CS313K: Logic, Sets, and Functions

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

Lecture 14 – Chap 4 (4.6, 4.7, 4.8)

Announcement

The book says problems 194, 199, 215, 216, 219 are due on March 16. Wrong!

They're due on March 23 – the Tuesday after Spring Break.

Lemmas are like methods (!). They allow you to avoid doing the same work repeatedly.

```
... (app (cons e a) b) ...
```
$\leftrightarrow$ $\{\text{def app}\}$
```
... (if (endp (cons e a))
        b
        (cons (first (cons e a))
              (app (rest (cons e a)) b))) ...
```
$\leftrightarrow$ $\{\texttt{endp-cons}\}$
```
... (if nil
        b
        (cons (first (cons e a))
              (app (rest (cons e a)) b))) ...
```

```
↔                                          {if-ax-2}
... (cons (first (cons e a))
          (app (rest (cons e a)) b)) ...
↔                                       {first-cons}
... (cons e
          (app (rest (cons e a)) b)) ...
↔                                        {rest-cons}
... (cons e
          (app a b)) ...
...
```

You have to repeat this work every time you expand `(app (cons ... ...) ...)`, *unless* you prove the lemma:

Lemma `app-cons`:
`(app (cons x y) z)` $=$ `(cons x (app y z))`

```
... (app (cons e a) b) ...
```
$\leftrightarrow$ {def app}
```
... (if (endp (cons e a)) ...) ...
```
$\leftrightarrow$ {endp-cons}
```
... (if nil ...) ...
```
$\leftrightarrow$ {if-ax-2}
```
... (cons (first (cons e a)) ...) ...
```
$\leftrightarrow$ {first-cons}
```
... (cons e (app (rest (cons e a)) b)) ...
```
$\leftrightarrow$ {rest-cons}
```
... (cons e (app a b)) ...
...
```

```
... (app (cons e a) b) ...
```

$\leftrightarrow$ {app-cons}

```
... (cons e (app a b)) ...
...
```

In your homework, use lemmas! Don't just grind out the same sequence of steps over and over again.

Proof Advice

"Promote and Forward Chain!"

$$
\begin{array}{l}
((A \rightarrow (B \wedge C)) \\
\ \wedge \\
\ (B \rightarrow D) \\
\ \wedge \\
\ (C \rightarrow E)) \\
\rightarrow \\
(A \rightarrow (D \wedge E))
\end{array}
$$

Writing Down Cases

Suppose you want to prove γ using a case split justified by $\alpha \vee \beta$.

`Case 1:` $(\alpha \rightarrow \gamma)$

...

`Case 2:` $(\beta \rightarrow \gamma)$

...

Proving Something by Rewriting to T

Remember, to prove γ you can show how to rewrite γ to `t`, i.e., you prove $\gamma \leftrightarrow$ `t`.

Writing Down Cases

Suppose you want to prove γ using a case split justified by $\alpha \vee \beta$.

Case 1:
$(\alpha \rightarrow \gamma)$
$\leftrightarrow (\alpha \rightarrow \gamma\prime)$
$\leftrightarrow (\alpha \rightarrow \gamma\prime\prime)$
...
$\leftrightarrow (\alpha \rightarrow \mathtt{t})$
$\leftrightarrow \mathtt{t}$
Case 2: ...

Writing Down Cases

Suppose you want to prove γ using a case split justified by $\alpha \vee \beta$.

Case 1:

$(\alpha \rightarrow \gamma)$

$\leftrightarrow (\alpha \rightarrow \gamma\prime)$

$\leftrightarrow (\alpha \rightarrow \gamma\prime\prime)$

...

$\leftrightarrow (\alpha \rightarrow \mathtt{t})$

$\leftrightarrow \mathtt{t}$

Case 2: ...

Writing Down Cases

Suppose you want to prove γ using a case split justified by $\alpha \vee \beta$.

`Case 1:` α

γ

$\leftrightarrow$ γ'

$\leftrightarrow$ γ''

`...`

$\leftrightarrow$ `t`

`Case 2:` `...`

Writing Down Cases

Suppose you want to prove γ using a case split justified by $\alpha \vee \beta$.

`Case 1:` α *(Remember:* α *is a hypothesis!)*

γ

$\leftrightarrow$ γ'

$\leftrightarrow$ γ''

...

$\leftrightarrow$ `t`

`Case 2:` ...

Setup

When I refer to the "basic axioms" below I mean the axioms about `if`, `t`, `nil`, `consp`, `cons`, `first`, `rest`, `endp`, etc in Section 4.3.