

seq.1 Derivations with Quantifiers

fol:seq:prq:
sec

Example seq.1. Give an **LK**-derivation of the sequent $\exists x \neg\varphi(x) \Rightarrow \neg\forall x \varphi(x)$.

When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be lower down in the finished proof). Also, it is a good idea to try and look ahead and try to guess what the initial sequent might look like. In our case, it will have to be something like $\varphi(a) \Rightarrow \varphi(a)$. That means that when we are “reversing” the quantifier rules, we will have to pick the same term—what we will call a —for both the \forall and the \exists rule. If we picked different terms for each rule, we would end up with something like $\varphi(a) \Rightarrow \varphi(b)$, which, of course, is not derivable.

Starting as usual, we write

$$\frac{}{\exists x \neg\varphi(x) \Rightarrow \neg\forall x \varphi(x)}$$

We could either carry out the $\exists\text{L}$ rule or the $\neg\text{R}$ rule. Since the $\exists\text{L}$ rule is subject to the eigenvariable condition, it’s a good idea to take care of it sooner rather than later, so we’ll do that one first.

$$\frac{\frac{}{\neg\varphi(a) \Rightarrow \neg\forall x \varphi(x)}}{\exists x \neg\varphi(x) \Rightarrow \neg\forall x \varphi(x)} \exists\text{L}$$

Applying the $\neg\text{L}$ and $\neg\text{R}$ rules backwards, we get

$$\frac{\frac{\frac{\frac{}{\forall x \varphi(x) \Rightarrow \varphi(a)}}{\neg\varphi(a), \forall x \varphi(x) \Rightarrow} \neg\text{L}}{\forall x \varphi(x), \neg\varphi(a) \Rightarrow} \text{XL}}{\neg\varphi(a) \Rightarrow \neg\forall x \varphi(x)} \neg\text{R}}{\exists x \neg\varphi(x) \Rightarrow \neg\forall x \varphi(x)} \exists\text{L}$$

At this point, our only option is to carry out the $\forall\text{L}$ rule. Since this rule is not subject to the eigenvariable restriction, we’re in the clear. Remember, we want to try and obtain an initial sequent (of the form $\varphi(a) \Rightarrow \varphi(a)$), so we should choose a as our argument for φ when we apply the rule.

$$\frac{\frac{\frac{\frac{\frac{}{\varphi(a) \Rightarrow \varphi(a)}}{\forall x \varphi(x) \Rightarrow \varphi(a)} \forall\text{L}}{\neg\varphi(a), \forall x \varphi(x) \Rightarrow} \neg\text{L}}{\forall x \varphi(x), \neg\varphi(a) \Rightarrow} \text{XL}}{\neg\varphi(a) \Rightarrow \neg\forall x \varphi(x)} \neg\text{R}}{\exists x \neg\varphi(x) \Rightarrow \neg\forall x \varphi(x)} \exists\text{L}$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was $\exists L$, and the eigenvariable a does not occur in its lower sequent (the end-sequent), this is a correct derivation.

Problem seq.1. Give **derivations** of the following sequents:

1. $\Rightarrow (\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall z (\varphi(z) \wedge \psi(z))$.
2. $\Rightarrow (\exists x \varphi(x) \vee \exists y \psi(y)) \rightarrow \exists z (\varphi(z) \vee \psi(z))$.
3. $\forall x (\varphi(x) \rightarrow \psi) \Rightarrow \exists y \varphi(y) \rightarrow \psi$.
4. $\forall x \neg\varphi(x) \Rightarrow \neg\exists x \varphi(x)$.
5. $\Rightarrow \neg\exists x \varphi(x) \rightarrow \forall x \neg\varphi(x)$.
6. $\Rightarrow \neg\exists x \forall y ((\varphi(x, y) \rightarrow \neg\varphi(y, y)) \wedge (\neg\varphi(y, y) \rightarrow \varphi(x, y)))$.

Problem seq.2. Give **derivations** of the following sequents:

1. $\Rightarrow \neg\forall x \varphi(x) \rightarrow \exists x \neg\varphi(x)$.
2. $(\forall x \varphi(x) \rightarrow \psi) \Rightarrow \exists y (\varphi(y) \rightarrow \psi)$.
3. $\Rightarrow \exists x (\varphi(x) \rightarrow \forall y \varphi(y))$.

(These all require the CR rule.)

Photo Credits

Bibliography