CS 81 Assignment 7 Solutions for Wed., Mar. 30
Hein p. 532-533:

---

1. 10.d

Given formula:

$$\exists x\ \forall y\ p(x, y) \land \forall x\ (p(x, x) \rightarrow \exists y\ q(y, x)) \rightarrow \exists y\ \exists x\ q(x, y)$$

According to Hein's syntax rules (p. 351), this is equivalent to:

$$(\exists x\ \forall y\ p(x, y) \land \forall x\ (p(x, x) \rightarrow \exists y\ q(y, x))) \rightarrow \exists y\ \exists x\ q(x, y)$$

Negate the entire formula:

$$\neg((\exists x\ \forall y\ p(x, y) \land \forall x\ (p(x, x) \rightarrow \exists y\ q(y, x))) \rightarrow \exists y\ \exists x\ q(x, y))$$

Replace the →s:

$$\neg(\neg(\exists x\ \forall y\ p(x, y) \land \forall x\ (\neg p(x, x) \lor \exists y\ q(y, x))) \lor \exists y\ \exists x\ q(x, y))$$

Push negations inward:

$$(\neg\neg(\exists x\ \forall y\ p(x, y) \land \forall x\ (\neg p(x, x) \lor \exists y\ q(y, x))) \land \neg\exists y\ \exists x\ q(x, y))$$

$$\exists x \forall y\ p(x, y) \ \land \forall x\ (\neg p(x, x) \lor \exists y\ q(y, x))) \ \land \ \forall y\ \forall x\ \neg q(x, y)$$

Separate into conjoined formulas, which will become clauses:

$\exists x\ \forall y\ p(x, y)$
$\forall x\ (\neg p(x, x) \lor \exists y\ q(y, x))$
$\forall y\ \forall x\ \neg q(x, y)$

Use prenex quantifier conversion rules in 2nd formula:

$\exists x\ \forall y\ p(x, y)$
$\forall x \exists y\ (\neg p(x, x) \lor q(y, x))$
$\forall y\ \forall x\ \neg q(x, y)$

Introduce skolem constant a for x in the first formula and function f(x) for y in the second, and drop universally-quantified variables.

$p(x, a)$
$\neg p(x, x) \lor q(f(x), x)$
$\neg q(x, y)$

The above is our clause form.

Conduct resolution:

1.	p(x, a)	Premise
2.	¬p(x, x) ∨ q(f(x), x)	Premise
3.	¬q(x, y)	Premise
4.	q(f(a), a)	Resolution 1, 2
5.	⊥	Resolution 3, 4

---

2. 13.b (Use otter to solve, by first translating to clause form by hand.)

    a. Every committee member is rich and famous.
    b. Some committee members are old.
    c. Therefore some committee members are old and famous.

Translation:

    a. ∀x (committeeMember(x) → (rich(x) ∧ famous(x)))
    b. ∃x (committeeMember(x) ∧ old(x))
    c. ∃x (committeeMember(x) ∧ (old(x) ∧ famous(x))

Replace → and negate conclusion:

    a. ∀x (¬committeeMember(x) ∨ (rich(x) ∧ famous(x)))
    b. ∃x (committeeMember(x) ∧ old(x))
    c. ∀x (¬committeeMember(x) ∨ ¬old(x) ∨ ¬famous(x))

Convert to clauses:

    d. ¬committeeMember(x) ∨ rich(x)
    e. ¬committeeMember(x) ∨ famous(x)
    f. committeeMember(a)
    g. old(a)
    h. ¬committeeMember(x) ∨ ¬old(x) ∨ ¬famous(x)

Otter version of clauses:

```
-committeeMember(x) | rich(x).

-committeeMember(x) | famous(x).

committeeMember(a).

old(a).

-committeeMember(x) | -old(x) | -famous(x).
```

Otter proof:

```
2 [] -committeeMember(x)|famous(x).
3 [] -committeeMember(x)| -old(x)| -famous(x).
4 [] committeeMember(a).
5 [] old(a).
6 [hyper,4,2] famous(a).
8 [hyper,6,3,4,5] $F.
```

The author of otter states that it is not intended to do inductive proofs, and refers the reader to theorem provers such as ACL2 for such tasks. The difficulty is that otter does not know how to automate the set up of a basis and induction step. However, otter should be able to automate the proof of those parts once they are set up for it. This part of the assignment involves demonstrating that otter can do this.

Refer to the number theory axioms used earlier in the course.
Express axioms N1-N6 in otter formula (not clause) form, and use it in the following problems.
Create a single file that displays the input and the proof part of the output in each case.

For all problems, I added to the following axioms as a base.

```
formula_list(usable).

all x (s(x) != 0).                                    % N1

all x all y (s(x) = s(y) -> x = y).                   % N2

all x (a(x, 0) = x).                                  % N3

all x all y (a(x, s(y)) = s(a(x, y))).                % N4

all x (m(x, 0) = 0).                                  % N5

all x all y all z (m(x, s(y)) = a(m(x, y), x)).       % N6

end_of_list.
```

---

3. Prove theorem T1 using otter, by separate proofs for the basis and induction step. Call these files num1.in and num2.in.

```
-(a(0, 0) = 0).                              % T1 basis
```

Proof of T1 basis:

```
3 [] a(0,0)!=0.
4 [] a(x,0)=x.
6 [binary,4.1,3.1] $F.
```

```
-(all x (a(0, x) = x -> a(0, s(x)) = s(x))).  % T1 induction step
```

Proof of T1 induction step:

```
3 [] a(0,s($c1))!=s($c1).
6 [] a(x,s(y))=s(a(x,y)).
7 [copy,6,flip.1] s(a(x,y))=a(x,s(y)).
13 [] a(0,$c1)=$c1.
33 [para_into,7.1.1.1,13.1.1,flip.1] a(0,s($c1))=s($c1).
35 [binary,33.1,3.1] $F.
```

---

4. Introduce T1 as an axiom, then prove T2 in a similar manner as used in the previous problem. Call these files num3.in and num4.in.

```
all x (a(0, x) = x).                          % T1

-(all x (a(s(x), 0) = s(a(x, 0)))).           % T2 basis
```

Proof of T2 basis:

```
3 [] a(s($c1),0)!=s(a($c1,0)).
4 [copy,3,flip.1] s(a($c1,0))!=a(s($c1),0).
6,5 [] a(x,0)=x.
16 [back_demod,4,demod,6,6] s($c1)!=s($c1).
18 [para_into,5.1.1,5.1.1] x=x.
19 [binary,18.1,16.1] $F.


-(
  all y                                       % T2 induction step
    (((all x (a(s(x), y) = s(a(x, y))))
   -> (all x (a(s(x), s(y)) = s(a(x, s(y)))))))
 ).
```

Proof of T2 induction step:

```
3 [] a(s($c1),s($c2))!=s(a($c1,s($c2))).
4 [copy,3,flip.1] s(a($c1,s($c2)))!=a(s($c1),s($c2)).
7 [] a(x,s(y))=s(a(x,y)).
9,8 [copy,7,flip.1] s(a(x,y))=a(x,s(y)).
18 [] a(s(x),$c2)=s(a(x,$c2)).
19 [copy,18,demod,9] a(s(x),$c2)=a(x,s($c2)).
20 [back_demod,4,demod,9] a($c1,s(s($c2)))!=a(s($c1),s($c2)).
86 [para_from,19.1.1,8.1.1.1,demod,9]
                              a(x,s(s($c2)))=a(s(x),s($c2)).
87 [binary,86.1,20.1] $F.
```

---

5. Introduce T2 as an axiom, then prove T3 in a similar manner as used in the previous problem. Call these files num5.in and num6.in.

```
all x (a(0, x) = x).                          % T1

all y all x (a(s(x), y) = s(a(x, y))).        % T2

-(all y (a(0, y) = a(y, 0))).                 % T3 basis
```

Proof of T3 basis:

```
3 [] a(0,$c1)!=a($c1,0).
5,4 [] a(x,0)=x.
14,13 [] a(0,x)=x.
17 [back_demod,3,demod,14,5] $c1!=$c1.
20 [para_into,4.1.1,4.1.1] x=x.
21 [binary,20.1,17.1] $F.


-(
 (all y (a(x, y) = a(y, x)))         % T3 induction step
  -> (all y (a(s(x), y) = a(y, s(x))))
 ).
```

Proof of T3 induction step:

```
3 [] a(s(x),$c1)!=a($c1,s(x)).
4 [] a(x,0)=x.
6 [] a(x,s(y))=s(a(x,y)).
8,7 [copy,6,flip.1] s(a(x,y))=a(x,s(y)).
15 [] a(s(x),y)=s(a(x,y)).
16 [copy,15,demod,8] a(s(x),y)=a(x,s(y)).
17 [] a(x,x)=a(x,x).
18 [copy,16,flip.1] a(x,s(y))=a(s(x),y).
27 [para_into,7.1.1.1,17.1.1,demod,8] a(x,s(x))=a(x,s(x)).
28 [para_into,7.1.1.1,4.1.1] s(x)=a(x,s(0)).
31 [copy,27,flip.1] a(x,s(x))=a(x,s(x)).
32 [copy,28,flip.1] a(x,s(0))=s(x).
59,58 [para_into,32.1.1,17.1.1,flip.1] s(x)=a(s(0),x).
66 [back_demod,31,demod,59] a(x,s(x))=a(x,a(s(0),x)).
69 [back_demod,3,demod,59,59] a(a(s(0),x),$c1)!=a($c1,a(s(0),x)).
1394 [para_into,66.1.1,18.1.1,demod,59]a(a(s(0),x),x)=a(x,a(s(0),x)).
1395 [binary,1394.1,69.1] $F.
```

---

6. Prove the associative law (which you did by hand in assignment 4) using otter. Call these files num7.in and num8.in.

```
all x (a(0, x) = x).                                  % T1

all y all x (a(s(x), y) = s(a(x, y))).                % T2

all x all y (a(x, y) = a(y, x)).                      % T3

-(
 all y (all z (a(0, a(y, z)) = a(a(0, y), z)))        % T4 basis
 ).
```

Proof of T4 basis:

```
3 [] a(0,a($c2,$c1))!=a(a(0,$c2),$c1).
4 [copy,3,flip.1] a(a(0,$c2),$c1)!=a(0,a($c2,$c1)).
5 [] a(x,0)=x.
15,14 [] a(0,x)=x.
19 [back_demod,4,demod,15,15] a($c2,$c1)!=a($c2,$c1).
22 [para_into,5.1.1,5.1.1] x=x.
23 [binary,22.1,19.1] $F.
```

```
-(all x                                          % T4 induction step
   (
       (all y (all z (a(x, a(y, z)) = a(a(x, y), z))))
    -> (all y (all z (a(s(x), a(y, z)) = a(a(s(x), y), z))))
     )
    ).
```

Proof of T4 induction step:

```
3 [] a(s($c3),a($c2,$c1))!=a(a(s($c3),$c2),$c1).
4 [copy,3,flip.1] a(a(s($c3),$c2),$c1)!=a(s($c3),a($c2,$c1)).
5 [] a(x,0)=x.
7 [] a(x,s(y))=s(a(x,y)).
9,8 [copy,7,flip.1] s(a(x,y))=a(x,s(y)).
16 [] a(s(x),y)=s(a(x,y)).
17 [copy,16,demod,9] a(s(x),y)=a(x,s(y)).
18 [] a(x,y)=a(y,x).
19 [] a($c3,a(x,y))=a(a($c3,x),y).
21,20 [copy,19,flip.1] a(a($c3,x),y)=a($c3,a(x,y)).
22 [copy,17,flip.1] a(x,s(y))=a(s(x),y).
32 [para_into,8.1.1.1,5.1.1] s(x)=a(x,s(0)).
35 [copy,32,flip.1] a(x,s(0))=s(x).
48 [para_from,32.1.1,4.1.1.1.1,demod,21,21]
a($c3,a(a(s(0),$c2),$c1))!=a(s($c3),a($c2,$c1)).
52 [para_into,35.1.1,18.1.1] a(s(0),x)=s(x).
54 [copy,52,flip.1] s(x)=a(s(0),x).
117 [para_into,17.1.1.1,54.1.1] a(a(s(0),x),y)=a(x,s(y)).
159 [para_into,22.1.1.2,8.1.1] a(x,a(y,s(z)))=a(s(x),a(y,z)).
168 [copy,159,flip.1] a(s(x),a(y,z))=a(x,a(y,s(z))).
1678 [para_from,117.1.1,48.1.1.2,flip.1]
              a(s($c3),a($c2,$c1))!=a($c3,a($c2,s($c1))).
1679 [binary,1678.1,168.1] $F.
```

Extra credit:

Translate the otter proof output in 3-6 back into a more readable presentational form.