CS313K: Logic, Sets, and Functions

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

Lecture 16 – Chap 4 (4.9, 4.10)

Announcement

The Midterm is on April 1 (*not* an April Fools joke!). That is 9 days from today.

Important Note

From now on, when justify our lines we won't necessarily list

- the axioms,
- the hypotheses, or
- the definitions

used.

You may explain just the *main steps*.

You must indicate *all the previously proved lemmas*.

Proof $(H1 \wedge H2) \rightarrow (lhs = rhs)$

$H1$: ¬(endp x)

$H2$: (rev (rev (rest x))) = (rest x)

lhs = (rev (rev x)) rhs = x

```
= (rev (app (rev (rest x))                [rev, H1]
            (cons (first x) nil)))
= (app (rev (cons (first x) nil))[rev-app]
       (rev (rev (rest x))))
= (app (cons (first x) nil)                [rev-id]
       (rev (rev (rest x))))
= (app (cons (first x) nil) (rest x)) [H2]
= (cons (first x) (app nil (rest x)))[...]
```

Note that if you have:

```
(((f a) = (g a))
 ∧
 ...)
```

$\longrightarrow$

```
(p (f a))
```

you may use Hyp to replace `(f a)` by `(g a)` (or vice versa).

But if you have:

```
(((f a) = (g a))
 ∧
 ...)
```

$\longrightarrow$

```
(p (f c))
```

you may *not* use Hyp to replace `(f c)` by `(g c)`.

The Hyp rule *does not allow instantiation* of the variables of the hypothesis.

But if you have a previously proved theorem

`T1:  (f a) = (g a)`

and wish to prove:

```
   (((f a) = (g a))
    ∧
    ...)
→
   (p (f c))
```

you may use Rewrite to replace `(f c)` by `(g c)`.

But if you have a previously proved theorem

T1: `(f a) = (g a)`

and wish to prove:

```
   (((f a) = (g a))
    ∧
    ...)
→
   (p (f c))
```

you may use Rewrite to replace `(f c)` by `(g c)`.

Difference between Hyp and Rewrite

Rewrite appeals to a previously proved theorem.

Theorems hold *for all* values of the variables.

Hyp appeals to a hypothesis of the current conjecture.

A hypothesis is about the current (even if unknown) values of the variables.

Here is a trivial algebraic theorem:

$$(x = 1) \rightarrow (x + 1 = 2)$$

which is easy proved via Hyp.

Here is a formula that is *not* a theorem.

$$(x = 1) \rightarrow (0 = 1)$$

If Hyp allowed instantiation, we could prove it!
(And *we don't want to be able to prove this!*)