Andreas van Cranenburgh, 0440949
Basic Logic, predicate logic 3

Exercise 1

Theorem: if phi is a formula in a first-order language, then there is an equivalent formula phi' using only implication, negation and existential and universal quantification as logical constants.

Proof: induction on complexity of phi
base case: phi = phi', since no logical constants occur
induction step:
assume that for psi and chi we have psi' and chi', such that psi is equivalent
with psi' and psi' uses only implication negation and existential quantification, and mutatis mutantdis for chi.
The following cases occur:
- phi = ~psi
  phi' = ~psi'
- phi = ∃x psi:
  phi' = ∃x psi'
- phi = ∀x psi
  phi' = ∀x psi'
- phi = psi -> chi
  phi = psi' -> chi'
- phi = psi ^ chi
  phi' = ~(psi' -> ~chi')
- phi = psi v chi
  phi' = ~psi' -> chi'

The last three equivalences follow from propositional logic.

Lemma:

if x does not occur freely in A
A -> ∀x: B <=> ∀x: (A -> B)

```
1.    A -> ∀x: B              pr.
2. +- A                       A
3. |  ∀x: B                   E->1, 2
4. |  B                       E∀
   +--------------
5.    A -> B                  I->
6.    ∀x: (A -> B)            I∀
```

```
1.    ∀x: (A -> B)            pr.
2.    A - > B                 E∀
3. +- A                       A
4. |  B                       E->
5. |  ∀x: B                   I∀
   +--------------
6.    A -> ∀x: B              I->
```

```
∃x: B -> A <=> ∀x: (B -> A)

1.     ∃x: B -> A                 pr.
2. +- B                           A
3. |  ∃x B                        I∃2
4. |  A                           E->1,3
   +-------------------
5.   B -> A                       I->
6.   ∀x: (B -> A)                 I∀

1.     ∀x: (B -> A)               pr.
2.     B -> A                     E∀
3. +- ∃x: B                       A
4. |  A                           E∃ 2,3
   +-------------------
5.     ∃x: B -> A                 I->

A -> ∃x: B <=> ∃x: (A -> B)

1.       A -> ∃x: B               pr.
2. +--   A                        A
3. |     ∃x: B                    E->1,2
4. |+-   B                        A
6. ||    B                        rep. 4
   |+-----------------
7. |     B -> B                   I->
8. |     B                        E∃ 3,7
   +------------------
9.       A -> B                   I->
10.      ∃x (A -> B)              I∃

1.      ∃x: (A -> B)               pr.
2. +-- A -> B                      A
3. |+- A                           A
4. ||  B                           E->2,3
5. ||  ∃x: B                       I∃
   |+------------------
6. |   A -> ∃x: B                  I->
   +-------------------
7.      (A -> B) -> (A -> ∃x: B)   I->
8.      A -> ∃x: B                 E∃ 1,7

∀x: B -> A <=> ∃x: (B -> A)

1.     ∀x: B -> A                  pr.
2.    ~∃x: (B -> A)                A
3. [...]

1.     ∃x: (B -> A)                pr.
2.+-- ∀x: B                        A
3.|   B                            E∀
4.|+- B -> A                       A
5.||  A                            E->3,4
  |+---------------
6.|    (B -> A) -> A               I->
7.|   A                            E∃ 1,6
 +----------------
8.     ∀x: B -> A                  I->
```

By the completeness theorem we know that these derivations show that these equivalences hold.

Exercise 2

i) Given the Completeness theorem Δ has a model. Now, consider an arbitrary model M = D, I of Δ. We construct a model M' = D', I' such that M' is a restriction of M.

D' is a subset of D, such that for every d in D' and d' in D', either d = d', or there is some P such that d is in I(P) whereas d' is not in I(P), or vice versa. In other words, the objects in D' are discernable by at least one predicate.
I' is defined thus: For every d in D', I'(d) = I(d)

Because we have a finite number of predicates, there are only a finite number of distinct objects required. In other words, if we have n predicates, then each object in our domain can be characterized by a n-digit bit string, of which there are only a finite number possible. Hence the restricted model is finite.

ii) Given the previous theorem we can decide for any formula in the language L whether it is a logical validity. We do this by enumerating all possible models, starting by interpreting each predicate occuring in the formula as the empty set, and then adding objects step-by-step. After a finite amount of steps we reach the interpretation where each predicate is interpreted with the whole domain, because both the domain and number of predicates is finite. If we have not encountered an interpretation where the formula is false, then the formula is a logical validity.

Exercise 3

i)
Given that phi is valid in all infinite structures, phi cannot specify anything in particular, so the sentence phi must express that there are at least n objects in the domain, for an arbitrary n.

Consider a sentence phi such that:
phi = ∃x1 ... ∃xn: x1 =/= x2 ... x1 =/= xn ... x2 =/= x3 ... xn-1 =/= xn

All inifinite structures model phi, but for every value of n we can construct a model with n+1 objects which will also be a model of phi.

ii)
We can imagine Sigma as consisting of the following infinite set:
phi_2 = ∃x1 ∃x2: x1 =/= x2
phi_3 = ∃x1 ∃x2 ∃x3: x1 =/= x2 ^ x1 =/= x3 ^ x2 =/= x3
....

Now by the compactness theorem it may seem that this theory has a finite model, but in fact it is not possible to find an n such that it is big enough to satisfy all the formulas in sigma, because for every finite n, we simply present the sentence phi_n+1 to contradict the claim that a model with n elements could satisfy Sigma.