Chapter 8
Relational Proofs

§8.1 Introduction

Formal proofs in relational logic are analogous to formal proofs in propositional logic. The major difference is that there are additional rules of inference and/or axiom schemata to deal with quantifiers and quantified sentences.

§8.2 Rules of Inference

As with propositional logic, proofs in relational logic are based on rules of inference. In the case of relational logic, a rule of inference consists of (1) a set of relational sentence schemas called *premises*, and (2) another set of sentence patterns called *conclusions*. Whenever we have sentences that *match* the conditions of the rule, then it is acceptable to infer sentences matching the conclusions.

Since relational logic includes all of the logical operators in propositional logic, we have all of the same rules of inference. The following paragraphs descibe some of these rules.

The rule shown below is called *Modus ponens* (MP). The significance of this rule is that, whenever sentences of the form $(\phi \Rightarrow \psi)$ and ϕ have been established, then it is acceptable to infer the sentence ψ as well.

$$\begin{array}{l} \phi \Rightarrow \psi \\ \underline{\phi \qquad\quad} \\ \psi \end{array}$$

Modus tolens (MT) is the reverse of modus ponens. If we believe that ϕ implies ψ and we believe that ψ is false, then we can infer that ϕ must be false as well.

$$\begin{array}{l} \phi \Rightarrow \psi \\ \underline{\neg\psi \qquad} \\ \neg\phi \end{array}$$

And elimination (AE) states that, whenever we believe a conjunction of sentences, then we can infer each of the conjuncts. In this case, note that there are multiple conclusions.

$$\begin{array}{l} \underline{\phi \wedge \psi} \\ \phi \\ \psi \end{array}$$

And introduction (AI) states that, whenever we believe some sentences, we can infer their conjunction.

$$\frac{\begin{array}{l}\phi\\ \psi\end{array}}{\phi \wedge \psi}$$

In addition to these rules of inference for logical operators, we have some rules of inference appropriate to quantified sentences.

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

$$\frac{\forall \nu . \phi}{\phi_{\nu/\tau} \text{where } \tau \text{ is free for } \nu \text{ in } \phi}$$

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 herself, i.e. $hates(jane, jane)$. We can even infer than Jane hates her mother, i.e. $hates(jane, mom(jane))$.

In addition, we can use universal instantiation to create conclusions with free variables. For example, from $\forall y.hates(jane, y)$, we can infer $hates(jane, y)$. In doing so, however, we have to be careful to avoid conflicts with other variables in the quantified sentence. This is the reason for the constraint on the replacement term. As an example, consider the expression $\forall y.\exists z.hates(y, z)$; i.e., everybody hates somebody. From this expression, it makes sense to infer $\exists z.hates(mom(x), z)$; i.e., everybody's mother hates somebody. However, we do not want to infer $\exists z.hates(mom(z), z)$; i.e., there is someone who is hated by his mother.

We can avoid this problem by obeying the restriction on the universal instantiation rule. We say that a term τ is *free* for a variable ν in an expression ϕ if and only if ν does not occur within the scope of a quantifier of some variable in τ. For example, the term `Mom(x)` is free for `y` in $\exists z.hates(y, z)$. However, the term `Mom(z)` is not free for `y`, since `y` occurs within the scope of a quantifier of `z`. Thus, we cannot substitute `Mom(z)` for `y` in this sentence, and we avoid the problem we have just described.

Existential instantiation (EI) allows us to eliminate existential quantifiers. Like universal instantiation, this rule states that we can infer an instance of the quantified sentence in which the existentially quantified variable is replaced by a suitable term. There are two cases of existential instantiation.

The first case of existential instantiation covers the situation when the sentence within the quantifier contains no free variables other than the quantified variable. In this case, the quantifier can be dropped and the variable can be replaced by an arbitrary new constant.

$$\frac{\exists \nu . \phi}{\phi_{\nu/\tau} \text{where } \tau \text{ is a new object constant}}$$

When there are free variables other than the quantified variable in the quantified sentence, those variables must be taken into account. In this case the variable is replaced by a functional term consisting of a new function constant applied to the non-quantified free variables in the quantified sentence.

$$\frac{\exists \nu . \phi}{\phi_{\nu/\pi(\nu_1,\ldots,\nu_n)}} \quad \begin{array}{l} \text{where } \pi \text{ is a new function constant} \\ \text{where } \nu_1, \ldots, \nu_n \text{ are the free variables in } \phi \end{array}$$

For example, if we have the premise $\exists z.hates(y,z)$ and if `Foe` is a new function constant, we can use existential instantiation to infer the sentence `Hates(y,Foe(y))`. The term `Foe(y)` here is a term designating the person `y` hates.

The mention of free variables in the replacement term is intended to capture the relationship between the value of the existentially quantified variable and the values for the free variables in the expression. Without this restriction, we would be able to instantiate the sentence $\forall x.\exists y.hates(x,y)$ and the sentence $\exists y.\forall x.hates(x,y)$ in the same way, despite their very different meanings.

Of course, when there are no free variables in an expression, the variable can be replaced by a function of no arguments or, equivalently, by a new constant. For example, if we have the sentence $\exists y.\forall x.hates(x,y)$, and `Mike` is a new object constant, we can infer $\forall x.hates(x,Mike)$; i.e., everyone hates `Mike`.

Note that, in performing existential instantiation, it is extremely important to avoid object and function constants that have been used already. Without this restriction, we would be able to infer $hates(Jill,Jill)$ from the somewhat weaker fact $\exists z.hates(Jill,z)$.

Although these rules cover many cases of inference, they are not by themselves exhaustive. Later, we define a criterion of exhaustiveness and present rules that satisfy our criterion.

Given any set of inference rules, we say that a conclusion ϕ is *derivable* from a set of premises Δ if and only if (1) ϕ is a member of Δ, or (2) ϕ is the result of applying a rule of inference to sentences derivable from Δ. A *derivation* of ϕ from Δ is a sequence of sentences in which each sentence either is a member of Δ or is the result of applying a rule of inference to elements earlier in the sequence.

§8.3 Axiom Schemata for Relational Logic

The *universal distribution* schema (UD) allows us to distribute quantification over implication.

$$(\forall \nu.\phi \Rightarrow \psi) \Rightarrow ((\forall \nu.\phi) \Rightarrow (\forall \nu.\psi))$$

The *universal generalization* schema (UG) allows us to derive universally quantified statements. If a sentence ϕ does not contain ν as a free variable, then it is certainly permissible to conclude that $\forall \nu.\phi$.

$$\phi \Rightarrow \forall \nu.\phi$$
where ν does not occur free in ϕ

The *universal instantiation* schema (UI) states that, whenever the database contains a universally quantified sentence $\forall\nu.\phi$, it is acceptable to add a copy of ϕ in which all occurrences of ν have been replaced by any suitable term.

$$(\forall\nu.\phi \Rightarrow \phi_{\nu/\tau})$$
where τ is free for ν in ϕ

Note that the universal instantiation schema is very similar to the universal instantiation rule of inference. In fact, given modus ponens, it allows us to draw all the same conclusions. This is the reason we can drop that rule in our definition of proof. We can drop the other rules of inference for similar reasons.

§8.4 Example

As an illustration of these concepts, 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 are adding two facts about the world not stated explicitly in the problem: that greyhounds are dogs and that our speed relationship is transitive.

$$\forall x.\forall y.(horse(x) \wedge dog(y) \Rightarrow faster(x, y))$$
$$\exists y.(greyhound(y) \wedge (\forall z.(rabbit(z) \Rightarrow faster(y, z)))$$
$$\forall y.(greyhound(y) \Rightarrow dog(y))$$
$$\forall x\forall y.\forall z.(faster(x, y) \wedge faster(y, z) \Rightarrow faster(x, z))$$
$$horse(harry)$$
$$rabbit(ralph)$$

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:

$$faster(harry, ralph)$$

The derivation of this conclusion goes as shown below. The first six lines correspond to the premises just formalized. The seventh line is the result of applying existential instantiation to the second sentence. Because there are no free variables, we replace the quantified variable by the new object constant `greg`. The eighth and nine lines come from and elimination. The tenth line is a universal instantiation of the ninth line. In the eleventh line, we use modus ponens to infer that Greg is faster than Ralph. Next, we instantiate the sentence about greyhounds and dogs and infer that Greg 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 Greg. In the final sequence, we instantiate the transitivity sentence, again form the necessary conjunction, and infer the desired conclusion.

1.	$\forall x.\forall y.(horse(x) \wedge dog(y) \Rightarrow faster(x, y))$	Premise
2.	$\exists y.(greyhound(y) \wedge (\forall z.(rabbit(z) \Rightarrow faster(y, z)))$	Premise
3.	$\forall y.(greyhound(y) \Rightarrow dog(y))$	Premise
4.	$\forall x.\forall y.\forall z.(faster(x, y) \wedge faster(y, z) \Rightarrow faster(x, z))$	Premise
5.	$horse(harry)$	Premise
6.	$rabbit(ralph)$	Premise
7.	$greyhound(greg) \wedge \forall z.(rabbit(z) \Rightarrow faster(greg, z))$	EI: 2
8.	$greyhound(greg)$	AE: 7
9.	$\forall z.(rabbit(z) \Rightarrow faster(greg, z))$	AE: 7
10.	$rabbit(ralph) \Rightarrow faster(greg, ralph)$	UI: 9
11.	$faster(greg, ralph)$	MP: 10, 6
12.	$greyhound(greg) \Rightarrow dog(greg)$	UI: 3
13.	$dog(greg)$	MP: 12, 8
14.	$horse(harry) \wedge dog(greg) \Rightarrow faster(harry, greg)$	UI: 1
15.	$horse(harry) \wedge dog(greg)$	AI: 5, 13
16.	$faster(harry, greg)$	MP: 14, 15
17.	$faster(harry, greg) \wedge faster(greg, ralph) \Rightarrow faster(harry, ralph)$	UI: 4
18.	$faster(harry, greg) \wedge faster(greg, ralph)$	AI: 16,11
19.	$faster(harry, ralph)$	MP: 17, 19

As with propositional logic, this derivation 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 automating the process of inference

Exercises

1. *Proofs.* The law says that it is a crime to sell an unregistered gun. Red has several unregistered guns, and all of them were purchased from Lefty. Give a formal proof that Lefty is a criminal.

2. *Proofs.* Give a formal proof of the sentence $\forall x.(p(x) \Rightarrow r(x))$ from the premises $\forall x.(p(x) \Rightarrow q(x))$ and $\forall x.(q(x) \Rightarrow r(x))$. Note that the generalization theorem does *not* solve this problem. We need to use a generalized axiom schema.

   If you are unhappy with the contorted steps you must take to solve this problem, that is as it should be. One of the points here is to show the awkwardness of using axiom schemata as a way of motivating the simpler proof methods discussed in the next chapters.