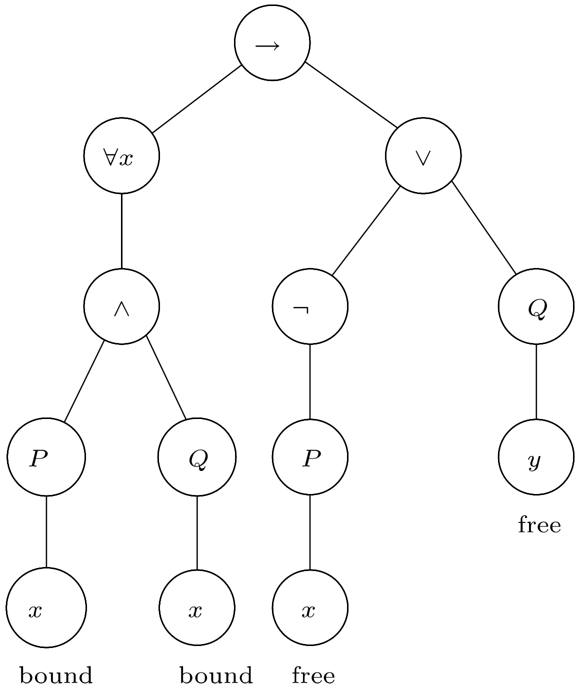


Figure 2.2. A parse tree of a predicate logic formula illustrating free and bound occurrences of variables.

quantifier for x; e.g. the scope of $\forall x$ in $\forall x\,(P(x) \to \exists x\, Q(x))$ is $P(x)$. It is quite possible, and common, that a variable is bound and free in a formula. Consider the formula

$$(\forall x\,(P(x) \land Q(x))) \to (\neg P(x) \lor Q(y))$$

and its parse tree in Figure 2.2. The two x leaves in the subtree of $\forall x$ are bound since they are in the scope of $\forall x$, but the leaf x in the right subtree of $\to$ is free since it is *not* in the scope of any quantifier $\forall x$ or $\exists x$. Note, however, that a single leaf either is under the scope of a quantifier, or it isn't. Hence *individual* occurrences of variables are either free or bound, never both at the same time.

2.2.4 Substitution

Variables are place holders so we must have some means of *replacing* them with more concrete information. On the syntactic side, we often need to replace a leaf node x by the parse tree of an entire term t. Recall from the definition of formulas that any replacement of x may only be a term; it could not be a predicate expression, or a more complex formula, for x serves as a term to a predicate symbol one step higher up in the parse tree (see Definition 2.1 and the grammar in (2.2)). In substituting t for x we have to leave untouched the *bound* leaves x since they are in the scope of some $\exists x$ or $\forall x$, i.e. they stand for *some unspecified* or *all* values respectively.

Definition 2.7 Given a variable x, a term t and a formula ϕ we define $\phi[t/x]$ to be the formula obtained by replacing each free occurrence of variable x in ϕ with t.

Substitutions are easily understood by looking at some examples. Let f be a function symbol with two arguments and ϕ the formula with the parse tree in Figure 2.1. Then $f(x, y)$ is a term and $\phi[f(x, y)/x]$ is just ϕ again. This is true because *all* occurrences of x are bound in ϕ, so *none* of them gets substituted.

Now consider ϕ to be the formula with the parse tree in Figure 2.2. Here we have one free occurrence of x in ϕ, so we substitute the parse tree of $f(x, y)$ for that free leaf node x and obtain the parse tree in Figure 2.3. Note that the bound x leaves are unaffected by this operation. You can see that the process of substitution is straightforward, but requires that it be applied *only to the free occurrences* of the variable to be substituted.

A word on notation: in writing $\phi[t/x]$, we really mean this to be the formula *obtained* by performing the operation $[t/x]$ on ϕ. Strictly speaking, the chain of symbols $\phi[t/x]$ is *not* a logical formula, but its *result* will be a formula, provided that ϕ was one in the first place.

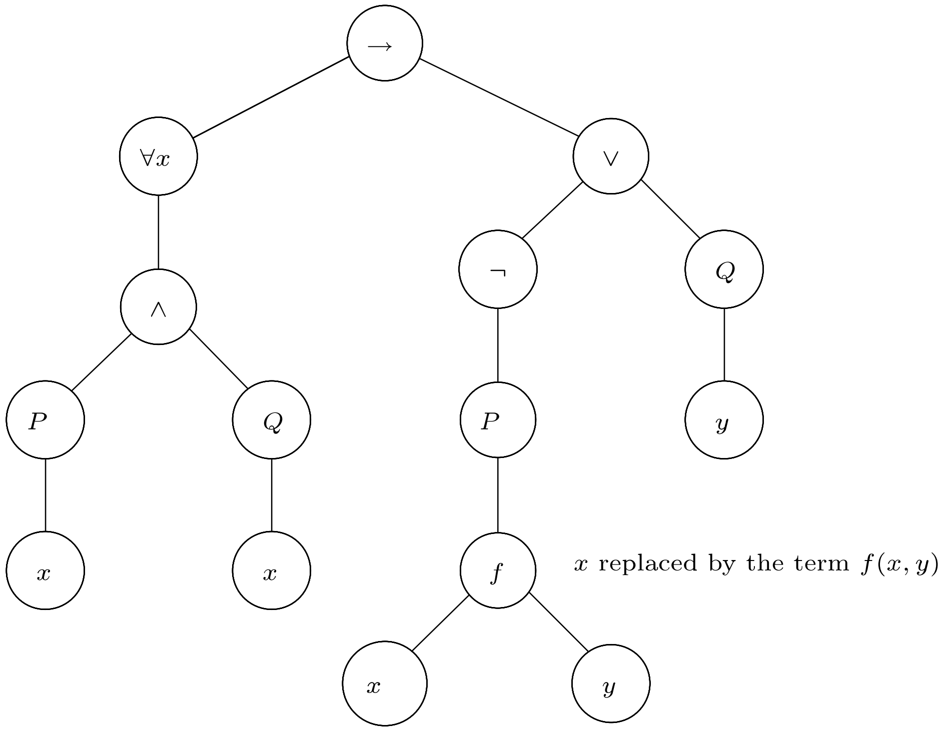


Figure 2.3. A parse tree of a formula resulting from substitution.

Unfortunately, substitutions can give rise to undesired side effects. In performing a substitution $\phi[t/x]$, the term t may contain a variable y, where free occurrences of x in ϕ are under the scope of $\exists y$ or $\forall y$ in ϕ. By carrying out this substitution $\phi[t/x]$, the value y, which might have been fixed by a concrete context, gets caught in the scope of $\exists y$ or $\forall y$. This binding capture overrides the context specification of the concrete value of y, for it will now stand for '*some unspecified*' or '*all*,' respectively. Such undesired variable captures are to be avoided at all costs.

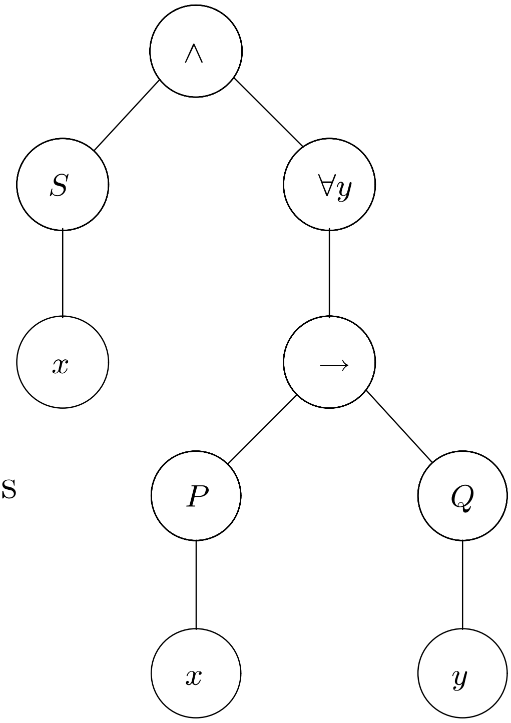
Definition 2.8 Given a term t, a variable x and a formula ϕ, we say that t is free for x in ϕ if no free x leaf in ϕ occurs in the scope of $\forall y$ or $\exists y$ for any variable y occurring in t.

This definition is maybe hard to swallow. Let us think of it in terms of parse trees. Given the parse tree of ϕ and the parse tree of t, we can perform the substitution $[t/x]$ on ϕ to obtain the formula $\phi[t/x]$. The latter has a parse tree where all free x leaves of the parse tree of ϕ are replaced by the parse tree of t. What 't is free for x in ϕ' means is that the variable leaves of the parse tree of t won't become bound if placed into the bigger parse tree of $\phi[t/x]$. For example, if we consider x, t and ϕ in Figure 2.3, then t is free for x in ϕ since the *new* leaf variables x and y of t are not under the scope of any quantifiers involving x or y.

Example 2.9 Consider the ϕ with parse tree in Figure 2.4 and let t be $f(y, y)$. All two occurrences of x in ϕ are free. The leftmost occurrence of x could be substituted since it is not in the scope of any quantifier, but substituting the rightmost x leaf introduces a new variable y in t which becomes bound by $\forall y$. Therefore, $f(y, y)$ is not free for x in ϕ.

What if there are no free occurrences of x in ϕ? Inspecting the definition of 't is free for x in ϕ,' we see that *every* term t is free for x in ϕ in that case, since no free variable x of ϕ is below some quantifier in the parse tree of ϕ. So the problematic situation of variable capture in performing $\phi[t/x]$ cannot occur. Of course, in that case $\phi[t/x]$ is just ϕ again.

It might be helpful to compare 't is free for x in ϕ' with a precondition of calling a procedure for substitution. If you are asked to compute $\phi[t/x]$ in your exercises or exams, then that is what you should do; but any reasonable implementation of substitution used in a theorem prover would have to check whether t is free for x in ϕ and, if not, rename some variables with fresh ones to avoid the undesirable capture of variables.



the term $f(y, y)$ is not free for x in this formula

Figure 2.4. A parse tree for which a substitution has dire consequences.

2.3 Proof theory of predicate logic

2.3.1 Natural deduction rules

Proofs in the natural deduction calculus for predicate logic are similar to those for propositional logic in Chapter 1, except that we have new proof rules for dealing with the quantifiers and with the equality symbol. Strictly speaking, we are *overloading* the previously established proof rules for the propositional connectives $\wedge$, $\vee$ etc. That simply means that any proof rule of Chapter 1 is still valid for logical formulas of predicate logic (we originally defined those rules for logical formulas of propositional logic). As in the natural deduction calculus for propositional logic, the additional rules for the quantifiers and equality will come in two flavours: introduction and elimination rules.

The proof rules for equality First, let us state the proof rules for equality. Here equality does not mean syntactic, or intensional, equality, but equality in terms of computation results. In either of these senses, any term t has to be equal to itself. This is expressed by the introduction rule for equality:

$$\frac{}{t = t} \; {=}\mathrm{i} \tag{2.5}$$

which is an axiom (as it does not depend on any premises). Notice that it may be invoked only if t is a term, our language doesn't permit us to talk about equality between formulas.

This rule is quite evidently sound, but it is not very useful on its own. What we need is a principle that allows us to substitute equals for equals repeatedly. For example, suppose that $y * (w + 2)$ equals $y * w + y * 2$; then it certainly must be the case that $z \geq y * (w + 2)$ implies $z \geq y * w + y * 2$ and vice versa. We may now express this substitution principle as the rule =e:

$$\frac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]} =\mathrm{e}.$$

Note that t_1 and t_2 have to be free for x in ϕ, whenever we want to apply the rule =e; this is an example of a *side condition* of a proof rule.

Convention 2.10 Throughout this section, when we write a substitution in the form $\phi[t/x]$, we implicitly assume that t is free for x in ϕ; for, as we saw in the last section, a substitution doesn't make sense otherwise.

We obtain proof

1	$(x + 1) = (1 + x)$	premise
2	$(x + 1 > 1) \rightarrow (x + 1 > 0)$	premise
3	$(1 + x > 1) \rightarrow (1 + x > 0)$	$=$e $1, 2$

establishing the validity of the sequent

$$x + 1 = 1 + x,\ (x + 1 > 1) \rightarrow (x + 1 > 0)\ \vdash\ (1 + x) > 1 \rightarrow (1 + x) > 0.$$

In this particular proof t_1 is $(x + 1)$, t_2 is $(1 + x)$ and ϕ is $(x > 1) \rightarrow (x > 0)$. We used the name =e since it reflects what this rule is doing to data: it eliminates the equality in $t_1 = t_2$ by replacing all t_1 in $\phi[t_1/x]$ with t_2. This is a sound substitution principle, since the assumption that t_1 equals t_2 guarantees that the logical meanings of $\phi[t_1/x]$ and $\phi[t_2/x]$ match.

The principle of substitution, in the guise of the rule =e, is quite powerful. Together with the rule =i, it allows us to show the sequents

$$t_1 = t_2 \vdash t_2 = t_1 \tag{2.6}$$

$$t_1 = t_2,\ t_2 = t_3 \vdash t_1 = t_3. \tag{2.7}$$

A proof for (2.6) is:

1	$t_1 = t_2$	premise
2	$t_1 = t_1$	=i
3	$t_2 = t_1$	=e $1, 2$

where ϕ is $x = t_1$. A proof for (2.7) is:

1	$t_2 = t_3$	premise
2	$t_1 = t_2$	premise
3	$t_1 = t_3$	=e $1, 2$

where ϕ is $t_1 = x$, so in line 2 we have $\phi[t_2/x]$ and in line 3 we obtain $\phi[t_3/x]$, as given by the rule =e applied to lines 1 and 2. Notice how we applied the scheme =e with several different instantiations.

Our discussion of the rules =i and =e has shown that they force equality to be *reflexive* (2.5), *symmetric* (2.6) and *transitive* (2.7). These are minimal and necessary requirements for any sane concept of (extensional) equality. We leave the topic of equality for now to move on to the proof rules for quantifiers.

The proof rules for universal quantification The rule for eliminating $\forall$ is the following:

$$\frac{\forall x\, \phi}{\phi[t/x]}\ \forall x\, \mathrm{e}.$$

It says: If $\forall x\, \phi$ is true, then you could replace the x in ϕ by any term t (given, as usual, the side condition that t be free for x in ϕ) and conclude that $\phi[t/x]$ is true as well. The intuitive soundness of this rule is self-evident.

Recall that $\phi[t/x]$ is obtained by replacing all free occurrences of x in ϕ by t. You may think of the term t as a more concrete *instance* of x. Since ϕ is assumed to be true for all x, that should also be the case for any term t.

Example 2.11 To see the necessity of the proviso that t be free for x in ϕ, consider the case that ϕ is $\exists y\,(x < y)$ and the term to be substituted for x is y. Let's suppose we are reasoning about numbers with the usual 'smaller than' relation. The statement $\forall x\, \phi$ then says that for all numbers n there is some bigger number m, which is indeed true of integers or real numbers. However, $\phi[y/x]$ is the formula $\exists y\,(y < y)$ saying that there is a number which is bigger than itself. This is wrong; and we must not allow a proof rule which derives semantically wrong things from semantically valid ones. Clearly, what went wrong was that y became bound in the process of substitution; y is not free for x in ϕ. Thus, in going from $\forall x\,\phi$ to $\phi[t/x]$, we have to enforce the side condition that t be free for x in ϕ: use a fresh variable for y to change ϕ to, say, $\exists z\,(x < z)$ and then apply $[y/x]$ to that formula, rendering $\exists z\,(y < z)$.

The rule $\forall x\,\mathrm{i}$ is a bit more complicated. It employs a proof box similar to those we have already seen in natural deduction for propositional logic, but this time the box is to stipulate the scope of the 'dummy variable' x_0 rather than the scope of an assumption. The rule $\forall x\,\mathrm{i}$ is written

$$\frac{\boxed{\begin{array}{l} x_0 \\ \qquad\quad \vdots \\ \qquad \phi[x_0/x] \end{array}}}{\forall x\,\phi}\ \forall x\,\mathrm{i}.$$

It says: If, starting with a 'fresh' variable x_0, you are able to prove some formula $\phi[x_0/x]$ with x_0 in it, then (*because* x_0 *is fresh*) you can derive $\forall x\,\phi$. The important point is that x_0 is a new variable which doesn't occur *anywhere outside its box*; we think of it as an *arbitrary* term. Since we assumed nothing about this x_0, anything would work in its place; hence the conclusion $\forall x\,\phi$.

It takes a while to understand this rule, since it seems to be going from the particular case of ϕ to the general case $\forall x\,\phi$. The side condition, that x_0 does not occur outside the box, is what allows us to get away with this.

To understand this, think of the following analogy. If you want to prove to someone that you can, say, split a tennis ball in your hand by squashing it, you might say 'OK, give me a tennis ball and I'll split it.' So we give you one and you do it. But how can we be sure that you could split *any* tennis ball in this way? Of course, we can't give you *all of them*, so how could we be sure that you could split any one? Well, we assume that the one you did split was an arbitrary, or 'random,' one, i.e. that it wasn't special in any way – like a ball which you may have 'prepared' beforehand; and that is enough to convince us that you could split *any* tennis ball. Our rule says that if you can prove ϕ about an x_0 that isn't special in any way, then you could prove it for any x whatsoever.

To put it another way, the step from ϕ to $\forall x\,\phi$ is legitimate only if we have arrived at ϕ in such a way that none of its assumptions contain x as a free variable. Any assumption which has a free occurrence of x puts constraints on such an x. For example, the assumption $\mathrm{bird}(x)$ confines x to the realm of birds and anything we can prove about x using this formula will have to be a statement restricted to birds and not about anything else we might have had in mind.

It is time we looked at an example of these proof rules at work. Here is a proof of the sequent $\forall x\,(P(x) \to Q(x)),\ \forall x\,P(x) \ \vdash\ \forall x\,Q(x)$:

1		$\forall x\,(P(x) \to Q(x))$	premise
2		$\forall x\,P(x)$	premise
3	x_0	$P(x_0) \to Q(x_0)$	$\forall x\,\mathrm{e}\ 1$
4		$P(x_0)$	$\forall x\,\mathrm{e}\ 2$
5		$Q(x_0)$	$\to\!\mathrm{e}\ 3, 4$
6		$\forall x\,Q(x)$	$\forall x\,\mathrm{i}\ 3\text{–}5$

The structure of this proof is guided by the fact that the conclusion is a $\forall$ formula. To arrive at this, we will need an application of $\forall x\,\mathrm{i}$, so we set up the box controlling the scope of x_0. The rest is now mechanical: we prove $\forall x\,Q(x)$ by proving $Q(x_0)$; but the latter we can prove as soon as we can prove $P(x_0)$ and $P(x_0) \to Q(x_0)$, which themselves are instances of the premises (obtained by $\forall\mathrm{e}$ with the term x_0). Note that we wrote the name of the dummy variable to the left of the first proof line in its scope box.

Here is a simpler example which uses only $\forall x\,\mathrm{e}$: we show the validity of the sequent $P(t),\ \forall x\,(P(x) \to \neg Q(x)) \ \vdash\ \neg Q(t)$ for any term t:

1	$P(t)$	premise
2	$\forall x\,(P(x) \to \neg Q(x))$	premise
3	$P(t) \to \neg Q(t)$	$\forall x\,\mathrm{e}\ 2$
4	$\neg Q(t)$	$\to\!\mathrm{e}\ 3, 1$

Note that we invoked $\forall x\,\mathrm{e}$ with the same instance t as in the assumption $P(t)$. If we had invoked $\forall x\,\mathrm{e}$ with y, say, and obtained $P(y) \to \neg Q(y)$, then that would have been valid, but it would not have been helpful in the case that y was different from t. Thus, $\forall x\,\mathrm{e}$ is really a *scheme* of rules, one for each term t (free for x in ϕ), and we should make our choice on the basis of consistent pattern matching. Further, note that we have rules $\forall x\,\mathrm{i}$ and $\forall x\,\mathrm{e}$ *for each variable x*. In particular, there are rules $\forall y\,\mathrm{i}$, $\forall y\,\mathrm{e}$ and so on. We will write $\forall$i and $\forall$e when we speak about such rules without concern for the actual quantifier variable.

Notice also that, although the square brackets representing substitution appear in the rules $\forall$i and $\forall$e, they do not appear when we use those rules. The reason for this is that we actually carry out the substitution that is asked for. In the rules, the expression $\phi[t/x]$ means: 'ϕ, but with free occurrences of x replaced by t.' Thus, if ϕ is $P(x, y) \to Q(y, z)$ and the rule refers to $\phi[a/y]$, we carry out the substitution and write $P(x, a) \to Q(a, z)$ in the proof.

A helpful way of understanding the universal quantifier rules is to compare the rules for $\forall$ with those for $\wedge$. The rules for $\forall$ are in some sense generalisations of those for $\wedge$; whereas $\wedge$ has just two conjuncts, $\forall$ acts like it conjoins lots of formulas (one for each substitution instance of its variable). Thus, whereas $\wedge$i has two premises, $\forall x$ i has a premise $\phi[x_0/x]$ for each possible 'value' of x_0. Similarly, where and-elimination allows you to deduce from $\phi \wedge \psi$ whichever of ϕ and ψ you like, forall-elimination allows you to deduce $\phi[t/x]$ from $\forall x\, \phi$, for whichever t you (and the side condition) like. To say the same thing another way: think of $\forall x$ i as saying: to prove $\forall x\, \phi$, you have to prove $\phi[x_0/x]$ for every possible value x_0; while $\wedge$i says that to prove $\phi_1 \wedge \phi_2$ you have to prove ϕ_i for every $i = 1, 2$.

The proof rules for existential quantification The analogy between $\forall$ and $\wedge$ extends also to $\exists$ and $\vee$; and you could even try to guess the rules for $\exists$ by starting from the rules for $\vee$ and applying the same ideas as those that related $\wedge$ to $\forall$. For example, we saw that the rules for or-introduction were a sort of dual of those for and-elimination; to emphasise this point, we could write them as

$$\frac{\phi_1 \wedge \phi_2}{\phi_k}\ \wedge\mathrm{e_k} \qquad\qquad \frac{\phi_k}{\phi_1 \vee \phi_2}\ \vee\mathrm{i_k}$$

where k can be chosen to be either 1 or 2. Therefore, given the form of forall-elimination, we can infer that exists-introduction must be simply

$$\frac{\phi[t/x]}{\exists x \phi}\ \exists x\,\mathrm{i}.$$

Indeed, this is correct: it simply says that we can deduce $\exists x\, \phi$ whenever we have $\phi[t/x]$ for some term t (naturally, we impose the side condition that t be free for x in ϕ).

In the rule $\exists$i, we see that the formula $\phi[t/x]$ contains, from a computational point of view, more information than $\exists x\, \phi$. The latter merely says that ϕ holds for some, unspecified, value of x; whereas $\phi[t/x]$ has a witness t at its disposal. Recall that the square-bracket notation asks us actually to carry out the substitution. However, the notation $\phi[t/x]$ is somewhat misleading since it suggests not only the right witness t but also the formula ϕ itself. For example, consider the situation in which t equals y such that $\phi[y/x]$ is $y = y$. Then you can check for yourself that ϕ could be a number of things, like $x = x$ or $x = y$. Thus, $\exists x\,\phi$ will depend on which of these ϕ you were thinking of.

Extending the analogy between $\exists$ and $\vee$, the rule $\vee$e leads us to the following formulation of $\exists$e:

$$\frac{\exists x\,\phi \qquad \boxed{\begin{array}{l} x_0\ \phi[x_0/x] \\ \qquad\vdots \\ \qquad\chi \end{array}}}{\chi}\ \exists\mathrm{e}.$$

Like $\vee$e, it involves a case analysis. The reasoning goes: We know $\exists x\,\phi$ is true, so ϕ is true for at least one ‘value’ of x. So we do a case analysis over all those possible values, writing x_0 as a generic value representing them all. If assuming $\phi[x_0/x]$ allows us to prove some χ which doesn’t mention x_0, then this χ must be true whichever x_0 makes $\phi[x_0/x]$ true. And that’s precisely what the rule $\exists$e allows us to deduce. Of course, we impose the side condition that x_0 can’t occur outside its box (therefore, in particular, it cannot occur in χ). The box is controlling two things: the scope of x_0 and also the scope of the assumption $\phi[x_0/x]$.

Just as $\vee$e says that to use $\phi_1 \vee \phi_2$, you have to be prepared for either of the ϕ_i, so $\exists$e says that to use $\exists x\,\phi$ you have to be prepared for any possible $\phi[x_0/x]$. Another way of thinking about $\exists$e goes like this: If you know $\exists x\,\phi$ and you can derive some χ from $\phi[x_0/x]$, i.e. by giving a name to the thing you know exists, then you can derive χ even without giving that thing a name (provided that χ does not refer to the name x_0).

The rule $\exists x\,$e is also similar to $\vee$e in the sense that both of them are elimination rules which don’t have to conclude a *subformula* of the formula they are about to eliminate. Please verify that all other elimination rules introduced so far have this *subformula property.*[2] This property is computationally very pleasant, for it allows us to narrow down the search space for a proof dramatically. Unfortunately, $\exists x\,$e, like its cousin $\vee$e, is not of that computationally benign kind.

[2] For $\forall x\,$e we perform a substitution $[t/x]$, but it preserves the logical structure of ϕ.

Let us practice these rules on a couple of examples. Certainly, we should be able to prove the validity of the sequent $\forall x\, \phi \;\vdash\; \exists x\, \phi$. The proof

1	$\forall x\, \phi$	premise
2	$\phi[x/x]$	$\forall x\,$e 1
3	$\exists x\, \phi$	$\exists x\,$i 2

demonstrates that, where we chose t to be x with respect to both $\forall x\,$e and to $\exists x\,$i (and note that x is free for x in ϕ and that $\phi[x/x]$ is simply ϕ again).

Proving the validity of the sequent $\forall x\,(P(x) \to Q(x)),\ \exists x\, P(x) \;\vdash\; \exists x\, Q(x)$ is more complicated:

1		$\forall x\,(P(x) \to Q(x))$	premise
2		$\exists x\, P(x)$	premise
3	x_0	$P(x_0)$	assumption
4		$P(x_0) \to Q(x_0)$	$\forall x\,$e 1
5		$Q(x_0)$	$\to$e 4, 3
6		$\exists x\, Q(x)$	$\exists x\,$i 5
7		$\exists x\, Q(x)$	$\exists x\,$e 2, 3–6

The motivation for introducing the box in line 3 of this proof is the existential quantifier in the premise $\exists x\, P(x)$ which has to be eliminated. Notice that the $\exists$ in the conclusion has to be introduced *within the box* and observe the nesting of these two steps. The formula $\exists x\, Q(x)$ in line 6 is the instantiation of χ in the rule $\exists$e and does not contain an occurrence of x_0, so it is allowed to leave the box to line 7. The almost identical 'proof'

1		$\forall x\,(P(x) \to Q(x))$	premise
2		$\exists x\, P(x)$	premise
3	x_0	$P(x_0)$	assumption
4		$P(x_0) \to Q(x_0)$	$\forall x\,$e 1
5		$Q(x_0)$	$\to$e 4, 3
6		$Q(x_0)$	$\exists x\,$e 2, 3–5
7		$\exists x\, Q(x)$	$\exists x\,$i 6

is illegal! Line 6 allows the fresh parameter x_0 to escape the scope of the box which declares it. This is not permissible and we will see on page 116 an example where such illicit use of proof rules results in unsound arguments.

A sequent with a slightly more complex proof is

$$\forall x\,(Q(x) \to R(x)),\ \exists x\,(P(x) \land Q(x)) \ \vdash\ \exists x\,(P(x) \land R(x))$$

and could model some argument such as

> *If all quakers are reformists and if there is a protestant who is also a quaker, then there must be a protestant who is also a reformist.*

One possible proof strategy is to assume $P(x_0) \land Q(x_0)$, get the instance $Q(x_0) \to R(x_0)$ from $\forall x\,(Q(x) \to R(x))$ and use $\land\mathrm{e}_2$ to get our hands on $Q(x_0)$, which gives us $R(x_0)$ via $\to$e . . . :

1		$\forall x\,(Q(x) \to R(x))$	premise
2		$\exists x\,(P(x) \land Q(x))$	premise
3	x_0	$P(x_0) \land Q(x_0)$	assumption
4		$Q(x_0) \to R(x_0)$	$\forall x\,\mathrm{e}\ 1$
5		$Q(x_0)$	$\land\mathrm{e}_2\ 3$
6		$R(x_0)$	$\to\mathrm{e}\ 4,5$
7		$P(x_0)$	$\land\mathrm{e}_1\ 3$
8		$P(x_0) \land R(x_0)$	$\land\mathrm{i}\ 7,6$
9		$\exists x\,(P(x) \land R(x))$	$\exists x\,\mathrm{i}\ 8$
10		$\exists x\,(P(x) \land R(x))$	$\exists x\,\mathrm{e}\ 2, 3{-}9$

Note the strategy of this proof: We list the two premises. The second premise is of use here only if we apply $\exists x\,$e to it. This sets up the proof box in lines 3−9 as well as the fresh parameter name x_0. Since we want to prove $\exists x\,(P(x) \land R(x))$, this formula has to be the last one in the box (our goal) and the rest involves $\forall x\,$e and $\exists x\,$i.

The rules $\forall$i and $\exists$e both have the side condition that the dummy variable cannot occur outside the box in the rule. Of course, these rules may still be nested, by choosing another fresh name (e.g. y_0) for the dummy variable. For example, consider the sequent $\exists x\,P(x),\ \forall x\,\forall y\,(P(x) \to Q(y)) \ \vdash\ \forall y\,Q(y)$. (Look how strong the second premise is, by the way: given any x, y, if $P(x)$, then $Q(y)$. This means that, if there is any object with the property P, then all objects shall have the property Q.) Its proof goes as follows: We take an arbitrary y_0 and prove $Q(y_0)$; this we do by observing that, since some x

satisfies P, so by the second premise any y satisfies Q:

1		$\exists x\, P(x)$	premise
2		$\forall x \forall y\, (P(x) \to Q(y))$	premise
3	y_0		
4	x_0	$P(x_0)$	assumption
5		$\forall y\, (P(x_0) \to Q(y))$	$\forall x$ e 2
6		$P(x_0) \to Q(y_0)$	$\forall y$ e 5
7		$Q(y_0)$	$\to$e 6, 4
8		$Q(y_0)$	$\exists x$ e 1, 4–7
9		$\forall y\, Q(y)$	$\forall y$ i 3–8

There is no special reason for picking x_0 as a name for the dummy variable we use for $\forall x$ and $\exists x$ and y_0 as a name for $\forall y$ and $\exists y$. We do this only because it makes it easier for us humans. Again, study the strategy of this proof. We ultimately have to show a $\forall y$ formula which requires us to use $\forall y$ i, i.e. we need to open up a proof box (lines 3–8) whose subgoal is to prove a generic instance $Q(y_0)$. Within that box we want to make use of the premise $\exists x\, P(x)$ which results in the proof box set-up of lines 4–7. Notice that, in line 8, we may well move $Q(y_0)$ out of the box controlled by x_0.

We have repeatedly emphasised the point that the dummy variables in the rules $\exists$e and $\forall$i must not occur outside their boxes. Here is an example which shows how things would go wrong if we didn't have this side condition. Consider the invalid sequent $\exists x\, P(x), \forall x\, (P(x) \to Q(x)) \vdash \forall y\, Q(y)$. (Compare it with the previous sequent; the second premise is now much weaker, allowing us to conclude Q only for those objects for which we know P.) Here is an alleged 'proof' of its validity:

1		$\exists x\, P(x)$	premise
2		$\forall x\, (P(x) \to Q(x))$	premise
3	x_0		
4	x_0	$P(x_0)$	assumption
5		$P(x_0) \to Q(x_0)$	$\forall x$ e 2
6		$Q(x_0)$	$\to$e 5, 4
7		$Q(x_0)$	$\exists x$ e 1, 4–6
8		$\forall y\, Q(y)$	$\forall y$ i 3–7

The last step introducing $\forall y$ is *not* the bad one; that step is fine. The bad one is the second from last one, concluding $Q(x_0)$ by $\exists x\,\mathrm{e}$ and violating the side condition that x_0 may not leave the scope of its box. You can try a few other ways of 'proving' this sequent, but none of them should work (assuming that our proof system is sound with respect to semantic entailment, which we define in the next section). Without this side condition, we would also be able to prove that 'all x satisfy the property P as soon as one of them does so,' a semantic disaster of biblical proportions!

2.3.2 Quantifier equivalences

We have already hinted at semantic equivalences between certain forms of quantification. Now we want to provide formal proofs for some of the most commonly used quantifier equivalences. Quite a few of them involve several quantifications over more than just one variable. Thus, this topic is also good practice for using the proof rules for quantifiers in a nested fashion.

For example, the formula $\forall x\, \forall y\, \phi$ should be equivalent to $\forall y\, \forall x\, \phi$ since both say that ϕ should hold for all values of x and y. What about $(\forall x\, \phi) \wedge (\forall x\, \psi)$ versus $\forall x\, (\phi \wedge \psi)$? A moment's thought reveals that they should have the same meaning as well. But what if the second conjunct does not start with $\forall x$? So what if we are looking at $(\forall x\, \phi) \wedge \psi$ in general and want to compare it with $\forall x\, (\phi \wedge \psi)$? Here we need to be careful, since x might be free in ψ and would then become bound in the formula $\forall x\, (\phi \wedge \psi)$.

Example 2.12 We may specify 'Not all birds can fly.' as $\neg\forall x\, (B(x) \to F(x))$ or as $\exists x\, (B(x) \wedge \neg F(x))$. The former formal specification is closer to the structure of the English specification, but the latter is logically equivalent to the former. Quantifier equivalences help us in establishing that specifications that 'look' different are really saying the same thing.

Here are some quantifier equivalences which you should become familiar with. As in Chapter 1, we write $\phi_1 \dashv\vdash \phi_2$ as an abbreviation for the validity of $\phi_1 \vdash \phi_2$ and $\phi_2 \vdash \phi_1$.

Theorem 2.13 Let ϕ and ψ be formulas of predicate logic. Then we have the following equivalences:

1. (a) $\neg\forall x\, \phi \dashv\vdash \exists x\, \neg\phi$
   (b) $\neg\exists x\, \phi \dashv\vdash \forall x\, \neg\phi$.
2. Assuming that x is not free in ψ: