David Pick
CM 2403

2 b)

```
fact {
#Person >= 4
}
```

2 c)

nonempty: There is a friend
transitive: a friend of my friend is my friend
irreflexive: I am not my own friend
symmetric: Barney says, "I love you, you love me..."
functional: No person likes more than one person
injective: Only liked by one. Also, "a face only a mother could love"
total: Every friend has a friend
onto: Every person is liked.

Information taken from
www.purplemath.com/modules/fcns.htm
www.wikipedia.org

2 d)

dropping nonempty: No instance found
dropping transitive:
{Person$0->Person$3, Person$1->Person$2, Person$2->Person$1, Person$3->Person$0}
dropping irreflexive:
{Person$0->Person$0, Person$1->Person$1, Person$2->Person$2, Person$3->Person$3}
dropping symmetric: No instance found
dropping functional: No instance found
dropping injective: No instance found
dropping total: No instance found
dropping onto: No instance found

3)

```
module homework/hw02_2

sig Person { likes: set Person }

assert NewNonEmptinessOK {
  some likes iff
    (some p1, p2: Person | p1->p2 in likes)
}
check NewNonEmptinessOK

//Transitive
assert NewTransitivenessOK {
  (likes.likes in likes) iff
    (all p1, p2, p3: Person | p1->p2 in likes and p2->p3 in likes implies p1->p3 in likes)//
replace constraint with your answer
```

```
}
check NewTransitivenessOK

//Irreflexive
assert NewIrreflexiveOK {
  (no iden & likes) iff
    (all p1: Person | p1->p1 not in likes)   // replace constraint with your answer
}
check NewIrreflexiveOK

//Symmetric
assert NewSymmetricOK {
  (~likes in likes) iff
    (all p1, p2: Person | p1->p2 in likes implies p2->p1 in likes)  // replace constraint with
your answer
}
check NewSymmetricOK

//Functional
assert NewFunctionalOK {
  (~likes.likes in iden) iff
    (all p1: Person | not #p1.likes > 1)  // replace constraint with your answer
}
check NewFunctionalOK

//Injective
assert NewInjectiveOK {
  (likes.~likes in iden) iff
    (all p1: Person | not #p1.~likes > 1)  // replace constraint with your answer
}
check NewInjectiveOK

//Total
assert NewTotalOK {
  (Person in likes.Person) iff
    (all p1: Person | p1 in likes.Person)  // replace constraint with your answer
}
check NewTotalOK

//Onto
assert NewOntoOK {
  (Person in Person.likes) iff
    (all p1: Person | p1 in Person.likes)  // replace constraint with your answer
}
check NewOntoOK
```

5)
2.5 hours