CS313K: Logic, Sets, and Functions

J Strother Moore
Department of Computer Sciences
University of Texas at Austin

(Lecture 19)

Universal Closure

Suppose you have a formula γ with some free variables $v_1, v_2, \ldots$ in it. If somebody says "γ is a theorem" they mean its *universal closure* is a theorem, $(\forall v_1 : (\forall v_2 : (\ldots \gamma \ldots)))$.

Often we write
$(\forall v_1 : (\forall v_2 : (\ldots \gamma \ldots)))$
as
$(\forall v_1, v_2, \ldots : \gamma)$.

So to prove:

```
(∀ a,b,c : (app (app a b) c)
            =
            (app a (app b c))
```

just strip off the outermost quantifiers and prove that formula:

```
            (app (app a b) c)
            =
            (app a (app b c))
```

Warning

$(\forall \mathtt{x} : \gamma) \not\leftrightarrow \gamma$

For example,

$(\forall \mathtt{x} : \mathtt{x} = 1) \not\leftrightarrow \mathtt{x} = 1$

$$(\forall \mathtt{x} : \mathtt{x} = 1) \not\leftrightarrow \mathtt{x} = 1$$

So

Theorem:
$$(\forall \mathtt{x} : \mathtt{x} = 1) \rightarrow 4 = 5$$

But

$$\mathtt{x} = 1 \rightarrow 4 = 5$$

is not a theorem!

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge\ (\forall\ v\ :\ \gamma)\ \wedge \ldots] \ \rightarrow\ \psi$
2. $[\ldots \wedge\ (\exists\ v\ :\ \gamma)\ \wedge \ldots] \ \rightarrow\ \psi$
3. $\psi\ \rightarrow\ (\forall\ v\ :\ \gamma)$
4. $\psi\ \rightarrow\ (\exists\ v\ :\ \gamma)$

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge \; (\forall \; v \; : \; \gamma) \; \wedge \ldots] \; \rightarrow \; \psi$
2. $[\ldots \wedge \; \neg\psi \; \wedge \ldots] \; \rightarrow \; \neg \; (\exists \; v \; : \; \gamma)$
3. $\psi \; \rightarrow \; (\forall \; v \; : \; \gamma)$
4. $\psi \; \rightarrow \; (\exists \; v \; : \; \gamma)$

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge \ (\forall \ v \ : \ \gamma) \ \wedge \ldots] \ \rightarrow \ \psi$
2. $[\ldots \wedge \ \neg\psi \ \wedge \ldots] \ \rightarrow \ (\forall \ v \ : \ \neg\gamma)$
3. $\psi \ \rightarrow \ (\forall \ v \ : \ \gamma)$
4. $\psi \ \rightarrow \ (\exists \ v \ : \ \gamma)$

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge \ (\forall \ v \ : \ \gamma) \ \wedge \ldots] \ \rightarrow \ \psi$

3. $\psi \ \rightarrow \ (\forall \ v \ : \ \gamma)$
4. $\psi \ \rightarrow \ (\exists \ v \ : \ \gamma)$

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge\ (\forall\ v\ :\ \gamma)\ \wedge \ldots] \ \rightarrow\ \psi$

3. $\psi\ \rightarrow\ (\forall\ v\ :\ \gamma)$
4. $\neg(\exists\ v\ :\ \gamma)\ \rightarrow\ \neg\psi$

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge \ (\forall\ v\ :\ \gamma)\ \wedge \ldots] \ \rightarrow\ \psi$

3. $\psi\ \rightarrow\ (\forall\ v\ :\ \gamma)$
4. $(\forall\ v\ :\ \neg\gamma)\ \rightarrow\ \neg\psi$

The most interesting formulas have quantifiers in the hypotheses and/or the conclusions.

If you factor your formulas into our standard shape, there are four classes of formulas:

1. $[\ldots \wedge\ (\forall\ v\ :\ \gamma)\ \wedge \ldots]\ \rightarrow\ \psi$

3. $\psi\ \rightarrow\ (\forall\ v\ :\ \gamma)$

Formula Class 1

To prove

$$\ldots \wedge [\forall v : \gamma_v] \wedge \ldots \rightarrow \psi$$

you can give yourself as many instances of γ as you want:

$$\ldots \wedge [\gamma_{\alpha_1} \wedge \ldots \gamma_{\alpha_n} \wedge (\forall v : \gamma_v)] \wedge \ldots \rightarrow \psi$$

Formula Class 1

To prove

$$\ldots \wedge [\forall v : \gamma_v] \wedge \ldots \rightarrow \psi$$

you can give yourself as many instances of γ as you want:

$$\ldots \wedge [\gamma_{\alpha_1} \wedge \ldots \gamma_{\alpha_n} \wedge (\forall v : \gamma_v)] \wedge \ldots \rightarrow \psi$$

Formula Class 1

To prove

$$
\begin{array}{l}
\quad \texttt{(consp a)} \\
\quad \wedge \\
\quad \texttt{(}\forall\ \texttt{e : (mem e a)} \longrightarrow \texttt{(natp e))} \\
\longrightarrow \\
\quad \texttt{(natp (car a))}
\end{array}
$$

Formula Class 1

To prove

```
(consp a)
∧
???
∧
(∀ e : (mem e a) → (natp e))
→
(natp (car a))
```

Formula Class 1

To prove

$$
\begin{array}{l}
\quad \texttt{(consp a)} \\
\quad \wedge \\
\quad [\texttt{(mem (car a) a)} \rightarrow \texttt{(natp (car a))}] \\
\quad \wedge \\
\quad (\forall\ \texttt{e : (mem e a)} \rightarrow \texttt{(natp e)}) \\
\rightarrow \\
\quad \texttt{(natp (car a))}
\end{array}
$$

Formula Class 3

To prove

$\psi \rightarrow (\forall v : \gamma_v)$

you must prove

$\psi \rightarrow \gamma_z$

where z is a *new* variable symbol.

Formula Class 3

To prove

$\psi \rightarrow (\forall v : \gamma_v)$

you must prove

$\psi \rightarrow \gamma_z$

where z is a variable *not free in* $\psi \rightarrow (\forall v : \gamma_v)$

Formula Class 3

To prove

$\psi \rightarrow (\forall v : \gamma_v)$

you must prove

$\psi \rightarrow \gamma_z$

where z is a new variable symbol.

Formula Class 3

To prove

$$\texttt{(natsp a)} \longrightarrow (\forall\ \texttt{e} : \texttt{(mem e a)} \longrightarrow \texttt{(natp e)})$$

Note that `e` is not free in the formula. So we can just prove:

$$\texttt{(natsp a)} \longrightarrow (\texttt{(mem e a)} \longrightarrow \texttt{(natp e)})$$

Formula Class 3

Prove

$$\texttt{(natp e)} \longrightarrow (\forall \texttt{ e : (mem e a)} \longrightarrow \texttt{(natp e))}$$

Formula Class 3

Prove

$$\texttt{(natp e)} \longrightarrow \texttt{(}\forall\ \texttt{z : (mem z a)} \longrightarrow \texttt{(natp z))}$$

Formula Class 3

Prove

$$\texttt{(natp e)} \longrightarrow \texttt{((mem z a)} \longrightarrow \texttt{(natp z))}$$

Formula Class 2 (like 3)

To prove

$$\ldots \wedge [\exists v : \gamma_v] \wedge \ldots \rightarrow \psi$$

you must prove

$$\ldots \wedge \gamma_z \wedge \ldots \rightarrow \psi$$

where z is a new variable.

Formula Class 2 (like 3)

(∃ e : (mem e a)) ⟶ (consp a)

you must prove

(mem e a) ⟶ (consp a)

Formula Class 4 (like 1)

To prove

$\psi \longrightarrow (\exists\ v : \gamma_v)$

you must prove it for some α:

$\psi \longrightarrow \gamma_\alpha$

Formula Class 4 (like 1)

`(consp a)` $\longrightarrow$ `(`$\exists$ `e : (mem e a))`

you can prove

`(consp a)` $\longrightarrow$ `(mem (car a) a)`

Other Examples

`(∃ x : x=1)`

Use the ∃-Concl rule to reduce this to

`1=1)`

Other Examples

`(∀ x: (∃ y : x+y=0))`

Use ∀-Hyp to get:

`(∃ y : x+y=0)`

Then use ∃-Concl:

`x+(- x)=0`

Other Examples

`(∃ x: (∀ y : x+y=0))`

Use ∃-Hyp, by dropping ∃ and substituting `(- y)` for `x`:

`(∀ y0 : (- y)+y0=0)`

Note that our attempt to capture `y` failed! We're left with something we cannot prove.

Other Examples

How would you prove?

`(natsp a)` $\rightarrow$ `(`$\exists$ `e : (`$\forall$ `d : (mem d a)` $\rightarrow$ `e` $>$ `d)`

Answer: You'd use $\exists$-Concl to replace e by some expression that returns something larger than the largest element in `a`. To write this expression you'd have to define a function, e.g., `max`, that takes a list and returns the largest element, and then write `(+ 1 (max a))`.

Lessons

There are other suitable expressions, e.g., the sum of all the elements of a plus 23.

When choosing the α terms for $\forall$-Hyp and $\exists$-Concl: you may have to define new functions!

Lessons

You can't swap quantifiers arbitrarily:

$(\forall x : (\exists y : \gamma))$

is not necessarily the same as

$(\exists x : (\forall y : \gamma))$

or

$(\exists y : (\forall x : \gamma))$

It is helpful for you to think of γs for which these are different.

Lessons

When you use the rules to eliminate quantifiers, you must pay attention to the order in which the quantifiers are nested. You can apply ∀-Concl but not ∃-Concl to

$$\psi \rightarrow (\forall x : (\exists y : \gamma))$$

because the outermost quantifier in the conclusion is “∀.”