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.

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 five new rules of inference. In the next section, we introduce two rules of inference for universally quantified sentences. We then introduce two rules of inference for existentially quantified sentences. After that, we describe a new type of rule called *Domain Closure*. Finally, we illustrate the system with a few examples.

If you are 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.

8.2 Rules for Universal Quantifiers

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

$\forall \nu.\phi[\nu]$

$\phi[\tau]$

where τ is substitutable for ν 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.

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

Universal Introduction

ϕ

$\forall \nu.\phi$

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

8.3 Rules for Existential Quantifiers

Existential Elimination (EE) allows us to reason from an existentially quantified sentence to an instance of the scope of the quantified sentence. Once this is done, we can manipulate the instance and derive conclusions that would not be possible with the quantified sentence as a whole.

The main problem in eliminating an existential quantifier is choosing an expression to replace the existential variable. Should we replace it with a variable? How about an object constant? Unfortunately, neither of these solutions works.

We cannot use a variable since the resulting instance would be equivalent to a universally quantified sentence (given our convention that our free variables are universally quantified). For example, given the sentence $\exists x.p(x)$, we cannot infer $p(x)$ since that would say that p is true of everything.

And we cannot replace the quantified variable with an object constant. Although we know the sentence is true of some object, we do not know which it is. For example, given a language with just two object constants a and b and given the sentence $\exists x.p(x)$, we cannot infer $p(a)$ since we do

not know whether or not p is true of a and we cannot infer $p(b)$ since we do not know whether or not p is true of b.

The solution to this problem is to replace the variable in an existentially quantified sentence with a new type of expression, called a Skolem term. A Skolem term is effectively a placeholder for an actual object constant and is treated like an ordinary term by our rules of inference (except for Domain Closure, as described in Section 8.4). For example, in the case above, given the sentence $\exists x.p(x)$, we would infer $p([c])$, where c is a new constant (one that is not part of our language) and where the square brackets are used to make clear that $[c]$ is a Skolem term rather than an ordinary object constant.

Using the idea of Skolem terms, we can formulate a rule of inference for eliminating existential quantifiers. There are two cases to consider - the case where the quantified sentence is closed and the case where the quantified sentence contains one or more free variables.

Existential Elimination for closed sentences. Suppose we have a closed existentially quantified sentence with variable ν and scope $\phi[\nu]$. Then, we can infer an instance of ϕ in which ν is replaced by a new constant τ. In what follows, we refer to τ as a *Skolem constant*.

Existential Elimination

$\exists \nu.\phi[\nu]$

$\phi[[\tau]]$

where τ is a not an existing object constant

For example, if we have the sentence $\exists x.hates(jane,x)$, then we can conclude $hates(jane,[c])$, provided that c is not an existing object constant. This sentence tells us that Jane hates someone; we just do not know who that someone is. If we have the sentence $\exists x.hates(x,x)$, then we can conclude $hates([c],[c])$, provided that c is not an existing object constant. This sentence tells us that someone hates himself; we just do not know who that someone is.

Unfortunately, things get a little more difficult when the existentially quantified sentence contains free variables. Suppose, for example, we had the sentence $\exists y.hates(x,y)$. Given this sentence, we know that everyone hates someone, but it is possible that different people hate different people. If we were to replace the existential variable with a simple Skolem term $[c]$ we would get the conclusion $hates(x,[c])$, this would say that there is a single person that everyone hates, which is not the same as the original meaning.

Fortunately, we can solve this problem through the use of more complex Skolem terms that include not just a new constant but also include free variables in the Skolem term.

Existential Elimination for sentences with free variables. Suppose we have an existentially quantified sentence with quantified variable ν and scope $\phi[\nu_1,...,\nu_n,\nu]$ where $\nu_1, ..., \nu_n$ are free variables. Then, we can infer an instance of ϕ in which ν is replaced by a new term $[\pi(\nu_1,...,\nu_n)]$, where π is a new constant. In what follows, we refer to π as a *Skolem function*.

Existential Elimination

$\exists \nu.\phi[\nu_1,...,\nu_n,\nu]$

$\phi[[\pi(\nu_1,...,\nu_n)]]$

where π is a not an existing constant

For example, if we have the sentence $\exists y.hates(x,y)$, then we can conclude $hates(x,[f(x)])$. This sentence tells us that everyone hates someone but it allows for the possibility that the hated person is different in each case.

In what follows, we combine these two cases into a single rule, called *Existential Elimination*, with the understanding that the first case applies in the case of closed existentially quantified sentences and the second applies in the case of existentially quantified sentences with free variables.

By comparison to Existential Elimination, *Existential Introduction* (EI) is easy. If we believe a sentence involving a 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 provided that τ does not contain any explicitly quantified variables.

Existential Introduction

$\phi[\tau]$

$\exists \nu.\phi[\nu]$

where τ does not contain any universally quantified variables

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. $\exists y.$*hates*(*jill*,y), and there is someone who hates Jill, i.e. $\exists x.$*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.

Note also that the rule applies only in cases where the term being replaced does not contain any universally quantified variables. This restriction is necessary to avoid incorrect conclusions. Suppose, for example, we had the sentence $\forall x.$*hates*(x,[*f*(x)]). If we were to use EI in this case, we would get the conclusion $\exists y.\forall x.$*hates*(x,y). The original sentence says that everyone hates someone but not necessarily the same person whereas this false conclusion states that there is a single person that everyone hates.

8.4 Domain Closure

Finally, for languages with finite Herbrand bases, we have the *Domain Closure* (DC) rule. For a language with object constants σ_1, ... , σ_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

$\phi[\sigma_1]$

...

$\phi[\sigma_n]$

$\forall \nu.\phi[\nu]$

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

Note that in applying Domain Closure, we do *not* need to worry about Skolem terms. Skolem terms are just place holders for constants in our language. If we have managed to prove that a condition holds for all of our object constants, then it must be true of everything, including whatever objects our Skolem terms represent.

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.

8.5 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 $loves(x,y)$.

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

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 Implication Elimination to produce our conclusion on line 6.

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

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

8.6 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.

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* relationship is transitive.

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

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 use existential elimination to derive an instance of the second premise, here using the Skolem term [c] to refer to the greyhound that is faster than every rabbit. Lines 8 and 9 come from And Elimination applied to line 7. 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 [c] is faster than Ralph. Next, we instantiate the sentence about greyhounds and dogs and infer that [c] 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 [c]. We then instantiate the transitivity sentence, again form the necessary conjunction, and infer the desired conclusion.

1.	$\forall x.\forall y.(h(x) \wedge d(y) \Rightarrow f(x,y))$	Premise
2.	$\exists y.(g(y) \wedge \forall z.(r(z) \Rightarrow f(y,z)))$	Premise
3.	$\forall y.(g(y) \Rightarrow d(y))$	Premise
4.	$\forall x.\forall y.\forall z.(f(x,y) \wedge f(y,z) \Rightarrow f(x,z))$	Premise
5.	$h(harry)$	Premise
6.	$r(ralph)$	Premise
7.	$g([c]) \wedge \forall z.(r(z) \Rightarrow f([c],z))$	EE: 2
8.	$g([c])$	AE: 7
9.	$\forall z.(r(z) \Rightarrow f([c],z))$	AE: 7
10.	$r(ralph) \Rightarrow f([c],ralph)$	UE: 9
11.	$f([c],ralph)$	IE: 10, 6
12.	$g([c]) \Rightarrow d([c])$	UE: 3
13.	$d([c])$	IE: 12, 8
14.	$\forall y.(h(harry) \wedge d(y) \Rightarrow f(harry,y))$	UE: 1
15.	$h(harry) \wedge d([c]) \Rightarrow f(harry,[c])$	UE: 14
16.	$h(harry) \wedge d([c])$	AI: 5, 13
17.	$f(harry,[c])$	IE: 15, 16
18.	$\forall y.\forall z.(f(harry,y) \wedge f(y,z) \Rightarrow f(harry,z))$	UE: 4
19.	$\forall z.(f(harry,[c]) \wedge f([c],z) \Rightarrow f(harry,z))$	UE: 18
20.	$f(harry,[c]) \wedge f([c],ralph) \Rightarrow f(harry,ralph)$	UE: 19
21.	$f(harry,[c]) \wedge f([c],ralph)$	AI: 17, 11
22.	$f(harry,ralph)$	IE: 20, 21

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.

8.7 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. We use Existential Elimination to eliminate the existential quantifier. We then use two applications of Universal Elimination to strip away the quantifiers from

the premise. And we use Implication Elimination to derive $q(x)$. 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.	$p(x,[c(x)])$	EE: 2
4.	$\forall y.(p(x,y) \Rightarrow q(x))$	UE: 1
5.	$p(x,[c(x)]) \Rightarrow q(x)$	UE: 4
6.	$q(x)$	IE: 3, 5
7.	$\exists y.p(x,y) \Rightarrow q(x)$	II: 2, 6
8.	$\forall x.(\exists y.p(x,y) \Rightarrow q(x))$	UI: 7

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

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 $\exists \nu.\phi[\nu]$ and a universally quantified implication $\forall \nu.(\phi[\nu] \Rightarrow \psi)$ to the consequent ψ, under the condition that ν 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 $\neg\exists x.\neg p(x)$.