CS313K: Logic, Sets, and Functions

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

Lecture 22 – Chap 7 (7.6, 7.7)

Why Instantiate a Formula with Same Vars?

Recall Quick Sort...

```
(defun qsort (x)  ; Quick Sort
 (if (endp x)
     nil
     (if (endp (rest x))
         x
         (app
          (qsort (all-smaller (first x) (rest x)))
          (cons (first x)
                (qsort (all-others (first x) (rest x))))
```

Imagine an inductive proof of `(ordp (qsort x))`.

```
(defun qsort (x)  ; Quick Sort
 (if (endp x)
     nil
     (if (endp (rest x))
         x
         (app
          (qsort (all-smaller (first x) (rest x)))
          (cons (first x)
                (qsort (all-others (first x) (rest x))))))))
```

Imagine an inductive proof of `(ordp (qsort x))`.

```
((¬(endp x)) ∧ ...
 ∧
 (ordp (qsort ..x..))
 ∧
 (ordp (qsort ..x..)))
→
(ordp (qsort x))
```

$$((\neg\texttt{(endp x)}) \wedge \ldots$$

$$\wedge$$

$$\texttt{(ordp (qsort ..x..))}$$

$$\wedge$$

$$\texttt{(ordp (qsort ..x..)))}$$

$$\longrightarrow$$

```
(ordp (app (qsort ..x..)
           (cons (first x) (qsort ..x..))))
```

Lemma: "If two lists are ordered and everything in the first is less than everything in the second, their concatenation (in that order) is ordered."

```
T1:  ((ordp lo)  ∧ (ordp hi)  ∧
      (∀ x : ( (mem x lo)
               →
               (∀ y : ((mem y hi) → (lexorder x y))))))
     →
      ((ordp (app lo hi)) ↔ t)
```

Goal:

```
(... ∧
  (ordp (qsort ..x..))
  ∧
  (ordp (qsort ..x..)))
→
(ordp (app (qsort ..x..)
           (cons (first x) (qsort ..x..))))
```

The conclusion matches the pattern in T1 under

σ = { `lo` ← `(qsort ..x..)`,
`hi` ← `(cons (first x) (qsort ..x..))`}

So apply

σ = { `lo` ← `(qsort ..x..)`,
`hi` ← `(cons (first x) (qsort ..x..))`}

to

```
T1:  ((ordp lo)  ∧ (ordp hi)  ∧
      (∀ x : ( (mem x lo)
                →
                (∀ y : ((mem y hi) → (lexorder x y)))))
     →
      ((ordp (app lo hi)) ↔ t)
```

Problem 285

THM: $(\exists$ `v : (P v))` $\leftrightarrow$ $(\neg(\forall$ `v :` $(\neg$`(P v))))`.

Proof: This formula is of the form $\alpha \leftrightarrow \beta$ and I'll prove it by proving $\alpha \rightarrow \beta$ and $\beta \rightarrow \alpha$. In office hours yesterday, I proved the first and then got lost trying to prove the second. Here is that proof.

$$(\neg(\forall\ v : (\neg(P\ v)))) \longrightarrow (\exists\ v : (P\ v))$$

$\leftrightarrow$ {Basic}

$$(\neg\ (\exists\ v : (P\ v))) \longrightarrow (\forall\ v : (\neg(P\ v)))$$

$\dashv$ {$\forall$-Concl}

$$(\neg\ (\exists\ v : (P\ v))) \longrightarrow (\neg(P\ d))$$

$\leftrightarrow$ {Basic}

$$(P\ d) \longrightarrow (\exists\ v : (P\ v))$$

$\dashv$ {$\exists$-Concl}

$$(P\ d) \longrightarrow (P\ d)$$

$\leftrightarrow$ T $\square$ {Basic}