure out which person, used which mode of transportation.

Exercise 7.3: Sudoku is a puzzle consisting of a 9x9 board divided into nine 3x3 subboards. In a typical puzzle, several of the squares are already filled, as in the example shown below. The goal of the puzzle is to place the numerals 1 through 9 into the remaining squares of the board in such a way that no numeral is repeated in any row or column or 3x3 subboard.

5	8	6					1	2
				5	2	8	6	
2	4		8	1				3
			5		3		9	
				8	1	2	4	
4		5	6			7	3	8
	5		2	3			8	1
7					8			
3	6				5			

Use the techniques described in the Chapter to solve this puzzle.

CHAPTER 8

Relational Proofs

8.1 Introduction

As with Propositional Logic, we can demonstrate logical entailment in Relational Logic by writing proofs. As with Propositional Logic, it is possible to show that a set of Relational Logic premises logically entails a Relational Logic conclusion if and only if there is a finite proof of the conclusion from the premises. Moreover, it is possible to find such proofs in a finite time.

In this chapter, we start by extending the Fitch system from Propositional Logic to Relational Logic. We then illustrate the system with a few examples. Finally, we talk about soundness and completeness.

8.2 Proofs

Formal proofs in Relational Logic are analogous to formal proofs in Propositional Logic. The major difference is that there are additional mechanisms to deal with quantified sentences.

The Fitch system for Relational Logic is an extension of the Fitch system for Propositional Logic. In addition to the ten logical rules of inference, there are four ordinary rules of inference for quantified sentences and one additional rule for finite languages. Let's look at each of these in turn. (If you're like me, the prospect of going through a discussion of so many rules of inference sounds a little repetitive and boring. However, it is not so bad. Each of the rules has its own quirks and idiosyncrasies, its own personality. In fact, a couple of the rules suffer from a distinct excess of personality. If we are to use the rules correctly, we need to understand these idiosyncrasies.)

Universal Introduction (UI) allows us to reason from arbitrary sentences to universally quantified versions of those sentences.

Universal Introduction

φ

$\forall \nu.\varphi$

where ν does not occur free in both φ and an active assumption

Typically, UI is used on sentences with free variables to make their quantification explicit. For example, if we have the sentence *hates(jane,y)*, then, we can infer $\forall y.hates(jane,y)$.

Note that we can also apply the rule to sentences that do not contain the variable that is quantified in the conclusion. For example, from the sentence *hates(jane,jill)*, we can infer $\forall x.hates(jane,jill)$. And, from the sentence *hates(jane,y)*, we can infer $\forall x.hates(jane,y)$. These are not particularly sensible conclusions. However, the results are correct, and the deduction of such results is necessary to ensure that our proof system is complete.

There is one important restriction on the use of Universal Introduction. If the variable being quantified appears in the sentence being quantified, it must not appear free in any *active assumption*, i.e. an assumption in the current subproof or any superproof of that subproof. For example, if there is a subproof with assumption $p(x)$ and from that we have managed to derive $q(x)$, then we cannot just write $\forall x.q(x)$.

If we want to quantify a sentence in this situation, we must first use Implication Introduction to discharge the assumption and then we can apply Universal Introduction. For example, in the case just described, we can first apply Implication Introduction to derive the result $(p(x) \Rightarrow q(x))$ in the parent of the subproof containing our assumption, and we can then apply Universal Introduction to derive $\forall x.(p(x) \Rightarrow q(x))$.

Universal Elimination (UE) allows us to reason from the general to the particular. It states that, whenever we believe a universally quantified sentence, we can infer a version of the target of that sentence in which the universally quantified variable is replaced by an appropriate term.

Universal Elimination

$$\frac{\forall v.\varphi[v]}{\varphi[\tau]}$$

where τ is substitutable for v in φ

For example, consider the sentence $\forall y.hates(jane,y)$. From this premise, we can infer that Jane hates Jill, i.e. *hates(jane,jill)*. We also can infer that Jane hates her mother, i.e. *hates(jane,mother(jane))*. We can even infer than Jane hates herself, i.e. *hates(jane,jane)*.

In addition, we can use Universal Elimination to create conclusions with free variables. For example, from $\forall x.hates(jane,x)$, we can infer *hates(jane,x)* or, equivalently, *hates(jane,y)*.

In using Universal Elimination, we have to be careful to avoid conflicts with other variables and quantifiers in the quantified sentence. This is the reason for the constraint on the replacement term.

As an example of what can go wrong without this constraint, consider the sentence $\forall x.\exists y.hates(x,y)$, i.e. everybody hates somebody. From this sentence, it makes sense to infer $\exists y.hates(jane,y)$, i.e. Jane hates somebody. However, we do not want to infer $\exists y.hates(y,y)$; i.e., there is someone who hates herself.

We can avoid this problem by obeying the restriction on the Universal Elimination rule. We say that a term τ is *free* for a variable ν in a sentence φ if and only if no free occurrence of ν occurs within the scope of a quantifier of some variable in τ. For example, the term x is free for y in $\exists z.hates(y,z)$. However, the term z is not free for y, since y is being replaced by z and y occurs within the scope of a quantifier of z. Thus, we cannot substitute z for y in this sentence, and we avoid the problem we have just described.

Existential Introduction (EI) is easy. If we believe a sentence involving a ground term τ, then we can infer an existentially quantified sentence in which one, some, or all occurrences of τ have been replaced by the existentially quantified variable.

Existential Introduction

$$\frac{\varphi[\tau]}{\exists \nu.\varphi[\nu]}$$

For example, from the sentence *hates(jill,jill)*, we can infer that there is someone who hates herself, i.e. $\exists x.hates(x,x)$. We can also infer that there is someone Jill hates, i.e. $\exists x.hates(jill,x)$, and there is someone who hates Jill, i.e. $\exists y.hates(x,jill)$. And, by two applications of Existential Introduction, we can infer that someone hates someone, i.e. $\exists x.\exists y.hates(x,y)$.

Note that, in Existential Introduction, it is important to avoid variables that appear in the sentence being quantified. Without this restriction, starting from $\exists x.hates(jane,x)$, we might deduce $\exists x.\exists x.hates(x,x)$. It is an odd sentence since it contains nested quantifiers of the same variable. However, it is a legal sentence, and it states that there is someone who hates himself, which does not follow from the premise in this case.

Existential Elimination (EE). Suppose we have an existentially quantified sentence with target φ; and suppose we have a universally quantified implication in which the antecedent is the same as the scope of our existentially quantified sentence and the conclusion does not contain any occurrences of the quantified variable. Then, we can use Existential Elimination to infer the consequent.

Existential Elimination

$\exists \nu.\varphi[\nu]$

$$\frac{\forall v.(\varphi[v] \Rightarrow \psi)}{\psi}$$

where v does not occur free in ψ

For example, if we have the sentence $\forall x.(hates(jane,x) \Rightarrow \neg nice(jane))$ and we have the sentence $\exists x.hates(jane,x)$, then we can conclude $\neg nice(jane)$).

It is interesting to note that Existential Elimination is analogous to Or Elimination. This is as it should, as an existential sentence is effectively a disjunction. Recall that, in Or Elimination, we start with a disjunction with *n* disjuncts and *n* implications, one for each of the disjuncts and produce as conclusion the consequent shared by all of the implications. An existential sentence (like the first premise in any instance of Existential Elimination) is effectively a disjunction over the set of all ground terms; and a universal implication (like the second premise in any instance of Existential Elimination) is effectively a set of implications, one for each ground term in the language. The conclusion of Existential Elimination is the common consequent of these implications, just as in Or Elimination.

Finally, for languages with finite Herbrand bases, we have the *Domain Closure* (DC) rule. For a language with object constants $\sigma_1, \ldots, \sigma_n$, the rule is written as shown below. If we believe a schema is true for every instance, then we can infer a universally quantified version of that schema.

Domain Closure

$$\frac{\begin{array}{l}\varphi[\sigma_1] \\ \ldots \\ \varphi[\sigma_n]\end{array}}{\forall v.\varphi[v]}$$

For example, in a language with four object constants *a* and *b* and *c* and *d*, we can derive the conclusion $\forall x.\varphi[x]$ whenever we have $\varphi[a]$ and $\varphi[b]$ and $\varphi[c]$ and $\varphi[d]$.

Why restrict DC to languages with finitely many ground terms? Why not use domain closure rules for languages with infinitely many ground terms as well? It would be good if we could, but this would require rules of infinite length, and we do not allow infinitely large sentences in our language. We can get the effect of such sentences through the use of *induction*, which is discussed in a later chapter.

As in Chapter 4, we define a *structured proof* of a conclusion from a set of premises to be a sequence of (possibly nested) sentences terminating in an occurrence of the conclusion at the *top level* of the proof. Each step in the proof must be either (1) a premise (at the top level) or an assumption (other than at the top level) or (2) the result of applying an ordinary or structured rule of inference to earlier items in the sequence (subject to the constraints given above and in Chapter 3).

8.3 Example

As an illustration of these concepts, consider the following problem. Suppose we believe that everybody loves somebody. And suppose we believe that everyone loves a lover. Our job is to prove that Jack loves Jill.

First, we need to formalize our premises. Everybody loves somebody. For all y, there exists a $z$such that $loves(y,z)$.

$$\forall y.\exists z.loves(y,z)$$

Everybody loves a lover. If a person is a lover, then everyone loves him. If a person loves another person, then everyone loves him. For all x and for all y and for all z, $loves(y,z)$ implies $loves(x,y)$.

$$\forall x.\forall y.\forall z.(loves(y,z) \Rightarrow loves(x,y))$$

Our goal is to show that Jack loves Jill. In other words, starting with the preceding sentences, we want to derive the following sentence.

$$loves(jack,jill)$$

A proof of this result is shown below. Our premises appear on lines 1 and 2. The sentence on line 3 is the result of applying Universal Elimination to the sentence on line 1, substituting *jill* for y. The sentence on line 4 is the result of applying Universal Elimination to the sentence on line 2, substituting *jack* for x. The sentence on line 5 is the result of applying Universal Elimination to the sentence on line 4, substituting *jill* for y. Finally, we apply Existential Elimination to produce our conclusion on line 6.

1.	$\forall y.\exists z.loves(y,z)$	Premise
2.	$\forall x.\forall y.\forall z.(loves(y,z) \Rightarrow loves(x,y))$	Premise
3.	$\exists z.loves(jill,z)$	UE: 1
4.	$\forall y.\forall z.(loves(y,z) \Rightarrow loves(jack,y))$	UE: 2
5.	$\forall z.(loves(jill,z) \Rightarrow loves(jack,jill))$	UE: 4

6. *loves(jack,jill)* EE: 3, 5

Now, let's consider a slightly more interesting version of this problem. We start with the same premises. However, our goal now is to prove the somewhat stronger result that everyone loves everyone. For all x and for all y, x loves y.

$$\forall x.\forall y.loves(x,y)$$

The proof shown below is analogous to the proof above. The only difference is that we have free variables in place of object constants at various points in the proof. Once again, our premises appear on lines 1 and 2. Once again, we use Universal Elimination to produce the result on line 3; but this time the result contains a free variable. We get the results on lines 4 and 5 by successive application of Universal Elimination to the sentence on line 2. We deduce the result on line 6 using Existential Elimination. Finally, we use two applications of Universal Introduction to generalize our result and produce the desired conclusion.

1.	$\forall y.\exists z.loves(y,z)$	Premise
2.	$\forall x.\forall y.\forall z.(loves(y,z) \Rightarrow loves(x,y))$	Premise
3.	$\exists z.loves(y,z)$	UE: 1
4.	$\forall y.\forall z.(loves(y,z) \Rightarrow loves(x,y))$	UE: 2
5.	$\forall z.(loves(y,z) \Rightarrow loves(x,y))$	UE: 4
6.	$loves(x,y)$	EE: 3, 5
7.	$\forall y.loves(x,y)$	UI: 6
8.	$\forall x.\forall y.loves(x,y)$	UI: 7

This second example illustrates the power of free variables. We can manipulate them as though we are talking about specific individuals (though each one could be any object); and, when we are done, we can universalize them to derive universally quantified conclusions.

h(harry)

r(ralph)

8.4 Example

As another illustration of Relational Logic proofs, consider the following problem. We know that horses are faster than dogs and that there is a greyhound that is faster than every rabbit. We know that Harry is a horse and that Ralph is a rabbit. Our job is to derive the fact that Harry is faster than Ralph.

horses faster than dogs $\forall x.\forall y.(h(x) \wedge d(y)) \rightarrow f(x,y)$

$\exists y.(g(y) \wedge \forall z.(r(z) \rightarrow f(y,z))$

$\forall y.(g(y) \rightarrow d(y))$ All greyhounds are dogs

$\forall x.\forall y.\forall z.(f(x,y) \wedge f(y,z) \rightarrow f(x,z))$

First, we need to formalize our premises. The relevant sentences follow. Note that we have added two facts about the world not stated explicitly in the problem: that greyhounds are dogs and that our *faster than* relationship is transitive.

$$\forall x.\forall y.(h(x) \land d(y) \Rightarrow f(x,y))$$
$$\exists y.(g(y) \land \forall z.(r(z) \Rightarrow f(y,z)))$$
$$\forall y.(g(y) \Rightarrow d(y))$$
$$\forall x.\forall y.\forall z.(f(x,y) \land f(y,z) \Rightarrow f(x,z))$$
$$h(harry)$$
$$r(ralph)$$

Our goal is to show that Harry is faster than Ralph. In other words, starting with the preceding sentences, we want to derive the following sentence.

$$f(harry,ralph)$$

The derivation of this conclusion goes as shown below. The first six lines correspond to the premises just formalized. On line 7, we start a subproof with an assumption corresponding to the scope of the existential on line 2, with the idea of using Existential Elimination later on in the proof. Lines 8 and 9 come from And Elimination. Line 10 is the result of applying Universal Elimination to the sentence on line 9. On line 11, we use Implication Elimination to infer that y is faster than Ralph. Next, we instantiate the sentence about greyhounds and dogs and infer that y is a dog. Then, we instantiate the sentence about horses and dogs; we use And Introduction to form a conjunction matching the antecedent of this instantiated sentence; and we infer that Harry is faster than y. We then instantiate the transitivity sentence, again form the necessary conjunction, and infer the desired conclusion. Finally, we use Implication Introduction to discharge our subproof; we use Universal Introduction to universalize the results; and we use Existential Elimination to produce our desired conclusion.

1.	$\forall x.\forall y.(h(x) \land d(y) \Rightarrow f(x,y))$	Premise
2.	$\exists y.(g(y) \land \forall z.(r(z) \Rightarrow f(y,z)))$	Premise
3.	$\forall y.(g(y) \Rightarrow d(y))$	Premise
4.	$\forall x.\forall y.\forall z.(f(x,y) \land f(y,z) \Rightarrow f(x,z))$	Premise
5.	$h(harry)$	Premise
6.	$r(ralph)$	Premise
7.	$g(y) \land \forall z.(r(z) \Rightarrow f(y,z))$	Assumption
8.	$g(y)$	AE: 7
9.	$\forall z.(r(z) \Rightarrow f(y,z))$	AE: 7

10. $r(ralph) \Rightarrow f(y,ralph)$ UE: 9
11. $f(y,ralph)$ IE: 10, 6
12. $g(y) \Rightarrow d(y)$ UE: 3
13. $d(y)$ IE: 12, 8
14. $\forall y.(h(harry) \land d(y) \Rightarrow f(harry,y))$ UE: 1
15. $h(harry) \land d(y) \Rightarrow f(harry,y)$ UE: 14
16. $h(harry) \land d(y)$ AI: 5, 13
17. $f(harry,y)$ IE: 15, 16
18. $\forall y.\forall z.(f(harry,y) \land f(y,z) \Rightarrow f(harry,z))$ UE: 4
19. $\forall z.(f(harry,y) \land f(y,z) \Rightarrow f(harry,z))$ UE: 18
20. $f(harry,y) \land f(y,ralph) \Rightarrow f(harry,ralph)$ UE: 19
21. $f(harry,y) \land f(y,ralph)$ AI: 17, 11
22. $f(harry,ralph)$ IE: 20, 21
23. $g(y) \land \forall z.(r(z) \Rightarrow f(y,z)) \Rightarrow f(harry,ralph)$ II: 7, 22
24. $\forall y.(g(y) \land \forall z.(r(z) \Rightarrow f(y,z)) \Rightarrow f(harry,ralph))$UI: 23
25. $f(harry,ralph)$ EE: 2, 24

This derivation is somewhat lengthy, but it is completely mechanical. Each conclusion follows from previous conclusions by a mechanical application of a rule of inference. On the other hand, in producing this derivation, we rejected numerous alternative inferences. Making these choices intelligently is one of the key problems in the process of inference.

8.5 Example

In this section, we use our proof system to prove some basic results involving quantifiers.

Given $\forall x.\forall y.(p(x,y) \Rightarrow q(x))$, we know that $\forall x.(\exists y.p(x,y) \Rightarrow q(x))$. In general, given a universally quantified implication, it is okay to drop a universal quantifier of a variable outside the implication and apply an existential quantifier of that variable to the antecedent of the implication, provided that the variable does not occur in the consequent of the implication.

Our proof is shown below. As usual, we start with our premise. We start a subproof with an existential sentence as assumption. Then, we use Universal Elimination to strip away the outer quantifier from the premise. This allows us to derive $q(x)$ using Existential Elimination. Finally, we create an implication with Implication Introduction, and we generalize using Universal Introduction.

1. $\forall x.\forall y.(p(x,y) \Rightarrow q(x))$ Premise

2. $\exists y.p(x,y)$ Assumption

3. $\forall y.(p(x,y) \Rightarrow q(x))$ UE: 1

4. $q(x)$ EE: 2, 3

5. $\exists y.p(x,y) \Rightarrow q(x)$ II: 4

6. $\forall x.(\exists y.p(x,y) \Rightarrow q(x))$ UI: 5

The relationship holds the other way around as well. Given $\forall x.(\exists y.p(x,y) \Rightarrow q(x))$, we know that $\forall x.\forall y.(p(x,y) \Rightarrow q(x))$. We can convert an existential quantifier in the antecedent of an implication into a universal quantifier outside the implication.

Our proof is shown below. As usual, we start with our premise. We start a subproof by making an assumption. Then we turn the assumption into an existential sentence to match the antecedent of the premise. We use Universal Implication to strip away the quantifier in the premise to expose the implication. Then, we apply Implication Elimination to derive $q(x)$. Finally, we create an implication with Implication Introduction, and we generalize using two applications of Universal Introduction.

1. $\forall x.(\exists y.p(x,y) \Rightarrow q(x))$ Premise

2. $p(x,y)$ Assumption

3. $\exists y.p(x,y)$ EI: 2

4. $\exists y.p(x,y) \Rightarrow q(x)$ UE: 1

5. $q(x)$ IE: 4, 3

6. $p(x,y) \Rightarrow q(x)$ II: 5

7. $\forall x.\forall y.(p(x,y) \Rightarrow q(x))$ 2 x UI: 6

Recap

A Fitch system for Relational Logic can be obtained by extending the Fitch system for Propositional Logic with four additional rules to deal with quantifiers. The *Universal Introduction*rule allows us to reason from arbitrary sentences to universally quantified versions of those sentences. The *Universal Elimination* rule allows us to reason from a universally quantified sentence to a version of the target of that sentence in which the universally quantified variable is replaced by an appropriate term. The *Existential Introduction* rule allows us to reason from a sentence involving a term τ to an existentially quantified sentence in which one, some, or all occurrences of τ have been replaced by the existentially quantified variable. Finally, the *Existential Elimination* rule allows us to reason from an existentially quantified sentence $\exists v.\varphi[v]$ and a universally quantified implication $\forall v.(\varphi[v] \Rightarrow \psi)$ to the consequent ψ, under the condition that v does not occur in ψ.

Exercises

Exercise 8.1: Given $\forall x.(p(x) \wedge q(x))$, use the Fitch System to prove $\forall x.p(x) \wedge \forall x.q(x)$.

Exercise 8.2: Given $\forall x.(p(x) \Rightarrow q(x))$, use the Fitch System to prove $\forall x.p(x) \Rightarrow \forall x.q(x)$.

Exercise 8.3: Given the premises $\forall x.(p(x) \Rightarrow q(x))$ and $\forall x.(q(x) \Rightarrow r(x))$, use the Fitch system to prove the conclusion $\forall x.(p(x) \Rightarrow r(x))$.

Exercise 8.4: Given $\forall x.\forall y.p(x,y)$, use the Fitch System to prove $\forall y.\forall x.p(x,y)$.

Exercise 8.5: Given $\forall x.\forall y.p(x,y)$, use the Fitch System to prove $\forall x.\forall y.p(y,x)$.

Exercise 8.6: Given $\exists y.\forall x.p(x,y)$, use the Fitch system to prove $\forall x.\exists y.p(x,y)$.

Exercise 8.7: Given $\exists x.\neg p(x)$, use the Fitch System to prove $\neg\forall x.p(x)$.

Exercise 8.8: Given $\forall x.p(x)$, use the Fitch System to prove $\neg\exists x.\neg p(x)$.

try $\exists x.\neg p(x) \rightarrow \psi$

$\exists x.\neg p(x) \rightarrow \neg\psi$

$\neg\exists x.\neg p(x)$

$\forall x.(\neg p(x) \rightarrow \neg\forall x.p(x))$

$\exists y.(q(y) \wedge p(y))$

$q(y) \wedge p(y) \rightarrow$

$q(a)$

$\exists y.q(y)$

$\forall y\ q(y) \rightarrow r(y)$

$r(y)$